

Bayes-tilastotiede 1

Jouni Helske, Santtu Tikka

Copyright © 2021, 2022 Jouni Helske, Santtu Tikka, päivitetty 16.9.2022

Contents

Opintojakson kuvaus	2
Osaamistavoitteet	2
Kirjallisuutta	2
Ohjelmistoja	2
1 Johdanto	4
1.1 Bayes-tilastotiede?	4
1.2 Käsitteitä ja merkintöjä	7
2 Todennäköisyys epävarmuuden mittana	7
2.1 Todennäköisyyden aksioomat (Kolmogorovin aksioomat)	7
2.2 Ehdollinen todennäköisyys	7
2.3 Vedonlyönti ja subjektiivinen todennäköisyys	8
2.4 Todennäköisyyksien päivittyminen	8
2.5 Riippumattomuus	9
2.6 Bayesin kaava	9
2.7 Suora ja käänteinen todennäköisyys	10
3 Bayes-päätelyn filosofiaa	10
Esimerkki 4: Kolikonheitto (vaihdannaisuus)	11
Esimerkki 5: Luottamusväli ja posterioriväli	12
4 Posteriori päätelyn välineenä	13
4.1 Data-analyysin vaiheet Bayes-näkökulmasta	19
4.2 Priorijakauma	26
4.3 Prediktiiiviset jakaumat	32
4.4 Mallinvalinta ja -vertailu	65
4.5 Monimutkaisempia brms-malleja	66
4.6 Stan-mallien kirjoittaminen	72
5 Yleisiä periaatteita	75
5.1 Epäoleellisuus	75
5.2 Tyhjentyvyys	75
5.3 Uskottavuusperiaate ja epäinformatiivinen pysäyttäminen	76
6 Lisää esimerkkejä	77
6.1 Kalalohiset	77
6.2 Perhoslajimallinnus	83

Opintojakson kuvaus

Kurssi käsittelee Bayes-tilastotieteen perusteita: todennäköisyys epävarmuuden mittana, priori- posteriori- ja prediktiviset jakaumat, priorin valinta ja sen vaikutus posterioriin, Bayes-mallin sovitus aineistoon ja tulosten tulkinta.

Tarvittavia ja suositeltuja esitietoja:

- Todennäköisyyslaskennan perusteet (ehdollinen todennäköisyys, riippumattomuus, odotusarvon määrittelmä integraalina

$$E(g(x)) = \int_x g(x)p(x) dx$$

- R (noin YLM-kurssien taso)
- Yleistetyt lineaariset mallit 1 ja 2 (mallinnuksen perusteet, Poisson, binomi- ym. jakaumat)
- Tilastollinen päättely 1 (uskottavuusfunktio)

Osaamistavoitteet

Kurssin suorittanut

- ymmärtää Bayes-lähestymistavan peruseriaatteet ja erot suurimman uskottavuuden estimoinnin lähestymistapaan
- ymmärtää priori- ja posteriorijakauman käsitteet ja niiden yhteydet
- ymmärtää simuloinnin merkityksen Bayes-tilastotieteessä
- osaa tehdä tulkintoja priori- posteriorijakaumien perusteella
- osaa kuvata hierarkkisen Bayes-mallin rakenteen matemaattisesti
- osaa sovittaa, tulkita ja vertailla Bayes-malleja valmiilla ohjelmistoilla, erityisesti yleistettyjen lineaaristen mallien kehikon tapauksissa

Kirjallisuutta

- Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari ja Donald B Rubin (2013). *Bayesian Data Analysis*, Third Edition, CRC Press. [Saatavilla Gelmanin sivuilta ilmaisena pdf:nä](#)
- Richard McElreath (2020). *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC press. [Myös luentovideot saatavilla youtubesta](#)
- Paul-Christian Bürkner (2017). *brms: An R Package for Bayesian Multilevel Models Using Stan*. Journal of Statistical Software, 80(1), 1–28. [doi:http://dx.doi.org/10.18637/jss.v080.i01](https://doi.org/10.18637/jss.v080.i01)
- Jonah Gabry, Dan Simpson, Aki Vehtari, Michael Betancourt ja Andrew Gelman (2019). *Visualization in Bayesian workflow*. J. R. Stat. Soc. A, 182: 389–402. [doi:10.1111/rssa.12378](https://doi.org/10.1111/rssa.12378)
- Andrew Gelman, Jennifer Hill, Aki Vehtari (2020). *Regression and Other Stories*. <https://avehtari.github.io/ROS-Examples/>

Ohjelmistoja

Käytännön Bayes-tilastotiede on usein laskennallisesti intensiivistä, ja niinpä Bayes-mallinnukseen on kehitetty (ja kehitetään edelleen) lukuisia erilaisia algoritmeja ja niiden implementaatioita. Tällä kurssilla käytämme probabilistiseen ohjelmointikieleen nimeltä *Stan* pohjautuvia R-paketteja. Muita vaihtoehtoisia, yleisiä Bayes-mallinnusohjelmistoja ovat mm. *BUGS*, *JAGS* ja *Nimble* (ja niiden R-käyttöliittymät). Lisäksi erityyppiselle malleille ja ongelmille on olemassa omia räätälöityjä algoritmeja ja niiden implementointeja eri kielillä.

Listaus kurssilla käytettävistä paketeista

Paketti	Käyttötarkoitus
<code>brms</code>	Paketti Bayes-monitasomallien sovitukseen
<code>rstan</code>	R-käyttöliittymäpaketti Stan-mallien käyttöön
<code>bayesplot</code>	Bayes-analyysiin liittyvien kuvien piirtäminen
<code>tidybayes</code>	Apuvälineitä Bayes-analyysiin
<code>loo</code>	Bayes-ristiinvalidointi
<code>rstanarm</code>	<code>brms</code> -paketin kaltainen mallinnuspaketti
<code>ggplot2</code>	Yleinen graafien piirto (jota <code>bayesplot</code> kin käyttää)
<code>dplyr</code>	Datan muokkaus

Lisäksi edellä mainitut paketit tarvitsevat suuren joukon muita paketteja toimiakseen. Windows-koneen käyttäjät tarvitsevat myös [Rtools-ohjelman](#).

Koska yllämainitut paketit päivittyvät usein, on hyvä pitää myös R ajantasaisena. Luentomonisteen kirjoittamishetkellä vähintään R 4.2.0 olisi hyvä olla asennettuna, mutta vanhemmallakin voi ehkä pärjätä (jos ei, niin pakettien asennusvaiheessa tulee kyllä tästä ilmoitus).

Kokeile ensin toimiiko pakettien rakentaminen:

```
library(pkgbuild) # asenna normaalisti jos puuttuu
check_build_tools(TRUE)
```

```
## Found in Rtools 4.2 installation folder
```

```
## Your system is ready to build packages!
```

RStudiota käytettäessä tämä aiheuttaa Rtools-ohjelman asennuksen jos se puuttuu koneelta. Voit myös asentaa Rtoolsin suoraan lataamalla ohjelman täältä: <https://cran.r-project.org/bin/windows/Rtools/>

Asenna sitten `rstan`-paketti. Periaatteessa tämän pitäisi onnistua normaalisti komennolla

```
install.packages("rstan")
```

Kokeile sitten toimiiko `rstan` ajamalla `stan`-funktion dokumentaation esimerkki.

Lisää ohjeita: <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>

1 Johdanto

1.1 Bayes-tilastotiede?

Tilastollisen päättelyn päämääränä on tehdä havaintoaineiston perusteella päätelmiä ei-havaituista suureista. Nämä estimoitavat suuret voivat olla mahdollisesti havaittavia, kuten puuttuvat tai tulevaisuuden havainnot, tai ne voivat olla parametreja joita ei edes periaatteessa voida suoraan havaita. Peruseriaatteiltaan Bayes-päättelyn voi jakaa kolmeen vaiheeseen:

1. Määritellään yhteinen todennäköisyysmalli kaikille relevanteille suureille (havaituille tai havaitsemattomille).
2. Lasketaan havaitsemattomien suureiden ehdollinen jakauma ehdolla havaintoaineisto.
3. Tutkitaan mallin sopivuutta ja tehdään johtopäätöksiä edellä saadun ehdollisen jakauman perusteella. Tarvittaessa muokataan mallia ja toistetaan vaiheet 1–3.

Pääpiirteissään Bayes-analyysi noudattelee siis samoja työvaiheita kuin perinteinen suurimman uskottavuuden (SU) menetelmään perustuva data-analyysi: Valitaan jokin sopivalta vaikuttava malli, estimoidaan, tulkitaan ja toistetaan kunnes olemme tyytyväisiä. Teknisten erojen lisäksi eräs suuri ero on tulosten tulkinnoissa: Bayes-päättelyssä tilastolliset johtopäätökset ovat usein helpommin tulkittavissa sillä kaikki tulokset pohjautuvat ehdollisiin todennäköisyyksiin ilman ns. frekventistisen tilastotieteen kiertelyä (vrt. luottamusväli ja p -arvo).

Bayes-tilastotieteessä kaikki tuntemattomat suuret ajatellaan satunnaismuuttujiksi ja niihin liittyvää epävarmuutta mallinnetaan todennäköisyysjakaumilla. Tuntemattomien parametrien, puuttuvien havaintojen ja tulevaisuuden havaintojen välillä ei ole periaatteellista eroa. Todennäköisyys tulkitaan siis Bayes-päättelyssä *uskomusasteena* (degree of belief) joka toimii epävarmuuden mittana. Tämä lähestymistapa mahdollistaa erilaisten ennakkotietojen ja havaintoaineiston yhdistämisen:

Ennakkotieto, joka voi pohjautua esimerkiksi aiempiin tutkimuksiin, tutkijan omiin uskomuksiin tai “maalaisjärkeen”, pyritään muotoilemaan todennäköisyysjakaumana (*priorijakauma* $p(\theta)$) jota päivitetään havaintoaineistolla (*uskottavuusfunktioilla/otantajakaumalla/havaintomallilla* $p(y|\theta)$). Näin päädyimme nk. *posteriorijakaumaan* $p(\theta|y)$. Matemaattisesti

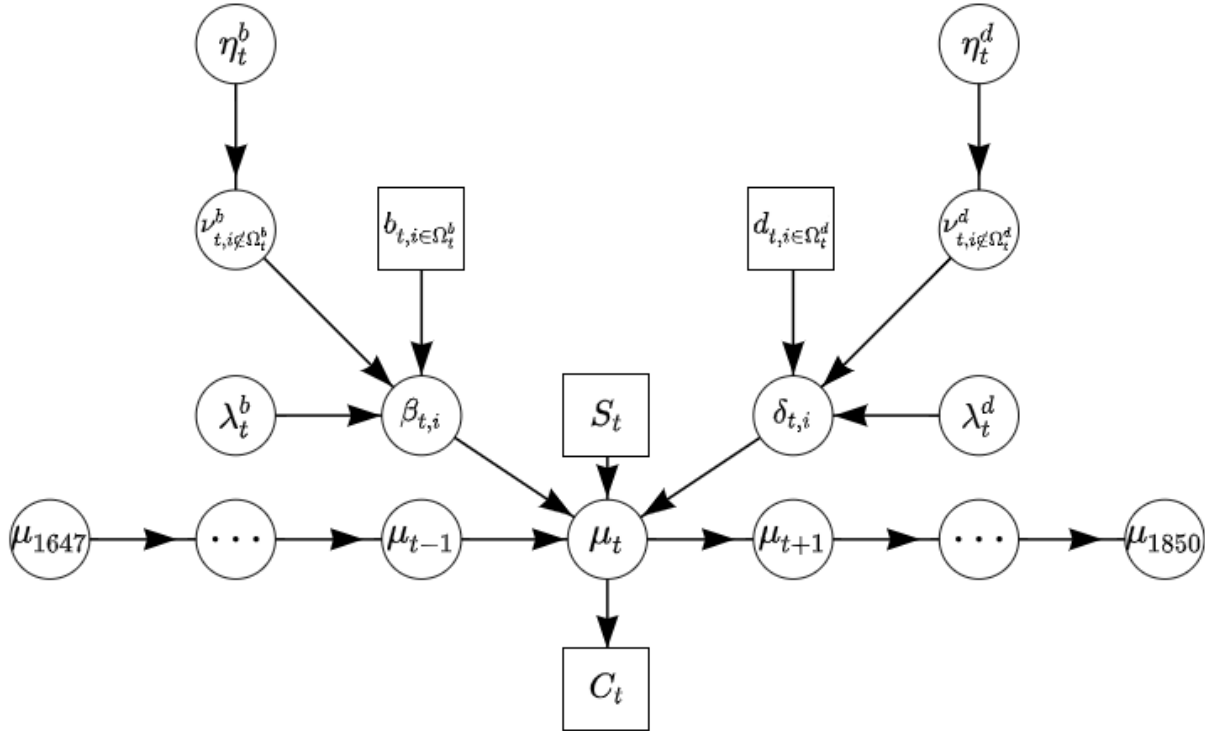
$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}.$$

Prioritiedon hyödyntämisen lisäksi Bayes-tilastotieteen menetelmät mahdollistavat kompleksisten, hierarkkisten mallien estimoinnin, jotka esimerkiksi suurimman uskottavuuden (SU) menetelmällä olisivat erittäin hankalia tai jopa mahdottomia estimoida. Bayes-menetelmät mahdollistavat myös parametrien estimoinnista johtuvan epävarmuuden huomioonottamisen esimerkiksi ennustamisessa, ja todennäköisyyksiin pohjautuvien tulkintojen tekemisen (tämä koskee myös mielivaltaisia parametrien ja havaintojen funktioita).

Bayes-tilastotieteen leviämistä jarrutti pitkään mallien estimointiin liittyvät laskennalliset ongelmat. Nykyään on kuitenkin olemassa useita tehokkaita ja helppokäyttöisiä menetelmiä ja ohjelmistoja kompleksisten Bayes-mallien sovittamiseen, mm. yleiskäyttöiset JAGS, Nimble ja Stan ohjelmistot, sekä esimerkiksi Staniin pohjautuva brms R-paketti joka mahdollistaa perinteisten yleistettyjen lineaaristen (seka)mallien laajentamisen Bayes-kehikkoon (ja enemmänkin).

Esimerkki 1: Monimutkainen hierarkkinen Bayes-malli

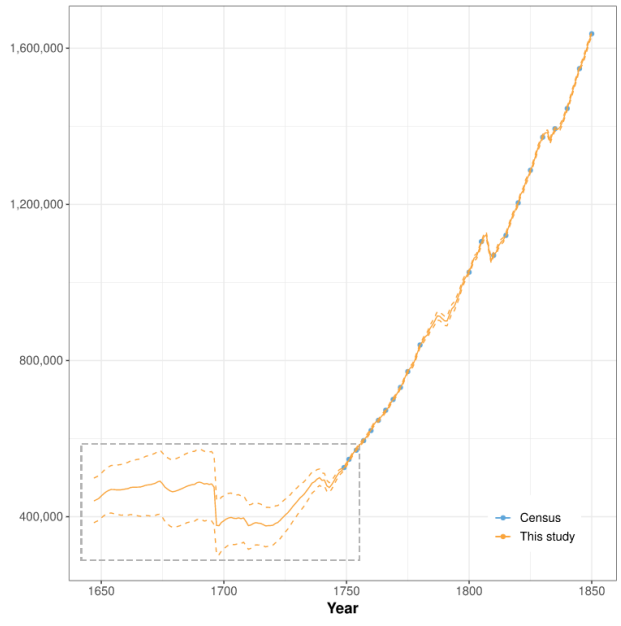
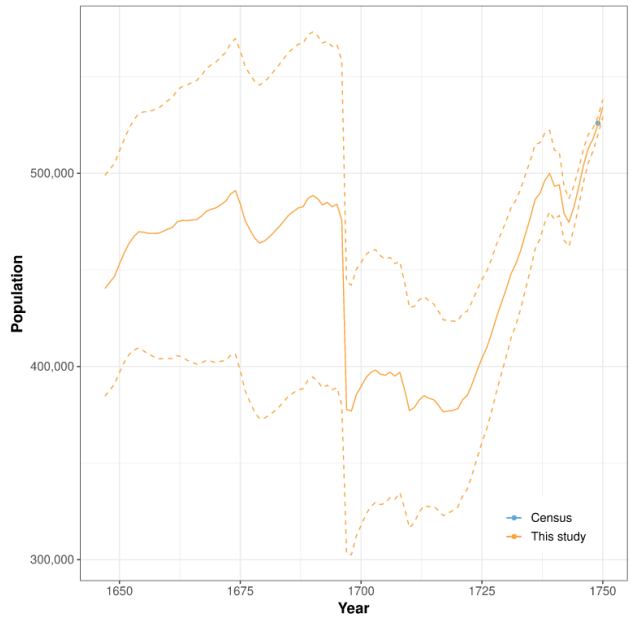
Alla oleva kuva on ns. suunnattu silmukaton graafi (directed acyclic graph, DAG), joka kuvastaa tuntemattomien parametrien (ympyrät) ja havaintojen (neliöt) välisiä yhteyksiä. Tarkemmin sanottuna graafi kuvaa hierarkkista Bayes-aikasarjamallia jota käytettiin Suomen väkiluvun estimoimiseen¹. Vaikkakin mallin estimointi Bayes-menetelmilläkin on haastavaa, SU-menetelmillä se olisi ollut lähes mahdotonta.



Huomaa mallin hierarkkinen rakenne: havainnot ja tuntemattomat parametrit liittyvät toisiinsa useilla eri tasoilla. Monimutkaisia Bayes-malleja kutsutaankin joskus Bayes-monitasomalleiksi (multilevel model).

Ylläolevan mallin avulla saatiin Bayes-estimaatit Suomen väkiluvulle ja estimaatteihin liittyvät 95 % epävarmuusvälit:

¹[Linkki artikkeliin Demography-lehdessä](#)



1.2 Käsitteitä ja merkintöjä

- Merkitään tuntemattomia parametreja yleisesti kirjaimella θ , joka voi tilanteesta riippuen olla skalaari, vektori tai yleisemmin kokoelma edellä mainittuja.
- Käytetään kirjainta y viittaamaan mallinnettaviin havaintoihin y_1, \dots, y_n joista jokainen y_i voi olla skalaari tai vektori ja jossa n on havaintojen lukumäärä.
- Joskus käytämme merkintää \tilde{y} viittaamaan estimoitaviin mahdollisesti havaittaviin suureisiin kuten tulevaisuuden havaintoihin.
- Käytetään kirjainta x viittaamaan selittäviin muuttujiin (kovariaatteihin), jotka (yleensä) ajattelemme kiinteinä (ei-satunnaisina).
- Kuten usein tilastotieteessä on tapana, käytämme usein lyhennettyä merkintää $p(Y = y|\theta) = p(y|\theta)$, eli emme tee erottelua satunnaisuuttujan Y ja sen realisaation y välillä ellei erottelu ole tarpeen selkeyden vuoksi.
- Yleensä oletetaan, että havainnot y_i ovat *vaihdannaisia* (*exchangeable*), mikä tarkoittaa sitä, että havaintojen yhteisjakauma $p(y_1, \dots, y_n)$ on riippumaton havaintojen järjestyksestä. Vaihdannaisuus pätee esimerkiksi silloin kun havainnot ovat riippumattomia ja samoin jakautuneita (eli *independent and identically distributed, iid.*) kiinteällä parametrin θ arvolla.
- Merkitään reunajakauman tiheys- tai todennäköisyysfunktioita $p(\cdot)$:llä ja ehdollista tiheys- tai pistetodennäköisyysfunktioita $p(\cdot|\cdot)$:llä riippumatta siitä, onko kyseessä parametrin vai havaintojen jakauma.
- Myöhemmin usein esiintyvä merkintä \propto tarkoittaa “suoraan verrannollinen”.

2 Todennäköisyys epävarmuuden mittana

Bayesiläisessä tilastotieteessä todennäköisyyskäsitteen tulkinta on laajempi kuin klassisessa tilastotieteessä. Klassinen tilastotiede perustuu yleensä todennäköisyyden frekventistiseen tulkintaan, jonka mukaan tapahtuman todennäköisyys on tapahtuman esiintymisen suhteellinen osuus riippumattomissa toistokokeissa. Esim. kolikon heitossa kruunan todennäköisyys on $1/2$, koska kruuna esiintyy noin puolessa tapauksista pitkässä heittosarjassa. Bayesiläisessä tilastotieteessä todennäköisyyttä käytetään yleisimmin epävarmuuden mittarina ja todennäköisyydestä voidaan puhua sellaistenkin tapahtumien yhteydessä, jotka eivät ole toistettavissa. Esim. voidaan pohtia, mikä on todennäköisyys, että huomenna sataa tai että joku tietty henkilö voittaa presidentinvaalit (ks. perusteluja todennäköisyyden käytöstä epävarmuuden mittana Gelmanin kirjasta).

2.1 Todennäköisyyden aksioomat (Kolmogorovin aksioomat)

Todennäköisyyslaskennassa *tapahtumat* (*events*) E ovat perusjoukon Ω osajoukkoja. Tapahtumille E määriteltävä todennäköisyys(mitta) $P(\cdot)$ toteuttaa seuraavat aksioomat:

- $P(E) \geq 0$, missä E on jokin tapahtuma (perusjoukon osajoukko, $E \subseteq \Omega$),
- $P(\Omega) = 1$, missä Ω on perusjoukko,
- $P(\cup_i E_i) = \sum_i P(E_i)$, jokaiselle numeroituvalla joukolle pareittain erillisiä tapahtumia $\{E_i\}$.

Tapahtuman E *komplementtitapahtuma* on $\bar{E} = \Omega \setminus E$.

2.2 Ehdollinen todennäköisyys

Bayesiläinen tilastotiede perustuu ehdollisten todennäköisyyksien laskentaan. Tapahtuman E todennäköisyys ehdolla H määritellään seuraavasti:

$$P(E|H) = \frac{P(E \cap H)}{P(H)}, \quad \text{kun } P(H) > 0.$$

Tapahtuma H edustaa taustatietoa tai joskus hypoteesia. Ehdollinen todennäköisyys $P(E|H)$ mittaa tapahtuman E epävarmuutta tiedon H valossa. Sitä voidaan tulkita seuraavasti:

- $P(E|H) = 1$, jos olet varma, että E tapahtuu,
- $P(E|H) = 0$, jos olet varma, että E ei tapahdu,

- $P(E|H) = p, 0 < p < 1$, jos tapahtumaan E liittyy epävarmuutta (mutta ei välttämättä satunnaisuutta),
- $P(E|H) > P(F|H)$, jos tapahtuman E epävarmuus on pienempi kuin tapahtuman F .

Kahdella tarkastelijalla voi olla kuitenkin eri käsitys epävarmuudesta, koska heidän taustatietonsa poikkeavat (eri H). Ehdollinen todennäköisyys $P(E|H)$ muuttuu, kun informaatio H muuttuu. Siksi Bayes-teoria perustuu *subjektiivisiin todennäköisyyksiin*.

Seuraavat ominaisuudet seuraavat todennäköisyyden aksioomista ja ehdollisen todennäköisyyden määritelmästä:

- $P(E|H) \geq 0$ kaikilla E ja H kun $P(H > 0)$,
- $P(H|H) = 1$ kaikilla H jos $P(H > 0)$,
- $P(E \cup F|H) = P(E|H) + P(F|H)$, kun E ja F ovat toisensa poissulkevia,
- $P(E \cap F|H) = P(F|H)P(E|F, H)$ kaikilla E, F ja H kun $P(H > 0)$ ja $P(F \cap H) > 0$.

2.3 Vedonlyönti ja subjektiivinen todennäköisyys

Subjektiivinen todennäköisyys voidaan määrittää vedonlyöntisuhteen avulla. Veto E :stä vedonlyöntisuhteella $\omega : 1$ (*at odds*) ja panoksella (*stake*) M tarkoittaa

- Jos E ei toteudu, menetät M ;
- Jos E toteutuu, voitat ωM .

Jos uskot vahvasti E :hen, hyväksyt vedon pienellä vedonlyöntisuhteella $\omega(H)$ (ω riippuu H :sta). Olkoon $\tilde{\omega}(H)$ *reilun pelin* vedonlyöntisuhde. Reilussa pelissä

$$P(E|H) \cdot \tilde{\omega}(H)M + P(\bar{E}|H) \cdot (-M) = 0.$$

Tapahtuman E epävarmuudeksi saadaan todennäköisyys

$$P(E|H) = \frac{1}{1 + \tilde{\omega}(H)}.$$

Näin siis voit periaatteessa määrittää henkilökohtaisen subjektiivisen todennäköisyytesi sen perusteella, minkä arvioit reilun pelin vedonlyöntisuhteeksi.

Huomaa, että tilastotieteessä ja todennäköisyyslaskennassa vedonlyöntisuhde määritellään hiukan eri tavalla. Vedonlyöntisuhde E :n puolesta on todennäköisyyksien suhde $P(E|H)/P(\bar{E}|H)$, ja E :tä vastaan se on $P(\bar{E}|H)/P(E|H)$.

2.4 Todennäköisyyksien päivittyminen

Voit määritellä tapahtumaan E liittyvän epävarmuuden (ehdollisena) todennäköisyytenä $P(E|H)$. Kun hankit uutta tietoa F , voit "oppia" siitä. Nyt E :hen liittyvä epävarmuus ei välttämättä enää ole $P(E|H)$ vaan $P(E|F, H)$.

Esimerkki 2: Pinssin heitto

Heitetään pinssiä, jonka toinen puoli (kuvapuoli) on kupera. Olkoon tulos 0, jos kuvapuoli on ylöspäin, ja 1 jos se on alaspäin. Merkitään E :llä tapahtumaa "1". Ennakkokäsitys epävarmuudesta olkoon $P(E|H) = \frac{1}{2}$. Olkoon F koesarja, jossa on 10 heittoa:

$$0, 0, 0, 0, 1, 1, 0, 0, 1, 0.$$

Tämän jälkeen epävarmuus $P(E|F, H) < \frac{1}{2}$, sillä tulos "0" esiintyi koesarjassa F tulosta "1" harvemmin. Bayes-tilastotiede tarjoaa tämän tapaiseen "oppimiseen" todennäköisyyslaskentaan perustuvan kehikon.

2.5 Riippumattomuus

Tapahtumat E ja F ovat *riippumattomia* (*independent*), jos $P(E \cap F) = P(E)P(F)$. Riippumattomuus on yhtäpitävää sen kanssa, että

- $P(E|F) = P(E) = P(E|\bar{F})$, kun $P(F > 0)$, tai
- $P(F|E) = P(F)$, kun $P(E > 0)$.

Tämä voidaan tulkita myös niin, että E ja F ovat riippumattomia, jos toisen (E tai F) tunteminen ei muuta käsitystä toisen epävarmuudesta). Tapahtumasta F ei opita, kun pohditaan tapahtuman E epävarmuutta. Vastaavasti määritellään *ehdollinen riippumattomuus* (*conditional independence*): E ja F ovat riippumattomia ehdolla H , jos

$$P(E \cap F|H) = P(E|H)P(F|H),$$

kun $P(H > 0)$. Riippumattomuus ei kuitenkaan ole aivan mutkaton asia, mikä voidaan ymmärtää seuraavasta esimerkistä

Esimerkki 3: Riippumattomuus otannassa

Laatikossa on r punaista ja k valkoista palloa. Tehdään otanta takaisinsijoittaen. Tarkastellaan tapahtumia

- E = “1. pallo on punainen”, ja
- F = “2. pallo on punainen”.

Tapaus 1: Punaisten pallojen osuus $\theta = r/(r+k)$ tunnetaan. Tällöin $P(F|E) = \theta = P(F)$, joten E ja F ovat riippumattomia. Näin ajatellaan “klassisessa” tilastotieteessä.

Tapaus 2: Punaisten osuutta ei tunneta. Tällöin E ja F eivät ole riippumattomia, vaikka otannat ovat. Syy on se, että E antaa informaatiota punaisten osuudesta ja se vaikuttaa F :n ehdolliseen todennäköisyyteen. Bayes-tilastotieteessä sanotaan, että tapahtumat ovat vaihdannaisia (ajattelemalla, että θ on satunnaismuuttuja):

$$P(F|E, \theta) = P(F|\theta) = \theta.$$

Siis E ja F ovat *ehdollisesti riippumattomia* ehdolla θ . Ehdollistaminen tuntemattomien parametrien suhteen (tässä tapauksessa θ :n suhteen) on oleellinen osa (bayesiläistä) tilastollista mallinnusta.

2.6 Bayesin kaava

Olkoon H_1, H_2, \dots perusjoukon Ω ositus ts. $H_i \cap H_j = \emptyset, i \neq j$, ja $\cup_i H_i = \Omega$. Oletetaan lisäksi, että $P(H_i) > 0$ kaikilla $i = 1, 2, \dots$. Tällöin on voimassa *kokonaistodennäköisyyden* kaava:

$$P(E) = \sum_i P(E|H_i)P(H_i).$$

Tästä kaavasta ja yhtälöstä

$$P(H_i \cap E) = P(H_i)P(E|H_i)$$

seuraa *Bayesin kaava*

$$P(H_i|E) = \frac{P(H_i)P(E|H_i)}{\sum_j P(H_j)P(E|H_j)}.$$

Bayesin kaava voidaan esittää myös suppeammassa muodossa

$$P(H_i|E) \propto P(H_i)P(E|H_i),$$

josta on tiputettu pois suhteellisuuseroin $1/P(E)$. Huomaa, että tarkasteltaessa vedonlyöntisuhteita suhteellisuuseroin supistuu pois:

$$\underbrace{\frac{P(H_j|E)}{P(H_k|E)}}_{\text{posteriorivedonlyöntisuhde}} = \underbrace{\frac{P(H_j)}{P(H_k)}}_{\text{priorivedonlyöntisuhde}} \times \underbrace{\frac{P(E|H_j)}{P(E|H_k)}}_{\text{uskottavuussuhde}}$$

Yleisemmässä tapauksessa, kun E edustaa havaintoaineistoa ja H_i :t eri malleja (tai hypoteeseja), posteriorivedonlyöntisuhteita voidaan käyttää mallien vertailussa (tai hypoteesien testauksessa).

2.7 Suora ja käänteinen todennäköisyys

Ajatellaan, että aiomme tehdä n riippumatonta satunnaiskoetta ja kussakin kokeessa onnistumisen todennäköisyys on θ . Nyt onnistumisten lukumäärä Y noudattaa binomijakaumaa: $Y \sim \text{Bin}(n, \theta)$, eli $p(Y = y|\theta) = \binom{n}{y} \theta^y (1 - \theta)^{n-y}$.

- Tämä “suoran” todennäköisyyden ratkaisu tunnettiin jo 1650-luvulta lähtien (Pascal ja Fermat).
- Käänteinen ongelma: mikä on parametrin θ jakauma, kun tunnetaan satunnaiskokeiden lopputulokset y eli on havaittu $Y = y$?

Thomas Bayes (1701–1761) oletti, että parametriin θ sisältyvä epävarmuus voidaan kuvata tasajakauman avulla: $\theta \sim U(0, 1)$. Tällöin hän saattoi määrittää havaintoihin perustuvan θ :n ehdollisen *posteriorijakauman*

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y|\theta)p(\theta)}{\int p(y, \theta) d\theta} \propto p(y|\theta)p(\theta),$$

missä $p(\theta)$ on ns. priorijakauma, $p(y|\theta)$ on uskottavuus ja $p(y) = \int p(y, \theta) d\theta$ on y :n marginaalijakauma, joka ei riipu θ :sta. Huomaa merkintä \propto : Kun tarkastelun kohteena on θ (ja havainnot y on kiinnitetty), posteriorin arvo on suoraan verrannollinen uskottavuuden ja priorijakauman tuloon. Nimittäjän integraalin arvoa emme yleisessä tapauksessa osaa laskea (mutta onneksi sitä ei usein tarvita). Bayesin kaavaa käyttäen voimme esimerkkinä tilanteessa päätellä, että posteriori $p(\theta|y)$ on $\text{Beta}(y + 1, n - y + 1)$ -jakauma.

Huomaa, että Bayes-analyysissä $p(y|\theta)$ on y :n todennäköisyysjakauma ehdolla θ , kun taas SU-estimoinnissa numeerisesti sama uskottavuusfunktio $p(y|\theta) = L(\theta; y)$ ei ole todennäköisyysjakauma vaan tulkitaan θ :n funktiona ehdolla y . Tarkemmin sanottuna Bayes-analyysissä käytämme havaintojen ehdollista jakaumaa $p(Y = y|\Theta = \theta)$ (havainnot ovat satunnaismuuttujia ja parametrit ovat kiinnitettyjä), kun taas SU-estimoinnissa käytämme funktiota $L(\theta; y)$. Molemmista käytetään kuitenkin usein termiä uskottavuusfunktio.

Historiaa: Thomas Bayesin rooli Bayes-tilastotieteen “perustajana” on todellisuudessa pienempi kuin nimen perusteella voisi päätellä. Bayes ei koskaan julkaissut pohdintojaan liittyen ehdollisiin todennäköisyyksiin, vaan hänen ystävänsä Richard Price julkaisi Bayesin muistiinpanot postuumisti käytettyään ensin kaksi vuotta korjaillessaan ja editoidessaan niitä. Toisaalta Bayes (Price) ei myöskään esittänyt muita esimerkkejä tai antanut mitään erityisempiä “filosofisia” tulkintoja laskuilleen. Pierre-Simon Laplace (1749–1827) kehittäi myöhemmin Bayesista riippumattomasti samoja ajatuksia, mutta hänen jälkeensä Bayes-rintamalla oli pitkään hiljaista toisaalta laskennallisten vaikeuksien ja toisaalta filosofisten argumenttien vuoksi.

3 Bayes-päätelyn filosofiaa

Bayesin kaavaan liittyy filosofinen aspekti: Priorijakauma $p(\theta)$ on tutkijan vapaasti määriteltävissä, joka johtaa *subjektiivisen todennäköisyyden* tai *uskomusasteen* käsitteeseen (*subjective probability, degree of belief*). Tätä pidettiin (pidetään) ei-toivottavana, sillä sen ajateltiin (ajatellaan) olevan ristiriidassa luonnontieteiden faktoihin pohjautuvan objektiivisuuden kanssa.

Toisaalta tilastolliseen päätelyyn liittyy lukuisia muitakin subjektiivisia valintoja liittyen esimerkiksi uskottavuusfunktion $p(y|\theta)$ muodon määrittämiseen, havaintojen (ehdolliseen) riippumattomuuteen, selittäjien ja selitettävän muuttujan lineaariseen suhteeseen, ja yleisemmin mallin- ja muuttujienväliin, hypoteesintestaukseen, koeasetelmien suunnitteluun ja datan keräämiseen. Nämäkin valinnat pohjautuvat tyypillisesti johonkin ennakkotietoon, ts. prioritietoa käytetään myös ei-Bayes-tilastotieteen tapauksessa.

Bayes-tilastotieteen mukaisen subjektiivisen todennäköisyyskäsitteen lisäksi todennäköisyyksiä voidaan tulkita ns. frekventistisen tai klassisen ajattelutavan mukaisesti (mm. Fisher ja Pearson).

- Nyt todennäköisyydet kuvaavat tapahtumien suhteellisten osuuksien raja-arvoja (tapahtuman suhteellisen osuus riippumattomissa toistoissa).
- “Frekventisti” ajattelee kiinnostuksen kohteena olevan parametrin θ olevan tuntematon vakio, kun taas “bayesiläinen” tulkitsee θ :n satunnaismuuttujaksi. Huomaa: Puhtaan frekventistisessä mielessä esimerkiksi kysymykset “Millä todennäköisyydellä $0 < \theta < 1$?” tai “Mikä on todennäköisyys, että huomenna ei sada” eivät ole mielekkäitä.

Oleellinen ero Bayes-tilastotieteen ja klassisen tilastotieteen välillä on siinä, että Bayes-tilastotieteessä kaikki tuntemattomat suureet (parametrit, puuttuvat/tulevaisuuden havainnot, joiden välillä ei ole periaatteellista eroa) käsitellään satunnaismuuttujina ja niihin liittyvää epävarmuutta mallinnetaan todennäköisyysjakaumilla. Bayes-malli laajentaa klassista tilastollista mallia siten, että se sisältää havaintojakauman lisäksi myös parametrien jakauman. Parametrien satunnaisuudesta seuraa se, että klassisen tilastotieteen oletus havaintojen *riippumattomuudesta* korvautuu niiden *vaihdannaisuudella* (havaintojono y_1, \dots, y_n voidaan järjestää uudelleen ilman, että havaintojen yhteisjakauma muuttuu).

Tavallisesti käytettävissä ovat havainnot y jostain vastemuuttujasta Y sekä jonkin joukko kiinteitä selittäjiä x . Yleensä ei ole tarvetta käsitellä myös kiinteitä selittäjiä siten, että nekin olisivat realisaatio jostain satunnaismuuttujasta X , eli Bayes-kehikon inferenssi ei millään tavalla kohdistu näihin selittäjiin. Selittäjille ei siis esimerkiksi aseteta prioria, eikä niille ole uskottavuusfunktiota mallissa. Tähän on kuitenkin joitakin poikkeuksia, esimerkiksi usean vastemuuttujan tapauksessa, jossa toinen vasteista selittää toista, ja halutaan mallintaa vasteiden yhteisjakaumaa. Toinen tapaus nousee esiin tapauksessa, jossa jokin selittäjä sisältää puuttuvaa tietoa. Tällöin voi olla tarpeen mallintaa puuttuvuusmekanismi ja selittäjän jakauma ehdolla tämä mekanismi.

Esimerkki 4: Kolikonheitto (vaihdannaisuus)

Tarkastellaan kolikonheittoa tilanteessa, jossa emme tiedä onko kolikko symmetrinen. Olkoon Y tapahtuma “tulos on kruuna”. Klassisessa tilastotieteessä voidaan merkitä kruunan todennäköisyyttä (suhteellista frekvenssiä äärettömän pitkässä koesarjassa) parametrilla θ , ja tällöin heittojen tulokset ovat riippumattomia, kun θ tunnetaan, ja jokaisella heitolla kruunan todennäköisyys on θ .

Bayes-tilastotieteessä parametri θ ajatellaan satunnaismuuttujaksi, ja malli voidaan esittää ehdollisena todennäköisyytenä

$$P(Y|\theta) = \theta.$$

Edelleen havainnot (heittojen tulokset) ovat riippumattomia ehdolla θ . Sen sijaan ei-ehdolliset havainnot eivät ole riippumattomia vaan *vaihdannaisia*.

Heitetään nyt kolikkoa kaksi kertaa. Olkoon y_1 = “ensimmäinen heitto on kruuna” ja y_2 = “toinen heitto on kruuna”. Oletetaan edelleen $P(y_1|\theta) = \theta$. Nyt

$$P(y_1) = \int P(y_1|\theta)p(\theta) d\theta = \int \theta p(\theta) d\theta = E(\theta),$$

missä $p(\theta)$ on θ :n todennäköisyysjakauma. Lisäksi

$$P(y_2|y_1, \theta) = P(y_2|\theta) = \theta, \quad \text{ja } P(y_1 \cap y_2|\theta) = P(y_1|\theta)P(y_2|\theta) = \theta^2.$$

Jälkimmäisestä seuraa, että

$$P(y_1 \cap y_2) = E[P(y_1 \cap y_2|\theta)] = E(\theta^2).$$

Nyt voimme osoittaa, että $P(y_2|y_1) \geq P(y_2)$:

$$\begin{aligned} P(y_2|y_1) &= \frac{P(y_1 \cap y_2)}{P(y_1)} = \frac{E(\theta^2)}{E(\theta)} = \frac{\text{Var}(\theta) + E(\theta)^2}{E(\theta)} = \frac{\text{Var}(\theta)}{E(\theta)} + E(\theta) \\ &= \frac{\text{Var}(\theta)}{E(\theta)} + P(y_2) \geq P(y_2). \end{aligned}$$

Yhtäsuuruus on voimassa jos ja vain jos $\text{Var}(\theta) = 0$. Siis y_1 ja y_2 ovat riippumattomia vain, jos θ on ei-satunnainen.

