ÜLO MAIVÄLI

BAYESIAANLIK ANDMEANA

Sissejuhatus: maailm, teooria ja mudel

Suur ja väike maailm

Kuna maailmas on kõik kõigega seotud, on seda raske otse uurida. Teadlased püüavad asja lihtsamaks teha, lõigates reaalsuse väikesteks tükkideks, kasutades tordilabidana teaduslike hüpoteese. Nad teevad seda lootuses, et kui tükid on ükshaaval korralikult läbi nätsutatud, saab neist taas tordi kokku panna.[1] Tüüpiline teaduslik hüpotees pakub välja tavakeelse (mitte-matemaatilise) seletuse mõnele piiritletud loodusnähtusele. Neid väga erinevaid asju, mida me kutsume hüpoteesideks, saab sageli jagada osadeks, mida saab osaliselt kirjeldada matemaatiliste formalismide ehk mudelite abil. Kuigi hüpoteesid on üksteisest väga erinevad, on neid kirjeldavad mudelid sageli matemaatiliselt sarnased.

[1] Näiteks antibiootikume uuritakse keemilise sideme tasemel kasutades orgaanilise keemia meetodeid. Antibiootikumide molekulaarseid märklaudu uuritakse molekulaarbioloogiliste meetoditega, nende toimet uuritakse rakubioloogia ja füsioloogia meetoditega, aga kaasajal on väga olulised ka ökoloogilised, evolutsioonilised, meditsiinilised, põllumajanduslikud, majanduslikud ja psühholoogilised aspektid. Kõigil neil tasanditel on loodud hüpoteese, milledest kokku moodustub meie teadmine antibiootikumidest.

Kui mudel on teooria lihtsustus, siis teooria on maailma lihtsustus.

Mudeliteks nimetatakse väga erinevaid asju: skeeme, diagramme, füüsilisi mudeleid (näit Watsoni ja Cricki 1953. aasta nukleotiidimudelid), mudelorganisme, katsesüsteeme, võrrandisüsteeme jms. Üldiselt teeb mudeli mudeliks, et see asendab selle, mida teadlane tegelikult uurida tahab millegagi, mida on lihtsam mõista, manipuleerida või uurida. Meie räägime edaspidi ainult matemaatilisest mudelist ja eriti selle erijuhust, statistilisest ehk stohhastilisest mudelist.

Mis juhtub, kui teie mudel, ja seega ka hüpotees, mis mudeli genereeris, on andmetega kooskõlas? Kas see tähendab, et see hüpotees vastab tõele? Või, et see on tõenäoliselt tõene? Kahjuks on vastus mõlemale küsimusele eitav. Põhjuseks on asjaolu, et enamasti leiab iga nähtuse seletamiseks rohkem kui ühe alternatiivse teadusliku hüpoteesi ning rohkem kui üks üksteist välistav hüpotees võib olla olemasolevate andmetega võrdses kooskõlas. Asja teeb veelgi hullemaks, et teoreetiliselt on võimalik sõnastada lõpmata palju erinevaid teooriaid, mis kõik pakuvad alternatiivseid ja üksteist välistavaid seletusi samale nähtusele. Kuna hüpoteese, erinevalt andemtest, on lõpmata hulk, siis loogilise paratamatusena ei kata andmed kunagi

hüpoteese täielikut, ja igas teaduslikus faktis saab kahelda.

Samuti ei saa kindel olla, et parimad teooriad on meile üldse kunagi pähe torganud ning, et meie poolt kogutud andmed kajastavad hästi tegelikkust. Siit saab filosoof teha järelduse, et kui kindel teadmine maailma kohta peaks olema võimalik, siis on see, kuidas sellist teadmist kasvõi põhimõtteliselt hankida, meie jaoks sügavalt mõistatuslik.

(1) **Näide: politoloogia.**

Hüpotees (H_1) ütleb, et valijad käituvad ratsionaalselt, ehk lähtuvalt endi huvidest [@achen2016democracy]. Alternatiiv (H_2) ütleb, et valijad ei vali poliitikuid lähtuvalt oma tegelikest huvidest. Kuna H_1 on liiga hajus, et seda otse andmetega võrrelda, tuletame sellest kitsama alamhüpoteesi ($H_{1.1}$), mille kohaselt valijad eelistavad tagasi valida kandidaate, kes on ennast tõestanud sellega, et saavad hakkama majanduse edendamisega. Seega, poliitikud, kes on võimekad majanduse vallas, valitakse tagasi suurema tõenäosusega kui need, kes seda ei ole. Sellest hüpoteesist tuletati kaks andemete vastu testitavat järelmit: - $H_{1.1.1}$ – majandusel läheb keskeltläbi paremini juba tagasi valitud poliitikute all kui esimest korda valitud poliitkute all, kelle ridu ei ole veel elektoraadi poolt harvendatud ja - $H_{1.1.2}$ – majandusnäitajate varieeruvus on esimesel juhul väiksem, sest kehvemad poliitikud on juba valimist eemaldatud. Esimese järelmi testimiseks kasutati statistilise mudelina (m_1) aritmeetilist keskmist ja teise järelmi jaoks (m_2) standardhälvet.