Käytännössä Bayes-päätelystä on useita variaatioita, riippuen siitä kuinka tutkija suhtautuu priorijakauman määrittämiseen. Esimerkiksi ns. objektiivisessa lähestymistavassa pyritään käyttämään prioreita, jotka ovat jollain tapaa yleisesti hyväksyttävissä ja usein epäinformatiivisia (esim. edellä mainittu tasajakautunut prior). Empiirinen Bayes -menetelmä taas tarkoittaa lähestymistapaa, jossa priorijakaumat estimoidaan datasta, joka

on ristiriidassa ajatustavan kanssa, jossa priorijakauman ajatellaan kuvastavan aidosti tietoa ennen havaintoja. Tällainen tiukka ajattelu voi kuitenkin ääritapauksessa johtaa tilanteeseen, jossa priorijakauma on huonosti yhteensopiva havaintojen kanssa siten, että posteriorijakauman estimointi on käytännössä mahdotonta.

Ehkäpä yleisin variaatio on ns. pragmaattinen Bayes-päätely, jossa Bayes-menetelmiä käytetään koska niiden avulla pystytään vastaamaan paremmin käsillä olevaan ongelmaan kuin vaihtoehtoisilla tavoilla.

Esimerkki 5: Luottamusväli ja posterioriväli

Oletetaan $Y_1, \dots, Y_n \sim N(\mu, 1)$, jossa μ on tuntematon estimoitava parametri ja haluamme arvioida μ :n estimointiin liittyvää epävarmuutta jonkinlaisella epävarmuusvälillä.

Perinteinen frekventistinen tapa on tietysti laskea esimerkiksi 95 % luottamusväli $C_f = (\bar{Y} \pm 1.96/\sqrt{n})$.

- Nyt $P(C_f \ni \mu) = 0.95$ kaikille $\mu \in \mathbb{R}$.
- μ on tuntematon vakio, C_f on satunnainen väli (koska se on satunnaismuuttujien Y_1, \dots, Y_n funktio).
- Jos generoisimme datan $y_1^i, \dots, y_n^i \sim N(\mu, 1)$, $i = 1, \dots, K$, ja laskisimme jokaiselle otokselle välin c_f^i , keskimäärin 95 % kerroista c_f^i sisältäisi μ :n.
- Todennäköisyysväite koskee siis menetelmää jolla c_f^i on laskettu, ei yksittäistä väliä.

Bayes-versiossa meidän täytyy ensin määritellä priorin $p(\mu)$. Asetetaan nyt $\mu \sim U(-a, a)$ ja $a \rightarrow \infty$ (ts. $p(\mu) \propto 1$). Nyt

- $p(\mu|y_1, \dots, y_n) \propto p(y_1, \dots, y_n|\mu)p(\mu)$ (eli $N(\bar{y}, 1/n)$).
- Tämän todennäköisyysjakauman avulla voidaan määrittää posterioriväli (joskus luottoväli, credible interval) $c_b = (\bar{y} \pm 1.96/\sqrt{n})$.
- Nyt $P(\mu \in c_b|y_1, \dots, y_n) = 0.95$.

Huomaa suoraviivainen tulkinta: todennäköisyysväite koskee yksittäistä, juuri laskettua väliämme! Toisaalta huomaamme myös, että koska käytimme priorina koko reaaliarvoisen kattavaa tasajakaumaa, luottamusväli ja posterioriväli ovat numeerisesti identtisiä. Niinpä luottamusvälin tulkinta “luottamusväli sisältää todellisen odotusarvon 95 % todennäköisyydellä” ei ole tässä tapauksessa kovinkaan vaarallinen virhe. Luottamusväli ja posterioriväli eivät kuitenkaan aina ole identtisiä! Esimerkiksi valitsemalla $\mu \sim N(0, 1)$ saisimme $\mu|y_1, \dots, y_n \sim N\left(\frac{\sum_{i=1}^n y_i}{n+1}, \frac{1}{n+1}\right)$.

4 Posteriori päättelyn välineenä

Bayes-päätely pohjautuu posteriorijakauman $p(\theta|y) \propto p(\theta)p(y|\theta)$ tarkasteluun. Tästä jakaumasta voidaan koko jakauman (yleensä marginaalijakaumien) visualisoinnin lisäksi laskea erilaisia tunnuslukuja, posteriorivälejä, piste-estimaatteja ja kiinnostavia todennäköisyyksiä.

Usein suljetun muodon ratkaisua posteriorijakaumalle $p(\theta|y)$, ei ole olemassa käytännössä, vaan joudumme turvautumaan vaihtoehtoisin menetelmiin, joista tyypillisimpiä ovat erilaiset simulointimenetelmät, jotka tuottavat jollain numeerisella algoritmilla näytteitä (otoksia) tästä jakaumasta. Kontrastina frekventistinen tilastotiede hyödyntää SU-menetelmän avulla saatuja piste-estimaatteja, mikä tyypillisesti on huomattavasti helpompaa kuin posteriorijakauman simuloiminen. Toisaalta posteriorista saadun otoksen perusteella on helppoa tehdä monipuolisia päätelmiä tarkasteltavasta ilmiöstä.

Tämän kurssin puitteissa emme perehdy syvällisesti menetelmiin, joilla posteriorijakaumia simuloidaan, vaan käytämme R-paketteja, joissa tällaiset menetelmät ovat valmiiksi toteutettuna. Tarkastellaan seuraavaksi Bayes-mallin sovittamista `brms`-paketin avulla.

Esimerkki 6: Kasvien paino -aineisto

Tarkastellaan aineistoa Annette Dobsonin kirjasta *An introduction to generalized linear models* (aineisto löytyy myös `lm`-funktion dokumentaation esimerkeistä). Aineisto kerättiin jakamalla geneettisesti samanlaiset siemenet kahteen ryhmään, joista ensimmäisen ryhmän siemenet istutettiin tavalliseen maaperään, kun taas toisen ryhmän siemenet istutettiin lannoitettuun maaperään. Ennalta määrätyn ajan kuluttua kasvit kerättiin, kuivattiin ja punnittiin. Kiinnostuksen kohteena ovat ryhmien väliset erot painoissa, eli vaikuttiko ravintorikkaampi maaperä satoon.

Määritellään malli havainnolle i seuraavasti:

$$\text{sato}_i = \alpha + \beta \cdot \text{lannoite}_i + \epsilon_i, \quad \epsilon \sim N(0, \sigma^2),$$

missä $\text{lannoite}_i = 1$ jos siemen i istutettiin lannoitettuun maahan ja 0 muuten. Kyseessä on siis tavallinen lineaarinen regressiomalli.

Näkemättä aineistoa voisi olla luonnollista ajatella, että ravintorikkaamman maan vaikutus satovolyymiin olisi vähintäänkin ei-negatiivinen. Tämänkaltaisen prioritiedon käyttäminen on helppoa Bayes-tilastotieteessä esimerkiksi muuttamalla vastaavan regressiokertoimen priorijakaumaa. Yksi vaihtoehto olisi rajoittaa kyseisen regressiokertoimen priorijakauma positiiviseksi (esim. Gamma-jakauma tai katkaistu standardinormaali-jakauma). Tällaisen “kovan” rajoituksen käyttö voi olla perusteltua jos tiedämme, että parametrin negatiiviset arvot ovat teoreettisesti mahdottomia, mutta esimerkkinä tapauksessa on järkevämpää käyttää prioria joka painottaa priorikäsityksemme mukaisesti positiivisia arvoja mutta mahdollistaa kuitenkin negatiivisetkin arvot (todellisuudessa jos pyrkimyksenä olisi osoittaa lannoitteen hyödyllisyys, olisi reiluuden nimissä järkevä käyttää prioria joka on symmetrinen nollan suhteen).

Määritellään nyt priorin $\beta \sim N(1, 0.5^2)$. Nyt prioritodennäköisyys sille, että lannoite parantaa satoa (eli $\beta > 0$) on noin 98 %.

Sovitetaan tämä malli käyttäen `brms` R-pakettia. Myös mallin tasoparametrille α tulee määrittää priorin (tai määrittää priorin parametrin α ja β yhteisjakaumalle), mutta keskitytään nyt nimenomaan regressiokertoimeen β ja annetaan `brms`-paketin valita parametrin α priorin automaattisesti:

```
d1 <- data.frame(  
  sato = c(  
    4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14,  
    4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69  
  ),  
  lannoite = rep(factor(c("Ei", "Kyllä")), each = 10)  
)
```

```

library(brms)
fit_brm1 <- brm(
  sato ~ lannoite,
  data = d1,
  prior = prior(normal(1, 0.5), class = "b"),
  refresh = 0
)

```

Funktion `brm` syntaksi on lähellä tutun `lm` funktion syntaksia: ensin määritellään mallin kaava, aineisto (joka on pakollinen argumentti toisin kuin `lm` funktion kohdalla), mahdolliset priorit (palataan tähän syntaksiin myöhemmin) ja muita valinnaisia argumentteja (`refresh`-argumentti vaikuttaa laskennan aikana ruudulle tulostuvaan tekstiin). Myös sovituksen tulostus on samankaltainen `lm` ja `glm` funktioiden kanssa:

```
fit_brm1
```

```

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: sato ~ lannoite
## Data: d (Number of observations: 20)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept          4.81     0.24    4.31    5.25 1.00    2512    2134
## lannoiteKyllä       0.08     0.32   -0.50    0.74 1.00    2641    2427
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma          0.78     0.15    0.56    1.13 1.00    2130    2706
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

```

Huomaamme, että käsittelyn vaikutus on keskimäärin lievästi positiivinen (0.08, 95 % posterioriväli -0.5, 0.74), joka poikkeaa selvästi prioristamme (odotusarvo 1, 95 % prioriväli (0.02, 1.98)). Huolimatta verrattain vahvasta prioristamme (ottaen huomioon aineiston koko), näyttää siltä, että lannoitteella ei ole vaikutusta satoon.

Toistetaan nyt sama analyysi käyttäen `brm`-funktion epäinformatiivisia oletusprioreita:

```
fit_brm2 <- brm(sato ~ lannoite, data = d1, refresh = 0)
```

```
fit_brm2
```

```

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: sato ~ lannoite
## Data: d (Number of observations: 20)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept          5.03     0.25    4.55    5.52 1.00    3035    2457
## lannoiteKyllä     -0.37     0.34   -1.05    0.30 1.00    2861    2155

```

```
##
## Family Specific Parameters:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      0.75      0.13      0.54      1.06 1.00      2855      2447
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Huomaa miten tulokset muuttuivat! Sovitetaan malli myös suurimman uskottavuuden menetelmällä:

```
fit_lm <- lm(sato ~ lannoite, data = d1)
fit_lm
```

```
##
## Call:
## lm(formula = sato ~ lannoite, data = d1)
##
## Coefficients:
## (Intercept)  lannoiteKyllä
##          5.032          -0.371
```

Funktio brm käyttää oletuksena ns. “heikosti informatiivisia” prioreita, jolloin posteriorijakauma \approx uskottavuusfunktio ja niinpä eri lähestymistapojen tuottamat parameteriestimaatitkin ovat tässä tapauksessa hyvin lähellä toisiaan.

Lasketaan vielä posterioritodennäköisyydet, että maan lannoitus parantaa satoa käyttäen kahta Bayes-malliamme:

```
hypothesis(fit_brm1, "lannoiteKyllä > 0")
```

```
## Hypothesis Tests for class b:
##      Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob
## 1 (lannoiteKyllä) > 0      0.08      0.32     -0.41      0.61      1.38      0.58
##   Star
## 1
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

```
hypothesis(fit_brm2, "lannoiteKyllä > 0")
```

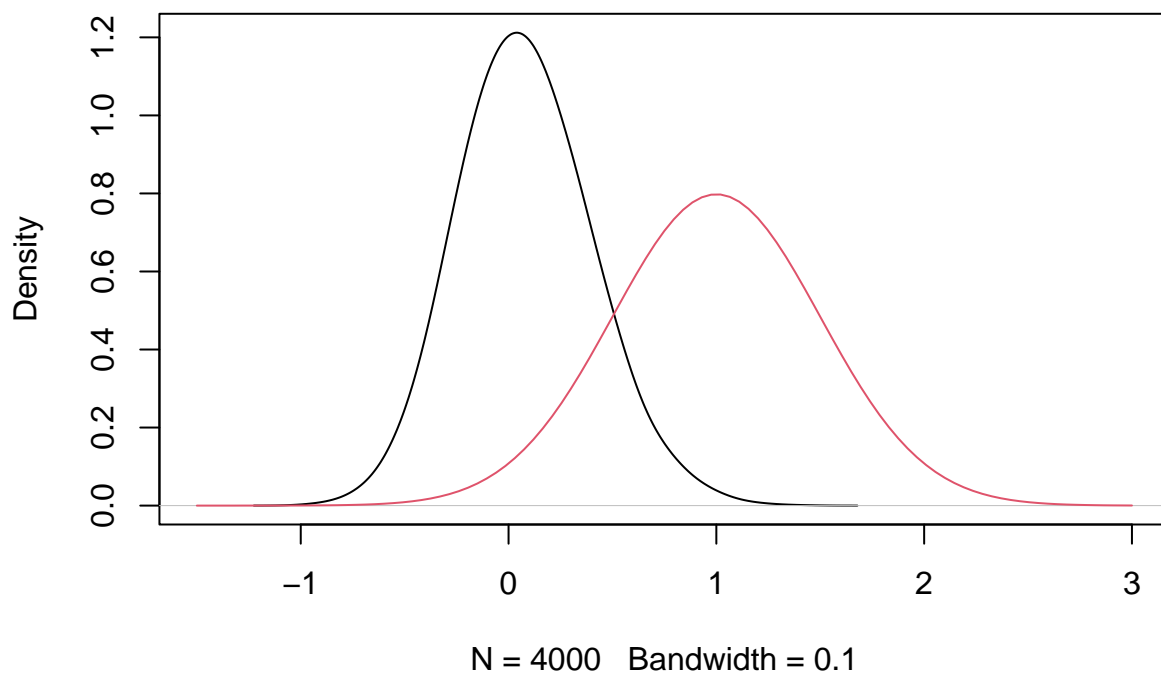
```
## Hypothesis Tests for class b:
##      Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob
## 1 (lannoiteKyllä) > 0    -0.37      0.34     -0.93      0.19      0.15      0.13
##   Star
## 1
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

Riippuen priorin valinnasta, lannoite siis kasvattaa satoa 58 % tai 13 % todennäköisyydellä (huomaa kuitenkin tämän dikotomisen tulokinnan ongelma verrattuna varsinaisten piste- ja väliestimaattien käyttöön). Tässä tapauksessa siis priorimme vaikutti voimakkaasti lopputulokseen, johtuen suhteellisen kapeasta priori-

jakaumasta verrattuna aineiston pieneen määrään. Piirretään vielä lopuksi ensimmäisestä mallista saatu vaikutuksen posteriorijakauma (musta) ja vastaava priorijakauma (punainen):

```
posteriori <- as_draws_df(fit_brm1)
plot(
  density(posteriori$b_lannoiteKyllä, bw = 0.1),
  xlim = c(-1.5, 3),
  main = "Lannoitteen vaikutuksen prior- ja posterioritiheydet"
)
curve(dnorm(x, 1, 0.5), col = 2, add = TRUE)
```

Lannoitteen vaikutuksen prior- ja posterioritiheydet



brms-paketti mahdollistaa myös suoraviivaisen laajennoksen edellä olevaan malliin. Ajatellaan, että hajontaparametrin σ arvo riippuu maaperästä. Tämä onnistuu brm-funktiolla määrittämällä kaava myös σ :lle (huomaa funktio bf):

```
# käytetään oletusprioreita
fit_brm3 <- brm(
  bf(sato ~ lannoite, sigma ~ lannoite),
  data = d1,
  refresh = 0
)
```

```
fit_brm3
```

```
## Family: gaussian
## Links: mu = identity; sigma = log
## Formula: sato ~ lannoite
##          sigma ~ lannoite
```



```

## Data: d (Number of observations: 20)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept 5.03 0.21 4.61 5.45 1.00 3791 2520
## sigma_Intercept -0.48 0.25 -0.90 0.05 1.00 3152 2859
## lannoiteKyllä -0.37 0.35 -1.06 0.29 1.00 2707 2513
## sigma_lannoiteKyllä 0.30 0.35 -0.41 0.97 1.00 2951 2898
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

```

Posteriorivälien perusteella näyttää siltä, että lannoitteella ei ole selvää vaikutusta myöskään satojen hajontaan (huomaa myös rivi "Links" joka kertoo, että σ :n jakauman kohdalla käytettiin log-linkkiä).

Piirretään vielä keskihajontaan liittyvän lannoitteen vaikutuksen kertoimen posteriorijakauma käyttäen `bayesplot`-pakettia. Huomaa, että osa `brms`-paketin funktiosta lyhentää mallin parametrien nimiä, kun taas `bayesplot` tarvitsee koko nimet, jotka saat selville `variables` funktion avulla:

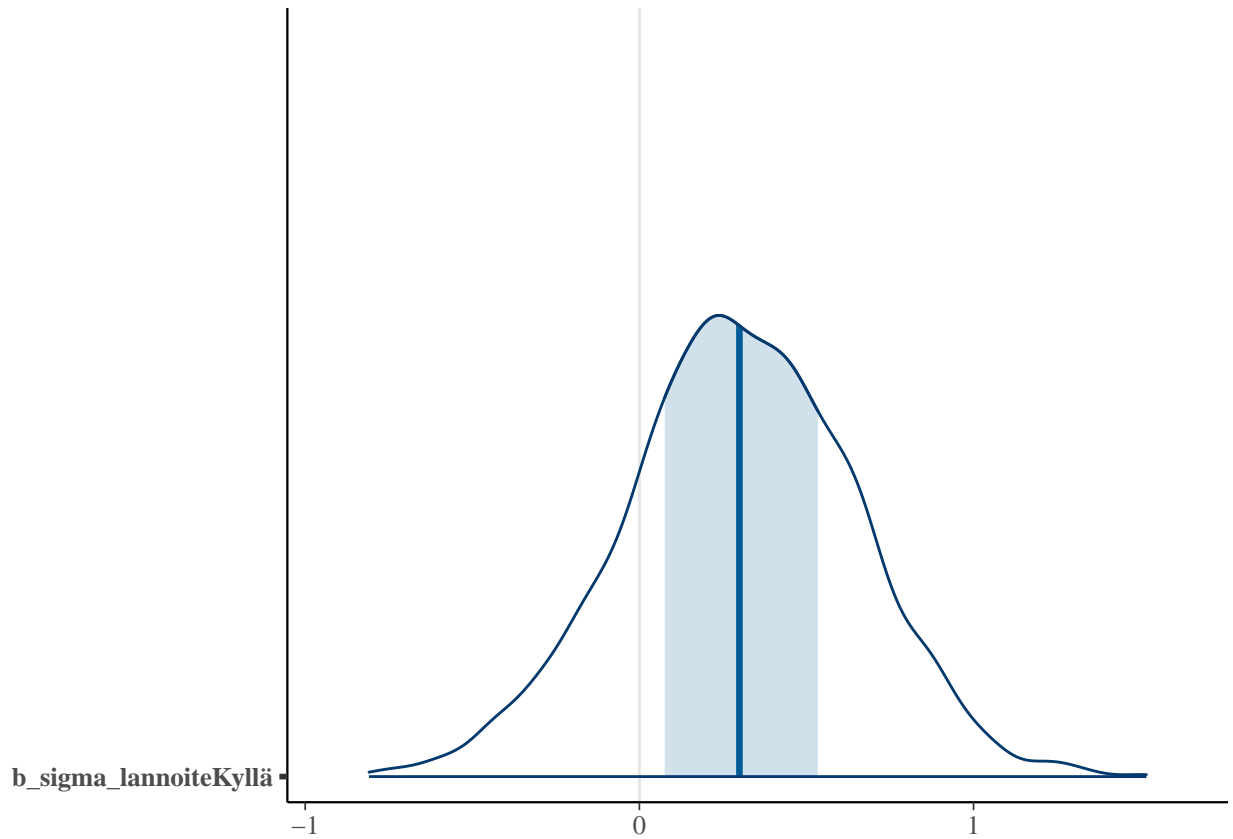
```

suppressMessages(library(bayesplot))
variables(fit_brm3)

## [1] "b_Intercept"          "b_sigma_Intercept"    "b_lannoiteKyllä"
## [4] "b_sigma_lannoiteKyllä" "lprior"                "lp_"

mcmc_areas(fit_brm3, "b_sigma_lannoiteKyllä")

```



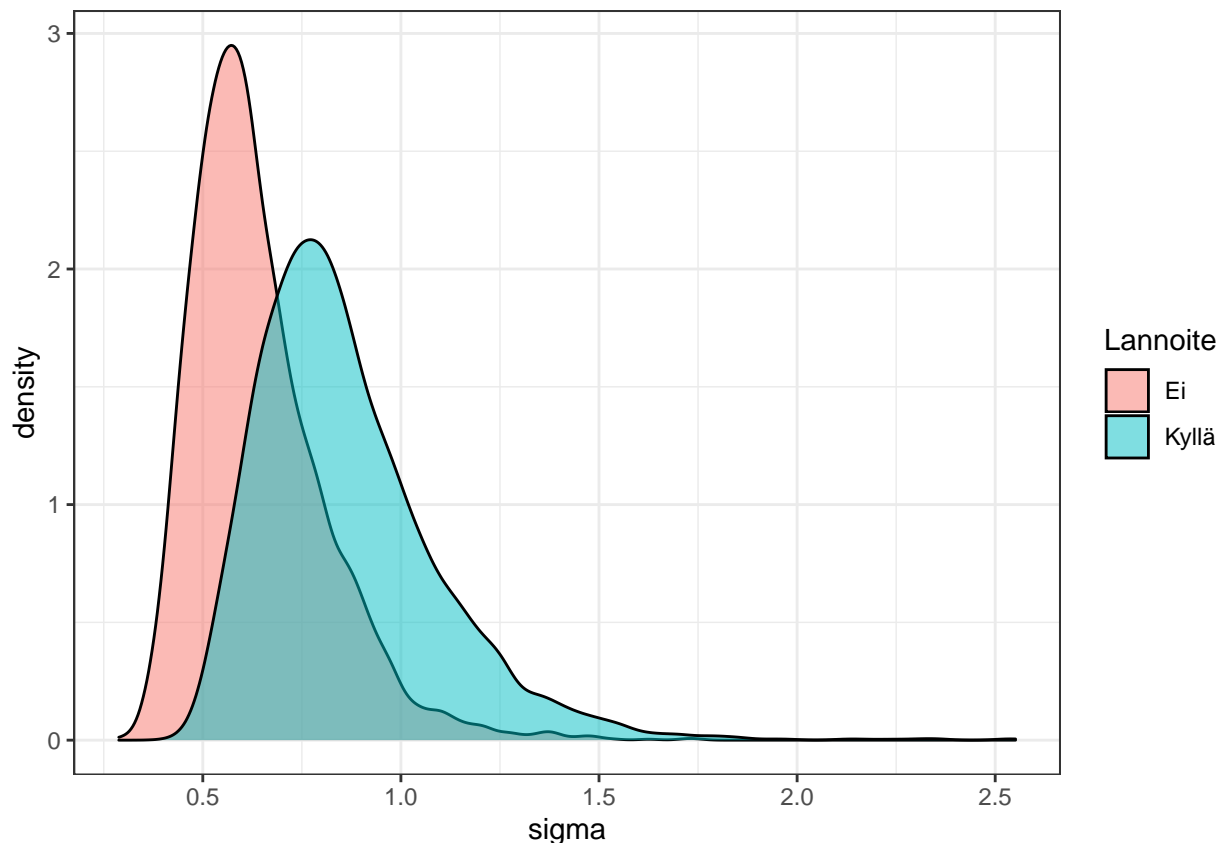
Ehkäpä mielenkiintoisempaa on kuitenkin verrata suoraan σ :n posteriorijakaumia käsittelyn suhteen:

```

samples <- as_draws_df(fit_brm3)
dens <- data.frame(
  sigma = c(
    exp(samples$b_sigma_Intercept),
    exp(samples$b_sigma_Intercept + samples$b_sigma_lannoiteKyllä)
  ),
  Lannoite = rep(c("Ei", "Kyllä"), each = nrow(samples))
)

suppressMessages(library(ggplot2))
ggplot(dens) +
  theme_bw() +
  geom_density(aes(x = sigma, fill = Lannoite), alpha = 0.5)

```



Jälkihuomaus: jos olisimme kuitenkin uskoneet vakaasti a priori siihen, että lannoitteen vaikutus on ei-negatiivinen, olisimme voineet edetä myös siten, että mallin sovittamisen jälkeen olisimme rajanneet posteriorinäytteistä pois tapaukset joissa $\beta^i < 0$ ja tarkastelleet vain loppuja näytteitä. Vastaava yleistyy tilanteeseen jossa meillä olisi useampia selittäjiä joiden pitäisi noudattaa jotain ennalta määrättyä järjestystä kuten $\beta_1 < \beta_2 < \dots < \beta_k$. Jossain tilanteessa tällainen jälkikäteen tapahtuva “hylkäysotanta” voi olla helpompi toteuttaa kuin vastaavien rajoitusten määrittäminen posteriorinäytteistyksen aikana (toisinaan taas tämä on erittäin tehontonta, jos data ei näytä noudattavan kyseisiä rajoituksia).

4.1 Data-analyysin vaiheet Bayes-näkökulmasta

Nykyaikaisen Bayes-päätelyn voidaan karkeasti ajatella etenevän seuraavalla tavalla:

- Tarkastellaan käsillä olevaa aineistoa alustavasti esimerkiksi visualisoinnein.
- Kuten suurimman uskottavuuden estimoinnin tapauksessa, valitaan todennäköisyysmalli (uskottavuus-funktio) havainnoille y ehdolla parametrit θ , $p(y|\theta)$.
- Määritellään priorijakauma $p(\theta)$.
- Simuloidaan keinotekoista dataa jakaumien $p(\theta)$ ja $p(y|\theta)$ perusteella ja arvioidaan mallin mielekkyyttä mm. visualisoimalla simuloitua dataa ja vertaamalla tätä ennakkotietoihimme. Muutetaan tarvittaessa priorijakaumia.
- Lasketaan ja tulkitaan posteriorijakauma $p(\theta|y)$.
- Arvioidaan mallin yhteensopivuutta havaintoaineiston kanssa ja posteriorijakauman perusteella tehtävien päätelmien järjestyks.
- Tutkitaan, miten herkkiä tulokset ovat mallin oletuksille ja vertaillaan tuloksia vaihtoehtoisin malleihin esimerkiksi ristiinvalidoinnilla.
- Tarvittaessa toistetaan edellä mainitut askeleet.

4.1.1 Piste-estimaatteja

Posteriorijakaumasta voidaan laskea erilaisia piste-estimaatteja:

- Posteriorimoodi $\text{argmax}_{\theta} p(\theta|y)$. MAP eli maximum a posteriori.
- Posteriorimediaani, jakauman $p(\theta|y)$ mediaani.
- Posterioriodotusarvo $E(\theta|y)$.
- Posteriorikeskihajonta $SD(\theta|y)$.

Esimerkki 7: Piste-estimaatteja

Palataan lannoite-esimerkkiin ja lasketaan “käsini” erilaisia piste-estimaatteja:

```
otos <- as_draws_df(fit_brm1)
mean(otos$b_lannoiteKyllä) # E(beta | y)

## [1] 0.07868443

sd(otos$b_lannoiteKyllä) # SD(beta | y)

## [1] 0.3152322

# betan posteriorimoodi
f <- density(otos$b_lannoiteKyllä) # approksimoidaan tiheysfunktiota
f$x[which.max(f$y)] # etsitään moodi

## [1] 0.02853931
```

Kuten pitäisikin, tulokset ovat yhtäpitäviä brms paketin tulostuksen kanssa:

```
fit_brm1

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: sato ~ lannoite
## Data: d (Number of observations: 20)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept 4.81 0.24 4.31 5.25 1.00 2512 2134
## lannoiteKyllä 0.08 0.32 -0.50 0.74 1.00 2641 2427
##
## Family Specific Parameters:
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma 0.78 0.15 0.56 1.13 1.00 2130 2706
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

4.1.2 Posterioriväli

Yleisesti posterioriväli (tai -alue) on sellainen parametriavaruuden osajoukko (väli) C_{ϵ} , jolle $P(\theta \in C_{\epsilon}|y) = 1 - \epsilon$, missä ϵ on kiinnitetty (esim. 0.05) (y on data).

Tämä mahdollistaa useita tapoja määrittää C_{ϵ} kun kyseessä on posterioriväli:

1. Molempiin jakauman häntiin jää yhtä paljon todennäköisyysmassaa, eli $C_\epsilon = [a, b]$ s.e. $P(\theta < a|y) = P(\theta > b|y) = \frac{\epsilon}{2}$ (posteriorin kvantiiliväli, quantile interval, joskus central posterior interval). Tämä on tyypillisin vaihtoehto.
2. Valitaan C_ϵ siten, että välin pituus on mahdollisimman pieni (SPI, shortest posterior interval).
3. Suurimman posterioritiheyden väli (highest density interval HDI): $C_\epsilon = \{\theta : p(\theta|y) \geq k\}$, missä k riippuu ϵ :sta.

HDI on luonnollinen vaihtoehto moniulotteisessa tapauksessa (highest density region, HDR), mutta huomaa, että HDI voi itseasiassa koostua useasta osavälistä monihuippuisen jakauman tilanteessa. Yksihuippuisessa tilanteessa SPI ja HDI ovat yhtäpitävät. Käytännössä posteriorivälejä lasketaan kuitenkin usein yksiulotteisen $p(\theta_i|y)$:n marginaalijakauman empiirisinä kvantiiliväleinä posteriorinäytteiden perusteella (ylläolevan listan ensimmäinen vaihtoehto).

Esimerkki 8: Posteriorivälejä

Lasketaan 95 % posterioriväli käyttäen jälleen lannoite-esimerkkiä käyttäen:

```
otos <- as_draws_df(fit_brm1)
quantile(otos$b_lannoiteKyllä, prob = c(0.025, 0.975))
```

```
##      2.5%      97.5%
## -0.5017509  0.7383396
```

HDI-välin laskeminen on hankalampaa (multimodaalisessa tapauksessa), joten toistetaan tidybayes R-paketin esimerkki multimodaalisen (monihuippuisen) posteriorin kuvaamiseen:

```
suppressMessages(library(tidybayes))
suppressMessages(library(dplyr))
set.seed(123)
multimodal_draws <- data.frame(
  x = c(rnorm(5000, 0, 1), rnorm(2500, 4, 1))
)
multimodal_draws |>
  mode_hdi(x, .width = .80)
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded
```

```
## # A tibble: 2 x 6
##       x .lower .upper .width .point .interval
##   <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 -0.0153 -1.52  1.60  0.8 mode  hdi
## 2 -0.0153  3.04  5.00  0.8 mode  hdi
```

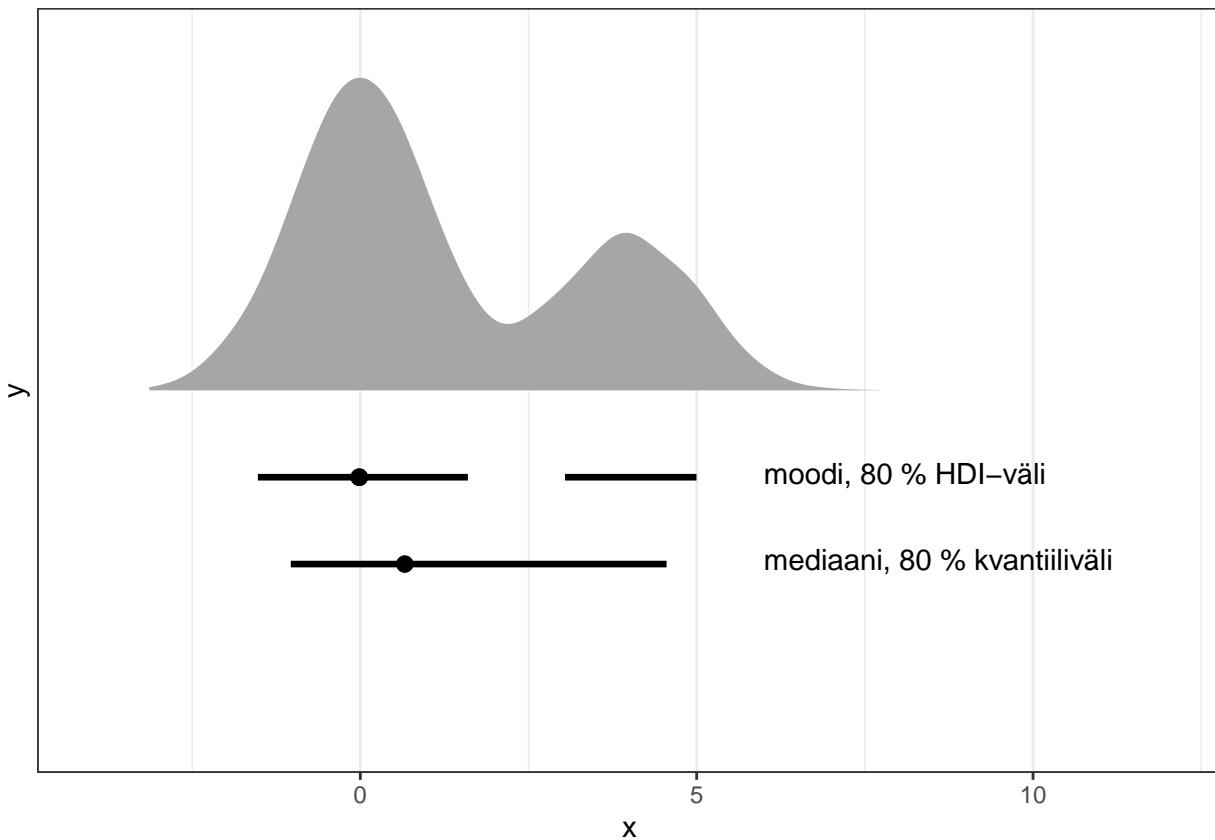
Huomaa kaksi väliä, jotka yhdessä muodostavat 80 % HDI:n. Piirretään posteriorijakaumasta myös kuva:

```
library(ggplot2)
multimodal_draws |>
  ggplot(aes(x = x)) +
  stat_slab(aes(y = 0)) +
  stat_pointinterval(aes(y = -0.5), point_interval = median_qi, .width = 0.8) +
  annotate(
    "text",
    label = "mediaani, 80 % kvantiiliväli",
    x = 6,
    y = -0.5,
    hjust = 0,
    vjust = 0.3
```

```

) +
stat_pointinterval(aes(y = -0.25), point_interval = mode_hdi, .width = 0.8) +
annotate(
  "text",
  label = "moodi, 80 % HDI-väli",
  x = 6,
  y = -0.25,
  hjust = 0,
  vjust = 0.3
) +
xlim(-4, 12) +
scale_y_continuous(breaks = NULL) +
theme_bw()

```



4.1.3 Mielivaltaiset todennäköisyydet ja odotusarvot

Yleisesti voimme laskea mielivaltaisia odotusarvoja posteriorin suhteen:

$$E_{\theta|y}[g(\theta)] = \int g(\theta)p(\theta|y) d\theta,$$

missä funktio $g(\theta)$ valitaan tapauskohtaisesti (esim. $g(\theta) = \theta$ antaa meille θ :n posterioriodotusarvon).

Voimme myös laskea helposti erilaisten hypoteesien todennäköisyyksiä, esim. $P(\theta > 0|y) = \int_0^\infty p(\theta|y) d\theta$. Huomaa, että tämä on oikea todennäköisyys sille, että θ on positiivinen, toisin kuin klassinen p -arvo joka ei ole nollahypoteesin todennäköisyys.

Esimerkki 9: Mielivaltaisen todennäköisyyden laskeminen

Kuvitellaan seuraavanlainen aineisto:

```
set.seed(1)
n <- 100
x <- rnorm(n)
z <- rep(factor(c("A", "B")), each = n / 2)
q <- plogis(-0.2 + x + 0.5 * (z == "B") + 0.2 * x * (z == "B"))
y2 <- rbinom(n, size = 1, prob = q)
d2 <- data.frame(y = y2, x = x, z = z)
```

Meillä on siis kaksi ryhmää (muuttuja z , ryhmät A ja B), jatkuva kovariaatti x , sekä Bernoulli-jakautunut vaste y , jossa “onnistumisen” todennäköisyys $q = P(y = 1|\theta)$ riippuu muuttujasta x ja ryhmästä z seuraavasti:

$$\text{logit}P(y_i = 1|\theta) = \beta_0 + \beta_1 x_i + \beta_2 I(z_i = B) + \beta_3 x_i I(z_i = B),$$

missä $I(\cdot)$ on indikaattorifunktio, joka saa arvon 1 kun funktion argumentin ehto on tosi ja arvon 0 muuten. Ajatellaan, että haluamme tutkia eroaako q ryhmien A ja B välillä. Sovitetaan Bayes-malli:

```
fit_q <- brm(y ~ x * z, family = bernoulli, data = d2, refresh = 0)
```

```
fit_q
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: y ~ x * z
## Data: d2 (Number of observations: 100)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.14     0.29   -0.74    0.42 1.00    4114    2967
## x              0.44     0.37   -0.25    1.19 1.00    3162    2608
## zB             0.83     0.47   -0.07    1.80 1.00    3276    2373
## x:zB          1.17     0.64   -0.01    2.48 1.00    3038    2922
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Huomaa argumentti `family`, joka määrittää havaintojakauman $p(y|\theta)$ muodon. Kertoimet ovat lineaarisen prediktorin tasolla ja mukana on vielä interaktiokin, joten tulkinta on hankalaa. Lasketaan nyt ennusteet q :lle eri ryhmissä ja marginalisoidaan yli x :n arvojen:

```
# lasketaan ennusteet kaikilla datassa olevilla x:n arvoilla kun z A ja B
pred_A <- posterior_epred(fit_q, data.frame(x = d2$x, z = "A"))
pred_B <- posterior_epred(fit_q, data.frame(x = d2$x, z = "B"))
dim(pred_A) # rivit vastaavat posteriorinäytteitä, sarakkeet x:n arvoja
```

```
## [1] 4000 100
```

```
# keskiarvoistetaan yli x-muuttujan ja kootaan tulokset yhteen:
results <- data.frame(
  q = c(rowMeans(pred_A), rowMeans(pred_B)),
  z = rep(c("A", "B"), each = nrow(pred_A))
)
```

```

results |>
  group_by(z) |>
  summarise(
    mean = mean(q),
    "2.5%" = quantile(q, 0.025),
    "97.5%" = quantile(q, 0.975)
  )

```

```

## # A tibble: 2 x 4
##   z      mean `2.5%` `97.5%`
##   <chr> <dbl> <dbl> <dbl>
## 1 A     0.478 0.347 0.608
## 2 B     0.649 0.531 0.758

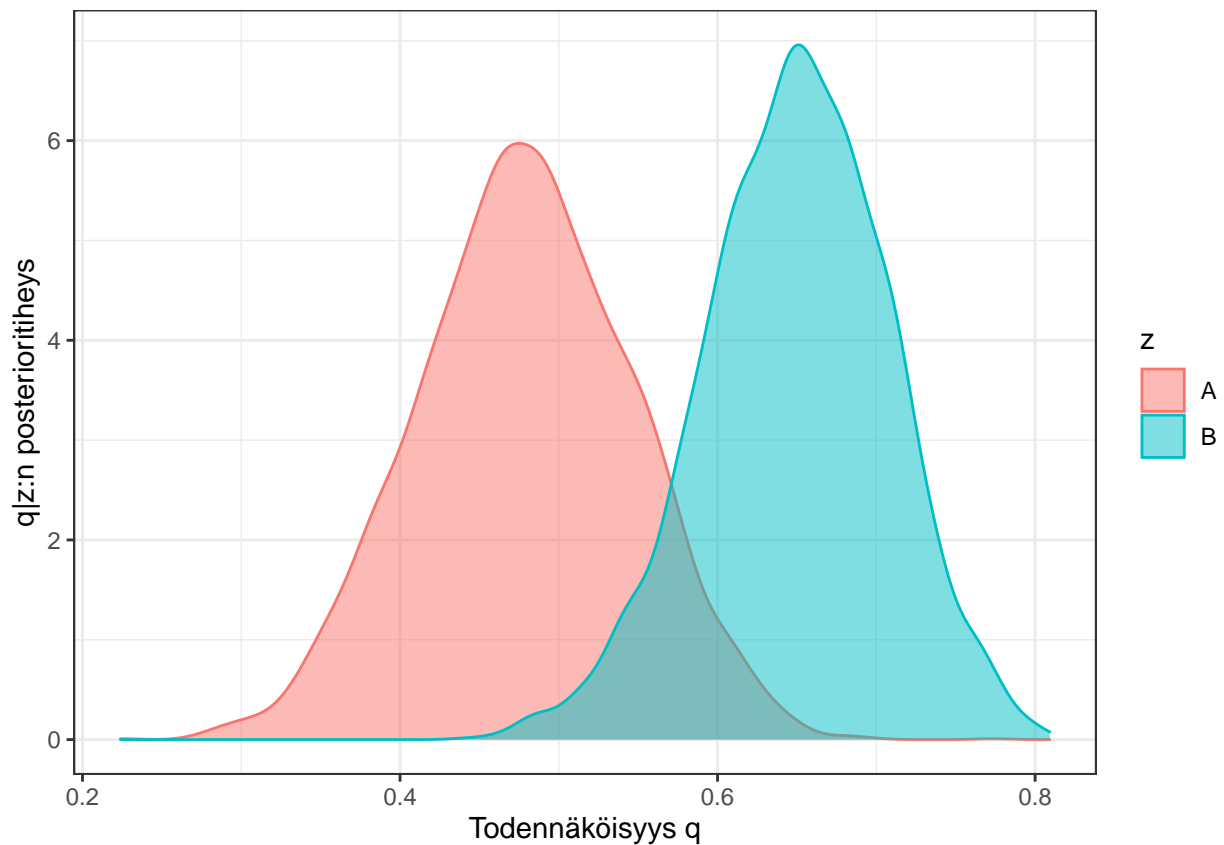
```

Piirretään q :n posteriorijakaumat ryhmittäin:

```

results |>
  ggplot(aes(x = q, fill = z, colour = z)) +
  geom_density(alpha = 0.5) +
  ylab("q|z:n posterioritiheys") +
  xlab("Todennäköisyys q") +
  theme_bw()

```



Olkoot q_A ja q_B ryhmäkohtaiset todennäköisyydet. Lasketaan vielä ryhmien todennäköisyyksien erotusten $q_B - q_A$ posteriorikeskiarvo ja 95 % posterioriväli, sekä todennäköisyys että $q_B > q_A$:

```

prob_table <- results |>
  summarise(

```



```

mean_diff = mean(q[z == "B"] - q[z == "A"]),
lwr_diff = quantile(q[z == "B"] - q[z == "A"], 0.025),
upr_diff = quantile(q[z == "B"] - q[z == "A"], 0.975),
mean_gt = mean(q[z == "B"] > q[z == "A"])
)
prob_table

```

```

## mean_diff lwr_diff upr_diff mean_gt
## 1 0.1706153 -0.004546387 0.3446934 0.972

```

Vaikka posteriorivälit ja -jakaumat menevät reilusti päällekkäin, ryhmässä B on onnistumisen todennäköisyys on kuitenkin keskimäärin 17 prosenttiyksikköä suurempi, ja olemme 97 % todennäköisyydellä varmoja siitä, että $q_B > q_A$.

4.1.4 Hyödyllisiä brms-funktioita

Listataan vielä hyödyllisiä funktiota posteriorinäytteiden käsittelyyn:

- `as_draws_df`: Palauttaa estimoidun mallin parametrien posteriorinäytteet $\theta^1, \dots, \theta^N$ datakehikkona, jossa N on näytteiden kokonaismäärä (kaikkien ketjujen kaikki lämmittelyjakson jälkeiset näytteet).
- `posterior_epred`: Palauttaa näytteitä havaintojen *odotusarvon* posterioriennustejakaumasta.
- `posterior_predict`: Palauttaa näytteitä havaintojen posterioriennustejakaumasta.

Kaksi jälkimmäistä funktiota ovat “apufunktioita”, voisimme laskea posterioriennusteita myös “käsin” käyttäen `as_draws_df` funktiosta saatavia näytteitä. Esimerkiksi tilanteessa, jossa $y \sim N(\alpha + \beta x, \sigma^2)$, $\theta = (\mu, \sigma)$, saamme näytteitä posterioriodotusarvon jakaumasta ehdolla x laskemalla $\tilde{y}_i = \alpha_i + \beta_i x$, $i = 1, \dots, n$. Vastaavasti havaintotason ennusteille $\tilde{y}_i = \alpha_i + \beta_i x + \epsilon_i$ jossa ϵ_i simuloidaan jakaumasta $N(0, \sigma_i^2)$.

4.1.5 Moniparametrisista malleista

Usein Bayes-mallimme sisältävät useita parametreja, joista osa ovat ns. *kiusaparametreja*. Esimerkiksi normaalijakauman tilanteessa olemme tyypillisesti kiinnostuneita odotusarvosta μ , keskihajonnan σ estimoinnin ollessa (välttämätön) sivuseikka. Voimme nyt jakaa parametrivektorin θ kahteen osaan, $\theta = (\theta_1, \theta_2)$, jossa θ_1 sisältää kiinnostavat parametrit (μ) ja θ_2 kiusaparametrit (σ). Nyt voimme kirjoittaa posteriorin muodossa

$$p(\theta|y) = p(\theta_1, \theta_2|y) \propto p(y|\theta_1, \theta_2)p(\theta_1, \theta_2).$$

Voimme nyt laskea kiinnostuksen kohteena olevan *marginaaliposteriorin* $p(\theta_1|y)$ integroimalla yli θ_2 :n:

$$p(\theta_1|y) = \int p(\theta_1, \theta_2|y) d\theta_2.$$

Jossain tapauksissa voimme laskea kyseisen posteriorin analyttisesti, usein käyttämällä hyväksi vaihtoehtoista muotoa

$$p(\theta_1|y) = \int p(\theta_1, \theta_2|y) d\theta_2 = \int p(\theta_1|\theta_2, y)p(\theta_2|y) d\theta_2.$$

Yleensä emme pyri laskemaan näitä(kään) integraaleja analyttisesti, vaan turvaudumme simulointiin: Olkoot θ_1 skalaari ja θ_2 k -ulotteinen. Olettaen, että meillä on m kappaletta näytteitä $\theta = (\theta_1, \theta_2)$:n yhteisposteriorista $p(\theta_1, \theta_2|y)$ $m \times (1 + k)$ matriisissa, saamme näytteitä marginaaliposteriorista $p(\theta_1|y)$ “pudottamalla” θ_2 :t pois, ts. poimimalla ensimmäisen sarakkeen posteriorinäytematriisista.

Usein teemme päätelmiä, ja varsinkin visualisointeja θ :n marginaaliposteriorien pohjalta, vaikka näin jätämmekin huomioimatta θ :n komponenttien välisiä posterioririippuvuuksia, koska esim. moniulotteisen yhteisposteriorin kuvaaminen on hankalaa (mahdotonta). Parittaisia posteriorijakauman visualisointeja (esim. hajontakuviot) voimme vielä tehdä helposti, ja ne voivatkin olla hyödyllisiä tutkittaessa mallin hyvyttä ja simulointiin liittyviä konvergenssiongelmiä. Huomaa kuitenkin, että esimerkiksi laskettaessa posterioriodotusarvoja $\mu^i = \alpha^i + \beta^i x$ käytämme jokaisen näytteen i kohdalla saman iteraation i näytteitä molemmille muuttujille α_i ja β_i , jolloin huomioimme näiden muuttujien mahdollisen korrelaation. Tällä on vaikutusta:

```

set.seed(1)
m <- 1e4
alpha <- rnorm(m) # p(alpha | y) (kuvitellaan näin)
# luodaan kuvitteellinen posterioririippuvuus parametrien välille:
beta <- rnorm(m, 2*alpha) # p(beta | alpha, y)
mu1 <- alpha + beta # mu_i = alpha_i + beta_i
mu2 <- alpha + sample(beta) # sekoitetaan betat, mu_i = alpha_i + beta_j
sd(mu1)

## [1] 3.199154

sd(mu2)

## [1] 2.488441

# epälineaarinen tapaus:
mu1_nl <- exp(alpha + beta) # mu_i = alpha_i + beta_i
mu2_nl <- exp(alpha + sample(beta)) # sekoitetaan betat, mu_i = alpha_i + beta_j
mean(mu1_nl)

## [1] 130.7941

mean(mu2_nl)

## [1] 21.49148

sd(mu1_nl)

## [1] 3506.396

sd(mu2_nl)

## [1] 252.9189

```

4.2 Priorijakauma

Ennen posteriorijakauman laskemista meidän täytyy määritellä havaintojen jakauma $p(y|\theta)$ ja tähän jakau-
maan liittyvien parametrien θ priorijakauma $p(\theta)$. Kuten SU-estimoinnissa, jakauma $p(y|\theta)$ saadaan tyypillis-
esti joko teoreettisin perustein tai datan ominaisuuksia tutkimalla. Huomaa, että usein $p(y|\theta)$ voidaan jakaa
esimerkiksi muotoon $p(y|z, \theta_1)p(z|\theta_2)$, eli mallimme sisältää jonkinlaisen hierarkian ja vastaavasti priorikin
on usein esim. muotoa $p(\theta) = p(\theta_1)p(\theta_2|\theta_1)$.

Priorijakauman määrittämiseen on useita lähestymistapoja, jotka voidaan jaotella eri tavoin. Eräs tapa on
tulkita priorijakauma samaan tapaan kuin havaintomalli $p(y|\theta)$, ts. ajatteleme, että on olemassa jonkin-
lainen meistä riippumaton prosessi $p(\theta)$ joka generoi parametreja θ . Voimme myös ajatella priorijakaumaa
subjektiivisena esityksenä kaikelle sille tiedolle mitä meillä on θ :sta. Joka tapauksessa on kuitenkin oleellista
perustella valittu priorijakauma, erityisesti jos priorin on hyvin *informatiivinen*.

Perinteisesti priorin pyrittiin määrittämään “objektiivisesti” tai “epäinformatiiviseksi” siten, että priorin
vaikutus posterioriin olisi verrattain pieni, sillä subjektiivisten uskomusten käyttö (ilman perusteluja) analyysin
perustana ajateltiin (ajatellaan) olevan ristiriidassa tieteen objektiivisuuden kanssa. Epäinformatiivinen
priorijakauma (*noninformative, vague, reference, diffuse, flat*) on priorin jota uskottavuusfunktio dominoi siten,
että priorin arvo ei juurikaan vaihdu “järkevästi” (uskottavuusfunktion kanssa yhteensopivilla) parametrien
arvoilla, eikä myöskään tuota suuria arvoja tämän alueen ulkopuolella. Esimerkkejä tällaisista prioreista ovat
mm. tasajakauma sekä normaalijakauma suurella varianssilla.

Huomaa, että esimerkiksi tasainen priorin $p(\theta) \propto 1$ on *epäaito (improper)*, ts. $p(\theta)$ ei integroidu ykköseksi
ja täten se ei ole oikea jakauma. Tämä ei kuitenkaan yleensä haittaa käytännössä sillä voimme silti saada
johdettua aidon posteriorijakauman, ellei uskottavuusfunktio ole tasainen.

Liian epäinformatiivisten priorien käyttö voi kuitenkin johtaa ongelmiin, sillä tarvitsemme usein priorijakaumia auttamaan monimutkaisten hierarkkisten Bayes-mallien parametrien identifioitumisessa ja parantamaan estimoinnin laskennallista tehokkuutta. Nykyään varsinaisten epäinformatiivisten prioreiden sijaan suositaankin usein ns. *heikosti informatiivisia* prioreita. Usein pystymme arviomaan teoreettisin perustein esimerkiksi joidenkin muuttujien suuruusluokan tai sen etumerkin ja tällaisen prioritiedon käyttö voi helpottaa mallin numeerista estimointia ilman, että posteriorijohtopäätöksistä tulisi erityisen subjektiivisia. Toisaalta järkevän, “yleisesti tunnetun” prioritiedon käyttämättä jättämisenkin voi ajatella olevan yhtä “väärin” kuin “liian informatiivisen” priorin käyttö. Heikosti informatiivisen priorin ajatuksena on olla kompromissi näiden eri lähestymistapojen välillä: Määritellään priorin siten, että se sisältää tarkoituksella vähemmän informaatiota kuin meillä on käytössä, mutta kuitenkin tarpeeksi, jotta posteriorijakauma on järkvyä myös tilanteessa, jossa aineisto ei dominoi täysin posterioria.

Huomaa, että emme useinkaan sisällytä kaikkea ennakkotietoa tutkivasta ilmiöstä priorijakaumaan. Osa tästä informaatiosta liittyy enemmän havaintomalliin $p(y|\theta)$, voimme jättää jotain pois, jotta malli olisi yksinkertaisemmin estimoitavissa tai tulkittavissa, tai osa informaatiosta voi olla liian vaikeaa muotoilla todennäköisyysjakauksen avulla. Käyttämätöntä informaatiota voimme kuitenkin hyödyntää myöhemmin mallin hyvyttä tutkiessa. Lisäksi joissain tapauksissa haluamme analysiimme olevan puolueeton ja jätämme siksi prioritietoa käyttämättä. Esimerkiksi jonkin lääkkeen tehoa tutkiessa on järkvyä asettaa efektille symmetrinen priorin (esim. $N(0, 1)$) sen sijaan, että käyttäisimme prioria, joka painottaa positiivista vaikutusta (joka todennäköisesti aiempien kokeiden perusteella olisi aivan perusteltu priorin).

Sopivien priorien määrittelyyn ei ole yleispäteviä ohjeita vaan tämä, kuten niin moni muukin asia tilastollisessa mallinnuksessa, riippuu tilanteesta (mm. tutkimuskysymyksestä). Esimerkiksi heikosti informatiivisten priorien käytössä voidaan lähteä kahdesta suunnasta liikkeelle: Määritellään ensin hyvin epäinformatiiviset priorit, joita “tiukennetaan” (estimoimalla malli uudestaan uusilla informatiivisemmilla prioreilla) tarpeen mukaan sen verran, että tulokset ovat järkvyiä ja estimointi onnistuu (numeerisesti). Tai vaihtoehtoisesti lähdetään liikkeelle tiukoista, informatiivista prioreista, ja “löystytetään” näitä niin, että tulokset ottavat huomioon epävarmuuden, joka liittyy aineistoon ja muuhun informaatioon, jonka perusteella tiukat priorit määriteltiin. Huomaa, että molemmissa tapauksissa prioreita muokataan ensimmäisten mallien tulosten pohjalta. Tämän voi ajatella olevan tiukan Bayes-tulkinnan vastaista jos priorin ajatellaan tarkoittavan informaatiota ennen datan näkemistä. Tämäntyyppinen priorien “viilaus” on kuitenkin usein välttämätöntä, sillä varsinkin monimutkaisissa (hierarkkisissa) malleissa on usein parametreja, joiden arvoista on hyvin vaikea sanoa mitään a priori. Tiukan tulkinnan mukaan voisi käydä niin, että malli ei estimoituisi lainkaan, tai huonosti valittu priorin rajoittaisi posteriorijakaumasta esim. negatiivisen osan pois, jolloin tämän lähestymistavan mukaan meillä ei olisi muuta vaihtoehtoa kuin hankkia uusi aineisto ja käyttää tästä epäonnistuneesta analyysistä saatuja tietoja uutena priorina. Huomaa kuitenkin toisen ääripään ongelmat: Jos haluamme estimoida muuttujan X odotusarvon μ pohjautuen dataan x_1, \dots, x_n , ja asetamme priorin $\mu \sim N(\bar{x}, s^2/n)$, käytämme dataa kahdesti ja aliarvioimme μ :n posterioriepävarmuutta.

Andrew Gelman vastaa [blogissaan](#) kysymykseen “Can you change your Bayesian prior” näin:

As we discuss in BDA and elsewhere, one can think of one’s statistical model, at any point in time, as a placeholder, an approximation or compromise given constraints of computation and of expressing one’s model. In many settings the following iterative procedure makes sense:

1. Set up a placeholder model (that is, whatever statistical model you might fit).
2. Perform inference (no problem, now that we have Stan!).
3. Look at the posterior inferences. If some of the inferences don’t “make sense,” this implies that you have additional information that has not been incorporated into the model. Improve the model and return to step 1.

If you look carefully you’ll see I said nothing about “prior,” just “model.” So my answer to your question is: Yes, you can change your statistical model. Nothing special about the “prior.” You can change your “likelihood” too.

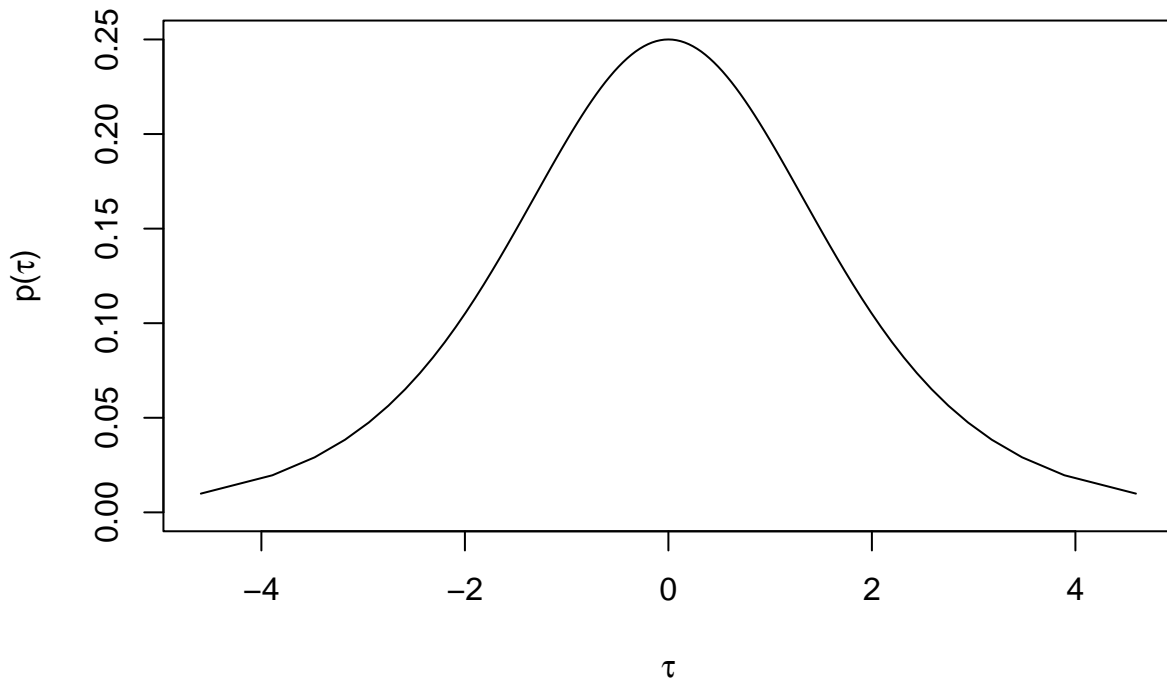
Esimerkki 10: Tasainen prior

Priorijakauman epäinformatiivisuus ei välttämättä ole invariantti parametrisoinnin suhteen. Olkoon $X \sim \text{Bernoulli}(\theta)$. Ajatellaan, että meillä ei ole mitään tietoa onnistumistodennäköisyydestä θ , joten asetamme $p(\theta) \propto 1$. Nyt kuitenkin jos uudelleenparametrisoimme mallin siten, että kiinnostuksen kohteena on parametri $\tau = \log(\theta/(1 - \theta))$, niin saamme

$$p(\tau) \propto \frac{e^\tau}{(1 + e^\tau)^2},$$

joka ei ole epäinformatiivinen τ :n suhteen:

```
theta <- seq(0, 1, by = 0.01)
tau <- log(theta / (1 - theta))
plot(
  x = tau,
  y = exp(tau) / (1 + exp(tau))^2,
  xlab = expression(tau),
  ylab = expression("p(" * tau * ")"),
  type = "l"
)
```



Tässä on ristiriita, sillä jos emme tiedä mitään θ :sta, ei meidän pitäisi tietää mitään myöskään τ :sta! Yleisesti ottaen tasaiset priorit eivät siis ole invariantteja muunnosten suhteen.

4.2.1 Jeffreysin prior

Palataan ongelmaan, jossa ennakkotietoa ei ole. Seuraavia ratkaisuja on tarjolla (esimerkiksi):

- Kun $y|\theta \sim N(\theta, v)$, missä v on tunnettu, käytetään prioria $\theta \sim N(0, w)$, missä w on suuri.

- Kun $y|\theta \sim N(\theta, v)$, missä v on tunnettu, käytetään prioria $p(\theta) \propto \kappa$.
- Kun $y|\phi \sim N(\mu, \phi)$, missä μ on tunnettu, käytetään prioria $p(\phi) \propto 1/\phi$.
- Kun $y|\theta \sim \text{Bin}(1, \theta)$, käytetään prioria $\theta \sim \text{Beta}(1, 1)$ (tasajakauma).

Tasajakauman käyttö priorijakaumana ei ole ongelmatonta, kuten tasan priorin tapauksessa huomasimme.

Sir Harold Jeffreys (1891–1989) oli brittiläinen matemaatikko, tilastotieteilijä, geofyysikko ja astronomi. Hän esitti periaatteen epäinformatiivisen priorin konstruioimiseksi: *säännön, jolla priorijakauma määritetään, tulisi johtaa samaan lopputulokseen, vaikka sääntöä sovellettaisiin muunnettuun parametriin.*

Kun määritellään epäinformatiiviseksi priorijakaumaksi $p(\theta) \propto J(\theta)^{1/2}$, missä $J(\theta)$ on parametrin θ Fisherin informaatio

$$J(\theta) = E \left[\left(\frac{\partial \log p(y|\theta)}{\partial \theta} \right)^2 \middle| \theta \right] = -E \left[\frac{\partial^2 \log p(y|\theta)}{\partial \theta^2} \middle| \theta \right],$$

niin määritelmästä seuraa, että prioria on muunnosinvariantti eli Jeffreysin prioriksi tulee sama kuin käyttämällä muuttujenvaihtokaavaa, ts. $p(\tau) = p(\theta) \left| \frac{\partial \theta}{\partial \tau} \right|$ missä $\theta = h^{-1}(\tau)$, ja tässä mielessä prioria on myös epäinformatiivinen (Jeffreys käytti itse termiä “minimaalisesti informatiivinen”). Voimme osoittaa, että prioria todella on muunnosinvariantti. Olkoon $\theta = h^{-1}(\tau)$, jolloin:

$$\begin{aligned} J(\tau) &= -E \left[\frac{\partial^2 \log p(y|\tau)}{\partial \tau^2} \middle| \tau \right] \\ &= -E \left[\frac{\partial^2 \log p(y|\theta = h^{-1}(\tau))}{\partial \theta^2} \left| \frac{\partial \theta}{\partial \tau} \right|^2 \middle| \theta \right] \\ &= J(\theta) \left| \frac{\partial \theta}{\partial \tau} \right|^2, \end{aligned}$$

eli $J(\tau)^{1/2} = J(\theta)^{1/2} \left| \frac{\partial \theta}{\partial \tau} \right|$, kuten haluttiin.

Jeffreysin periaate on mahdollista yleistää moniparametrisiin malleihin, mutta tulokset eivät ole yhtä suoraviivaisia kuin yhden parametrin tapauksessa. Lisäksi Jeffreysin prioria voi johtaa joissain tilanteissa epäaitoon posterioriin (esim. tietyissä normaalijakaumien sekoitusjakaumien tilanteissa).

Binomijakauman tapauksessa saamme

$$\begin{aligned} J(\theta) &= -E \left[\frac{\partial^2 \log p(y|\theta)}{\partial \theta^2} \middle| \theta \right] \\ &= -E \left[\frac{\partial^2}{\partial \theta^2} (y \log \theta + (n - y) \log(1 - \theta)) \middle| \theta \right] \\ &= -E \left[\frac{(2\theta - 1)y - n\theta^2}{(1 - \theta)^2 \theta^2} \middle| \theta \right] \\ &= \frac{-(2\theta - 1)n\theta + n\theta^2}{(1 - \theta)^2 \theta^2} \\ &= \frac{n}{\theta(1 - \theta)}, \end{aligned}$$

joten Jeffreysin prioria on siis $p(\theta) \propto \frac{1}{\sqrt{\theta(1-\theta)}}$, joka on $\text{Beta}(\frac{1}{2}, \frac{1}{2})$ -jakauma (muista, että tasajakauma vastaa $\text{Beta}(1, 1)$ -jakaumaa).

4.2.2 Konjugaattipriorit

Kuten aiemmin mainittiin, laskennalliset ongelmat olivat pitkään suuri este Bayes-menetelmien yleistymiselle. Yleisesti posteriorijakauman lauseketta

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y, \theta) d\theta}$$

ei voida ratkaista suljetussa muodossa nimittäjässä olevan hankalan moniulotteisen integraalin takia. Voimme kuitenkin joskus määrittää priorin siten, että posteriori ja priori kuuluvat samaan jakaumaperheeseen ja näin saamme yleensä analyttisen ratkaisun posteriorille. Tällaista prioria kutsutaan *konjugaattiprioriksi*. Tällainen konjugaattipriori on olemassa jos havaintomalli $p(y|\theta)$ kuuluu eksponenttiperheeseen (konjugaattisuus pätee myös jossain muissa erikoistapauksissa). Tästä esimerkki ovat edellä mainitut Beta-priori ja Binomi-uskottavuus, jotka tuottavat Beta-jakauman posterioriksi $p(\theta|y)$. Toinen esimerkitapaus on normaalijakauma tunnetulla varianssilla, joka yhdistettynä normaalijakautuneeseen prioriin odotusarvolle tuottaa posteriorin, joka on myös normaalijakauma.

Koska nykyaikaiset Bayes-analyysit pohjautuvat lähes aina simulointipohjaisiin menetelmiin, ei konjugaattipriorien käytöllä ole tänä päivänä yhtä suurta merkitystä kuin ennen. Niiden käyttö on myös usein hankalaa (tai mahdotonta) kompleksisille hierarkkisille malleille, ja toisaalta joskus ennakkotietomme θ :sta on helpompi kirjoittaa muilla tavoin kuin konjugaattipioreita käyttäen. Konjugaattipriorien käyttö voi kuitenkin joissain tapauksissa olla hyödyllistä laskennallisen tehokkuuden parantamiseksi myös osana isompaa mallia.

Esimerkki 11: Normaalijakauman konjugaattisuus

Oletetaan havainnot $y = (y_1, \dots, y_n)$ joille pätee $y_i|\mu \sim N(\mu, \sigma^2)$ ja oletetaan keskihajonta σ tunnetuksi. Nyt havaintojen yhteisjakauma ehdolla on muotoa

$$p(y|\mu) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right].$$

Valitaan nyt prioriksi $\mu \sim N(\mu_0, \sigma_0^2)$, jolloin Bayesin kaavan mukaisesti

$$p(\mu|y) \propto p(\mu)p(y|\mu) \propto \exp\left[-\frac{1}{2} \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right) \left(\mu - \frac{\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{y}}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}\right)^2\right],$$

missä \bar{y} on y :n otoskeskiarvo. Näin siis μ :n posteriorijakauma on normaalijakauma odotusarvolla

$$\frac{\frac{1}{\sigma_0^2}\mu_0 + \frac{n}{\sigma^2}\bar{y}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}$$

ja varianssilla

$$\frac{1}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}.$$

Parametrin μ posteriorijakauman odotusarvo voidaan ymmärtää painotettuna keskiarvona priorijakauman odotusarvosta ja otoskeskiarvosta, toisella tapaa kirjoitettuna

$$E(\mu|y) = \frac{\sigma_0^2}{\sigma_0^2 + \frac{n}{\sigma^2}}\mu_0 + \frac{\frac{n}{\sigma^2}}{\sigma_0^2 + \frac{n}{\sigma^2}}\bar{y},$$

$$\text{Var}(\mu|y) = (\tau_{\text{priori}} + \tau_{\text{data}})^{-1},$$

missä τ viittaa *tarkkuuteen* (varianssin käänteisluku).

Joitain ehdotuksia sopiviksi priorijakaumiksi erityisen prioritiedon puuttuessa on esitetty täällä: <https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations> (osa suosituksista liittyy erityisesti Stanin käyttämän MCMC-algoritmin tehokkuuden parantamiseen, mutta pääosin ehdotukset ovat verrattain yleisiä)

4.2.3 Yksiparametrin eksponenttisen perheen konjugaattiperhe

Oletetaan, että (yhden) havainnon jakauman tiheysfunktio on

$$p(y_i|\theta) = f(y_i)g(\theta)e^{\phi(\theta)u(y_i)},$$

missä $f(\cdot)$, $g(\cdot)$, $\phi(\cdot)$ ja $u(\cdot)$ ovat tunnettuja funktioita. Tällöin sanomme, että y_i :n jakauma kuuluu *eksponenttiseen perheeseen* (*exponential family*). Oletetaan nyt, että havainnot y_1, \dots, y_n ovat riippumattomia ehdolla θ ja noudattavat yksiparametriseen eksponenttiseen perheeseen kuuluvaa jakaumaa. Silloin uskottavuus on

$$\begin{aligned} p(y|\theta) &= \left[\prod_{i=1}^n f(y_i) \right] g(\theta)^n e^{\phi(\theta) \sum_{i=1}^n u(y_i)} \\ &= \left[\prod_{i=1}^n f(y_i) \right] g(\theta)^n e^{\phi(\theta)t(y)}, \end{aligned}$$

missä $t(y) = \sum_{i=1}^n u(y_i)$ on *tyhjentävä tunnusluku* (ks. Luku 5.2 tyhjentävyydestä) ja $\phi(\theta)$ on ns. *luonnollinen parametri*. Jos priorin on (normeerausta vaille) muotoa

$$p(\theta) \propto g(\theta)^\eta e^{\phi(\theta)\nu},$$

joillakin parametreilla η ja ν , niin silloin posteriori on

$$p(\theta|y) \propto g(\theta)^{\eta+n} e^{\phi(\theta)(\nu+t(y))},$$

mikä on samaa muotoa kuin priorijakauma. Erityisesti priorin parametrit päivittyvät posteriorissa seuraavasti:

$$\begin{aligned} \eta &\rightarrow \eta + n \\ \nu &\rightarrow \nu + t(y) \end{aligned}$$

4.2.4 Priorien määrittäminen brms-paketissa

brm-funktion argumentilla `prior` voidaan määrittellä malliin liittyvien parametrien priorijakaumat. Oletuksena kaikille parametreille käytetään joitain varsin epäinformatiivisia prioria. Voit tarkistaa oletuspriorit ja muuttujat, joille prioreita voi antaa komennolla `get_prior`:

```
x1 <- rnorm(10)
x2 <- rnorm(10)
y3 <- rnorm(10, x1 + x2)
d3 <- data.frame(y = y3, x1, x2)
```

```
get_prior(
  y ~ x1 + x2,
  data = d3
)
```

```
##           prior      class coef group resp dpar nlpar lb ub      source
##           (flat)         b             x1             (vectorized)
##           (flat)         b             x2             (vectorized)
## student_t(3, 0.6, 2.5) Intercept             default
## student_t(3, 0, 2.5)   sigma             0             default
```

Oletuksena regressiokertoimille (`class=b`) käytetään ns. diffuusio prioria ($p(\beta) \propto 1$). Huomaa, että tämä voi kuitenkin joskus tuottaa laskennallisia ongelmia. Vakiotermille `intercept` oletuspriori on yleinen t-jakauma vapausasteilla 3, jossa sijainti μ ja skaalaparametri ν riippuvat vasteesta y (`median(y)` ja `mad(y)`). Huomaa, että oletuksena `brms` keskittää kaikki selittäjät, ja vakiotermin priorin määrittämällä tätä keskistettyä tapausta vastaavalle vakiotermille, vaikka `brm` kuitenkin palauttaa vakiotermin arvon ilman tätä keskittämistä. Katso

lisää komennolla `?prior` ja varsinkin `?brmsformula` dokumentaation kappale “Parameterization of the population-level intercept”.

Prioreiden syöttäminen tapahtuu käyttäen funktiota `prior` (katso myös `?prior`) määrittelemällä haluttu priorijakauma (esim. `normal(0, 1)`), parametrin luokka (oletus `class = "b"`, populaatiotason kertoimet), yksittäinen kerroin (argumentti `coef`), ja muita mahdollisia argumentteja kuten `lb` joka määrittää priorijakauman alarajan (tämä vaihtoehto ei ole kuitenkaan käytössä kuin tietyn tyyppisille parametreille). Useita priorimäärittelyjä voidaan yhdistää joko `c()`-funktioilla tai yksinkertaisesti lisäämällä ne toisiinsa:

```
priors <- prior(normal(0, 1), class = "b", coef = x1) +
  prior(normal(1, 1), class = "b", coef = x2) +
  prior(cauchy(0, 2), class = "sigma")

fit <- brm(
  y ~ x1 + x2,
  prior = priors,
  data = d3,
  refresh = 0
)
```

Mallissa käytetyt priorit voi tarkistaa komennolla `prior_summary`:

```
prior_summary(fit)
```

```
##           prior      class coef group resp dpar nlpar lb ub  source
##           (flat)         b      x1
## normal(0, 1)         b      x1
## normal(1, 1)         b      x2
## student_t(3, -0.7, 2.5) Intercept
## cauchy(0, 2)      sigma                0      user
```

Vaihtoehtoja priorijakaumiksi (ja havaintojakaumiksi) löytyy komennolla `?brmsfamily`. Tarkempia tietoja jakaumien parametreista voi etsiä [Stanin dokumentaatiosta](#).

4.3 Prediktiiviset jakaumat

4.3.1 Prioriennustejakauma

Posteriorijakauman $p(\theta|y)$ lisäksi on usein hyödyllistä tarkastella tuntemattoman havainnon tai havaintojen \tilde{y} prediktiivisiä jakaumia. *Prioriennustejakauma* (prior predictive distribution) määritellään reunajakaumana

$$p(\tilde{y}) = \int p(\tilde{y}, \theta) d\theta = \int p(\tilde{y}|\theta)p(\theta) d\theta.$$

Huomaa ero ehdolliseen jakaumaan $p(\tilde{y}|\theta)$. Havainnon prioriennustejakauma ei käytä aineistoa lainkaan, se on “ennuste” tulevalle havainnolle, kun uskottavuus ja priorit on valittu (mutta havaintoa ei ole tehty). Prioriennustejakauma sisältää prioritiedosta johtuvan θ :n epävarmuuden, ja siten se on hyödyllinen työkalu esim. tutkittaessa priorijakauman järkevyyttä. Kuten posteriorijakauman kaavassa oleva normalisointivakio, tätäkään emme voi yleisesti laskea suljetussa muodossa, mutta jakaumasta simulointi on helppoa:

1. Ota näyte θ^i jakaumasta $p(\theta)$
2. Ota näyte \tilde{y}^i jakaumasta $p(\tilde{y}|\theta^i)$
3. Toista vaiheita 1–2 N kertaa, tallentaen \tilde{y}^i :t

Nyt $\tilde{y}^1, \dots, \tilde{y}^N \sim p(\tilde{y})$.

Esimerkki 13: Prediktiivinen priorijakauma normaalijakaumalle

Olkoon $\theta \sim N(0, 1)$ ja $\tilde{y}|\theta \sim N(\theta, 0.1)$. Tässä tapauksessa osaamme löytää marginaalijakauman integroimalla, mutta voimme myös välttää integroinnin toteamalla ensin että koska $p(y, \theta)$ on kaksiulotteinen normaalijakauma on myös $p(\tilde{y})$ normaalijakauma, ja tämän jälkeen voimme laskea jakauman parametrit käyttäen seuraavia hyödyllisiä kaavoja:

$$\begin{aligned} E_{\theta}(Y) &= E_{\theta}[E(Y|\theta)], \\ \text{Var}_{\theta}(Y) &= \text{Var}_{\theta}[E(Y|\theta)] + E_{\theta}[\text{Var}(Y|\theta)], \end{aligned}$$

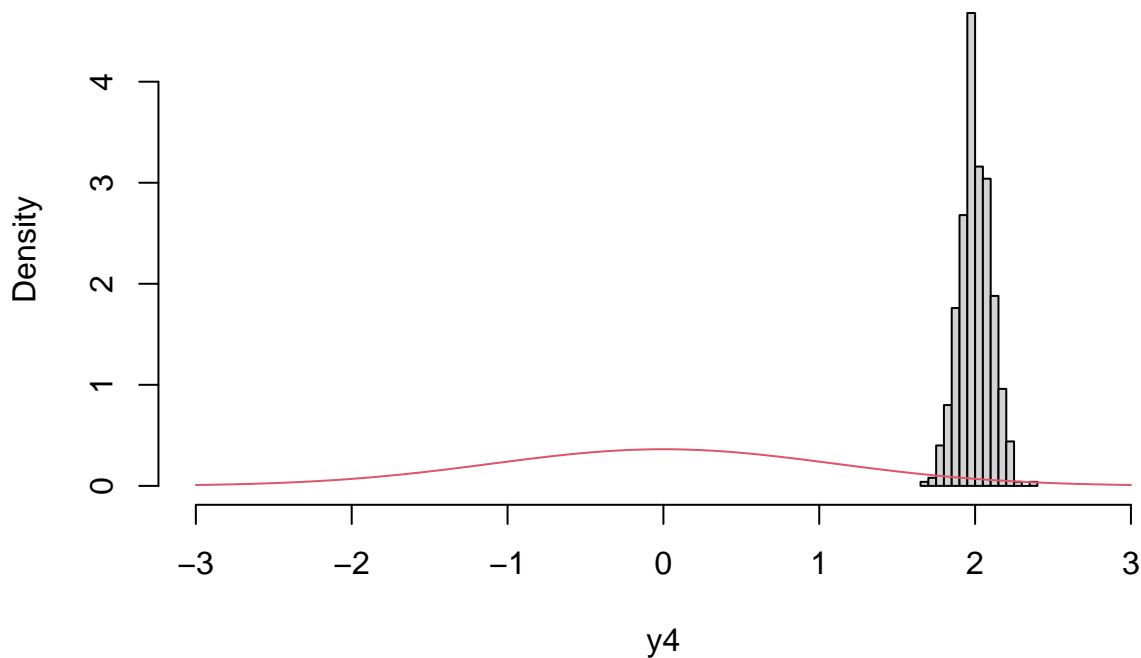
missä alaindeksi θ viittaa siihen että odotusarvo (varianssi) lasketaan suhteessa θ :n jakaumaan.

Nyt $E(Y|\theta) = \theta$ joten $E_{\theta}(\theta) = 0$ ja $\text{Var}_{\theta}(\theta) = 1$. Lisäksi $\text{Var}(Y|\theta) = 0.1$ joka on vakio joten $E_{\theta}[\text{Var}(Y|\theta)] = 0.1$. Niinpä $p(y)$ on normaalijakauma odotusarvolla 0 ja varianssilla 1.1.

Oletetaan että todellisuudessa $\theta = 2$ ja simuloidaan $y \sim N(\theta, 0.1)$, ja verrataan havaintojen histogrammia prediktiiviseen priorijakaumaamme:

```
set.seed(1)
y4 <- rnorm(500, 2, 0.1)
hist(y4, freq = FALSE, xlim = c(-3, 3))
curve(dnorm(x, 0, 1.1), from = -3, to = 3, col = 2, add = TRUE)
```

Histogram of y4



Huomataan että prediktiivisen priorijakauman keskiarvo on kaukana havaitusta keskiarvosta, mutta koska jakauma on myös erittäin leveä, ei tällaisen priorin käyttö ole tuota välttämättä ongelmia posteriorin simuloinnissa:

```
fit_norm <- brm(
  y ~ 1,
```

```
prior = prior(normal(0, 1), class = "Intercept"),
data = data.frame(y = y4),
refresh = 0
)
```

fit_norm

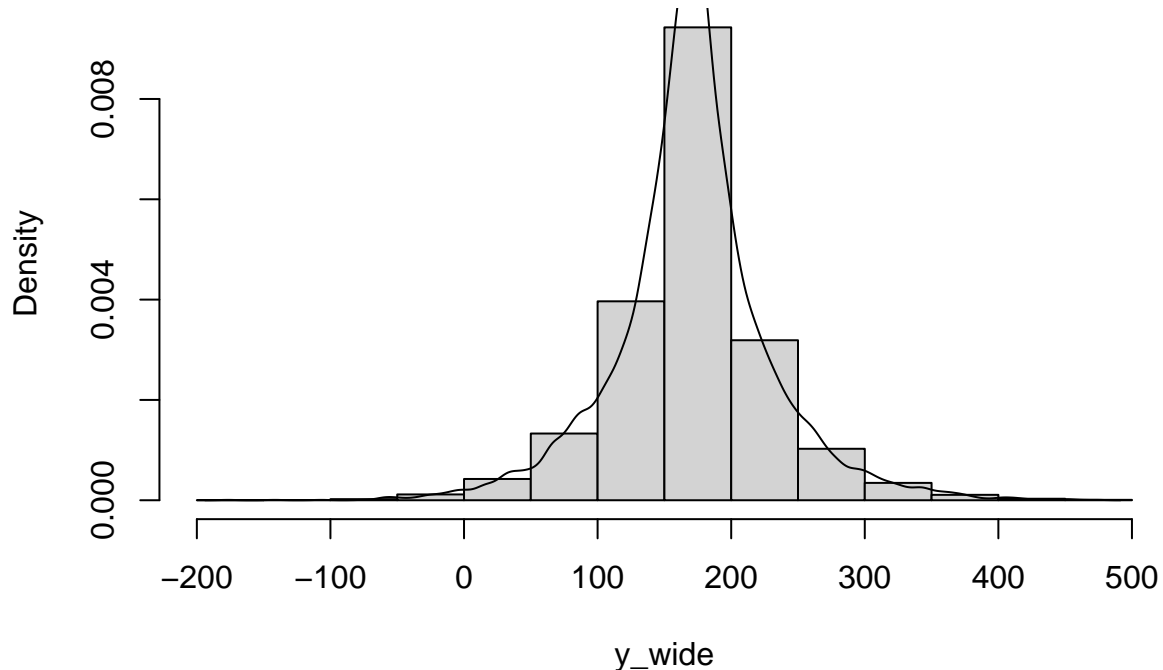
```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: y ~ 1
## Data: data.frame(y) (Number of observations: 10)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept -0.72 0.43 -1.53 0.17 1.00 2322 2338
##
## Family Specific Parameters:
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma 1.48 0.38 0.93 2.41 1.00 2266 2121
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Esimerkki 14: Liian leveä prior

Kuvitellaan että olemme kiinnostuneita aikuisten suomalaisten pituudesta y (senttimetriä). Oletetaan $y|\mu, \sigma^2 \sim N(\mu, \sigma^2)$, ja asetetaan priorit $\mu \sim N(170, 10)$ ja $\sigma \sim U(0, 100)$. Toisin sanoen olemme jokseenkin varmoja, että suomalaisten keskipituus jossain 150 cm ja 190 cm välissä. Toisaalta keskihajonnalle σ määrittämämme tasajakauma on erittäin leveä. Simuloidaan aineistoa tästä prediktiivisestä priorijakaumasta:

```
set.seed(1)
mu <- rnorm(1e4, 170, 10)
sigma <- runif(1e4, 0, 100)
y_wide <- rnorm(1e4, mu, sigma)
hist(y_wide, freq = FALSE)
lines(density(y_wide))
```

Histogram of y_wide



```
mean(y_wide < 0)
```

```
## [1] 0.0072
```

```
mean(y_wide > 200)
```

```
## [1] 0.2354
```

Huomaamme, että keinotekoinen aineistomme sisältää poikkeuksellisen paljon yli kaksimetrisiä ja useita negatiivisen pituuden omaavia henkilöitä! Tämantyyppiset leveät priorit ovat varsinkin ennen olleet erittäin suosittuja pyrkimässä priorijakauman vaikutusten minimoimiseen johtopäätöksiä tehtäessä. Prioriennustejakaumasta simuloimalla näemme kuitenkin heti, että priorivalintamme ovat objektiivisestikin mielettömiä. Huomaa kuitenkin, että yleisesti jos prioriennustejakauma näyttää olevan ristiriidassa aineiston tai (mallista puuttuvien) ennakkotietojen kanssa, vika voi olla niin priorijakaumissa kuin uskottavuusfunktion määrittelyssä. Toisaalta priorijakauman vaikutus posterioriin häviää aineiston koon kasvaessa, joten “liian leveät” priorit eivät välttämättä tuota käytännössä ongelmia, jos aineistossa on riittävästi informaatiota.