Tulemused paraku ei klappinud $H_{1.1.1}$ ja $H_{1.1.2}$ poolt ennustatuga, millest autorid tegid järelduse, et olemasolevad andmed ei toeta hüpoteesi $H_{1.1}$ (andmete vähesuse tõttu nad ei arvanud, et nad oleksid $H_{1.1}$ ümber lükanud). Seega, andmed fititi mudelitesse m_1 ja m_2, nende fittide põhjal tehti järeldused, et m_1 ja $H_{1.1.1}$ ning m_2 ja $H_{1.1.2}$ vahel puudub kooskõla, mille põhjal omakorda tehti järeldus, et $H_{1.1}$ ei õnnestunud kinnitada, mille põhjal ei tehtud järeldust H_1 kohta. H_1 vs. H_2 kohta tehakse järeldus alles raamatu lõpus, lähtudes $H_{1.1}$, $H_{1.2}$, ..., $H_{1.n}$ kohta tehtud järeldustest.

(2) **Näide: populatsioonigeneetika.**

Populatsioonigeneetikas on evolutsioon defineeritud kui alleelide sageduste muutumine põlvkonnast põlvkonda. Enne kui hakkame vaatama alleelisageduste muutusi, defineerime tingimused, mille korral alleelide sagedus ei muutu. Need on juhuslik sigimine lõpmata suures populatsioonis, mis koosneb diploidsetest organismidest, kellel on 1 geneetiline lookus ja 2 alleeli. See on Hardy-Weinbergi printsiip, millel põhineb enamus klassikalisest populatsioonigeneetikast ja mida kirjeldab võrrand

Ca. 1910 mõtlesid Bertrand Russell ja G.E. Moore välja tõe vastavusteooria, mille kohaselt tõest lausungit eristab väärast vastavus füüsikalisele maailmale. Seega on tõesed ainult need laused, mis vastavad asjadele. Ehkki keegi ei oska siiani öelda, mida vastavus selles kontekstis tähendab või kuidas seda saavutada, on vastavusteooria senini kõige populaarsem tõeteooria filosoofide hulgas (mis on kõnekas alternatiivide kohta). Samamoodi, kui lausete vastavusest maailmaga, võime rääkida ka võrrandite (ehk mudelite) vastavusest lausetega. Vastavusest lausetaga sellepärast, et mudelid on loodud kirjeldama teaduslikke teooriaid, mitte otse maailma. Seega ei pea me muretsema mudelite tõeväärtuse pärast. Võib isegi väita, et mudeli tõeväärtusest rääkimine on kohatu.

$$p^2 + 2pq + q^2 = 1$$

kus p^2, $2pq$ ja q^2 on genotüüpide AA, Aa ja aa sagedused sugurakkudes ning p ja q on alleelide A ja a sagedused (ning $p + q = 1$). Populatsioonis, mis on Hardy-Weinbergi tasakaalus, on p ja q põlvkondade vältel muutumatud. Selleks, et tasakaalu lõhkuda, toome mudelisse lisaparameetri w, mis iseloomustab valikusurvet ehk kohasust (fitnessi). Kohasus iseloomustab looduliku valiku poolt tingitud genotüüpide sageduste muutust populatsioonis. Nüüd saame deterministliku mudeli: [2]

$$p^2 wAA + 2pqwAa + q^2 waa = w_{mean}$$

kus w_{mean} on populatsiooni keskmine kohasus, wAA on genotüübi AA kohasus jne. Kui me teame parameetrite p, q, wAA, wAa ja waa väärtusi, saame hõlpsasti arvutada populatsiooni kohasuse.