Esimerkki 15: Prioriennustejakauman visualisointi

Tutkitaan `brms`-paketin epilepsia-aineistoa. Aikomuksena on sovittaa yksinkertainen Poisson-regressiomalli, jossa selitetään epilepsia-kohtausten määrää standardoidulla iällä, perustasolla (kohtausten lukumäärä tarkastelujakson alussa) sekä lääkityksellä (unohdetaan tässä kohtaa, että aineisto koostuu useasta henkilöstä ja mittaushetkestä). Poimitaan näytteitä prioriennustejakaumasta:

```
set.seed(1)
fit_prior <- brm(
  count ~ zAge + zBase * Trt,
  data = epilepsy,
```

```

family = poisson(),
sample_prior = "only",
prior = prior(student_t(5, 0, 10), class = b),
refresh = 0
)

```

```
fit_prior
```

```

## Family: poisson
## Links: mu = log
## Formula: count ~ zAge + zBase * Trt
## Data: epilepsy (Number of observations: 236)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      1.25      9.98  -14.89   16.97 1.00    2607    1390
## zAge            0.04     13.52  -26.06   27.22 1.00    4453    2252
## zBase           0.03     13.03  -26.30   26.36 1.00    4525    2081
## Trt1           -0.24     12.61  -25.54   25.55 1.00    4204    2372
## zBase:Trt1     0.18     12.78  -24.87   25.75 1.00    4632    2325
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

```

Argumenttivalinnalla `sample_prior = "only"` brm näytteistää vain priorijakaumaa ja mahdollistaa priorienustejakauman visualisoinnin esimerkiksi komennolla `pp_check`:

```
pp_check(fit_prior)
```

```
## Using 10 posterior draws for ppc type 'dens_overlay' by default.
```



Malli antaa niin suuria arvoja, että kuvan piirto ei onnistu (ks. x-akseli). Saamme prioriennustejakauman näytteet käyttöön komennolla `posterior_predict` (joka normaalisti palauttaa näytteitä *posterioriennustejakaumasta*, mutta koska `brm` näytteisti yllä prioria posteriorin sijaan, sama funktio palauttaa nyt näytteitä prioriennustejakaumasta):

```
py <- posterior_predict(fit_prior)
str(py)
```

```
## num [1:4000, 1:236] 9.67e+08 3.85e+03 4.38e+04 3.51e+08 9.47e+11 ...
## - attr(*, "dimnames")=List of 2
## ..$ : NULL
## ..$ : NULL
```

Koska annoimme `brm`-funktioille aineiston, jossa oli 236 havaintoa, funktio palauttaa 4000 kpl 236 havainnon otoksia prioriennustejakaumasta. Tutkitaan simuloitujen havaintojen vaihteluväliä ja verrataan replikaatioaineistojen maksimia oikean aineistomme maksimiin:

```
range(py)
```

```
## [1] 0.000000e+00 1.201289e+144
```

```
mean(apply(py, 1, max) > max(epilepsy$count))
```

```
## [1] 0.99175
```

Huomataan että replikaatioaineistoissamme on epärealistisen suuria arvoja, joten kokeillaan korjata prioreita:

```
priors2 <- prior(normal(0, 0.5), class = b) +
  prior(normal(2, 0.5), class = Intercept)
```

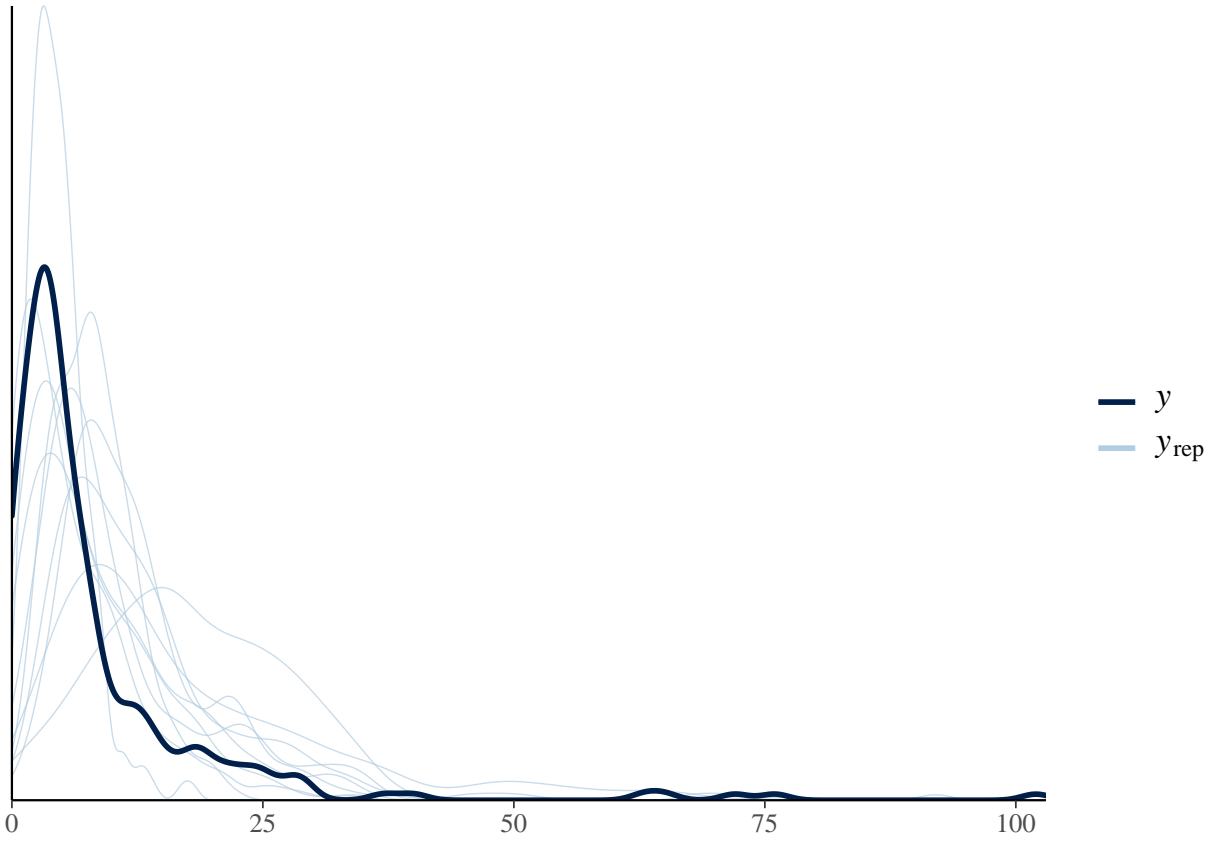
```
fit_prior2 <- brm(
  count ~ zAge + zBase * Trt,
  data = epilepsy,
  family = poisson(),
  sample_prior = "only",
  prior = priors2,
  refresh = 0
)
```

```
fit_prior2
```

```
## Family: poisson
## Links: mu = log
## Formula: count ~ zAge + zBase * Trt
## Data: epilepsy (Number of observations: 236)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      2.00     0.56   0.94   3.12 1.00   5548   3115
## zAge           0.01     0.48  -0.92   0.96 1.00   4933   3097
## zBase          -0.01     0.50  -0.97   0.97 1.00   5430   2834
## Trt1           0.01     0.48  -0.94   0.96 1.00   5463   3358
## zBase:Trt1     0.02     0.51  -1.01   1.02 1.00   5390   3002
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
pp_check(fit_prior2)
```

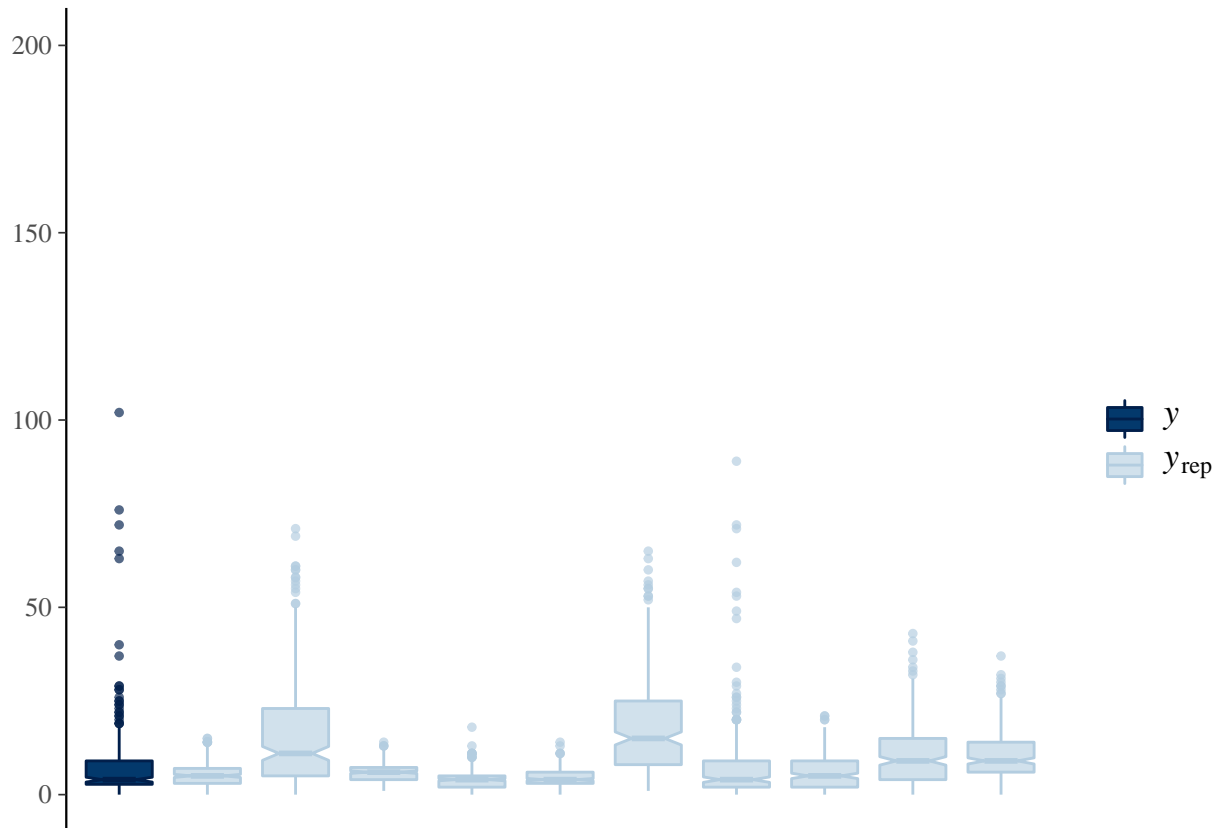
```
## Using 10 posterior draws for ppc type 'dens_overlay' by default.
```



```
pp_check(fit_prior2, type = "boxplot") + ggplot2::ylim(0, 200)
```

```
## Using 10 posterior draws for ppc type 'boxplot' by default.
```

```
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
```



```
mean(apply(posterior_predict(fit_prior2), 1, max) > max(epilepsy$count))
```

```
## [1] 0.27125
```

Priorinustejakauma on nyt selvästi uskottavampi.

Verrataan vielä posterioriestimaatteja näitä kahta priorivaihtoehtoa käyttäen:

```
fit1 <- brm(
  count ~ zAge + zBase * Trt,
  data = epilepsy,
  family = poisson(),
  prior = prior(student_t(5, 0, 10), class = b),
  refresh = 0
)
```

```
fit2 <- brm(
  count ~ zAge + zBase * Trt,
  data = epilepsy,
  family = poisson(),
  prior = priors2,
  refresh = 0
)
```

```
posterior_summary(fit1)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## b_Intercept 1.93542898 0.038156135 1.861235e+00 2.00953045
```



```
## b_zAge          0.14988475 0.025355432 1.010482e-01 0.20028918
## b_zBase         0.57091780 0.024205892 5.220952e-01 0.61744346
## b_Trt1          -0.19393093 0.055036031 -3.036431e-01 -0.08623433
## b_zBase:Trt1    0.04871247 0.029619196 -8.842615e-03 0.10767384
## lprior          -15.02441918 0.002447972 -1.502941e+01 -15.01982775
## lp__            -877.47043525 1.578856625 -8.813581e+02 -875.41626061
```

```
posterior_summary(fit2)
```

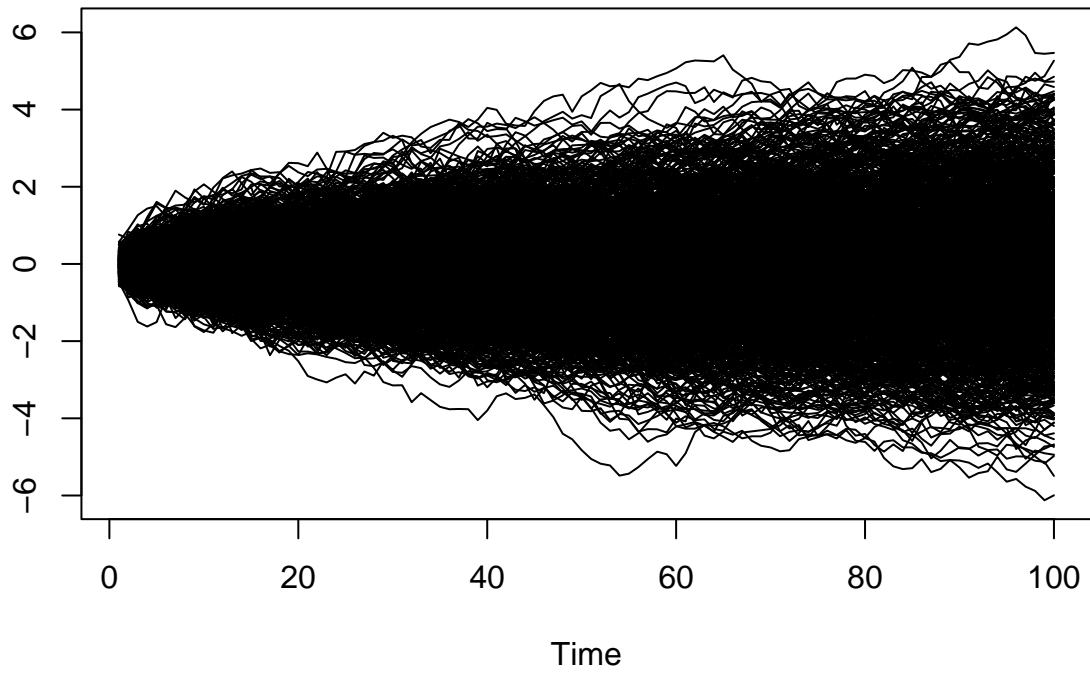
```
##           Estimate Est.Error      Q2.5      Q97.5
## b_Intercept 1.93547086 0.03714653 1.864367e+00 2.00617934
## b_zAge      0.14948864 0.02609750 9.829285e-02 0.20042123
## b_zBase     0.56971731 0.02378334 5.224398e-01 0.61653728
## b_Trt1      -0.19308175 0.05378579 -2.976023e-01 -0.08876349
## b_zBase:Trt1 0.04964447 0.02899327 -6.628362e-03 0.10700999
## lprior      -1.96851496 0.06439050 -2.101955e+00 -1.84958853
## lp__        -864.38335097 1.55357788 -8.682514e+02 -862.33883749
```

Huomataan että tässä tapauksessa data on kuitenkin niin informatiivista, että molemmat tapaukset tuottavat lähes identtiset tulokset.

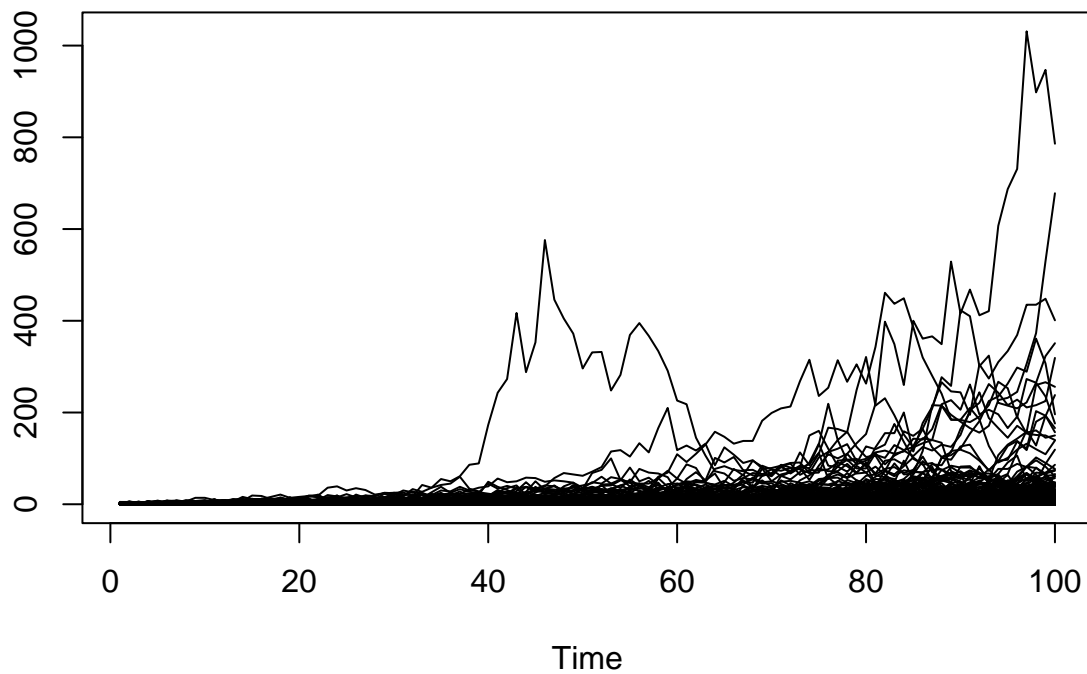
Esimerkki 16: Prioriennustejakauma satunnaiskulkumallille

Oletetaan että meillä on jokin latentti prosessi $\alpha_1, \dots, \alpha_T$ jossa $\alpha_t | \alpha_{t-1} \sim N(\alpha_{t-1}, \sigma^2)$. Lisäksi meillä on havaintoja $y_t | \alpha_t \sim \text{Poisson}(e^{\alpha_t})$. Simuloidaan 100 havainnon mittaisia aikasarjoja ehdolla $\sigma = 0.2$ ja $\alpha_1 \sim N(0, 0.2^2)$:

```
set.seed(1)
n <- 100
alpha <- replicate(1000, {
  cumsum(rnorm(n, 0, 0.2))
})
ts.plot(alpha)
```

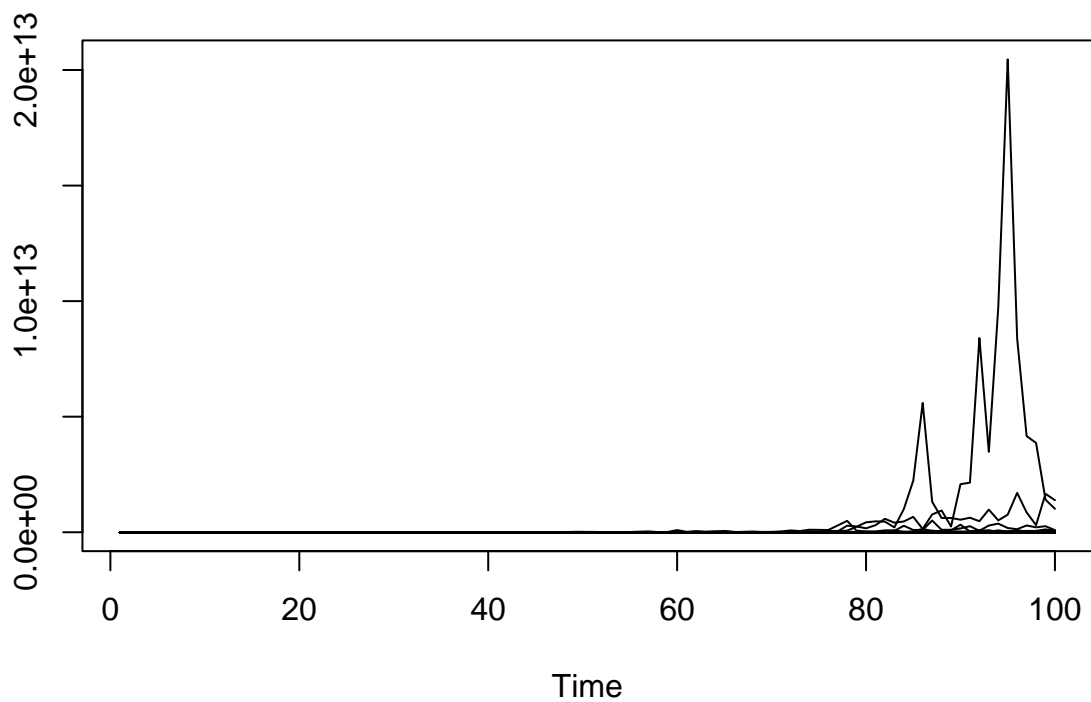


```
set.seed(1)
n <- 100
Y <- replicate(1000, {
  alpha <- cumsum(rnorm(n, 0, 0.2))
  rpois(n, exp(alpha))
})
ts.plot(Y)
```



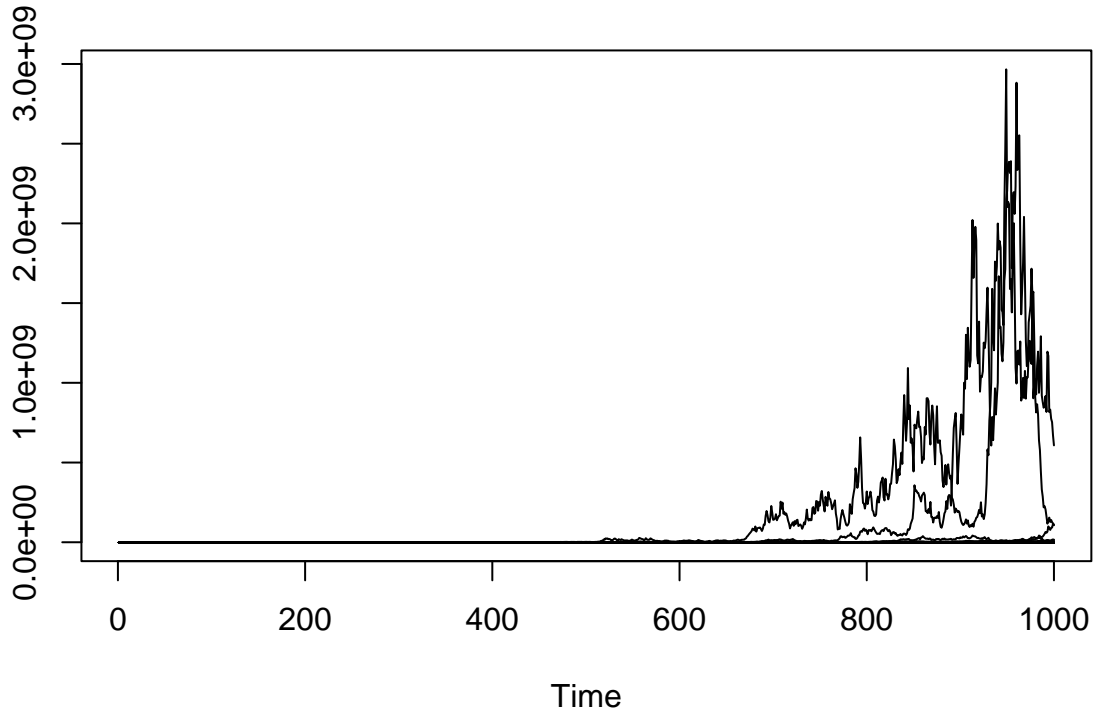
Kuvat ovat tietenkin epäselviä, mutta oleellista on, että havaintojen vaihteluväli on maltillinen. Vaihdetaan $\sigma = 1$:

```
set.seed(1)
Y <- replicate(1000, {
  alpha <- cumsum(rnorm(n, 0, 1))
  rpois(n, exp(alpha))
})
ts.plot(Y)
```



Selvästikään σ ei voi olla kovin suuri. Toisaalta sarjojen pituutta jatkamalla saadaan

```
set.seed(1)
n <- 1000
Y <- replicate(1000, {
  alpha <- cumsum(rnorm(n, 0, 0.2))
  rpois(n, exp(alpha))
})
ts.plot(Y)
```



Tästä voisi päätellä, että Gaussinen satunnaiskulku ei ole kovin hyvä malli Poisson-jakautuneille aikasarjoille, mutta käytännössä malli on kuitenkin yleisesti käytetty, sillä vaikka prediktiivinen priorijakauma on “liian joustava” mahdollistaen liiankin vaihtelevat ja räjähtävät aikasarjat, satunnaiskulkuprosessin (marginaaliset) posteriorijakaumat $p(\alpha_t|y)$ ovat kuitenkin yleensä maltillisia havaintojen informatiivisuuden takia.

4.3.2 Posterioriennustejakauma

Posteriorijakauman sovittamisen jälkeen voimme laskea *posterioriennustejakauman* (posterior predictive distribution):

$$p(\tilde{y}|y) = \int p(\tilde{y}, \theta|y) d\theta = \int p(\tilde{y}|\theta, y)p(\theta|y) d\theta.$$

Tapauksessa, jossa havainnot ovat toisistaan riippumattomia ehdolla θ , sievenee yllä oleva kaava muotoon

$$p(\tilde{y}|y) = \int p(\tilde{y}|\theta)p(\theta|y) d\theta.$$

Tästä jakaumasta saamme siis ennustejakauman uusille havainnoille ehdolla nykyinen havaintoaineisto (joka ottaa huomioon θ :n estimoimiseen liittyvän epävarmuuden), ja lisäksi posterioriennustejakauma on hyödyllinen mallin yhteensopivuustarkasteluissa. Tällöin on usein tapana simuloida uusia keinotekoisia havaintoaineistoja y_{rep}^i posterioriennustejakaumasta, ja verrata jotain oikeasta aineistosta y laskettua tunnuslukua $T(y)$ aineiston $(T(y_{\text{rep}}^1), \dots, T(y_{\text{rep}}^N))$ jakaumaan. (Huomaa, että tuntematon suure \tilde{y} voi tarkoittaa esimerkiksi joko yksittäistä havaintoa, yksittäistä aineistoa tai useaa aineistoa. Monet Bayes-R-paketit käyttävätkin merkintää y_{rep}^i , joka tarkoittaa yhtä yleensä samankokoista aineistoa kuin havaittu y .)

Esimerkki 17: Ennustaminen Bayes-mallilla

Tutkitaan ympäristömuuttujien vaikutusta muurahaisten lukumääriin Kaakkois-Australiassa Poisson-regression avulla. Aineisto sisältää seuraavat muuttujat:

- `counts`: abundances of 41 epigaeic ant species.
- `Bare.ground (bg)`: Percent cover of bare ground, as estimated from ten 1x1 metre quadrats.
- `Shrub.cover (sc)`: Percent shrub cover, as estimated from two 20x20m transects.
- `Feral.mammal.dung (fmd)` Proportion of quadrats including mammal dung, out of ten 1x1m quadrats.

Sovitetaan Poisson-malli, jossa selitetään muurahaisten lukumäärää paljaan maan osuudella (`bg`), kasvipeitteen osuudella (`sc`) ja jätöksillä:

```
ants <- read.table("http://users.jyu.fi/~jovetale/YLM2/ants.txt")
ants[1:10,]
```

```
##      counts    bg    sc  fmd
## 1         0  9.00  0.50  0.20
## 2         1  6.00  0.00  0.65
## 3         1  9.25  8.10  0.00
## 4         0 23.25  0.25  0.20
## 5         7  8.75  3.35  0.35
## 6         5 32.25  0.00  1.00
## 7         3 21.35  1.50  0.05
## 8         0 15.75  4.40  0.35
## 9         4  5.35  4.85  0.25
## 10        2 15.20  0.70  0.25
```

```
fit_ants <- brm(
  counts ~ bg + sc + fmd,
  data = ants,
  family = poisson,
  refresh = 0
)
```

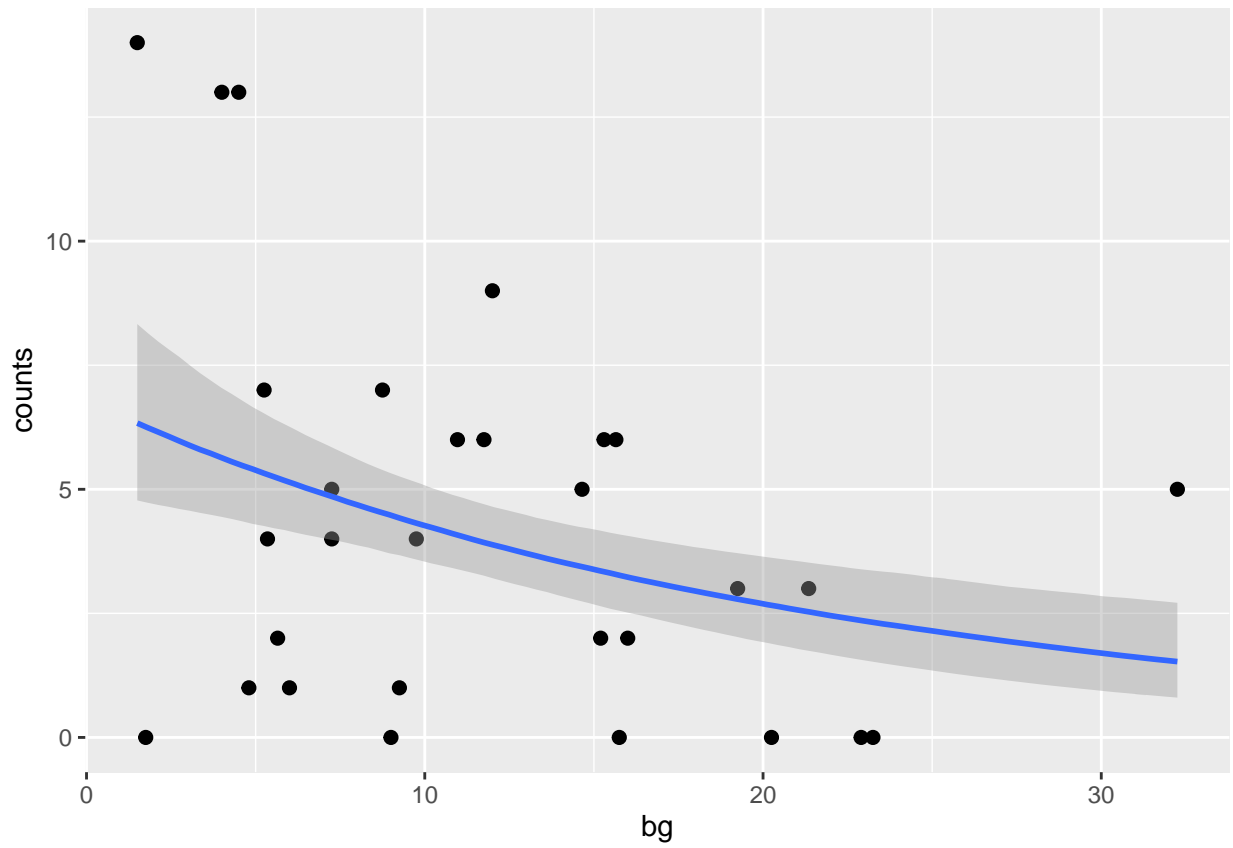
```
fit_ants
```

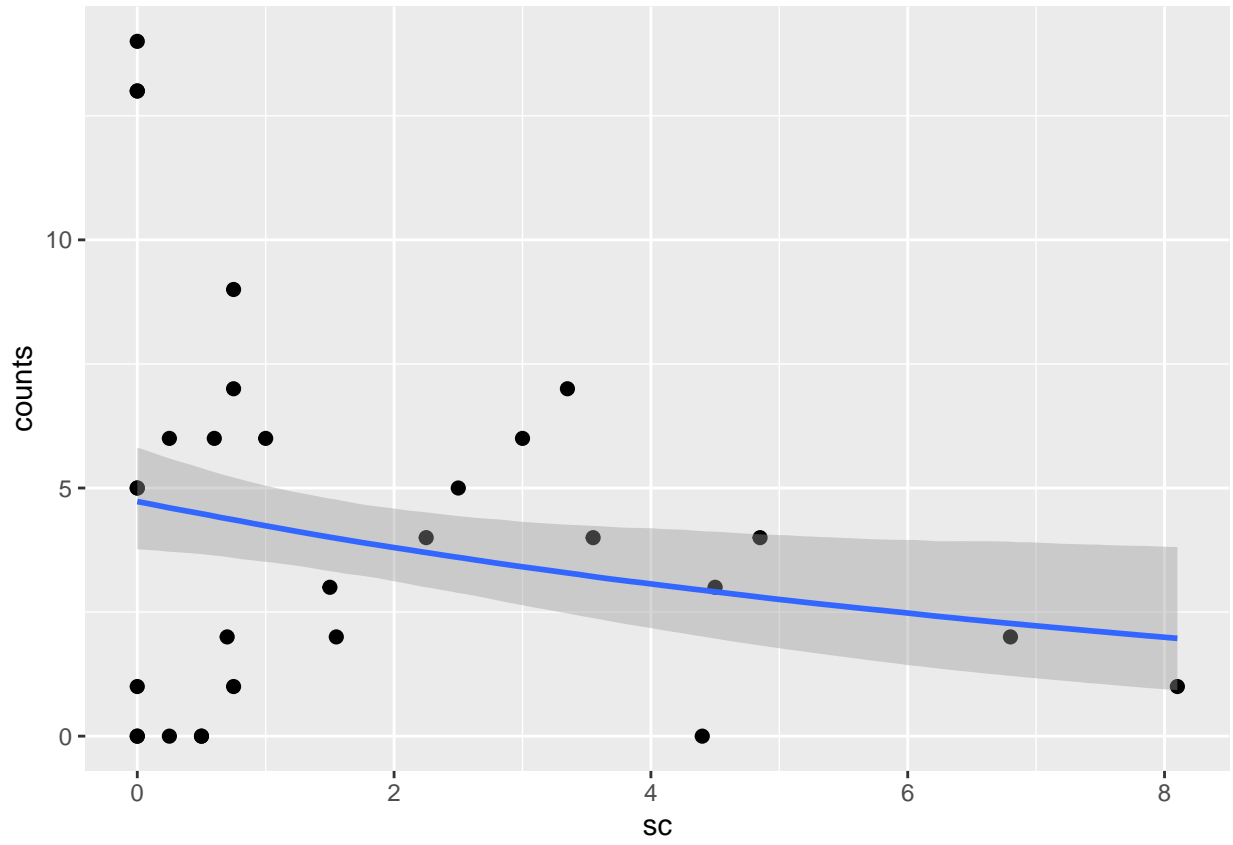
```
## Family: poisson
## Links: mu = log
## Formula: counts ~ bg + sc + fmd
## Data: ants (Number of observations: 30)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      2.10      0.17   1.77   2.42 1.00    4990    3690
## bg             -0.05      0.01  -0.07  -0.02 1.00    3531    3275
## sc             -0.11      0.05  -0.21  -0.01 1.00    3590    2688
## fmd             0.01      0.35  -0.71   0.67 1.00    3705    2715
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

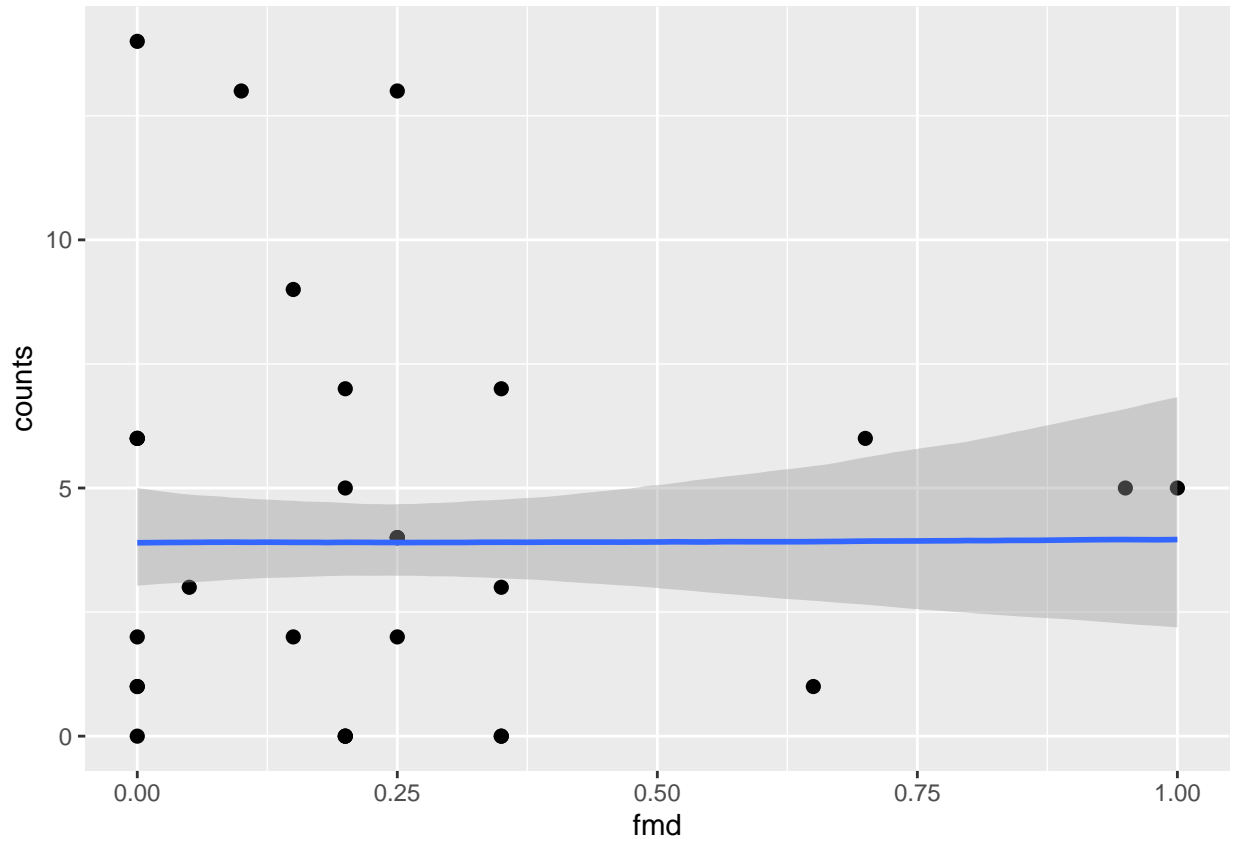
Parametrien posteriorivälien perusteella maan ominaisuuksilla näyttää olevan vaikutusta, jätöksien osalta posterioriväli on erittäin leveä ja sisältää nollan.

Piirretään myös ehdolliset posteriorimediaanit (oletuksena `conditional_effects` piirtää mediaanit, odotusarvot saa lisäämällä option `robust=FALSE`) eri muuttujien arvoilla (muut muuttujat on asetettu keskiarvokseen):

```
plot(conditional_effects(fit_ants), points = TRUE)
```

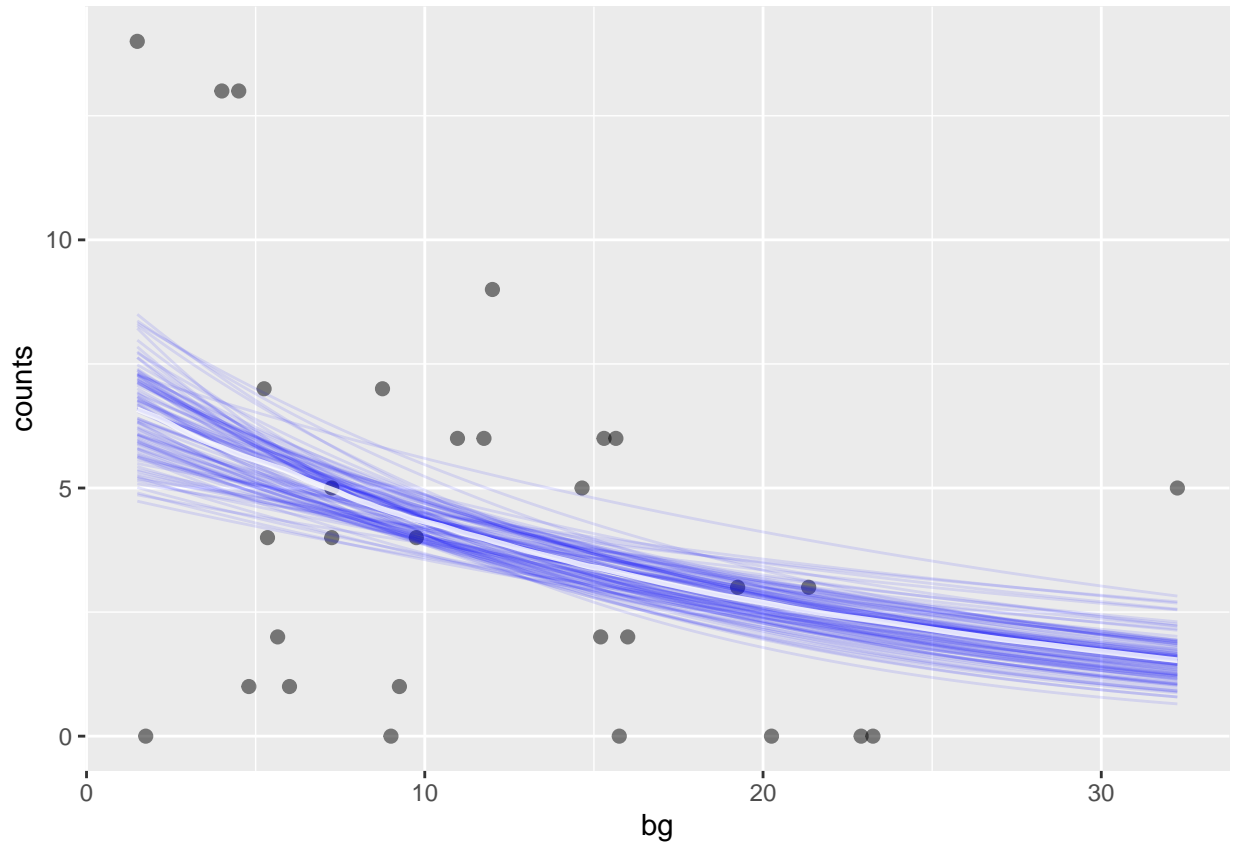






Piirretään vielä yksittäisiä käyriä 'bg'-muuttujan suhteen:

```
plot(
  conditional_effects(
    fit_ants, "bg",
    spaghetti = TRUE,
    ndraws = 100
  ),
  points = TRUE,
  point_args = c(alpha = 0.5)
)
```



Ennustetaan sitten muurahaisten lukumääriä tilanteessa, jossa $fmd = 0$, $sc = 0$ ja bg saa arvoja väliltä $[0,50]$. Functiolla `predict` voimme laskea havaintotason ennusteita, kun taas `fitted` funktiolla saamme näytteitä posterioriennustejakauman odotusarvoista ja niihin liittyvän epävarmuuden:

```
newdata <- data.frame(bg = seq(0, 50, by = 1), sc = 0, fmd = 0)
newobs <- predict(fit_ants, newdata = newdata)
expected <- fitted(fit_ants, newdata = newdata)
head(newobs)
```

```
##      Estimate Est.Error Q2.5 Q97.5
## [1,]  8.27675  3.159059   3    15
## [2,]  7.94075  3.163106   3    15
## [3,]  7.55900  2.979386   2    14
## [4,]  7.23025  2.861430   2    14
## [5,]  6.88550  2.796935   2    13
## [6,]  6.49100  2.712795   2    12
```

```
head(expected)
```

```
##      Estimate Est.Error   Q2.5   Q97.5
## [1,]  8.302329  1.3939073  5.894490 11.217198
## [2,]  7.915476  1.2676278  5.711894 10.577161
## [3,]  7.547996  1.1566245  5.511554  9.991111
## [4,]  7.198861  1.0601694  5.314803  9.438693
## [5,]  6.867097  0.9775336  5.151810  8.901082
## [6,]  6.551786  0.9079442  4.960716  8.432249
```

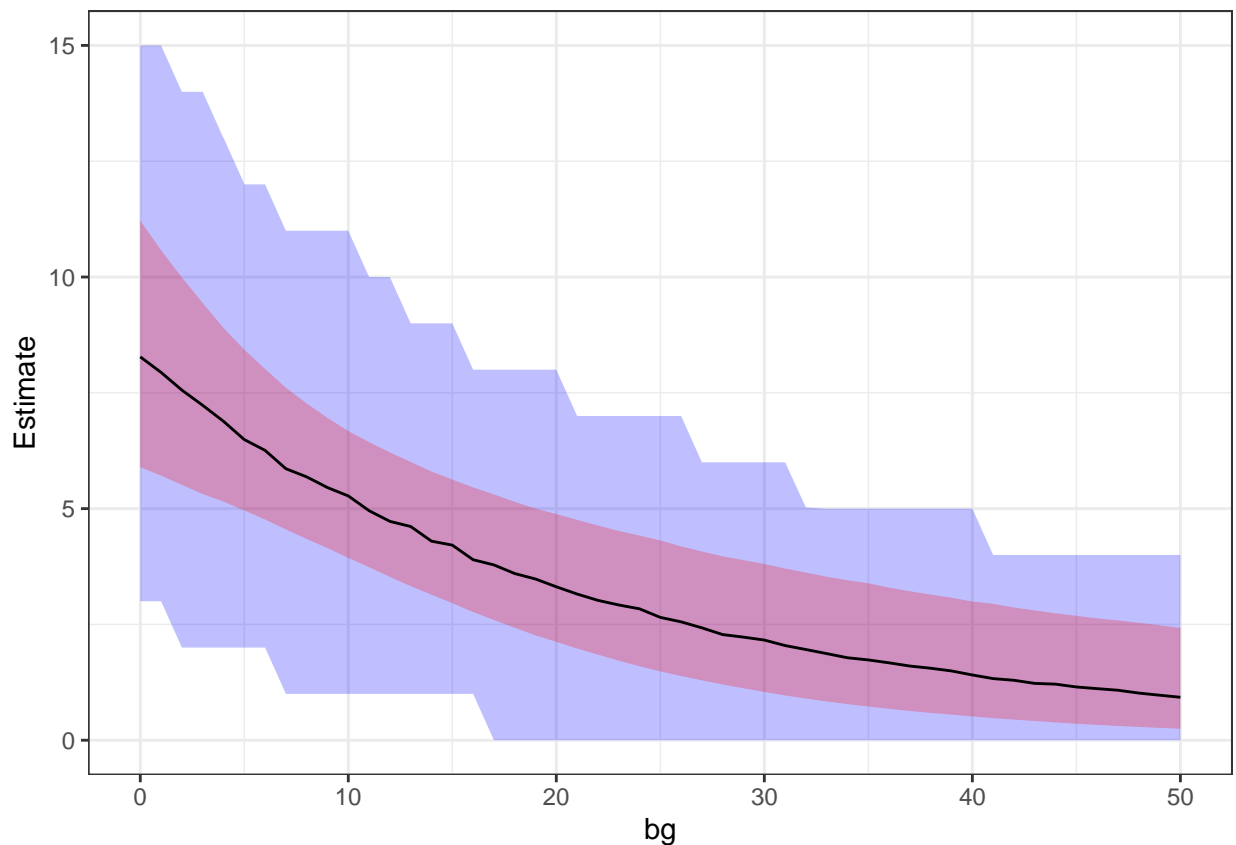
```
# saman voisi tehdä käyttäen alla olevia funktiota:  
str(posterior_predict(fit_ants, newdata = newdata))
```

```
## int [1:4000, 1:51] 10 8 8 10 11 3 11 12 11 11 ...  
## - attr(*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : NULL
```

```
str(posterior_epred(fit_ants, newdata = newdata))
```

```
## num [1:4000, 1:51] 9.07 7.29 6.16 10.78 7.72 ...
```

```
ggplot(cbind(newobs, newdata), aes(y = Estimate, x = bg)) +  
  geom_ribbon(aes(ymin = Q2.5, ymax = Q97.5), fill = "blue", alpha = 0.25) +  
  geom_ribbon(  
    data = cbind(newdata, expected),  
    aes(ymin = Q2.5, ymax = Q97.5),  
    fill = "red",  
    alpha = 0.25  
  ) +  
  geom_line() +  
  theme_bw()
```



Yllä `fitted`-funktiolla saimme näytteitä posterioriodotusarvon jakaumasta, josta sitten funktio laskee automaattisesti odotusarvon ja 95 % posteriorivälin. Funktion oletusargumentteja muuttamalla voimme laskea myös mediaanin ja muita välejä, sekä vaihtaa havaintoskaalasta lineaarisen prediktorin asteikolle (saman voisi tehdä “manuaalisesti” `posterior_predict` funktion avulla kuten aiemmin). Huomaa jälleen, että tämä ei ole sama kuin että laskisimme lineaarisen prediktorin jakaumasta vastaavat tunnusluvut, jotka

sitten muuntaisimme odotusarvoskaalalle:

```
linpred <- posterior_linpred(fit_ants, newdata = newdata)
trans_linpred <- exp(linpred) #log-linkin käänteisfunktio
summary_fun <- function(x) {
  c(
    Estimate = mean(x),
    Est.Error = sd(x),
    Q2.5 = unname(quantile(x, 0.025)),
    Q97.5 = unname(quantile(x, 0.975))
  )
}
head(exp(t(apply(linpred, 2, summary_fun))))
```

```
##      Estimate Est.Error    Q2.5    Q97.5
## [1,] 8.187349  1.181913  5.894490 11.217198
## [2,] 7.815529  1.172987  5.711894 10.577161
## [3,] 7.460595  1.165044  5.511554  9.991111
## [4,] 7.121780  1.158192  5.314803  9.438693
## [5,] 6.798352  1.152541  5.151810  8.901082
## [6,] 6.489613  1.148195  4.960716  8.432249
```

```
head(t(apply(trans_linpred, 2, summary_fun)))
```

```
##      Estimate Est.Error    Q2.5    Q97.5
## [1,] 8.302329  1.3939073  5.894490 11.217198
## [2,] 7.915476  1.2676278  5.711894 10.577161
## [3,] 7.547996  1.1566245  5.511554  9.991111
## [4,] 7.198861  1.0601694  5.314803  9.438693
## [5,] 6.867097  0.9775336  5.151810  8.901082
## [6,] 6.551786  0.9079442  4.960716  8.432249
```

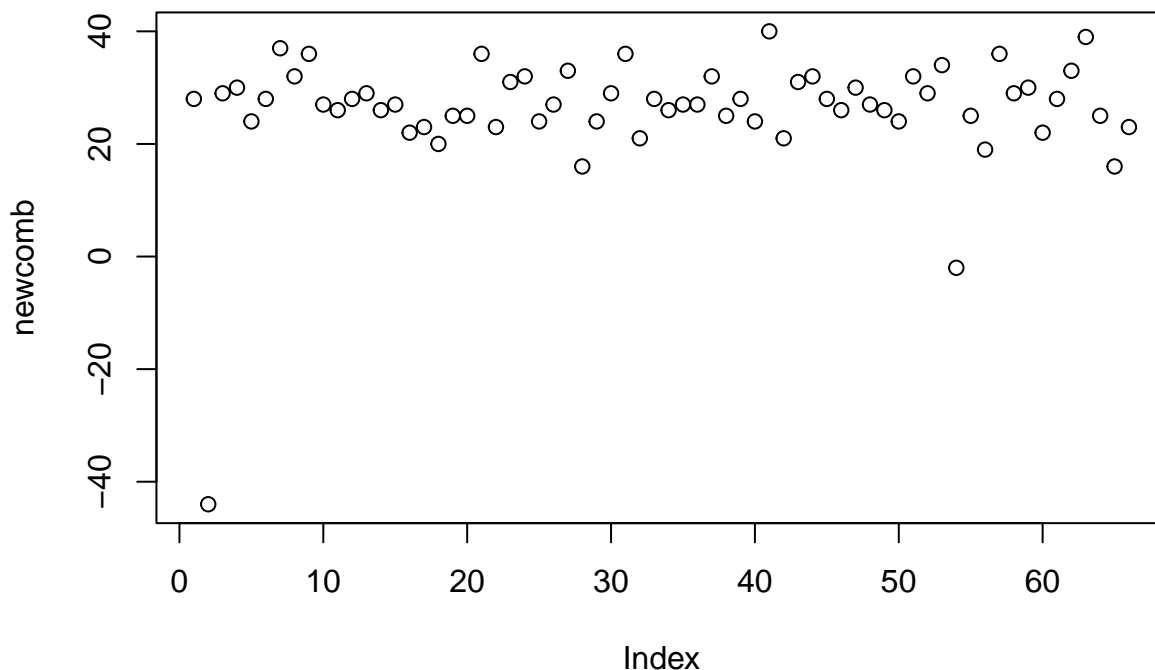
```
head(expected)
```

```
##      Estimate Est.Error    Q2.5    Q97.5
## [1,] 8.302329  1.3939073  5.894490 11.217198
## [2,] 7.915476  1.2676278  5.711894 10.577161
## [3,] 7.547996  1.1566245  5.511554  9.991111
## [4,] 7.198861  1.0601694  5.314803  9.438693
## [5,] 6.867097  0.9775336  5.151810  8.901082
## [6,] 6.551786  0.9079442  4.960716  8.432249
```

Esimerkki 18: Mallinsopivuuden tarkastelu posterioriennustejakauman avulla

Tutkitaan klassista Newcombin valonnopeuden määrittelyyn liittyvää koetta. Aineisto löytyy paketista MASS:

```
data(newcomb, package = "MASS")
plot(newcomb)
```



Mittaukset ovat skaalattuja valonnopeuksia, oikean arvon ollessa 33.02 (katso lisätietoja `?newcomb`).

Oletetaan että mittaukset y noudattavat normaalijakaumaa tuntemattomalla odotusarvolla μ ja varianssilla σ^2 (huomaa varianssiparametrisaatio keskihajonnan sijaan). Määrittelemällä prioriksi $p(\mu, \sigma^2) \propto \sigma^{-2}$, saamme analyttisen muodon posteriorille $p(\mu, \sigma^2)$:

$$\begin{aligned}
 p(\mu, \sigma^2 | y) &\propto (\sigma^2)^{-\left(\frac{n}{2}+1\right)} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right) \\
 &= (\sigma^2)^{-\left(\frac{n}{2}+1\right)} \exp\left(-\frac{1}{2\sigma^2} \left[\sum_{i=1}^n (y_i - \bar{y})^2 + n(\bar{y} - \mu)^2 \right]\right) \\
 &= (\sigma^2)^{-\left(\frac{n}{2}+1\right)} \exp\left(-\frac{1}{2\sigma^2} [(n-1)s^2 + n(\bar{y} - \mu)^2]\right) \\
 &= (\sigma^2)^{-\left(\frac{n-1}{2}+1\right)} \exp\left(-\frac{1}{2\sigma^2} (n-1)s^2\right) (\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{n}{2\sigma^2} (\bar{y} - \mu)^2\right) \\
 &= p(\sigma^2 | y) p(\mu | \sigma^2, y),
 \end{aligned}$$

missä

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$$

on otosvarianssi, \bar{y} on otoskeskiarvo ja n on mittausten lukumäärä. Varianssin σ^2 marginaaliposteriori on nyt [käänteinen Gamma-jakauma](#) $\text{Inv-Gamma}((n-1)/2, (n-1)s^2/2)$ (tunnetaan myös nimellä skaalattu käänteinen χ^2 -jakauma). Vastaavasti nähdään että ehdollinen posteriori $p(\mu | \sigma^2, y)$ on normaalijakauma $N(\bar{y}, \sigma^2/n)$.

Simuloidaan nyt $i = 1, \dots, 1000$ näytettä tästä posteriorista simuloimalla ensin $(\sigma^2)^i \sim p(\sigma^2|y)$ ja sitten $\mu^i \sim p(\mu|(\sigma^2)^i, y)$:

```
y <- newcomb
n <- length(y)

nsim <- 1000
# Posteriorijakauma p(sigma2)p(mu | sigma2)
# (saatu käyttäen prioria p(mu, sigma2) propto 1/sigma^2)
sigma2_sim <- (n - 1) * var(y) / rchisq(nsim, df = n - 1)
mu_sim <- rnorm(nsim, mean(y), sigma2_sim / n)
```

Nyt jokaista $(\mu^i, (\sigma^2)^i)$ paria kohden voimme simuloida uuden n havainnon kokoisen pseudoaineiston \tilde{y}_i ja laskea näistä erilaisia tunnuslukuja. Valitaan tässä testisuure $\frac{1}{n} \sum (\tilde{y} - \mu_i)^2$ ja tunnusluku $\min(\tilde{y})$. Lasketaan samat myös oikeasta aineistosta (korvaamalla \tilde{y} havaintodatalla y):

```
# pseudohavainnot
yrep <- matrix(NA, n, nsim)
# testisuure mean((y - mu)^2) replikaatioiden avulla laskettuna
var_rep <- numeric(nsim)
# testisuure datasta laskettuna
var_data <- numeric(nsim)
# testitunnusluku, aineiston (oikean tai simuloidun) minimiarvo
min_rep <- numeric(nsim)
min_data <- min(y)

for(i in 1:nsim){
  yrep[, i] <- rnorm(n, mu_sim[i], sqrt(sigma2_sim[i]))
  var_rep[i] <- mean((yrep[, i] - mu_sim[i])^2)
  var_data[i] <- mean((y - mu_sim[i])^2)
  min_rep[i] <- min(yrep[, i])
}
```

Näiden avulla voimme laskea Bayes “p-arvot” jotka, ehdolla mallimme, kertovat todennäköisyyden havaita yhtä äärimmäinen tunnusluvun arvo kuin mitä oikeasti havaitsimme:

```
mean(min_rep >= min_data)
```

```
## [1] 1
```

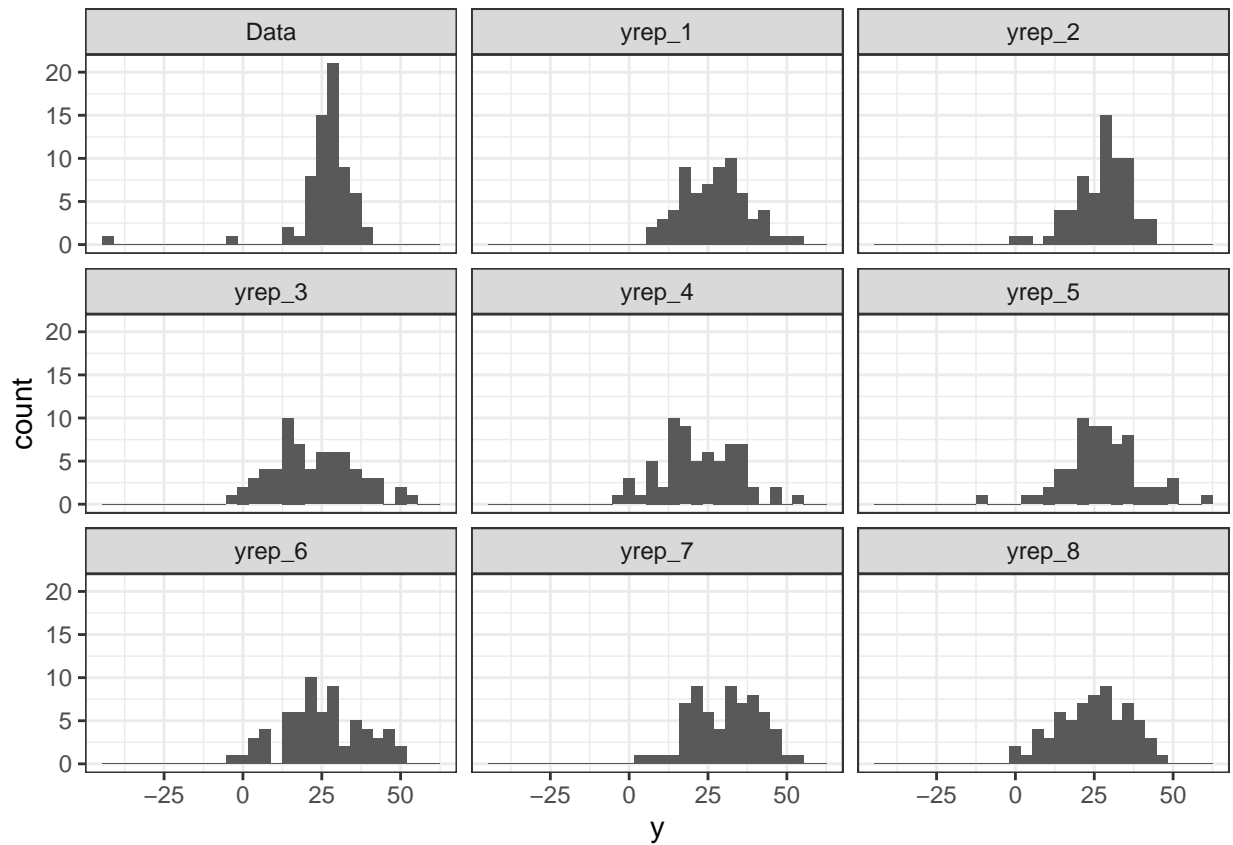
```
mean(var_rep >= var_data)
```

```
## [1] 0.471
```

Piirretään vielä kuvia, ensin oikean datan ja 8 replikaatiodatan histogrammit:

```
d <- data.frame(
  y = c(y, yrep[, 1:8]),
  replication = rep(c("Data", paste0("yrep_", 1:8)), each = n)
)
ggplot(d, aes(y)) +
  geom_histogram() +
  facet_wrap(~replication) +
  theme_bw()
```

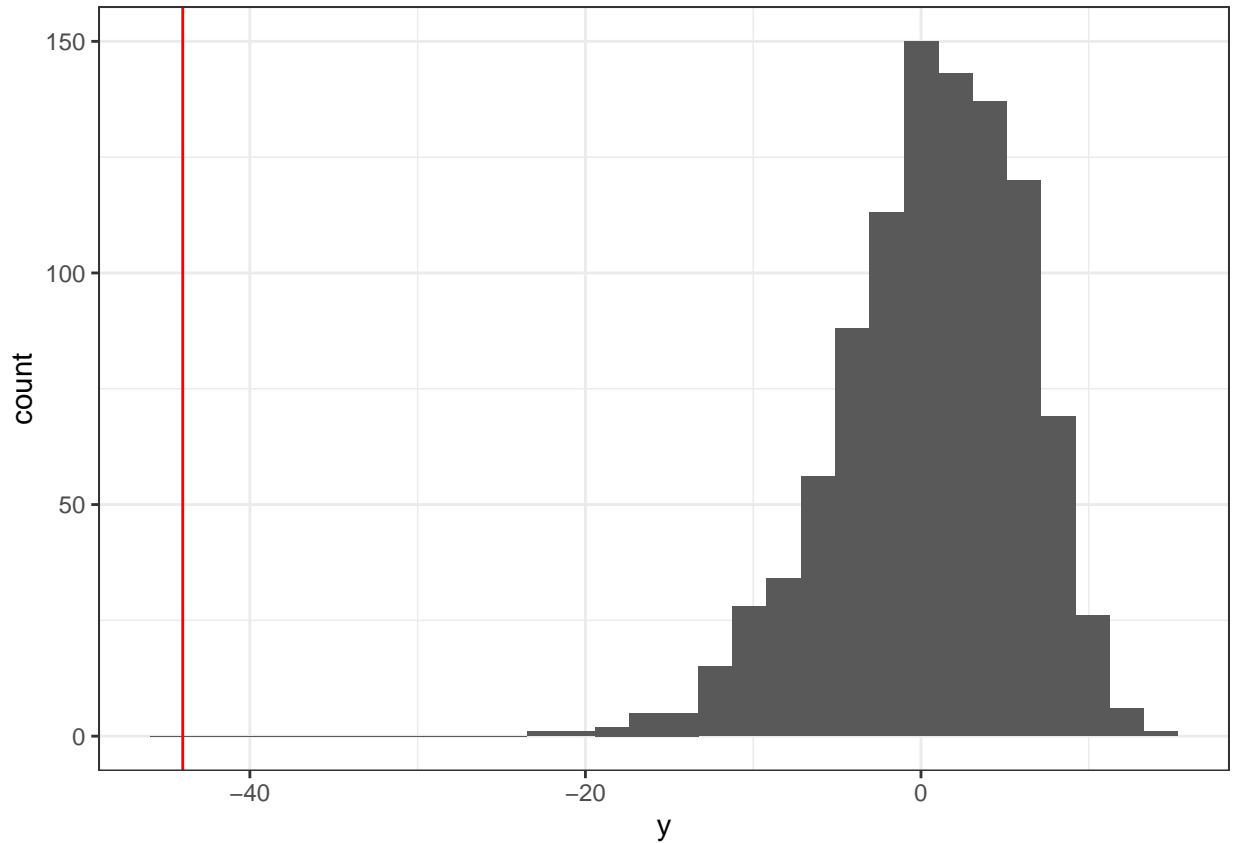
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Replikaatioaineistoista laskettujen minimiarvojen histogrammi ja datan minimi punaisena pystyviivana:

```
d <- data.frame(y = min_rep)
ggplot(d, aes(y)) +
  geom_histogram() +
  geom_vline(aes(xintercept = min_data), colour = "red") +
  theme_bw()
```

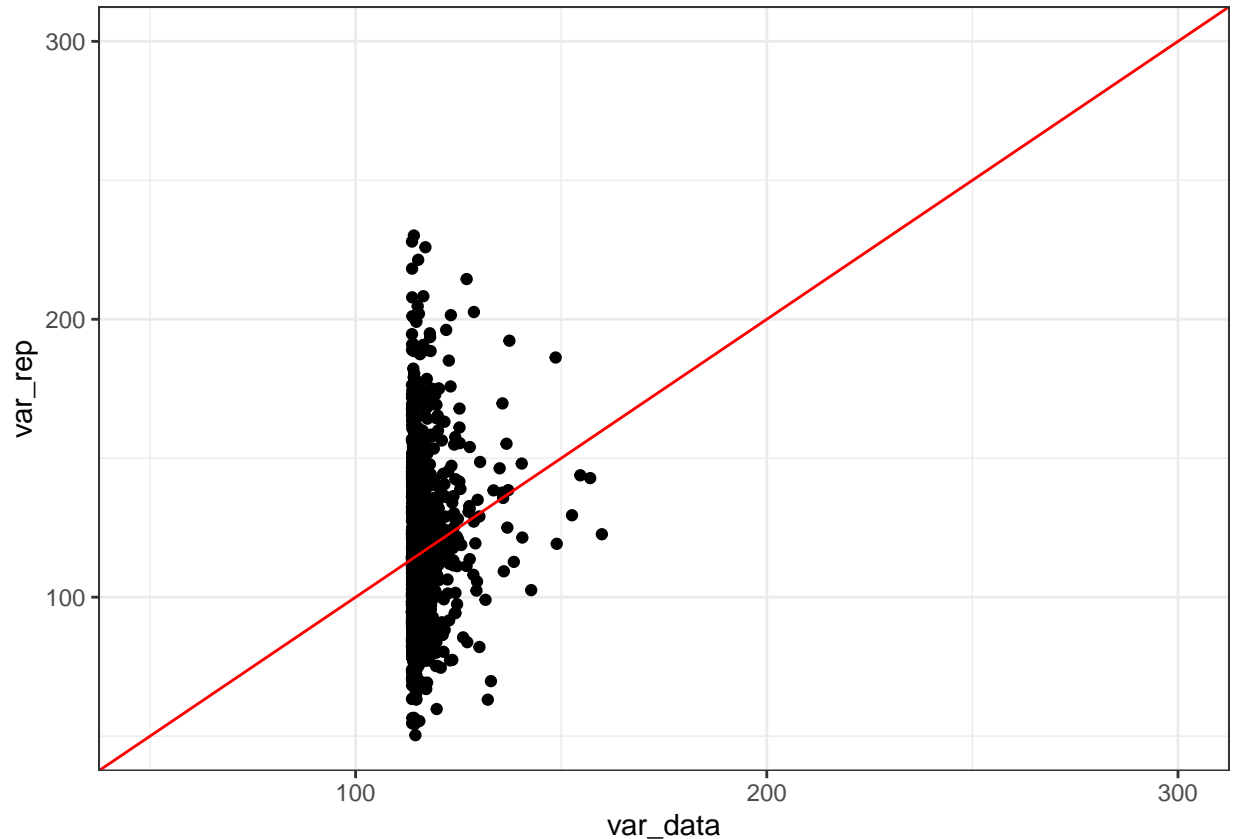
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Ja vielä jokaista posteriorinäytettä vastaavat varianssitunnusluvut dataa (x-akseli) ja pseudohavaintoja (y-akseli) käyttäen:

```
d <- data.frame(var_rep = var_rep, var_data = var_data)
ggplot(d, aes(x = var_data, y = var_rep)) +
  geom_point() +
  theme_bw() +
  geom_abline(aes(slope = 1, intercept = 0), colour = "red") +
  xlim(c(50, 300)) +
  ylim(c(50, 300))
```

Warning: Removed 1 rows containing missing values (geom_point).



Esimerkki 19: Posterioriennustejakauma mallinvalinnan apuna

Toistetaan [bayesplot-paketin esimerkki](#) posterioriennustetarkasteluista: Tarkoituksena on tutkia torakoiden torjunnan tehokkuutta (Gelman ja Hill, 2007). Aineisto löytyy paketista `rstanarm` ja se sisältää seuraavat muuttujat:

```
data("roaches", package = "rstanarm")
head(roaches)
```

```
##      y roach1 treatment senior exposure2
## 1 153 308.00         1     0 0.800000
## 2 127 331.25         1     0 0.600000
## 3   7   1.67         1     0 1.000000
## 4   7   3.00         1     0 1.000000
## 5   0   2.00         1     0 1.142857
## 6   0   0.00         1     0 1.000000
```

Muuttuja `y` kertoo kiinniotettujen torakoiden määrän, `roach1` kertoo kiinniotettujen torakoiden määrän ennen uutta “käsittelyä” `treatment` (1 jos käsittely tehty, 0 muuten), `senior` kertoo asuuko talossa vain vanhuksia ja `exposure2` sisältää tiedon montako päivää ansoja käytettiin. Sovitetaan aineistoon Poisson-regressiomalli `brms`-pakettia käyttäen:

```
roaches$roach100 <- roaches$roach1 / 100
fit_poisson <- brm(
  y ~ roach100 + treatment + senior + offset(log(exposure2)),
  family = poisson,
  data = roaches,
  refresh = 0
```

```
)
```

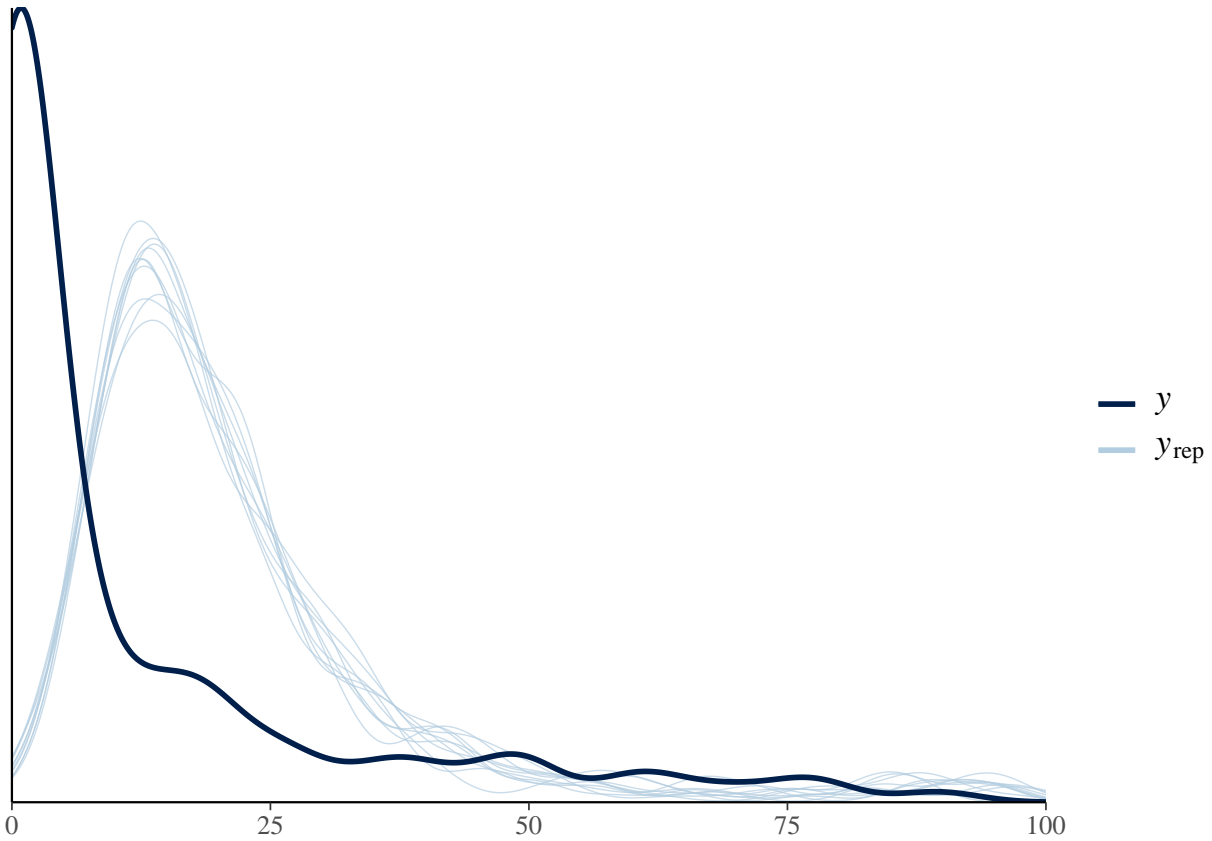
```
fit_poisson
```

```
## Family: poisson
## Links: mu = log
## Formula: y ~ roach100 + treatment + senior + offset(log(exposure2))
## Data: roaches (Number of observations: 262)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      3.09      0.02   3.05   3.13 1.00    4075    3216
## roach100        0.70      0.01   0.68   0.72 1.00    3985    2823
## treatment     -0.52      0.03  -0.57  -0.47 1.00    3934    2861
## senior         -0.38      0.03  -0.45  -0.31 1.00    3607    2499
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Verrataan posterioriennustejakaumasta saatuja näyttöjä aineistoomme:

```
pp_check(fit_poisson) + xlim(c(0, 100))
```

```
## Using 10 posterior draws for ppc type 'dens_overlay' by default.
## Warning: Removed 62 rows containing non-finite values (stat_density).
## Warning: Removed 23 rows containing non-finite values (stat_density).
```