[2] Deterministliku, sest mudeli parameetritele kindlad väärtused omistades ja mudeli läbi arvutades saame vastuseks sama arvu, ükskõik mitu korda me seda arvutust ka ei kordaks

Vaadates maailma mudeli pilgu läbi, juhul kui looduses mõõdetud genotüüpide sageduse muutus erineb mudelist arvutatud w_{mean}-ist, siis on meil tegemist geneetilise triiviga. Geneetiline triiv on genotüübisageduste juhuslik muutus populatsioonis, mis on seda suurem, mida väiksem on populatsioon ja mida väiksem on valikusurve populatsioonile. Seega oleks nagu võimalik geneetilise triivi olemasolu tuvastada alati, kui empiiriline genotüübisageduste muutuse kiirus erineb mudeli punktennustusest w_{mean}. Selle deterministliku mudeli järgi on valik ja triiv teineteist välistavad: kui empiiriline kohasus = w_{mean}, siis valik; muidu triiv.

Kui me aga eeldame, et populatsiooni suurus ei ole lõpmata suur, tuleb mudelisse sisse juhuslik valimiviga. Mida väiksem on populatsioon, seda suurema tõenäosusega ei anna juhuslik paljunemine ka ilma valikusurveta populatsioonis järgmist põlvkonda, mille genotüübisagedused vastaksid eelmise põlvkonna genotüübisagedustele. Seega muutub meie deterministlik mudel stohhastiliseks mudeliks, mille väljund ei ole enam punktväärtus w_{mean}-le vaid rida tõenäosusi erinevatele w_{mean}-i väärtustele. Selle mudeli järgi ei ole valik ja triiv enam erinevat tüüpi protsessid, vaid ühe kontiinumi kaks poolust; kontiinumi, mis sõltub populatsiooni suurusest ja valikusurve tugevusest. Kuna puhas looduslik valik saab mudeli järgi toimuda ainult lõpmata suures populatsioonis, milliseid looduses ei leidu, siis on alleeli a sageduse muutus teadlase poolt uuritavas looduslikus populatsioonis x ühtaegu nii loodusliku valiku kui geenitriivi tagajärg.

Mis juhtub, kui me ei tee mudeli struktuurist otse järeldusi maailma kohta? Nüüd alustame me eeldusest, et looduslik valik on looduses toimuv protsess. Näiteks Darwin nägi valikut loodusliku põhjus-

liku protsessina, mis on samas stohhastiline (mitte kõik kõrgema kohasusega organismid ei anna järglasi). Selle vaate kohaselt on loodusliku valiku tagajärjeks kallutatud valim genotüüpidest, mille avaldumise poolt põhjustatud erinevused organismides viivad nende erinevale paljunemisedukusele. Seega on valik ja triiv erinevat tüüpi looduslikud protsessid, mitte mudeli väljundid, mistõttu eristame valikut triivist nende põhjuste järgi. Kui tõuseb kasulike genotüüpidega organismide osakaal, siis on tegemist loodusliku valiku poolt tingitud evolutsiooniga. Kui aga genotüüpide sageduste muutumine ei ole põhjustatud indiviidide füüsilistest erinevustest, siis on tegu geneetilise triivi poolt tingitud evolutsiooniga.

Nõnda saame evolutsiooniteooriast lähtudes hoopis teistsuguse vaate bioloogiale kui mudeleid otse tõlgendades. Muidugi ei tähenda see, et me ei vaja mudeleid. Vajame küll, aga me peame neid ettevaatlikult tõlgendama, pidades silmas oma teooriate sisu.

Andemetega fititud mudelit tõlgendame teooria kaudu ja seda ei tohiks kunagi teha otse mudelist päris maailmale.

Mudeli väike maailm

Ülalmainitud teadusliku meetodi puudused tingivad, et meie huvides on oma teaduslikke probleeme veel ühe taseme võrra lihtsustada, taandades need statistilisteks probleemideks. Selleks tuletame tavakeelsest teaduslikust teooriast täpselt formuleeritud matemaatilise mudeli ning seejärel asume uurima oma mudelit lootuses, et mudeli kooskõla andmetega ütleb meile midagi teadusliku hüpoteesi kohta. Enamasti töötab selline lähenemine siis, kui mudeli ehitamisel arvestati võimaliku andmeid genereeriva mehhanismiga – ehk, kui mudeli matemaatiline struktuur koostati teaduslikku hüpoteesi silmas pidades. Mudelid, mis ehitatakse silmas pidades puhtalt matemaatilist sobivust andmetega, ei kipu omama teaduslikku seletusjõudu, kuigi neil võib olla väga hea ennustusjõud.