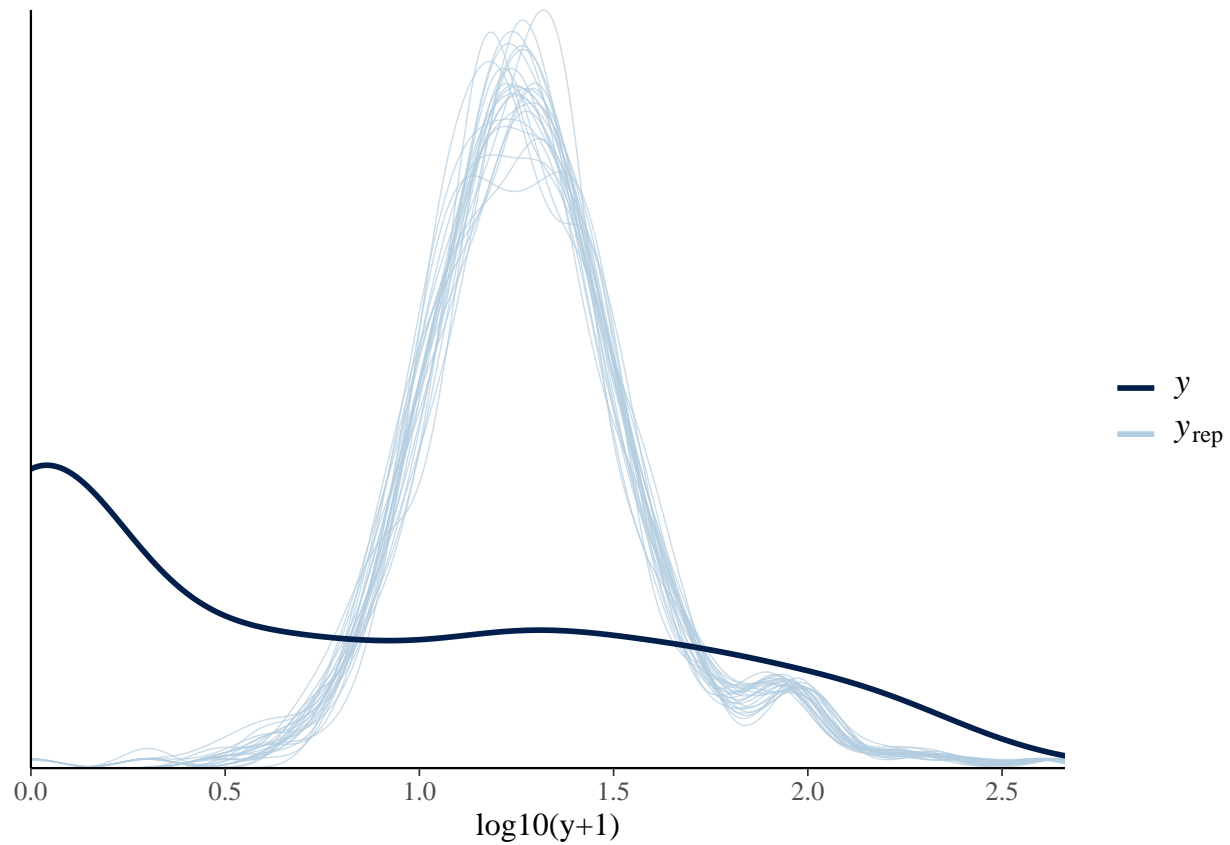


Funktio `pp_check` palauttaa `ggplot`-objektin johon lisäämme x-akselin katkaisevan `xlim`-komponentin. Huomaamme että aineistossa näyttää olevan enemmän nollia verrattuna posterioriennustejakauman tuottamiin näytteisiin. Piirretään vielä sama kuva käyttäen logaritmuunnosta ja `bayesplot`-pakettia:

```

yrep <- posterior_predict(fit_poisson)
idx <- sample(1:nrow(yrep), size = 25)
library(bayesplot)
ppc_dens_overlay(log10(roaches$y + 1), log10(yrep[idx,] + 1)) +
  xlab('log10(y+1)')

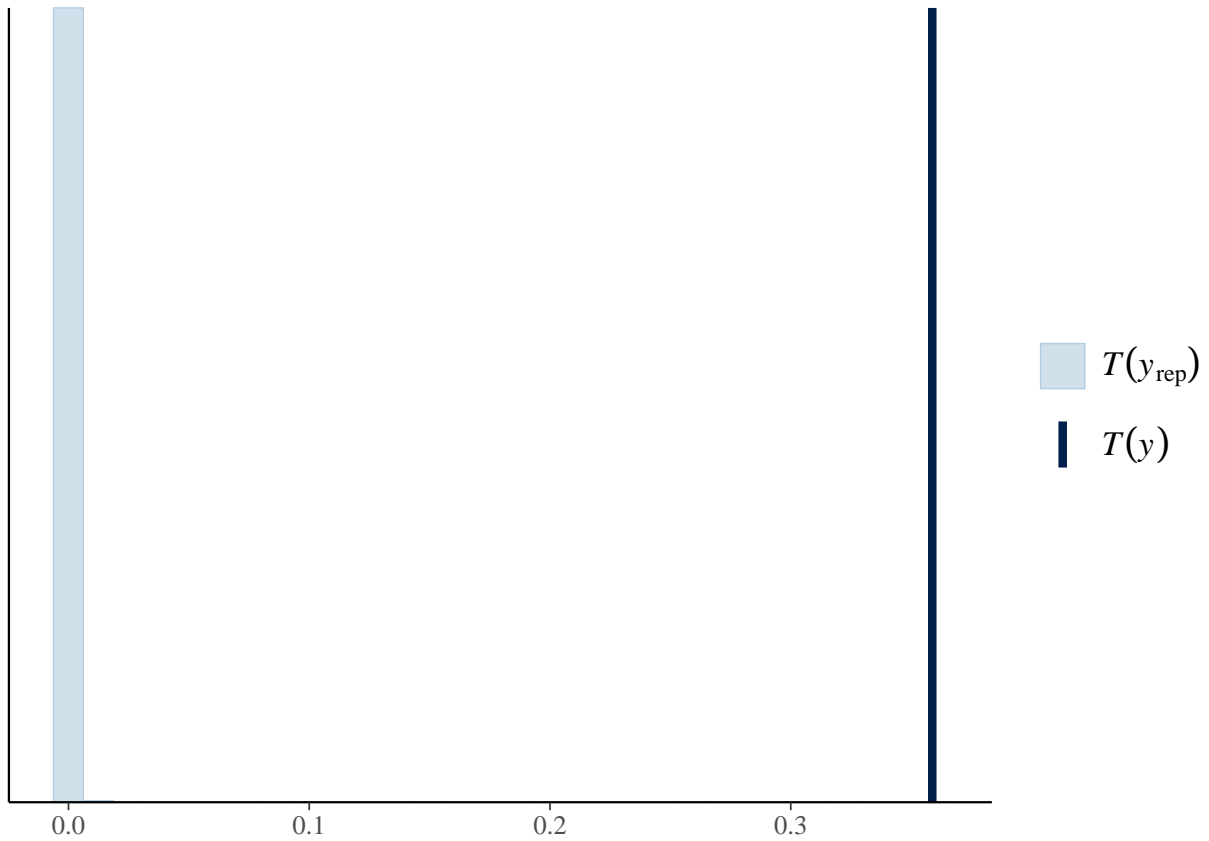
```



Verrataan myös nollien osuutta:

```
ppc_stat(roaches$y, yrep, stat = function(x) mean(x == 0))
```

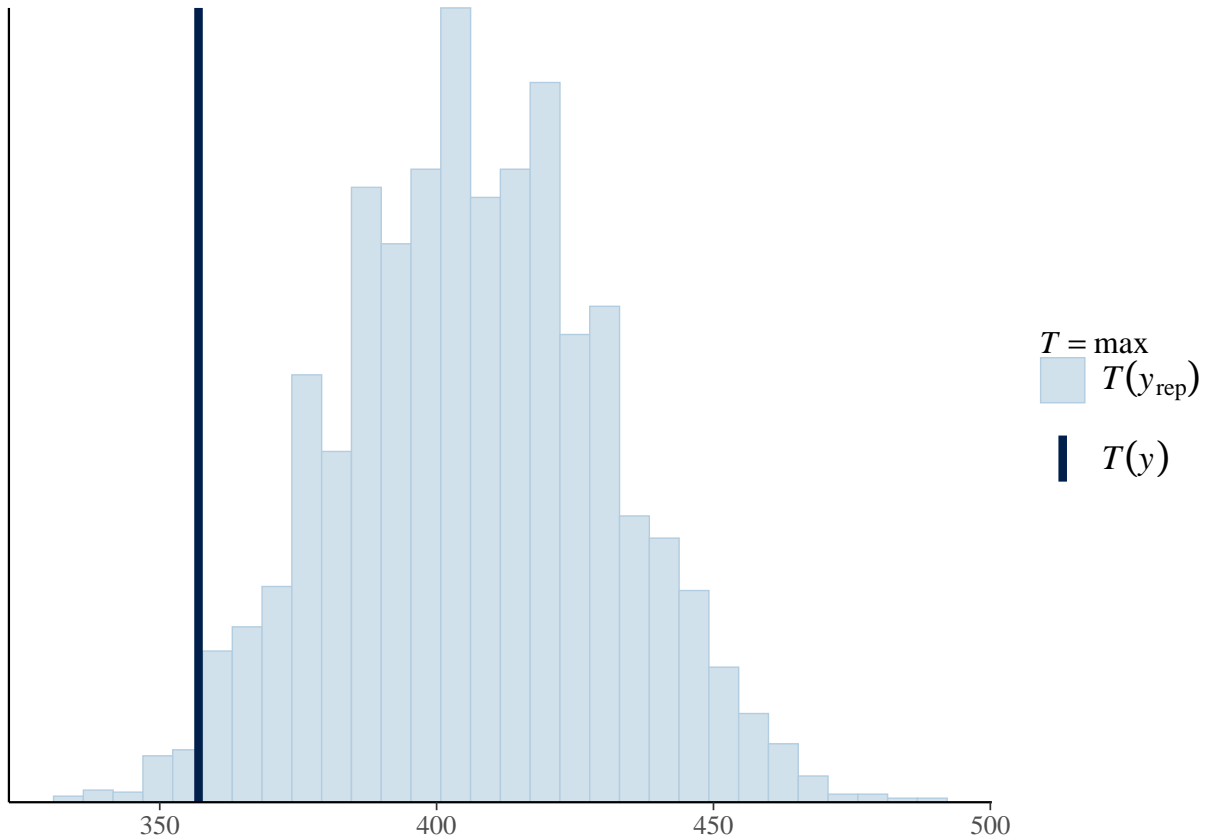
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Ja suurimman torakkamäärän jakaumaa:

```
ppc_stat(roaches$y, yrep, stat = max)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
max(roaches$y)
```

```
## [1] 357
```

```
max(yrep)
```

```
## [1] 491
```

Katsotaan miten tulokset muuttuvat kun vaihdetaan negatiiviseen binomijakaumaan:

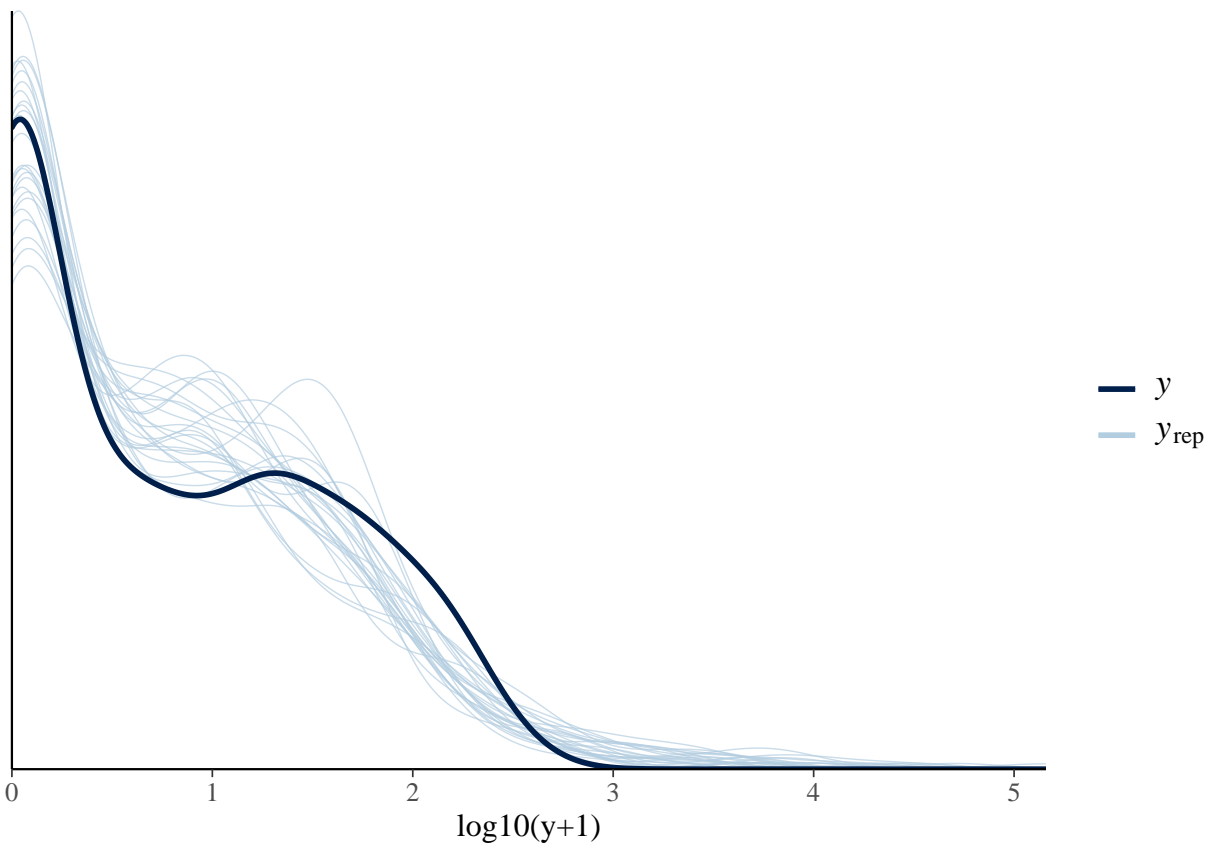
```
fit_nb <- brm(
  y ~ roach100 + treatment + senior + offset(log(exposure2)),
  family = negbinomial,
  data = roaches, refresh = 0
)
```

```
fit_nb
```

```
## Family: negbinomial
## Links: mu = log; shape = identity
## Formula: y ~ roach100 + treatment + senior + offset(log(exposure2))
## Data: roaches (Number of observations: 262)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept 2.85 0.23 2.42 3.30 1.00 5237 3070
## roach100 1.33 0.26 0.86 1.88 1.00 4122 2683
```

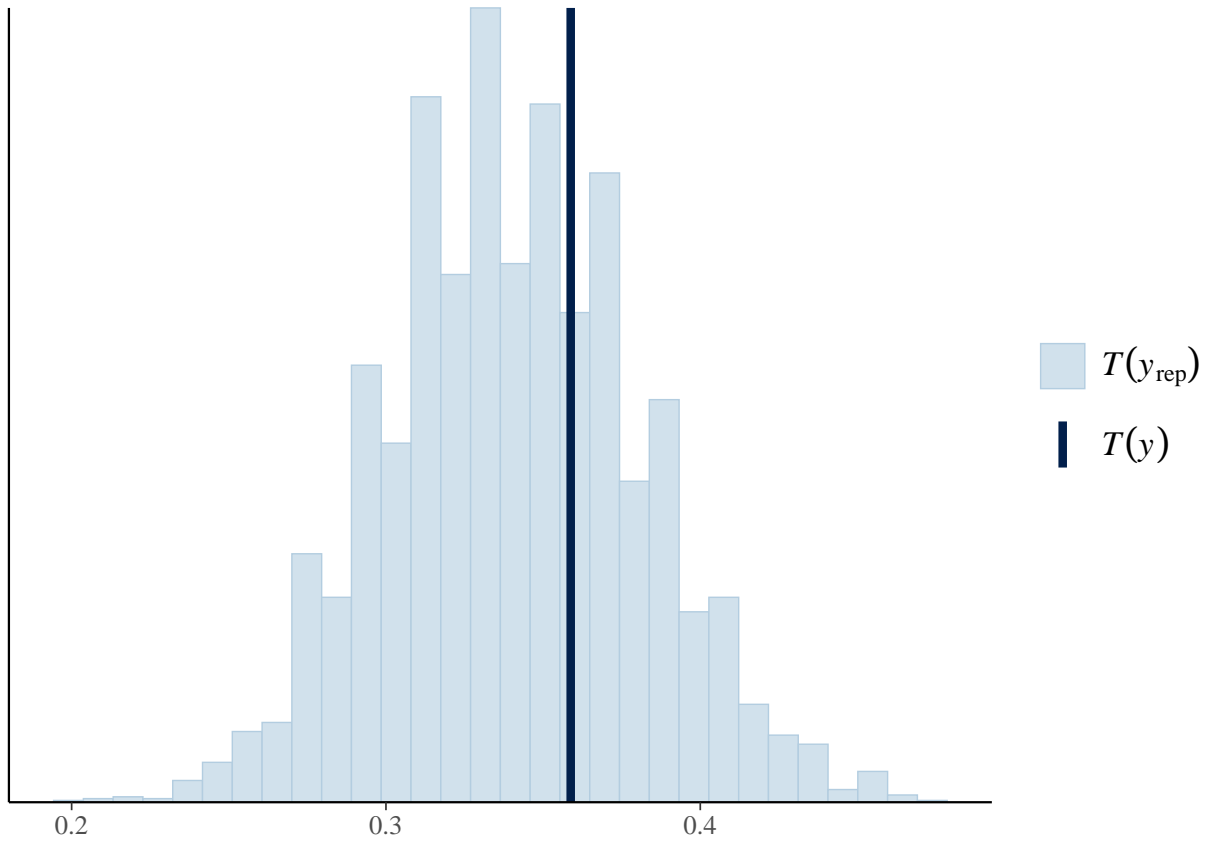
```
## treatment    -0.79      0.25    -1.27    -0.31 1.00     4838     3029
## senior       -0.33      0.28    -0.87     0.22 1.00     5453     3002
##
## Family Specific Parameters:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape    0.27      0.03    0.22    0.32 1.00     4608     2768
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
yrep <- posterior_predict(fit_nb)
ppc_dens_overlay(log10(roaches$y + 1), log10(yrep[idx,] + 1)) +
  xlab('log10(y+1)')
```



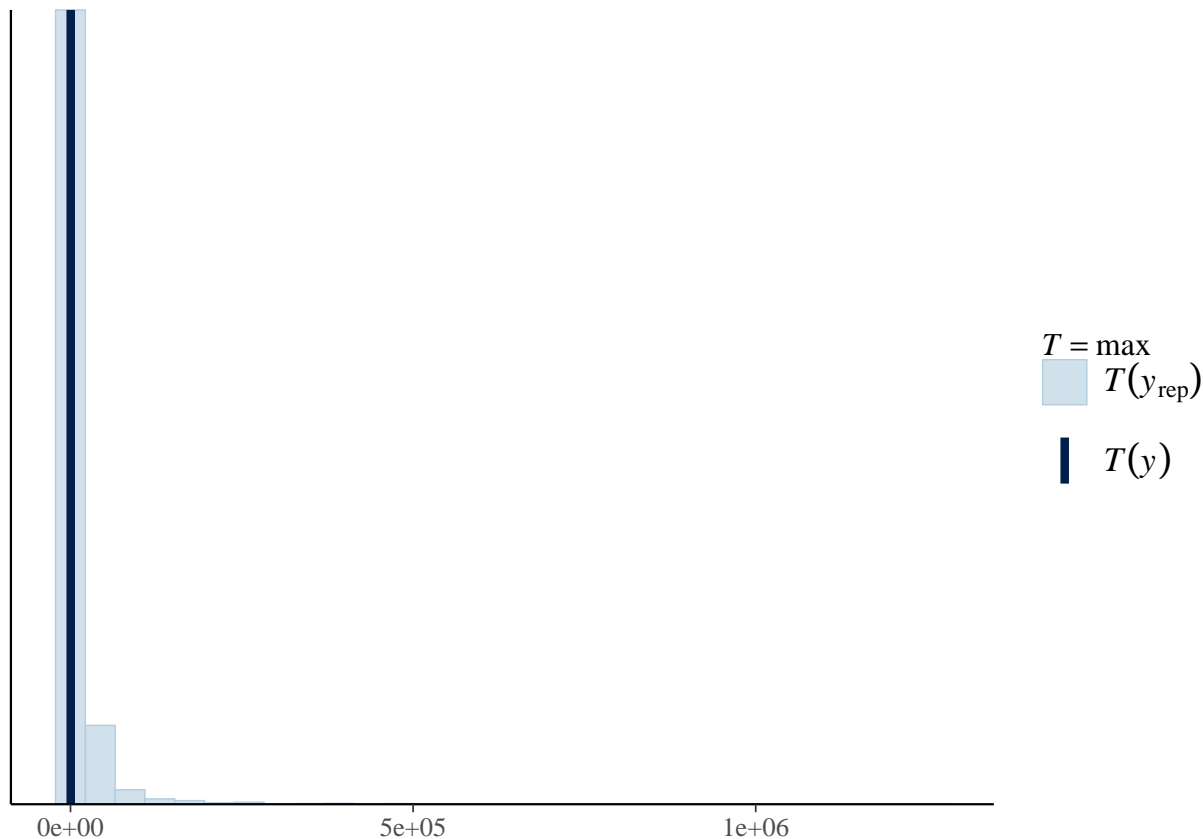
```
ppc_stat(roaches$y, yrep, stat = function(x) mean(x == 0))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ppc_stat(roaches$y, yrep, stat = max)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
max(roaches$y)
```

```
## [1] 357
```

```
max(yrep)
```

```
## [1] 1260734
```

Näyttää paremmalta nollien osuuden suhteen, mutta maksimiarvon jakauma on todella pitkähäntäinen.

4.4 Mallinvalinta ja -vertailu

Kuten SU-estimoinnin tapauksessa, mallinvalinta ja mallien vertailu on tärkeä (ja usein hankala) osa tilastollista analyysiä. Sisältötulkintojen ja prediktiiivisten jakaumien lisäksi Bayes-mallienkin kohdalla voidaan käyttää erilaisia informaatiokriteereitä kuten AIC ja BIC (on myös olemassa nimenomaan Bayes-käyttöön kehitettyjä kriteereitä kuten WAIC ja DIC). Lisäksi malleja voidaan vertailla käyttäen ns. Bayes-faktoreita sekä ristiinvalidointitekniikkoja käyttäen. Näistä enemmän Bayes tilastotiede 2 - kurssilla. Tutkitaan kuitenkin erästä ristiinvalidointitekniikkaa hieman lähemmin.

Leave-one-out-ristiinvalidointi (LOO) on suosittu laskennallinen menetelmä mallinvalinnassa ja -vertailussa varsinkin tilanteessa, jossa kiinnostuksen kohteena on mallin ennustuskyky. Menetelmä on kuitenkin laskennallisesti erittäin raskas Bayes-tilanteessa, sillä meidän pitäisi periaatteessa sovittaa malli n kertaa (jossa n on datapisteiden määrä). Varsinkin monimutkaisten Bayes-mallien kohdalla tämä olisi kohtuuttoman hidasta. Ns. PSIS-LOO-menetelmä (Vehtari, Gelman, ja Gabry, 2017) approksimoi oikeaa LOO-menetelmää tärkeyspainotuksen avulla siten että malli tarvitsee sovittaa vain kerran (tai muutamia kertoja). Menetelmä on suosittu varsinkin Stan-ohjelmiston käyttäjien keskuudessa, ja esimerkiksi `brms`-paketti sisältää vastaavan `loo`-metodin. Perinteisestä ristiinvalidoinnista poiketen, PSIS-LOO menetelmässä pyritään vertaamaan suoraan prediktiiivisen posteriorijakauman tiheysfunktion odotusarvoja, suuremman arvon ollessa parempi

(lisätietoa em. artikkelissa):

$$\text{elpd}_{\text{loo}} = \sum_{i=1}^n \log p(y_i | y_{-i}),$$

missä y_{-i} viittaa dataan ilman havaintoa y_i (n on datan koko). Karkeasti ajatellen tätä voi pitää samankaltaisena mallin hyvyyden mittarina kuin klassiset Bayes-analyysissä käytetyt informaatiokriteerit WAIC ja DIC, mutta LOO voi olla robustimpi esim. tilanteessa, jossa on käytetty heikosti informatiivisia prioreita. Lisäksi PSIS-LOO approksimaation sivutuotteena saadaan estimaateille myös keskivirheet, joiden avulla voidaan arvioida eroavatko mallit merkittävästi toisistaan (joskin nämä keskivirheet tyypillisesti aliarvioivat todellista epävarmuutta).

Edellämainittu LOO ei sellinaan ole kuitenkaan usein mielekäs mittari pitkittäisaineistomallien tai muiden aikasarjamallien yhteydessä jos päämääränä on esim. ennustaa uuden henkilön arvoja tai tulevaisuuden havaintoja (jälkimmäiseen ongelmaan luonnollinen valinta on leave-future-out-ristiinvaldointi, johon on myös kehitetty PSIS-LOO:n kaltaisia approksimaatioita).

Esimerkki 20: LOO

Funktion `loo.brmsfit` dokumentaation esimerkki, jossa sovitetaan Bayes-regressiomalli sekä malli jossa jokaiselle koehenkilölle on oma satunnainen vakiotermi:

```
# a) tavallinen lineaarinen Bayes-malli
fit_a <- brm(
  rating ~ treat + period + carry,
  data = inhaler,
  refresh = 0
)
loo_a <- loo(fit_a) # vaihtoehtoisesti fit1 <- add_criterion(fit1, "loo")

# b) malli jossa satunnainen vakiotermi joka koehenkilölle
fit_b <- brm(
  rating ~ treat + period + carry + (1 | subject),
  data = inhaler,
  refresh = 0
)
loo_b <- loo(fit_b)

# verrataan malleja
loo_compare(loo_a, loo_b)

##           elpd_diff se_diff
## fit_b    0.0         0.0
## fit_a  -7.6         4.7
```

Malli, jossa on mukana satunnaistermi näyttää hieman paremmalta (`loo_compare` vertaa `elpd`-muuttujien erotusta mallien välillä, mitä suurempi `elpd` sen parempi), mutta toisaalta suurehko keskivirhe kertoo että mallit eivät poikkea toisistaan kovinkaan paljoa. Varoitukset suurista `pareto_k`-arvoista kertovat siitä että PSIS-LOO menetelmä kohtasi ongelmia joidenkin havaintojen kohdalla koska posteriorit $p(\theta|y)$ ja $p(\theta|y_{-1})$ poikkeavat voimakkaasti toisistaan, joka voi viitata kyseisen havainnon poikkeuksellisuuteen tai ongelmiin mallin rakenteessa.

4.5 Monimutkaisempia brms-malleja

Edellä käsitellyjä `brms`-malleja voi laajentaa mm. mallintamalla myös havaintojakauman lisäparametreja, kuten normaalijakauman keskihajontaa tai negatiivisen binomijakauman dispersioparametria. Tämä onnistuu

lisäämällä useampi yhtälö brms-kutsuun bf-funktion sisään:

```
brm(
  bf(y ~ x + z + w, shape ~ x * w),
  family = negbinomial(link = "log", link_shape = "log"),
  data = data.frame(y, x, z, w)
)
```

Yllä myös määriteltiin linkkifunktiot eksplisiittisesti. Yksi brms-paketin vigneteistä käsittelee aihetta tarkemmin: https://cran.r-project.org/web/packages/brms/vignettes/brms_distreg.html

Voimme myös lisätä kaavaan sekamalleista tuttuja satunnaisefektejä, jossa | kertoo minkä muuttujan suhteen ryhmittely tehdään:

```
fit_mixed <- brm(
  count ~ zAge + zBase * Trt + (zAge | patient),
  data = epilepsy,
  family = poisson(),
  refresh = 0
)
```

Nyt jokaiselle potilaalle on oma vakiotermi $Intercept + Intercept_i$ ja regressiokerroin $\beta_{zAge} + \beta_{zAge,i}$ joissa jälkimmäiset potilaskohtaiset termit noudattavat nollakeskeisiä normaalijakaumia tuntemattomilla (estimoitavilla) keskihajonnoilla (oletuksena potilaskohtaiset termit annetaan myös korreloida):

```
summary(fit_mixed)
```

```
## Family: poisson
## Links: mu = log
## Formula: count ~ zAge + zBase * Trt + (zAge | patient)
## Data: epilepsy (Number of observations: 236)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Group-Level Effects:
## ~patient (Number of levels: 59)
##
```

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
## sd(Intercept)	0.58	0.07	0.46	0.74	1.00	1212	1857
## sd(zAge)	0.14	0.10	0.00	0.38	1.01	465	1042
## cor(Intercept,zAge)	0.21	0.50	-0.85	0.95	1.00	1798	1806
##							
## Population-Level Effects:							
##	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
## Intercept	1.77	0.12	1.53	2.01	1.00	1024	1556
## zAge	0.09	0.09	-0.09	0.26	1.00	1252	2061
## zBase	0.68	0.12	0.44	0.91	1.00	1060	1602
## Trt1	-0.26	0.16	-0.58	0.05	1.00	1009	1973
## zBase:Trt1	0.07	0.16	-0.26	0.38	1.00	1100	1594
##							

```
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Näiden satunnaistermien estimaatit saamme komennolla ranef (vastaavasti “kiinteät” efektit saadaan komennolla fixef):

```
head(ranef(fit_mixed)$patient)
```

```
## , , Intercept
##
##      Estimate Est.Error      Q2.5      Q97.5
## 1 -0.054010590 0.2707692 -0.6032880 0.4676098
## 2 -0.047413202 0.2707884 -0.5854379 0.4698597
## 3 -0.086023018 0.3038310 -0.7081310 0.5048838
## 4 -0.100354337 0.3056424 -0.7165202 0.4799416
## 5  0.046605760 0.2790092 -0.4983481 0.5907933
## 6  0.008691392 0.2247722 -0.4310939 0.4405584
##
## , , zAge
##
##      Estimate Est.Error      Q2.5      Q97.5
## 1 -0.0089017158 0.1498541 -0.3388695 0.2946387
## 2 -0.0074939003 0.1486474 -0.3417875 0.3286934
## 3 -0.0063292320 0.1507568 -0.3218861 0.3337767
## 4 -0.0163771650 0.1327055 -0.3278511 0.2608656
## 5  0.0016782711 0.1533245 -0.3411168 0.3608913
## 6  0.0004583859 0.1493012 -0.3305557 0.3238125
```

brms-paketilla voi sovittaa myös epälineaarisia malleja. Nämä vaativat usein tarkkuutta priorijakaumien ja joskus MCMC:n alkuarvojen suhteen (argumentti `inits`) jotta malli ylipäättensä konvergoi. Linearisessa mallissa siis vasteen odotusarvon (tai jakaumaparametrin kuten sigman) lineaarinen prediktori on muotoa $\eta_j = \sum_{i=1}^k b_i x_{ji}$ kun taas epälineaarisessa mallissa η_j ei ole kertoimien b ja selittäjien x lineaarikombinaatio, vaan esimerkiksi muotoa $\eta_j = b_1 \exp(b_2 x_j)$.

Toistetaan brms-paketin oppaan [Estimating Non-Linear Models with brms](#) esimerkki. Simuloidaan ensin aineistoa:

```
set.seed(1)
b <- c(2, 0.75)
x <- rnorm(100)
y5 <- rnorm(100, mean = b[1] * exp(b[2] * x))
d5 <- data.frame(x = x, y = y5)
```

Epälineaarisen mallin prioreiden määrittelyssä käytämme argumenttia `nlpar` nimeämään mihin parametriin priorin liittyy:

```
prior1 <- prior(normal(1, 2), nlpar = "b1") +
  prior(normal(0, 2), nlpar = "b2")
```

Lineaarisista malleista poiketen nyt merkitsemme mallin kaavan yhtälönä jossa myös kertoimet ovat näkyvissä ($y \sim b_1 * \exp(b_2 * x)$), ja lisäksi määritämme mitkä parametrit mallinnetaan ($b_1 + b_2 \sim 1$, joka on lyhenne muodosta $b_1 \sim 1$, $b_2 \sim 1$, näille parametreille voisi siis antaa selittäjiä...). Lisäksi kerromme `brm`-funktiolle että kaava on epälineaarinen argumentilla `nl = TRUE`:

```
fit_nl <- brm(
  bf(y ~ b1 * exp(b2 * x), b1 + b2 ~ 1, nl = TRUE),
  data = d5,
  prior = prior1,
  refresh = 0
)
```

```
fit_nl
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: y ~ b1 * exp(b2 * x)
```

```

##          b1 ~ 1
##          b2 ~ 1
##    Data: d (Number of observations: 100)
##    Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## b1_Intercept    2.00    0.11    1.80    2.21 1.00    1643    2013
## b2_Intercept    0.73    0.04    0.66    0.81 1.00    1652    2108
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma         0.97    0.07    0.84    1.13 1.00    2587    2168
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

```

brms tukee myös monia muita laajennoksia, esimerkiksi ([moniulotteisia vasteita](#)), ja [monotonisia efektejä](#) järjestyksasteikollisille muuttujille. Myös puuttuvaa tietoa voidaan mallintaa käyttäen `mi()`-funktioita:

```

data("nhanes", package = "mice")
N <- nrow(nhanes)

# simple model with missing data
bform1 <- bf(bmi | mi() ~ age * mi(chl)) +
  bf(chl | mi() ~ age) +
  set_rescor(FALSE)

fit_mi <- brm(bform1, data = nhanes, chains = 1, refresh = 0)

```

```
summary(fit_mi)
```

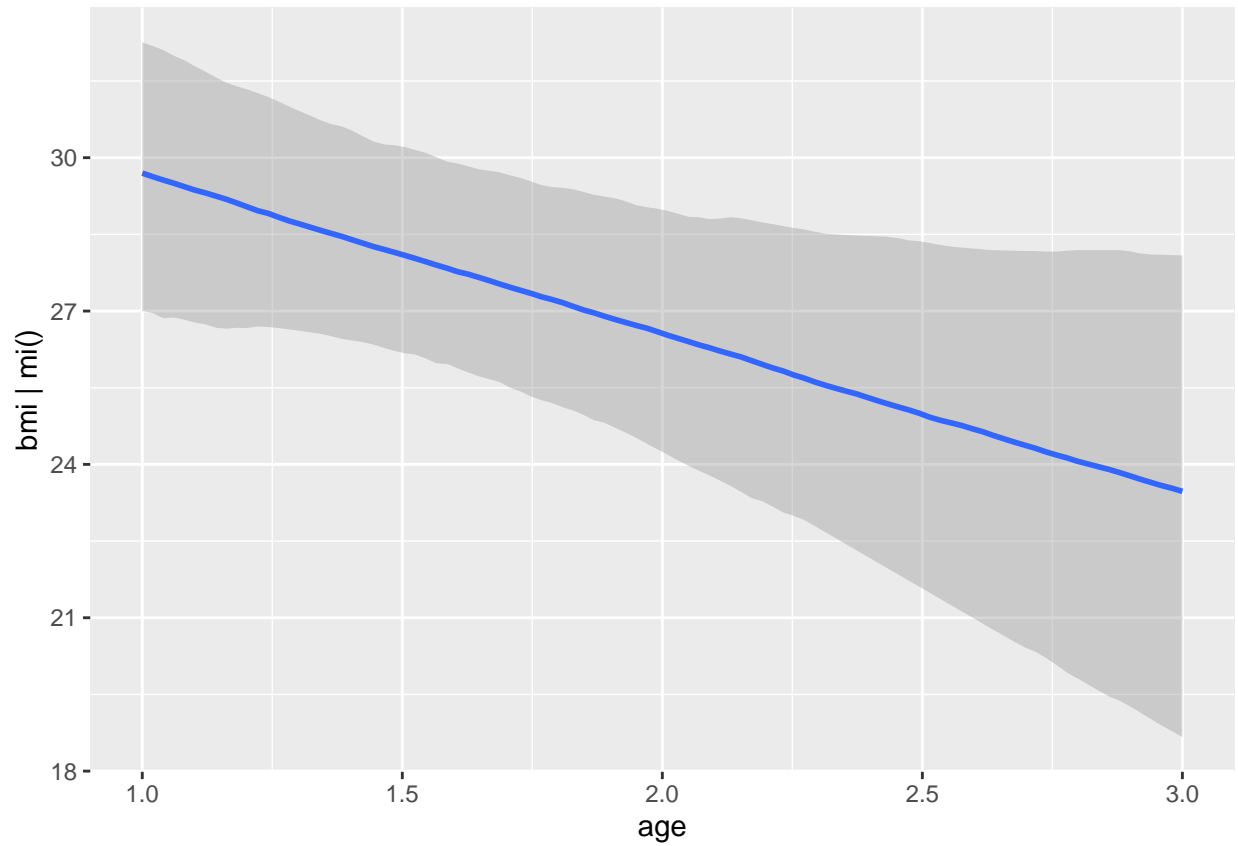
```

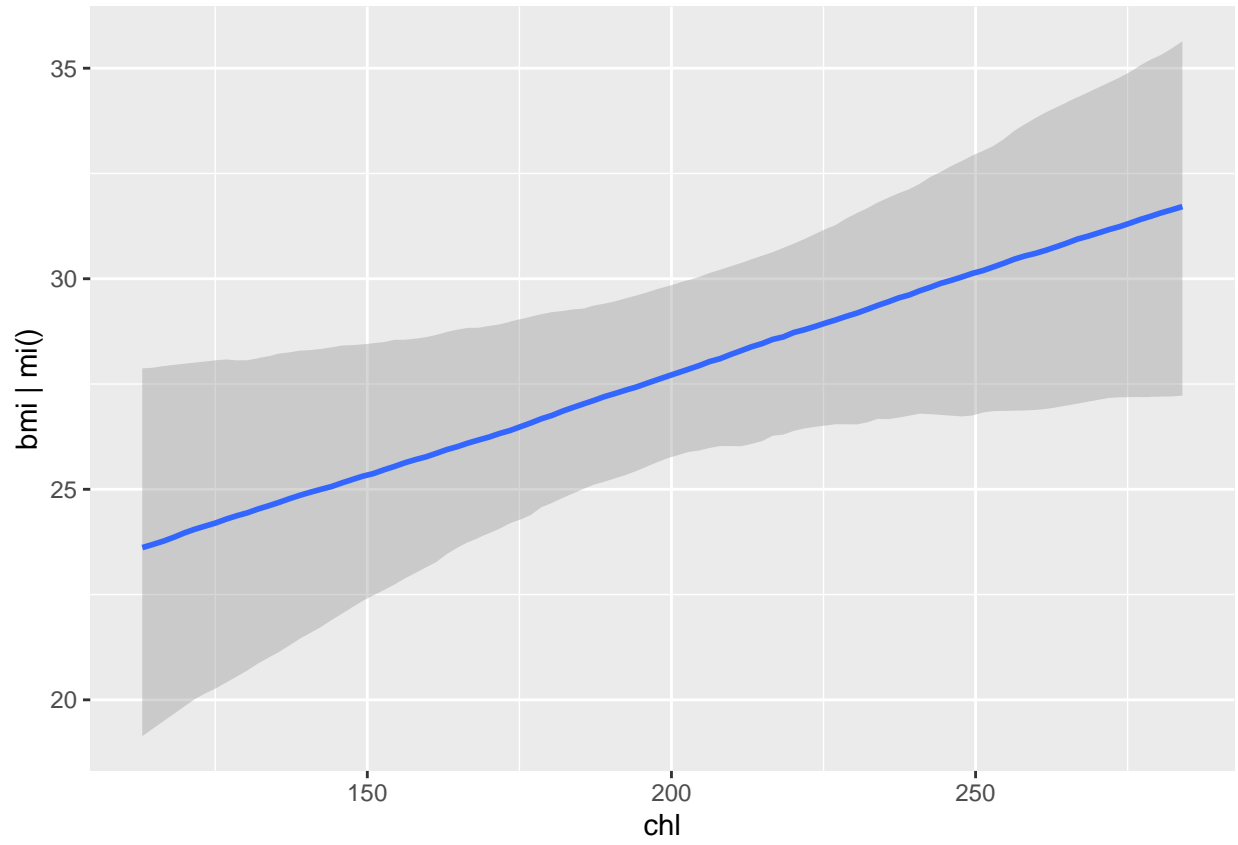
## Family: MV(gaussian, gaussian)
## Links: mu = identity; sigma = identity
##          mu = identity; sigma = identity
## Formula: bmi | mi() ~ age * mi(chl)
##          chl | mi() ~ age
##    Data: nhanes (Number of observations: 25)
##    Draws: 1 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 1000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## bmi_Intercept   13.68    8.42   -1.47   30.82 1.00    322    523
## chl_Intercept   141.56   24.83   90.33  192.44 1.00    641    586
## bmi_age         2.63    5.42   -8.61   12.76 1.00    289    419
## chl_age         28.53   12.91    3.79   54.93 1.00    622    552
## bmi_michl       0.10    0.04    0.01    0.18 1.00    324    501
## bmi_michl:age   -0.03    0.02   -0.07    0.02 1.00    288    502
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma_bmi      3.30    0.79    2.13    5.22 1.01    270    424

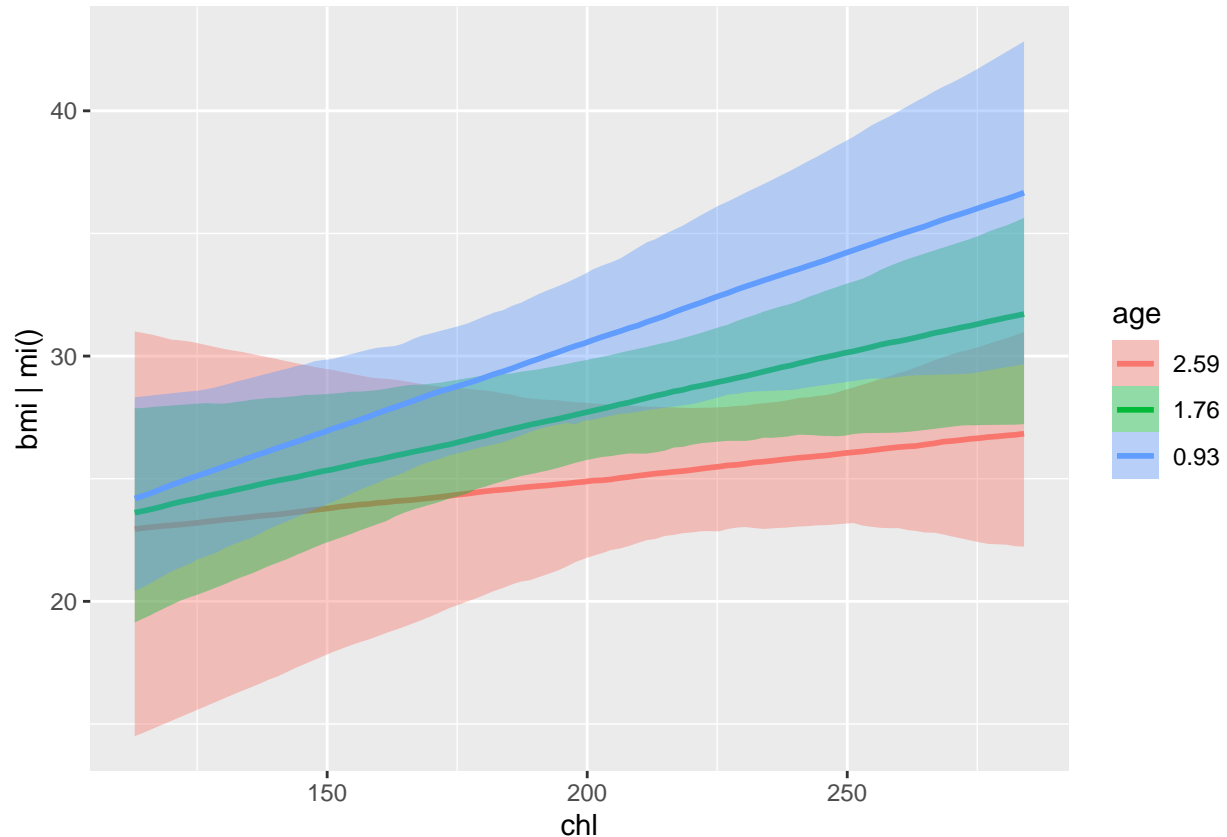
```

```
## sigma_ch1 39.78 7.51 28.18 57.13 1.00 532 563
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(conditional_effects(fit_mi, resp = "bmi"), ask = FALSE)
```







Määrittelimme mallissa “osa-mallin” muuttujalle `bmi` sekä muuttujalle `chl`. Merkintä `| mi()` kertoi että sen sijaan että poistaisimme datasta automaattisesti NA-arvoja sisältävät rivit (oletus), haluamme mallintaa puuttuvat havainnot: `chl`-muuttujan puuttuvat arvot generoidaan `chl ~ age`-mallin mukaisesti, kun taas `bmi`-muuttujalle käytämme `age`- ja `chl`-muuttujia (`chl`:n puuttuvat havainnot korvataan edellisen mallin mukaisilla arvoilla). [Katso lisää brms:n vignetestä.](#)

4.6 Stan-mallien kirjoittaminen

Tähän asti olemme käyttäneet mallien sovittamiseen `brms`-pakettia joka muodostaa käyttäjän antamista kaavoista Stan-kielellä kirjoitetun mallin, ja kutsuu sitten `rstan`-pakettia itse mallin sovittamiseen. Voimme tarkastella tätä Stan-mallia komennolla `stancode`:

```
y6 <- rnorm(10)
fit_code <- brm(y ~ 1, data = data.frame(y = y6), chains = 1, refresh = 0)

stancode(fit_code)
```

```
## // generated with brms 2.17.0
## functions {
## }
## data {
##   int<lower=1> N; // total number of observations
##   vector[N] Y; // response variable
##   int prior_only; // should the likelihood be ignored?
## }
## transformed data {
## }
```



```

## parameters {
##   real Intercept; // temporary intercept for centered predictors
##   real<lower=0> sigma; // dispersion parameter
## }
## transformed parameters {
##   real lprior = 0; // prior contributions to the log posterior
##   lprior += student_t_lpdf(Intercept | 3, 0.2, 2.5);
##   lprior += student_t_lpdf(sigma | 3, 0, 2.5)
##     - 1 * student_t_lccdf(0 | 3, 0, 2.5);
## }
## model {
##   // likelihood including constants
##   if (!prior_only) {
##     // initialize linear predictor term
##     vector[N] mu = Intercept + rep_vector(0.0, N);
##     target += normal_lpdf(Y | mu, sigma);
##   }
##   // priors including constants
##   target += lprior;
## }
## generated quantities {
##   // actual population-level intercept
##   real b_Intercept = Intercept;
## }

```

Stan-malli koostuu seuraavista koodilohkoista (osa on valinnaisia):

- `functions`-lohko sisältää omia myöhemmin käytettäviä funktiota.
- `data`-lohko sisältää aineiston joka syötetään `stan` funktion `data`-argumenttia käyttäen. `transformed data`-lohko taas sisältää mallinnuksen alussa muodostettavia lisämuuttujia.
- `parameters` ja `transformed parameters`-lohkot sisältävät mallin parametrit (jälkimmäiset ovat edellisten funktioita)
- `model`-lohko sisältää itse mallin määrittelyn.
- `generated quantities`-lohkossa voi laskea esimerkiksi parametreihin liittyviä tunnuslukuja tai generoida näytteitä posterioriennustejakaumasta.

`brms`-funktion sijaan voisimme määrittää mallimme suoraan Stan-koodina, jolloin käyttäisimme `rstan`-paketin `stan`-funktiota mallin sovittamiseen ja datan syöttämiseen:

```

fit_stan <- stan(
  model_code = stancode(fit),
  data = list(N = length(y6), Y = y6, prior_only = 0),
  chains = 1,
  refresh = 0
)

```

```
fit_stan
```

```

## Inference for Stan model: anon_model.
## 1 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=1000.
##
##           mean se_mean  sd  2.5%   25%   50%   75%  97.5% n_eff Rhat
## Intercept    0.26   0.02 0.41  -0.63   0.00   0.25   0.51   1.11  586   1
## sigma        1.27   0.01 0.34   0.79   1.02   1.21   1.44   2.12  546   1
## lprior       -3.33   0.00 0.11  -3.62  -3.37  -3.31  -3.26  -3.21  471   1
## b_Intercept  0.26   0.02 0.41  -0.63   0.00   0.25   0.51   1.11  586   1

```

```
## lp__          -19.01    0.05 1.07 -21.75 -19.48 -18.66 -18.22 -17.93   408    1
##
## Samples were drawn using NUTS(diag_e) at Tue Sep  6 08:27:51 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Stan-mallin voi syöttää joko merkkijonona kuten yllä, tai mallin voi kirjoittaa omaksi `.stan`-päätteiseksi tiedostoksi, jolloin esim. Rstudio osaa tallennettaessa varoittaa mallin syntaksivirheistä.

Stanin nettisivuilta löytyy lukuisia manuaaleja, esimerkkejä ja tutoriaaleja Stanin käyttöön liittyen: <https://mc-stan.org/users/documentation/>

Esimerkki 21: Normaalijakauman mallinnus Stanilla

Simuloidaan normaalijakautunutta dataa ja rakennetaan vastaava malli Stanilla:

```
malli <- "
data {
  int n;
  vector[n] y;
}

parameters {
  real mu;
  real<lower=0> sigma;
}

model {
  mu ~ normal(0, 100); // priori mu:lle
  sigma ~ normal(0, 10); // priori sigmalle
  y ~ normal(mu, sigma); // uskottavuusfunktio
}
"

fit_normal <- stan(
  model_code = malli,
  data = list(n = 100L, y = rnorm(100, 10, 2)),
  chains = 1, refresh = 0
)
```

```
fit_normal
```

```
## Inference for Stan model: anon_model.
## 1 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=1000.
##
##           mean se_mean  sd   2.5%   25%   50%   75%   97.5% n_eff Rhat
## mu         9.95   0.01 0.20   9.54   9.81   9.96  10.08  10.32  757   1
## sigma      1.91   0.01 0.15   1.66   1.81   1.90   2.01   2.22  636   1
## lp__     -113.47   0.05 1.00 -116.10 -113.97 -113.22 -112.70 -112.41  445   1
##
## Samples were drawn using NUTS(diag_e) at Tue Sep  6 08:29:13 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

5 Yleisiä periaatteita

Epäoleellisuus (ancillarity) ja *tyhjentyvyys (sufficiency)* ovat keskeisiä tilastollisen päättelyn käsitteitä ja ne on hyvä tuntea. *Uskottavuusperiaate (likelihood principle)* on seuraus näistä kahdesta edellä mainitusta

Seuraavassa oletetaan, että tuntematon suure on θ (sitä sen kummemmin spesifioimatta), priori θ :lle $p(\theta)$ ja uskottavuus $p(y|\theta)$.

5.1 Epäoleellisuus

Tarkastellaan seuraavaksi kahta erikoistapausta:

- Jos aineiston y jakauma ehdolla θ (uskottavuus) $p(y|\theta)$ ei riipu θ :sta, niin

$$p(\theta|y) \propto p(\theta),$$

eli posteriori on sama kuin priori. Tämä merkitsee sitä, että havainnosta y ei opita, tai toisin sanoin, y on epäinformatiivinen θ :lle. Sanomme, että y on epäoleellinen θ :lle

- Tarkastellaan tilannetta, jossa aineisto y on osittain epäinformatiivinen θ :lle. Oletetaan, että $y = (y_1, y_2)$. Voidaan ajatella, siten, että aineisto y havaitaan kahdessa osassa: ensin y_1 ja sitten y_2 . Havaitaan ensin y_1 , jolloin “opetus” on

$$p(\theta|y_1) \propto p(\theta)p(y_1|\theta).$$

Havaitaan tämän jälkeen y_2 , jolloin opetus on

$$\begin{aligned} p(\theta|y) &= p(\theta|y_1, y_2) \propto p(\theta|y_1)p(y_2|y_1, \theta) \\ &\propto p(\theta)p(y_1|\theta)p(y_2|y_1, \theta). \end{aligned}$$

Jos nyt $p(y_1|\theta) \propto 1$, niin y_1 on *epäoleellinen* θ :lle. Tällöin $p(\theta|y_1) = p(\theta)$ ja siten

$$p(\theta|y) = p(\theta|y_1, y_2) \propto p(\theta)p(y_2|y_1, \theta).$$

Siis tässä tapauksessa posteriori

$$p(\theta|y) \propto p(\theta)p(y_2|y_1, \theta)$$

riippuu y_2 :n ehdollisesta jakaumasta ehdolla y_1 (ja θ). Posteriori siis sisältää epäoleellisella aineiston osalla y_1 ehdollistetun jakauman. Huomaa, että vaikka epäoleellinen otossuure ei sisällä suoraa informaatiota θ :sta, se kuitenkin vaikuttaa epäsuoralla tavalla posteriorijakaumaan!

5.2 Tyhjentyvyys

Jos $p(y_2|y_1, \theta) \propto 1$, niin sanomme, että y_1 on *tyhjentyvä* θ :lle. Silloin

$$p(\theta|y_1, y_2) = p(\theta|y_1) \propto p(\theta)p(y_1|\theta).$$

Posteriori (ja siten johtopäätökset) eivät riipu lainkaan y_2 :sta.

“Klassisessa tilastotieteessä” pyritään ehdollistamaan päättelyt epäoleellisen tunnusluvun (kiasatunnusluvun) suhteen. Toisaalta pyritään etsimään tyhjentyvä tunnusluku ja käyttämään estimaattoreita ja testisuureita, jotka ovat sen funktioita. Bayes-tilastotieteessä näihin ei yleensä tarvitse kiinnittää huomiota, vaan kaikki hoituu itsestään posterioria laskettaessa.

Esimerkki 22: Sekoitusmittaukset

Oletetaan, että $y_i|\theta \sim N(\theta, 1)$, $i = 1, 2$. Oletetaan lisäksi, että havainnot koostuvat mittauksista

$$\begin{aligned} z &= y_1 - y_2 \\ w &= y_1 + y_2 \end{aligned}$$

eli muuttujia y_1 ja y_2 ei havaita suoraan. On helppo nähdä, että

$$p(z, w|\theta) = p(z)p(w|\theta),$$

sillä $z|\theta, w \sim N(0, 2)$ ja $w|\theta \sim N(2\theta, 2)$. Siis w on tyhjentävä θ :lle. Koska lisäksi $z|\theta \sim N(0, 2)$, jolloin $p(z|\theta)$ ei riipu θ :sta, on z epäoleellinen. Lisäksi koska $p(z|\theta) = p(z|w, \theta)$, ovat z ja w riippumattomia (ehdolla θ) ja myös $p(w|z, \theta) = p(w|\theta)$. Posteriori on nyt

$$p(\theta|z, w) \propto p(\theta)p(w|\theta) \propto p(\theta|w),$$

mistä nähdään, että posteriorijakauma y :n suhteen on sama kuin posteriorijakauma z :n suhteen, sillä

$$y_1 = \frac{1}{2}(z + w)$$

$$y_2 = \frac{1}{2}(w - z)$$

eli (z, w) sisältää saman informaation kuin $y = (y_1, y_2)$.

Esimerkki 23: Tyhjentävä tunnusluku Binomijakaumalle

Oletetaan, että $y_i|\theta \sim \text{Bin}(1, \theta)$ ja havainnot $y = (y_1, \dots, y_n)$ ovat riippumattomia ehdolla θ . Merkitään $s = \sum_{i=1}^n y_i$ ("onnistumisten" lukumäärä). Tällöin uskottavuus on

$$p(y|\theta) = \prod_{i=1}^n \theta^{y_i} (1 - \theta)^{1 - y_i} = \theta^s (1 - \theta)^{n - s}.$$

Tässä $p(y|s, \theta) = \binom{n}{s}^{-1} \propto 1$, joka ei riipu θ :sta. Siis s on tyhjentävä θ :lle.

Tulkinta: Kun $s = \sum_{i=1}^n y_i$ tunnetaan, niin havaintojen y_1, \dots, y_n tunteminen ei opeta lisää θ :sta. Voidaan siis toimia posteriorilla $p(\theta|y)$ tai $p(\theta|s)$.

5.3 Uskottavuusperiaate ja epäinformatiivinen pysäyttäminen

Uskottavuusperiaatteen mukaan kahden todennäköisyysmallin, joilla on vakiolla kertomista lukuun ottamatta sama uskottavuusfunktio, pitäisi johtaa samaan päättelyyn θ :n suhteen. Bayes-päätely toteuttaa tämän periaatteen, koska posteriorijakauma riippuu aineistosta ainoastaan uskottavuusfunktion välityksellä.

(Huomaa kuitenkin, että Jeffreysin periaatteella muodostettu priorijakauma riippuu otantajakaumasta. Tällöin priorit voivat poiketa vaikka uskottavuusfunktiot olisivat verrannolliset eli vakiokerrointa lukuun ottamatta samat. Siksi Jeffreysin periaate on ristiriidassa uskottavuusperiaatteen kanssa!)

Esimerkki 24: Tunnettu ja tuntematon toistojen määrä

Jatketaan edellistä esimerkkiä. Oletetaan, että n ei ole kiinnitetty. Jatketaan toistoja, kunnes saadaan r onnistumista. Nyt n on satunnainen. Uskottavuus on

$$p(n|\theta) = \binom{n-1}{r-1} \theta^{r-1} (1-\theta)^{(n-1)-(r-1)} \theta$$

$$= \binom{n-1}{r-1} \theta^r (1-\theta)^{n-r}$$

$$\propto \theta^r (1-\theta)^{n-r}, n = r, r+1, \dots$$

Kun n on kiinnitetty (ja r satunnainen), niin

$$p(r|\theta, n) = \binom{n}{r} \theta^r (1-\theta)^{n-r} \propto \theta^r (1-\theta)^{n-r}.$$

Koska kummassakin tapauksessa uskottavuus on sama, niin myös posteriori ja siten Bayes-tilastotieteen johtopäätökset ovat samat, jos priori on sama. Klassisessa tilastotieteessä SU-estimaattori on kummassakin tapauksessa $\hat{\theta} = r/n$, mutta jakaumateoria on eri (miksi?).

Tärkeä uskottavuusperiaatteen sovellus liittyy otannan pysäyttämissääntöihin. Pysäyttämissääntöä sanotaan epäinformatiiviseksi, jos peräkkäisotannanssa todennäköisyys keskeyttää otanta ei riipu parametrilla θ tai riippuu siitä ainoastaan havaintojen välityksellä. Tällöin pysäyttämissääntö ei vaikuta uskottavuusfunktioon eikä uskottavuusperiaatteen mukaan myöskään parametriin θ liittyvään tilastolliseen päättelyyn. Bayes-päättelyssä pysäyttämissääntö on epäinformatiivinen, jos pysäyttämissääntö ei vaikuta uskottavuusfunktioon eikä priorijakaumaan.

Oletetaan, että havainnot $y = (y_1, \dots, y_n)$ ovat riippumattomia ehdolla θ ja noudattavat jakaumaa $p(y_i|\theta)$. Oletetaan lisäksi, että pysäyttämissääntö i :n havainnon kohdalla riippuu edellisistä havainnoista y_1, \dots, y_{i-1} . Merkitään pysäyttämishetki käyttäen indikaattorimuuttujaa δ_i , $i = 1, \dots, n+1$, ja määritellään, että

$$\delta_i = \begin{cases} 1, & \text{jos pysäytetään otanta} \\ 0, & \text{jos poimitaan havainto } y_i. \end{cases}$$

Tällöin todennäköisyysmalli, joka sisältää sekä havainnot, että pysäyttämissääntö, voidaan kirjoittaa muodossa

$$p(y, \delta|\theta) = \left[\prod_{i=1}^n P(\delta_i = 0|\theta, y_1, \dots, y_{i-1})p(y_i|\theta) \right] \times P(\delta_{n+1} = 1|\theta, y_1, \dots, y_n).$$

Jos nyt pysäyttämistodennäköisyys i :n havainnon kohdalla määräytyy edellisten havaintojen y_1, \dots, y_{i-1} perusteella kaikilla $i = 1, \dots, n+1$, niin

$$P(\delta_i = 0|\theta, y_1, \dots, y_{i-1}) = P(\delta_i = 0|y_1, \dots, y_{i-1}), i = 1, 2, \dots$$

Tällöin uskottavuusfunktio on

$$p(y, \delta|\theta) \propto \prod_{i=1}^n p(y_i|\theta) = p(y|\theta),$$

joten pysäyttämissääntö ei vaikuta uskottavuuspäättelyyn.

Kun jatketaan Bernoulli-kokeita, kunnes "onnistumisia" on r kappaletta, pysäyttäminen määräytyy edellisistä havainnoista. Tällöin uskottavuuspäättelyssä ei tarvitse ottaa huomioon pysäyttämissääntöä. Jos otoskoko n on ennalta kiinnitetty, se on parametrilla θ riippumaton eikä pysäyttämissääntöä tarvitse ottaa huomioon.

6 Lisää esimerkkejä

6.1 Kalalohiset

Toistetaan YLM2-kurssin kalojen loiseliöiden esiintyvyyden mallinnus Bayesiläisittäin.

Aineisto on peräisin paperista [Crabs, leeches and trypanosomes: an unholy trinity?](#), abstrakti alla:

The red king crab *Paralithodes camtschaticus* was deliberately introduced to the Barents Sea in the 1960s and 1970s from its native area in the North Pacific. The carapace of these crabs is used by the leech *Johanssonia arctica* to deposit its eggs. The leech in turn is a vector for the blood parasite *Trypanosoma murmanensis* that can infect marine fish, including cod. Hemmingsen et al. (2005) examined cod for trypanosome infections during annual cruises along the coast of Finnmark in North Norway over three successive years and in four different areas. They show that trypanosome infections are strongest in the area of Varangerfjord where the density of red king crabs is highest. Thus, there is evidence that the introduction of the foreign red king crabs had an indirect detrimental effect on the health of the native cod population. This situation stands out because it is not an introduced parasite that is dangerous for a native host, but rather an introduced host that promotes transmission of two endemic parasites.

```
cod <- read.table("http://users.jyu.fi/~jovetale/YLM2/fish.txt", header = TRUE)
str(cod)
```

```
## 'data.frame': 1254 obs. of 11 variables:
## $ Sample : int 1 2 3 4 5 6 7 8 9 10 ...
## $ Intensity : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Prevalence: int 0 0 0 0 0 0 0 0 0 0 ...
## $ Year : int 1999 1999 1999 1999 1999 1999 1999 1999 1999 1999 ...
## $ Depth : int 220 220 220 220 220 220 220 194 194 194 ...
## $ Weight : int 148 144 146 138 40 68 52 3848 2576 1972 ...
## $ Length : int 26 26 27 26 17 20 19 77 67 60 ...
## $ Sex : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Stage : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Age : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Area : int 2 2 2 2 2 2 2 3 3 3 ...
```

Faktoroidaan alue ja vuosi:

```
cod$Area <- factor(cod$Area)
cod$Year <- factor(cod$Year)
```

Tutkitaan onko loistartuntojen esiintyvyydessä alueellisia ja/tai ajallisia eroja sovittamalla logistinen malli jossa vasteena on loisen esiintyvyys (Prevalence) ja selittäjinä ovat pituus, vuosi ja alue. Näistä pituusmuuttujassa on muutama puuttuva havainto:

```
colSums(is.na(cod))
```

```
##      Sample Intensity Prevalence      Year      Depth      Weight      Length
##         0         57           0         0           0           6           6
##      Sex      Stage      Age      Area
##         0          0          0         0
```

Puuttuvia on kokonaisuuteen nähden niin vähän että jätetään kyseiset rivit vain yksinkertaisuuden vuoksi pois (voisimme myös mallintaa puuttumista `mi()`-funktion avulla):

```
cod <- na.omit(cod)
```

Sovitetaan malli:

```
fit_cod1 <- brm(
  Prevalence ~ Length + Year + Area,
  data = cod,
  family = bernoulli,
  chains = 4,
  cores = 4,
  refresh = 0
)
```

```
fit_cod1
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: Prevalence ~ Length + Year + Area
## Data: cod (Number of observations: 1191)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
```

```
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.49     0.28   -1.04    0.05 1.00    5799    3420
## Length        0.01     0.00   -0.00    0.02 1.00    6102    3274
## Year2000       0.76     0.17    0.43    1.08 1.00    4342    3353
## Year2001      -0.47     0.14   -0.74   -0.19 1.00    4448    2855
## Area2         -0.72     0.19   -1.09   -0.36 1.00    3841    2836
## Area3         -0.66     0.17   -1.00   -0.32 1.00    3651    3275
## Area4         1.04     0.19    0.68    1.40 1.00    3816    3211
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Vaihtoehtoinen malli jossa mukana alueen ja vuoden yhdysvaikutus:

```
fit_cod2 <- brm(
  Prevalence ~ Length + Year * Area,
  data = cod,
  family = bernoulli,
  chains = 4,
  cores = 4,
  refresh = 0
)
```

```
fit_cod2
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: Prevalence ~ Length + Year * Area
## Data: cod (Number of observations: 1191)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      0.08     0.30   -0.50    0.65 1.00    3963    3245
## Length          0.01     0.00   -0.00    0.02 1.00    6501    2688
## Year2000         0.41     0.35   -0.27    1.12 1.00    1764    2566
## Year2001        -2.73     0.45   -3.68   -1.89 1.00    1533    1932
## Area2           -1.34     0.29   -1.91   -0.77 1.00    2293    2695
## Area3           -1.46     0.24   -1.94   -1.00 1.00    2472    2617
## Area4            0.31     0.27   -0.23    0.84 1.00    2665    2928
## Year2000:Area2  -0.09     0.52   -1.11    0.93 1.00    1897    2419
## Year2001:Area2  2.82     0.55    1.78    3.93 1.00    1645    1804
## Year2000:Area3  0.86     0.45   -0.04    1.77 1.00    1967    2569
## Year2001:Area3  2.80     0.51    1.85    3.86 1.00    1636    2215
## Year2000:Area4  0.92     0.61   -0.27    2.15 1.00    2501    2476
## Year2001:Area4  2.62     0.54    1.61    3.70 1.00    1622    1977
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Kokeillaan vielä ilman pituutta joka vaikuttaa olevan turha:

```
fit_cod3 <- brm(
  Prevalence ~ Year * Area,
  data = cod,
  family = bernoulli,
  chains = 4,
  cores = 4,
  refresh = 0
)
```

```
fit_cod3
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: Prevalence ~ Year * Area
## Data: cod (Number of observations: 1191)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      0.45      0.17   0.11   0.79 1.00    2222    2637
## Year2000        0.47      0.35  -0.19   1.16 1.00    1846    2576
## Year2001       -2.69      0.44  -3.59  -1.88 1.00    1664    1914
## Area2          -1.33      0.29  -1.91  -0.77 1.00    2651    2793
## Area3          -1.44      0.24  -1.92  -0.97 1.00    2235    2158
## Area4           0.27      0.27  -0.25   0.79 1.00    2186    2507
## Year2000:Area2  -0.14      0.52  -1.16   0.86 1.00    2303    2564
## Year2001:Area2  2.83      0.54   1.80   3.91 1.00    1817    2217
## Year2000:Area3  0.82      0.45  -0.04   1.71 1.00    2044    2605
## Year2001:Area3  2.77      0.51   1.82   3.80 1.00    1731    1781
## Year2000:Area4  0.90      0.62  -0.23   2.18 1.00    2485    2754
## Year2001:Area4  2.63      0.53   1.62   3.69 1.00    1691    2063
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Verrataan tuloksia ristiinvalidoinnilla:

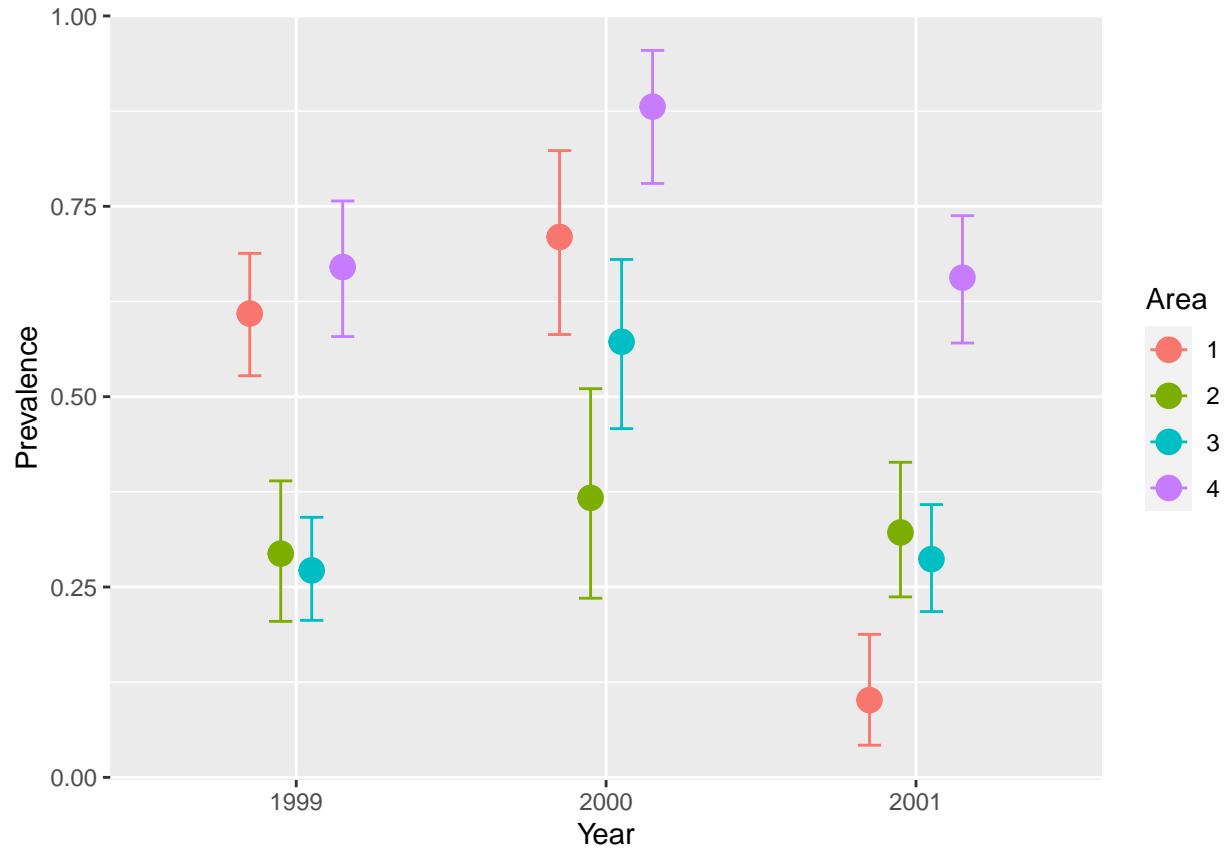
```
fit_cod1a <- add_criterion(fit_cod1, "loo")
fit_cod2a <- add_criterion(fit_cod2, "loo")
fit_cod3a <- add_criterion(fit_cod3, "loo")
loo_compare(fit_cod1a, fit_cod2a, fit_cod3a)
```

```
##           elpd_diff se_diff
## fit_cod2a  0.0         0.0
## fit_cod3a -0.2         1.6
## fit_cod1a -19.6        7.0
```

Tämän perusteella yhdysvaikutusermi on järkevä pitää mallissa, pituus-muuttuja ei juurikaan vaikuta mihinkään. Pudotetaan se pois jatko-analyysin helpottamiseksi, eli valitaan malli 3.

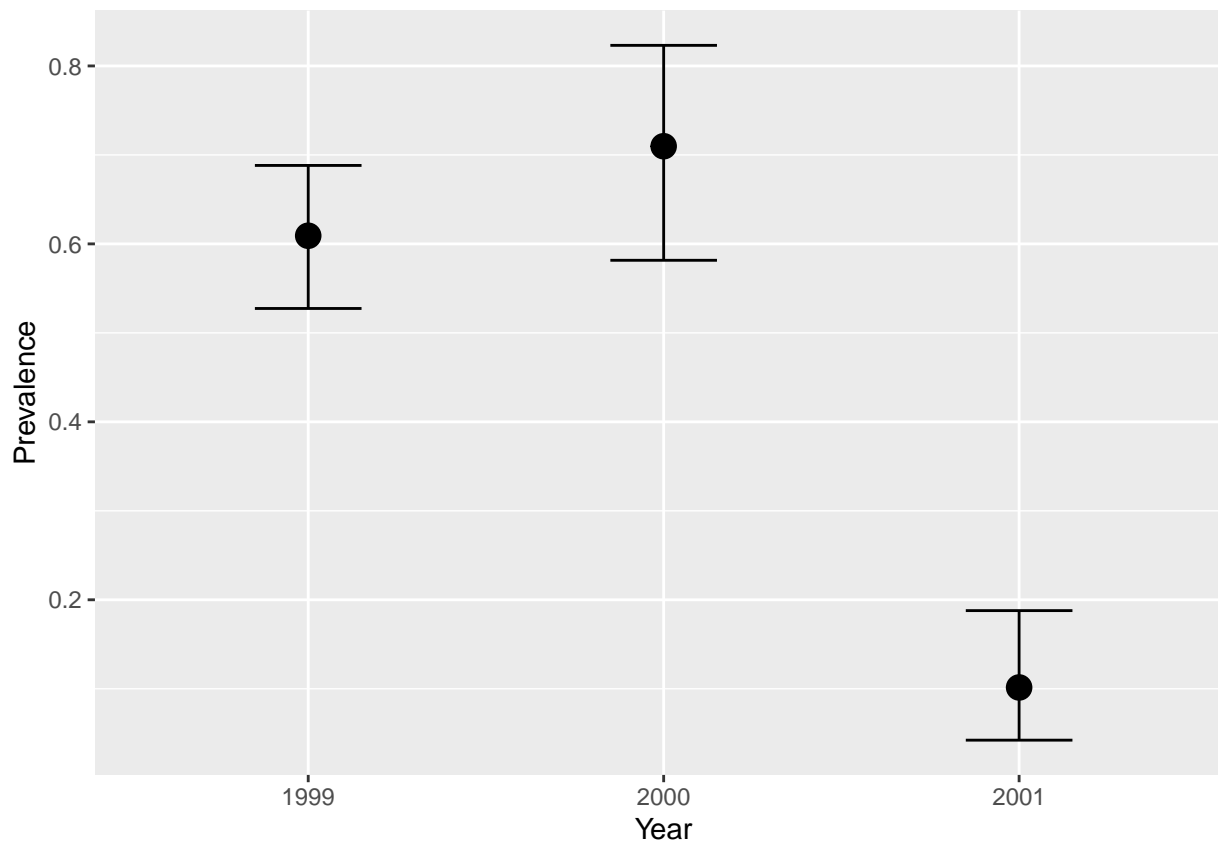
Hankalien ristitulosuhteiden ja log-oddsien sijaan tulkitaan tulokset suoraan todennäköisyyksien avulla jokaisessa ryhmässä (huomaa, että voisimme `effects`-argumentin avulla laskea ryhmäkohtaiset ennusteet myös mallin 1 tilanteessa vaikkei mallissa ollutkaan yhteisvaikutusermiä!):


```
conditional_effects(fit_cod3, effects = "Year:Area", robust = FALSE)
```



Ajatellaan että kiinnostuksena on nimenomaan vuoden vaikutuksesta. Kokeillaan ensin käyttää `conditional_effects`-funktiota:

```
conditional_effects(fit_cod3, effects = "Year", robust = FALSE)
```



Tämä ei ole se mitä halusimme, koska kuvan ennusteissa alue-muuttuja on kiinnitetty vertailuluokkaan 1. Efektit ovat siis ehdollisia todennäköisyyksiä $P(\text{prevalence}|\text{year} = t, \text{area} = 1)$, mutta meitä kiinnostaa marginaaliset todennäköisyydet $P(\text{prevalence}|\text{year} = t)$. Tämä pitää tehdä “käsini”:

```
newdata <- data.frame(
  Year = factor(rep(1999:2001, times = 4)),
  Area = factor(rep(1:4, each = 3))
)
sim <- posterior_epred(fit_cod3, newdata = newdata)
str(sim)
```

```
## num [1:4000, 1:12] 0.553 0.649 0.549 0.617 0.572 ...
```

Matriisi `sim` pitää sisällään joka rivillä yhteen posteriorinäytteen θ_i liittyvät `newdata` taulukon ennusteet. Muokataan data parempaan muotoon ja marginalisoidaan yli alueiden:

```
d <- data.frame(Prevalence = c(t(sim)), Year = newdata$Year, Area = newdata$Area)
d |>
  group_by(Year) |>
  summarise(
    mean = mean(Prevalence),
    lwr = quantile(Prevalence, 0.05),
    upr = quantile(Prevalence, 0.95)
  )
```

```
## # A tibble: 3 x 4
##   Year   mean   lwr   upr
##   <fct> <dbl> <dbl> <dbl>
```

```
## 1 1999 0.461 0.230 0.710
## 2 2000 0.632 0.306 0.921
## 3 2001 0.342 0.0702 0.693
```

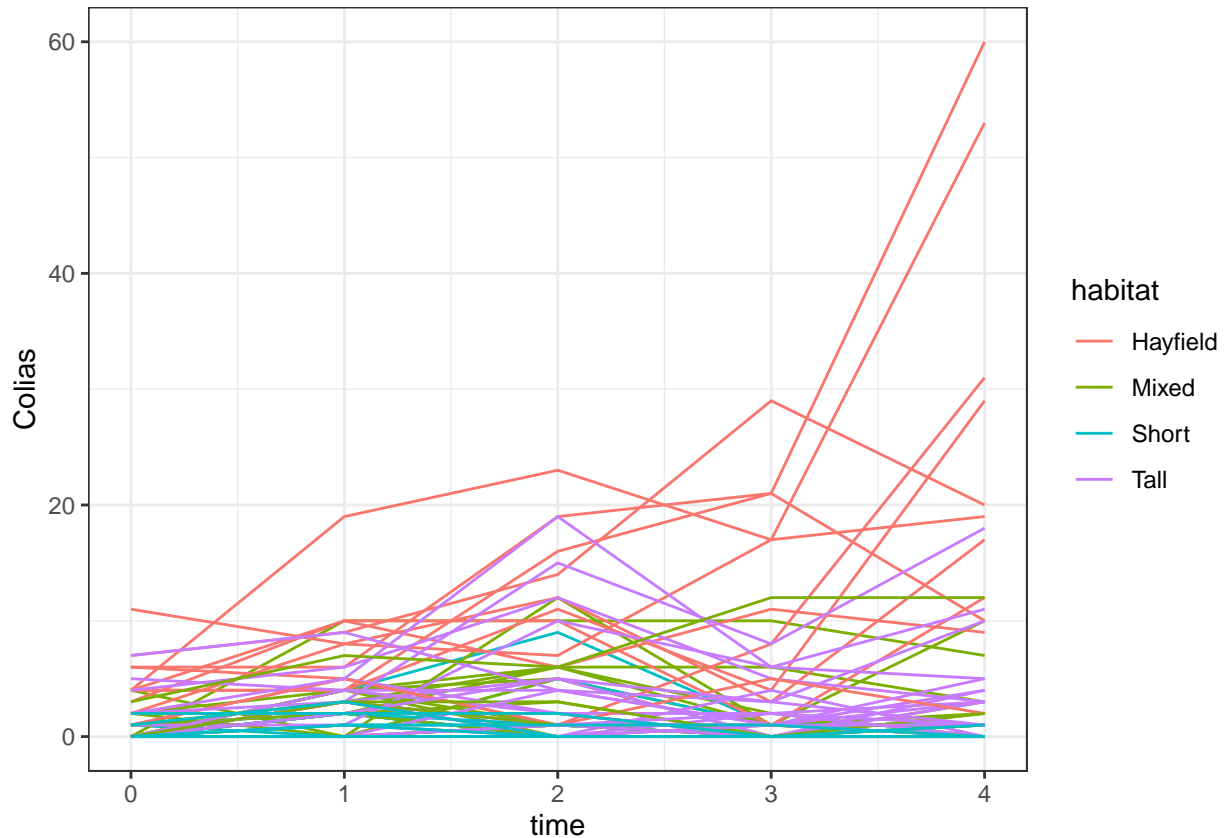
6.2 Perhoslajimallinnus

Tutkitaan ympäristötekijöiden yhteyttä erään perhoslajin runsauteen. Aineisto sisältää muuttujat

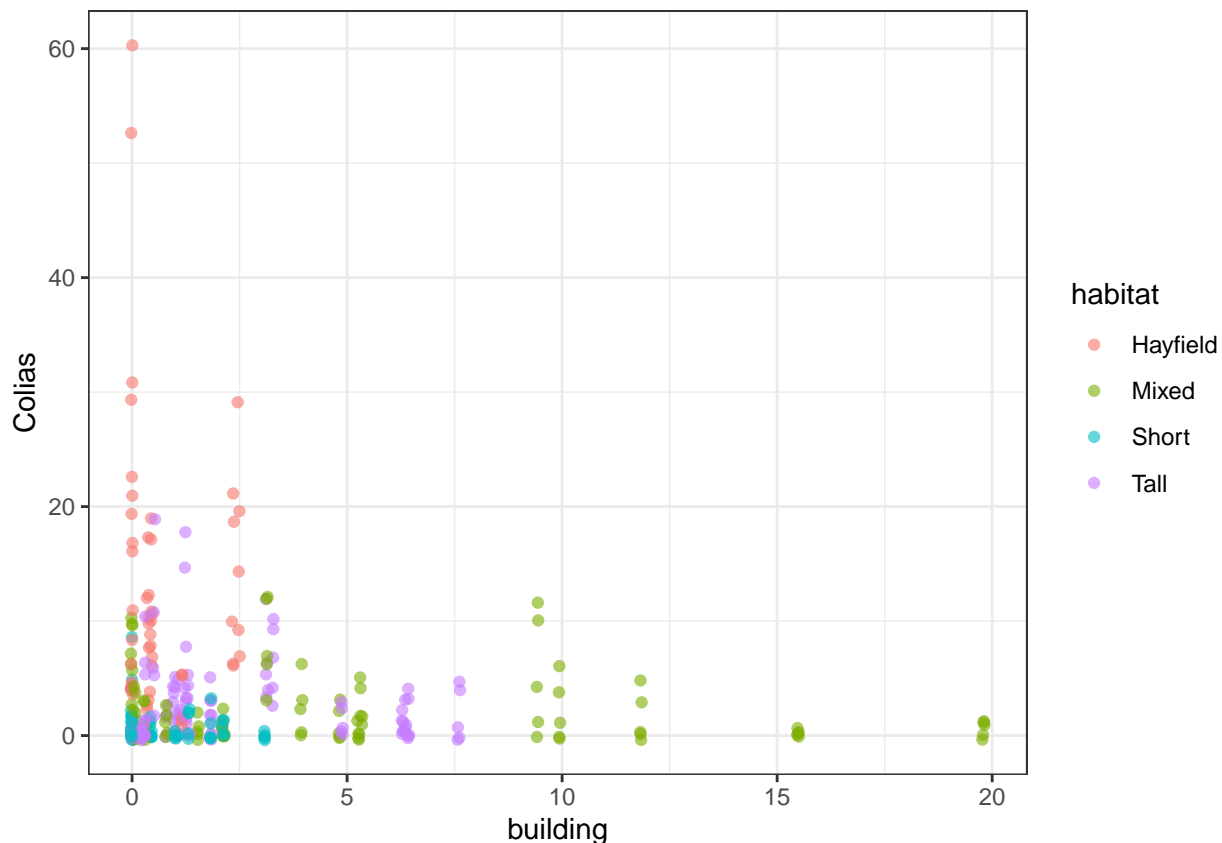
- site: havaintopaikka
- time: vuosi
- habitat: ruuhomaan tyyppi (hayfield, mixed, short, tall)
- building: rakennusten osuus ympäristössä prosentteina
- Colias: perhosten lukumäärä alueella

```
perhoset <- read.table("http://users.jyu.fi/~jovetale/YLM2/perhoset.txt", TRUE)
perhoset$time <- perhoset$time - 1 # ensimmäinen vuosi nollassi tulkinnallisista syistä

ggplot(perhoset, aes(x = time, y = Colias, group = site, colour = habitat)) +
  geom_line() + theme_bw()
```



```
ggplot(perhoset, aes(x = building, y = Colias, colour = habitat)) +
  geom_jitter(alpha=0.6) + theme_bw()
```



Tiedämme että sekä ruohomaan tyyppi että rakennusten osuus vaikuttavat perhosten lukumäärään, ja vuosien välillä on todennäköisesti eroa. Sovitetaan Poisson-malli jossa selittäjänä on ruohomaan tyyppi, rakennusten osuus, aika, ja ajan interaktio ruohomaan kanssa. Ajan vaikutus voi hyvinkin olla epälineaarinen, mutta toisaalta aikapisteitä on vain viisi, joten yksi vaihtoehto olisi mallintaa aika faktorina. Tehdään näin, mutta oletetaan kuitenkin että ajan vaikutus on monotoninen:

```
fit_pois_mo <- brm(
  Colias ~ mo(time) * habitat + building + (1 | site),
  data = perhoset,
  family = poisson,
  refresh = 0,
  chains = 4,
  cores = 4
)
```

```
fit_pois_mo
```

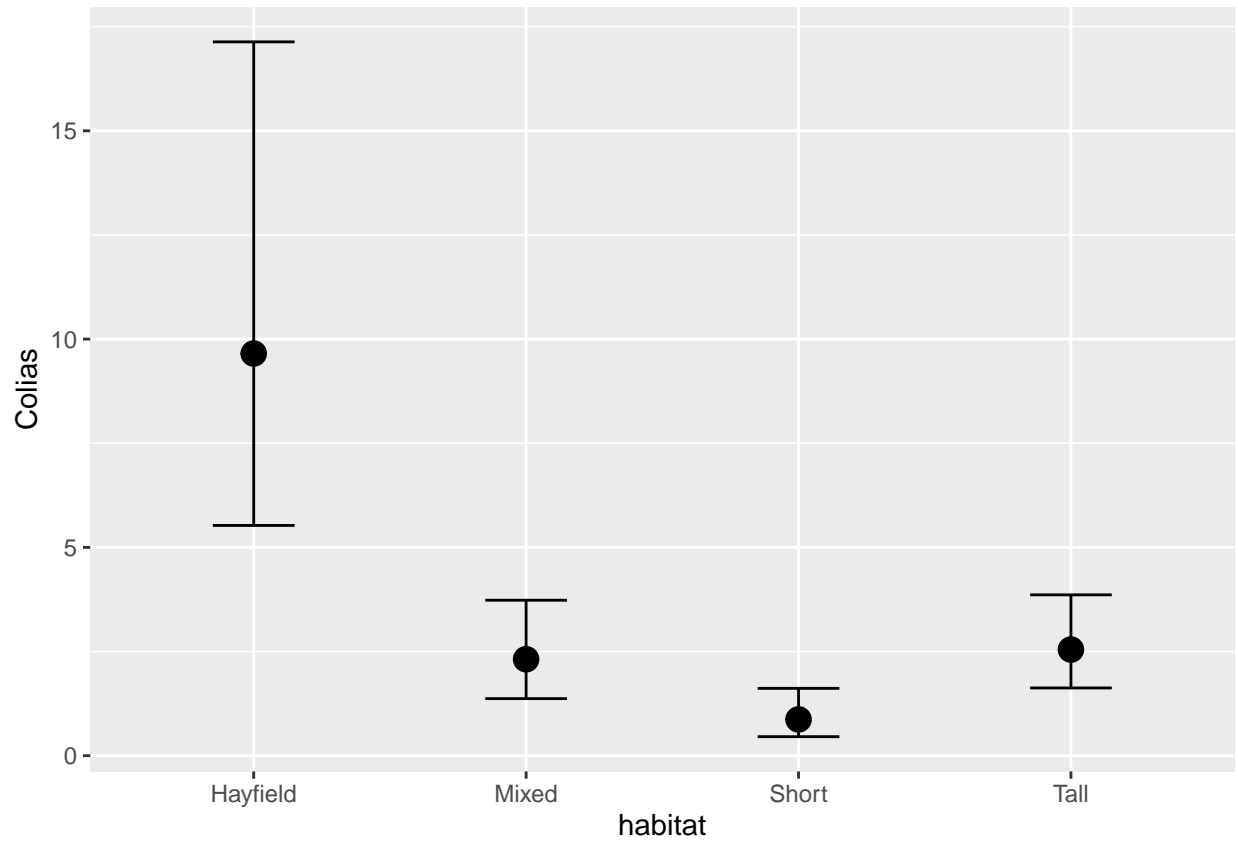
```
## Family: poisson
## Links: mu = log
## Formula: Colias ~ mo(time) * habitat + building + (1 | site)
## Data: perhoset (Number of observations: 330)
## Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 4000
##
## Group-Level Effects:
## ~site (Number of levels: 66)
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

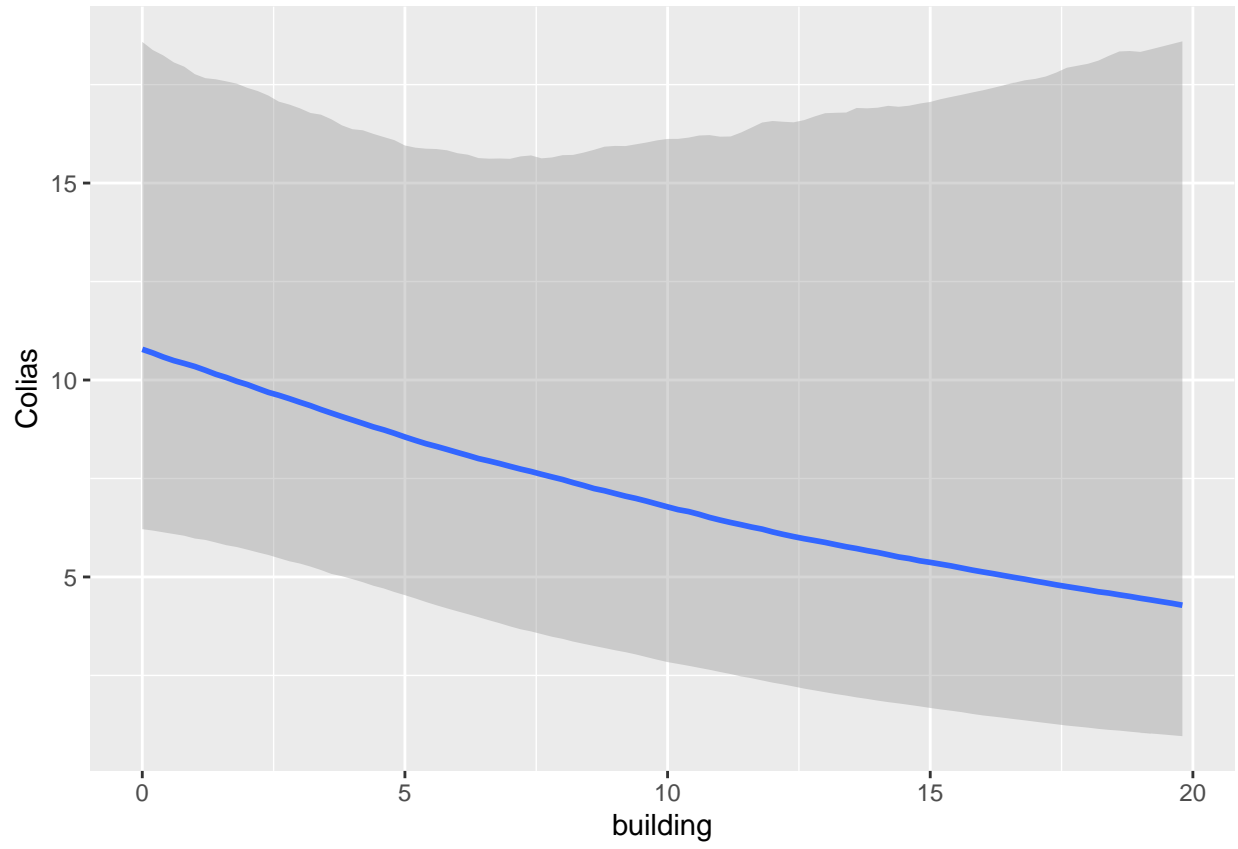
```

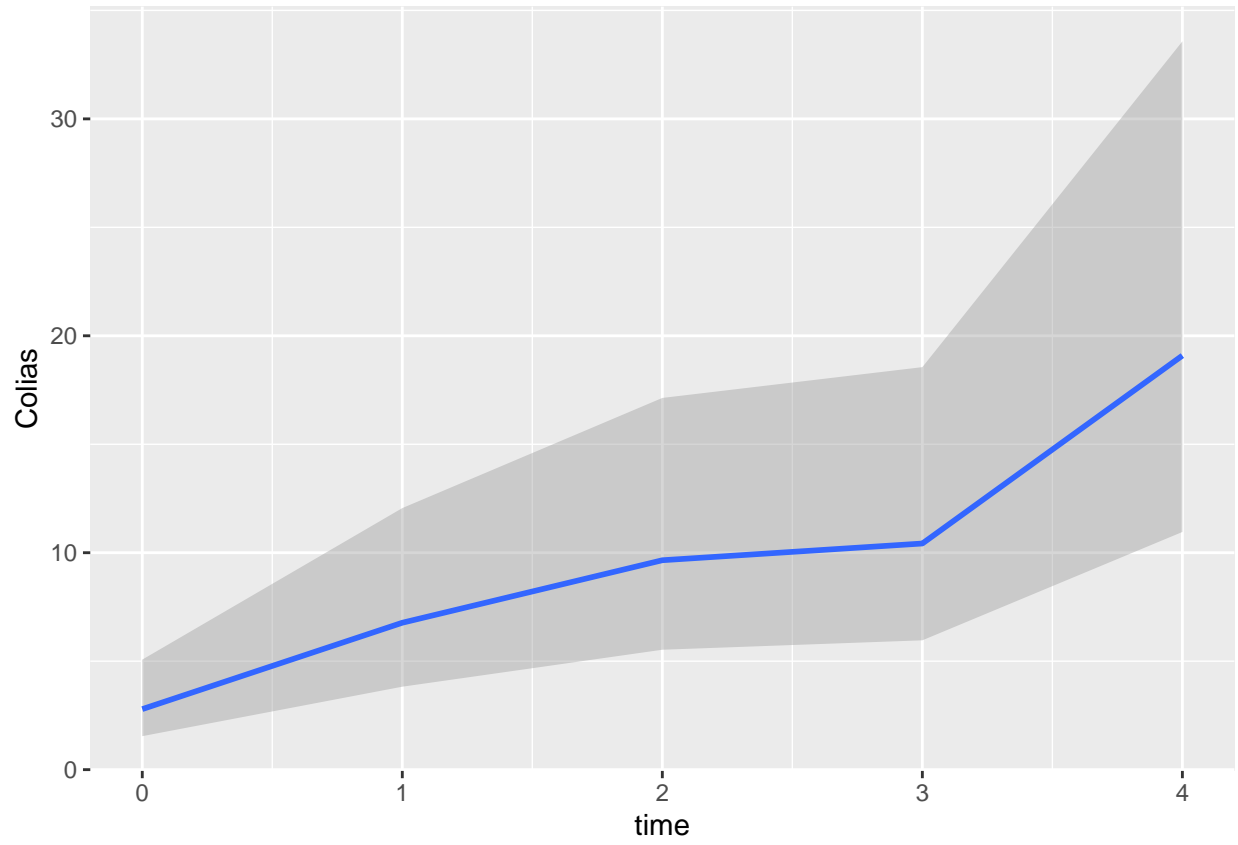
## sd(Intercept)    0.85    0.10    0.68    1.07 1.01    715    1400
##
## Population-Level Effects:
##              Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept          1.14    0.29    0.57    1.71 1.00    438    764
## habitatMixed       -1.24    0.40   -2.02   -0.45 1.01    532   1177
## habitatShort       -1.92    0.43   -2.76   -1.08 1.00    728   1461
## habitatTall        -1.13    0.36   -1.82   -0.41 1.01    470   1084
## building           -0.05    0.04   -0.12    0.02 1.01   1116   1974
## motime              0.48    0.03    0.42    0.55 1.00   2753   2881
## motime:habitatMixed -0.28    0.05   -0.39   -0.19 1.00   3245   2970
## motime:habitatShort -0.65    0.12   -0.90   -0.44 1.00   3706   2914
## motime:habitatTall  -0.24    0.05   -0.33   -0.15 1.00   2996   2751
##
## Simplex Parameters:
##              Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS
## motime1[1]          0.46    0.05    0.36    0.56 1.00    2994
## motime1[2]          0.19    0.05    0.09    0.28 1.00    2862
## motime1[3]          0.04    0.03    0.00    0.12 1.00    3184
## motime1[4]          0.31    0.04    0.23    0.40 1.00    5042
## motime:habitatMixed1[1] 0.08    0.07    0.00    0.24 1.00    3045
## motime:habitatMixed1[2] 0.09    0.08    0.00    0.29 1.00    4259
## motime:habitatMixed1[3] 0.50    0.16    0.19    0.82 1.00    4236
## motime:habitatMixed1[4] 0.33    0.16    0.05    0.64 1.00    3754
## motime:habitatShort1[1] 0.06    0.05    0.00    0.20 1.00    3541
## motime:habitatShort1[2] 0.12    0.08    0.01    0.32 1.00    3061
## motime:habitatShort1[3] 0.63    0.15    0.30    0.89 1.00    4758
## motime:habitatShort1[4] 0.19    0.13    0.01    0.50 1.00    4357
## motime:habitatTall1[1]  0.12    0.09    0.00    0.34 1.00    3134
## motime:habitatTall1[2]  0.09    0.08    0.00    0.28 1.00    3714
## motime:habitatTall1[3]  0.57    0.16    0.25    0.87 1.00    4582
## motime:habitatTall1[4]  0.22    0.14    0.01    0.52 1.00    3568
##
##              Tail_ESS
## motime1[1]          2664
## motime1[2]          1940
## motime1[3]          1996
## motime1[4]          3134
## motime:habitatMixed1[1] 1642
## motime:habitatMixed1[2] 2263
## motime:habitatMixed1[3] 2651
## motime:habitatMixed1[4] 2164
## motime:habitatShort1[1] 1695
## motime:habitatShort1[2] 1358
## motime:habitatShort1[3] 3007
## motime:habitatShort1[4] 1990
## motime:habitatTall1[1]  1553
## motime:habitatTall1[2]  2127
## motime:habitatTall1[3]  2722
## motime:habitatTall1[4]  1837
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

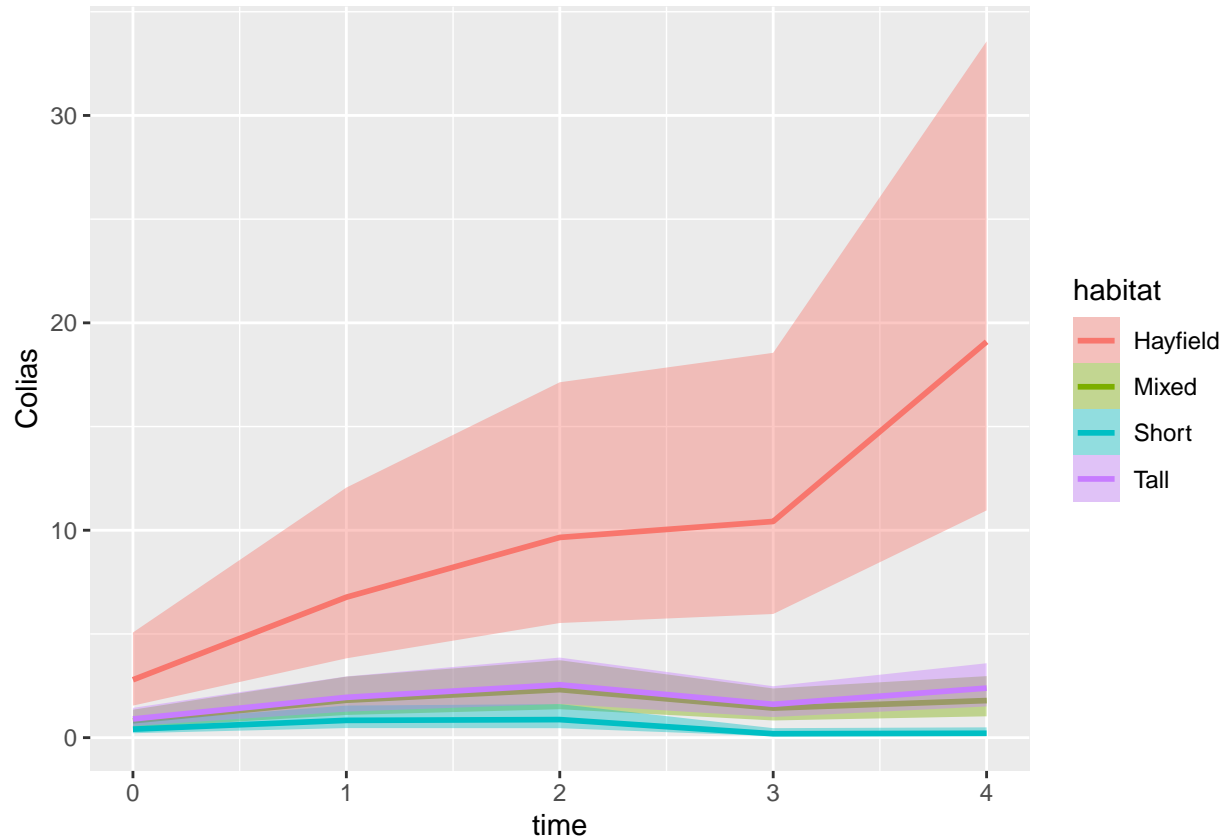
```

```
conditional_effects(fit_pois_mo)
```









Ajatellaan nyt että haluaisimme ennustaa seuraavaa aikapistettä. Tämä ei oikein onnistu (ainakaan helposti) mallillamme koska aika käsiteltiin faktorina. Ajan vaikutus näytti kuitenkin olevan suurin piirtein lineaarinen (pl. aikapisteen 3 notkahdus), joten vaihdetaan aika jatkuvaksi muuttujaksi ja sovitetaan malli uudestaan:

```
fit_pois <- brm(
  Colias ~ time * habitat + building + (1 | site),
  data = perhoset,
  family = poisson,
  refresh = 0,
  chains = 4,
  cores = 4,
  save_pars = save_pars(all = TRUE) # ennakoidaan LOO:ta ja tallennetaan kaikki parametrit
)
```

Aineistossa on aika paljon nollia, joten nolla-paisutettu Poisson tai negatiivinen binomijakauma voisikin olla parempi vaihtoehto. Sovitetaan myös nämä vaihtoehdot:

```
fit_pois0 <- brm(
  Colias ~ time * habitat + building + (1 | site),
  data = perhoset,
  family = zero_inflated_poisson(),
  refresh = 0,
  cores = 4,
  save_pars = save_pars(all = TRUE)
)
```

Negatiivinen binomijakauma ja sen nollapaisutettu versio:

```
fit_negbin <- brm(
  Colias ~ time * habitat + building + (1 | site),
  data = perhoset,
  family = negbinomial(),
  refresh = 0,
  cores = 4,
  save_pars = save_pars(all = TRUE)
)
```

Ja vielä nollapaisutettu NB:

```
fit_negbin0 <- brm(
  Colias ~ time * habitat + building + (1 | site),
  data = perhoset,
  family = zero_inflated_negbinomial(),
  refresh = 0,
  cores = 4,
  save_pars = save_pars(all = TRUE)
)
```

Verrataan malleja leave-one-out ristiinvalidoinnilla:

```
fp <- add_criterion(fit_pois, "loo", moment_match = TRUE)
fp0 <- add_criterion(fit_pois0, "loo", moment_match = TRUE)
nb <- add_criterion(fit_negbin, "loo", moment_match = TRUE)
nb0 <- add_criterion(fit_negbin0, "loo", moment_match = TRUE)
loo_compare(fp, fp0, nb, nb0)
```

```
perh_loo_res
```

```
##      elpd_diff se_diff
## nb      0.0      0.0
## nb0    -0.6      1.4
## fp0   -61.9     16.7
## fp    -79.4     17.2
```

Tässä hierarkkisessa pitkittäisaineistotilanteessa voisi olla myös mielekästä suorittaa ristiinvalidointi jättämällä yksi havaintopaikka kerrallaan pois, tai esimerkiksi viimeinen aikapiste (ks. `?loo::kfold_split_grouped`). Diagnostiikan mukaan LOO:n kanssa oli ongelmia, vaikka käytimme `moment_match`-argumenttia joka korjata tilannetta. Jatketaan kuitenkin NB-mallilla, nollapaisutettu NB ei näytä tuovan juuri muutosta (vastaava kerroin on itseasiassa lähellä nollaa).

Tutkitaan NB-mallia posterioriennustetarkasteluilla:

```
library(ggplot2)
library(patchwork)
p1 <- pp_check(fit_negbin, "bars", size = 0.1) +
  coord_cartesian(xlim = c(0, 50))
```

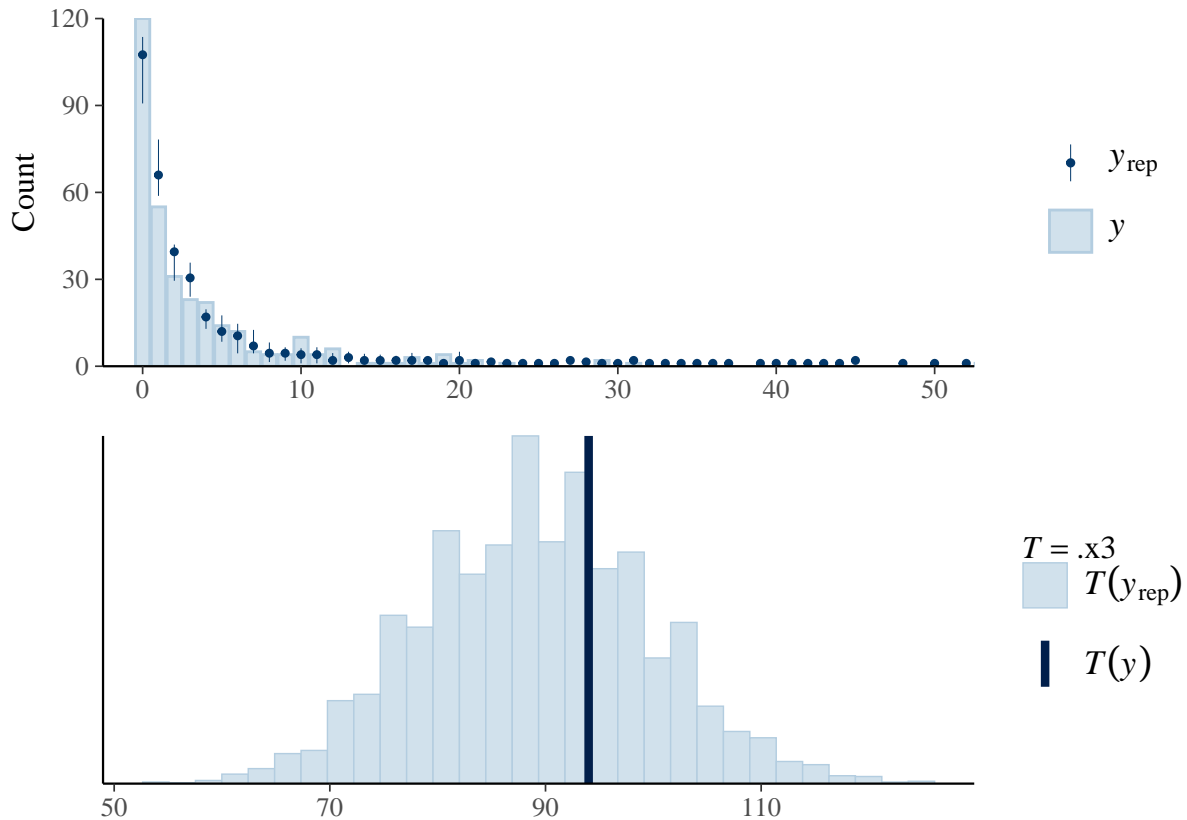
```
## Using 10 posterior draws for ppc type 'bars' by default.
```

```
zeros <- function(x) sum(x == 0)
p2 <- pp_check(fit_nb, "stat", stat = zeros)
```

```
## Using all posterior draws for ppc type 'stat' by default.
```

```
p1 + p2 + plot_layout(ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Ok. Tutkitaan populaatiotason keskiarvon ennusteita (jossa satunnaisfektit on asetettu nolliksi, `re_formula = NA`) aikapisteille 1–10:

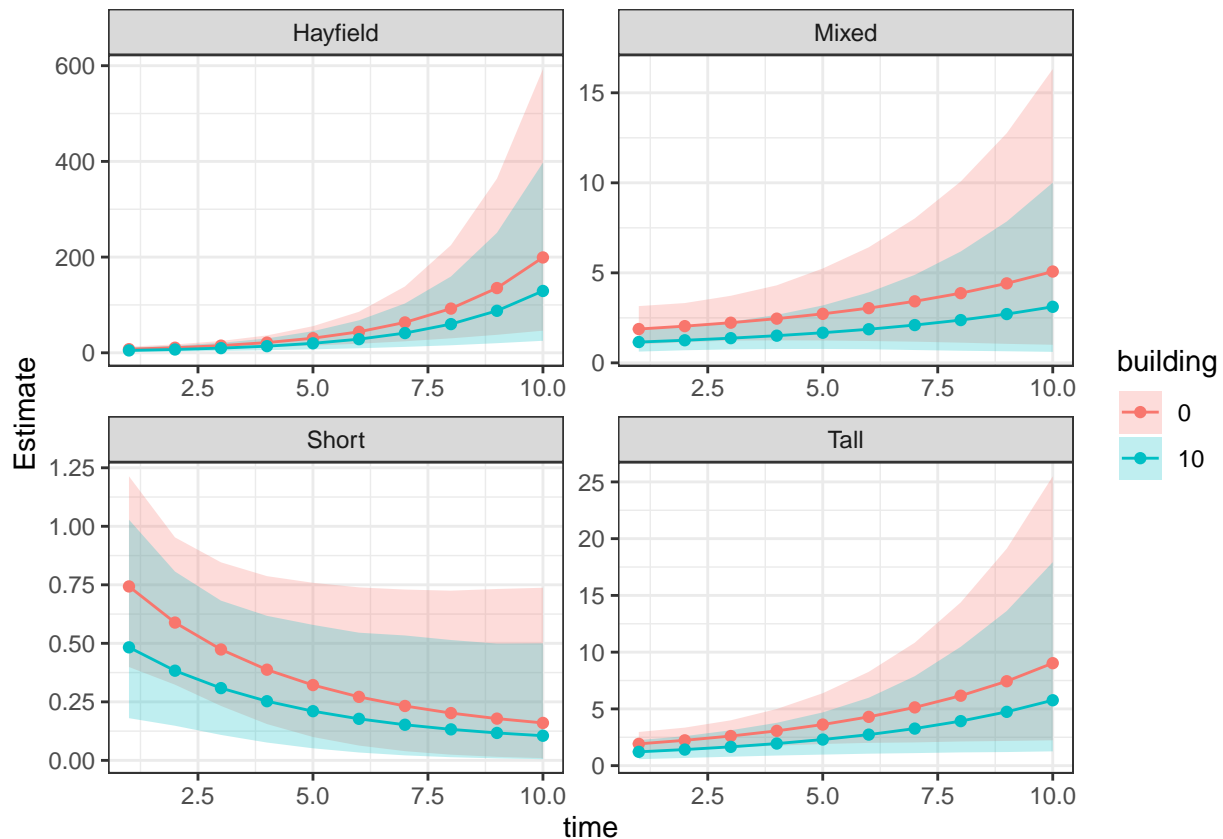
```
newdata <- data.frame(
  time = 1:10,
  building = rep(c(0, 10), each = 10),
  habitat = rep(unique(perhoset$habitat), each = 20)
)

pred_nb <- data.frame(
  newdata,
  fitted(fit_negbin, newdata = newdata, re_formula = NA)
)

pred_nb$building <- factor(pred_nb$building)

ggplot(
  pred_nb,
  aes(
    x = time, y = Estimate,
    fill = building, colour = building
  )
) +
  geom_ribbon(aes(ymin = Q2.5, ymax = Q97.5),
    colour = NA, alpha = 0.25
  ) +
  geom_line() +
  geom_point() +
```

```
theme_bw() +
facet_wrap(~habitat, scales = "free")
```



Sama kun `re_formula = NULL`, jolloin marginalisoidaan yli satunnaisefektien, ts. lasketaan ennusteet uudelle satunnaiselle havaintopaikalle:

```
newdata <- data.frame(
  time = 1:10,
  building = rep(c(0, 10), each = 10),
  habitat = rep(unique(perhoset$habitat), each = 20),
  site = 100 # uusi alue
)

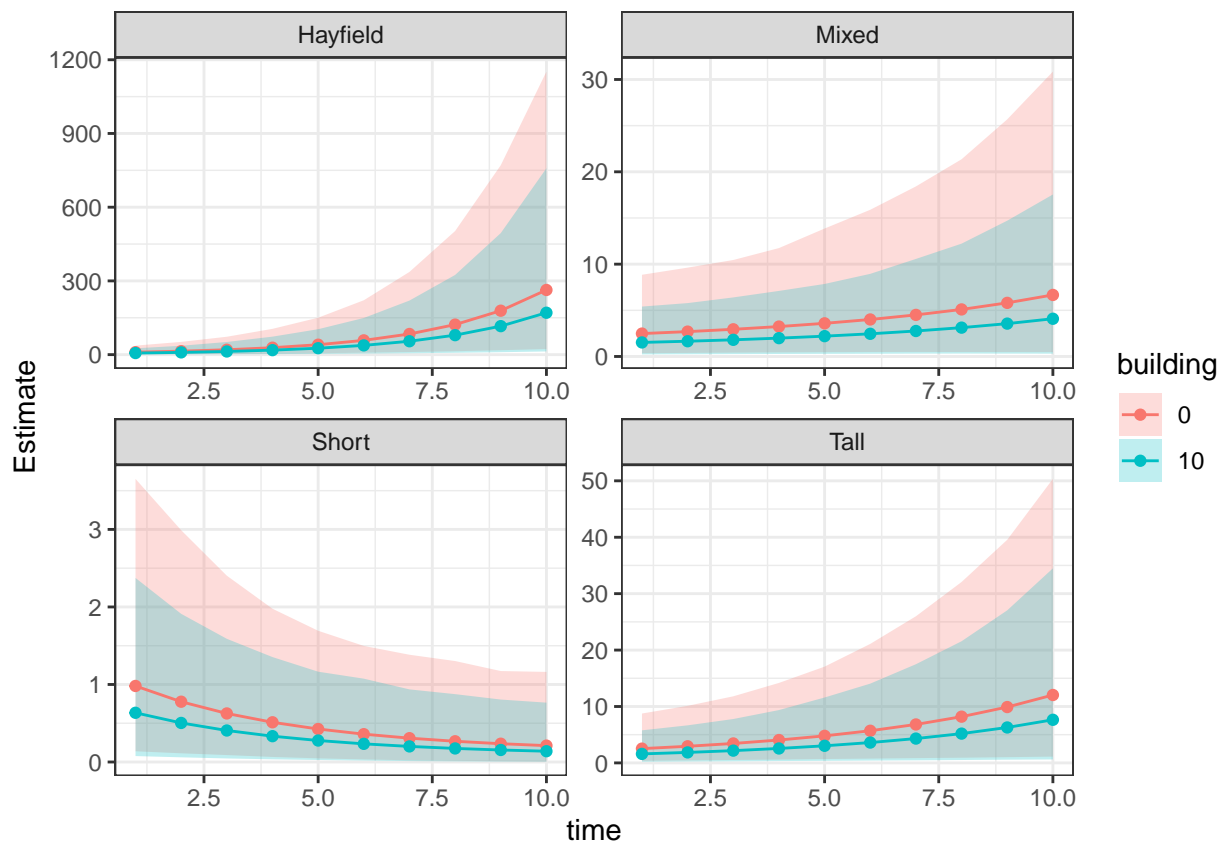
pred_nb <- data.frame(
  newdata,
  fitted(
    fit_negbin,
    newdata = newdata,
    re_formula = NULL,
    allow_new_levels = TRUE
  )
)
pred_nb$building <- factor(pred_nb$building)

ggplot(
  pred_nb,
```

```

aes(
  x = time, y = Estimate,
  fill = building,
  colour = building
)
) +
geom_ribbon(
  aes(ymin = Q2.5, ymax = Q97.5),
  colour = NA, alpha = 0.25
) +
geom_line() +
geom_point() +
theme_bw() +
facet_wrap(~habitat, scales = "free")

```



Ennusteet havaintopaikoille 1 ja 66 kun ruohotyyppi on "Mixed" ja rakennusprosentti on 0:

```

newdata <- data.frame(
  time = 1:10, building = 0, habitat = "Mixed",
  site = rep(c(3, 66), each = 10))

pred_nb2 <- data.frame(
  newdata,
  fitted(fit_negbin, newdata = newdata)
)
pred_nb2$site <- as.factor(pred_nb2$site)

```

```

ggplot(
  pred_nb2,
  aes(x = time, y = Estimate, fill = site, colour = site)
) +
  geom_ribbon(
    aes(ymin = Q2.5, ymax = Q97.5),
    colour = NA,
    alpha = 0.25
  ) +
  geom_line() +
  geom_point() +
  theme_bw()

```