Meil on kaks hüpoteesi, A ja B. Juhul kui A on tõene ja B on väär, kas on võimalik, et B on tõele lähemal kui A? Kui A ja B on teineteist välistavad punkthüpoteesid parameetri väärtuse kohta, siis on vastus eitav. Aga mis juhtub, kui A ja B on statistilised mudelid? Näiteks, kui tõde on, et eesti meeste keskmine pikkus on 178.3 cm ja A ütleb, et keskmine pikkus jääb kuhugi 150 cm ja 220 cm vahele ning B ütleb, et see jääb kuhugi 179 cm ja 182 cm vahele, siis on B "tõele lähemal" selles mõttes, et meil on temast teaduslikus mõttes rohkem kasu. Siit on näha oluline erinevus teadusliku hüpoteesi ja statistilise mudeli vahel: hüpotees on orienteeritud tõele, samal ajal kui mudel on orienteeritud kasule.

Mudeli maailm erineb päris maailmast selle poolest, et mudeli maailmas on kõik sündmused, mis põhimõtteliselt võivad juhtuda, juba ette teada ja üles loendatud (seda sündmuste kogu kutsutakse parameetriruumiks). Tehniliselt on mudeli maailmas üllatused võimatud.

Lisaks, tõenäosusteooriat, ja eriti Bayesi teoreemi kasutades on meil garantii, et me suudame mudelis leiduva informatsiooniga ümber käia parimal võimalikul viisil. Kõik see rõõm jääb siiski mudeli piiridesse. Mudeli eeliseks teooria ees on, et hästi konstrueeritud mudel on lihtsamini mõistetav — erinevalt vähegi keerulisemast teaduslikust hüpoteesist on mudeli eeldused ja ennustused läbinähtavad ja täpselt formuleeritavad. Mudeli puuduseks on aga, et er-

inevalt teooriast ei ole mingit võimalust, et mudel vastaks tegelikkusele.[3] Seda sellepärast, et mudel on taotluslikult lihtsustav (erandiks on puhtalt ennustuslikud mudelid, mis on aga sageli läbinähtamatu struktuuriga). Mudel on kas kasulik või kasutu; teooria on kas tõene või väär. Mudeli ja maailma vahel võib olla kaudne peegeldus, aga mitte kunagi otsene side. Seega, ükski number, mis arvutatakse mudeli raames, ei kandu sama numbrina üle teaduslikku ega päris maailma. Ja kogu statistika (ka mitteparameetriline) toimub mudeli väikses maailmas. Arvud, mida statistika teile pakub, elavad mudeli maailmas; samas kui teie teaduslik huvi on suunatud päris maailmale. Näiteks 95% usaldusintervall ei tähenda, et te peaksite olema 95% kindel, et tõde asub selles intervallis – sageli ei tohiks te seda nii julgelt tõlgendada isegi kitsas mudeli maailmas.

[3] Tõene mudel vastab kontseptuaalselt 1:1 mõõtkavas maakaardile (ja maakaart on samuti mudel).

(3) Näide: Aristoteles, Ptolemaios ja Kopernikus

Aristoteles (384–322 BC) lõi teooria maailma toimimise kohta, mis domineeris haritud eurooplase maailmapilti enam kui 1200 aasta vältel. Tema ühendteooria põhines maailmapildil, mis oli üldtunnustatud juba sajandeid enne Aristotelest ja mille kohaselt asub universumi keskpunktis statsionaarne maakera ning kõik, mida siin leida võib, on tehtud neljast elemendist: maa, vesi, õhk ja tuli. Maakera on liikumatult universumi keskpunktis, sest neli elementi liiguvad loomulikult ja iseenesest selle keskpunkti suunas Kõige raskem element, maa, jääb niiviisi keskele, selle peale koguneb vesi, mis on raskuselt järgmine, selle peale omakorda õhk, ja lõpuks kõige kergem element, tuli. Erinevalt sellest, mida me maa peal kogeme, on kogu maailmaruum alates kuu sfäärist tehtud viiendast elemendist (eeter), mida aga ei leidu maal (nagu nelja elementi ei leidu kuu peal ja sealt edasi). Taevakehad (kuu, päike, planeedid ja kinnistähed) tiirlevad ümber maa kontsentrilistes sfäärides, mille vahel pole vaba ruumi. Seega on kogu liikumine eetri sfäärides ühtlane ja ringikujuline ja just see liikumine põhjustab pika põhjus-tagajärg ahela kaudu kõiki liikumisi, mida maapeal kohtame, kaasa arvatud sündimine, elukäik ja surm. Kõik, mis maapeal huvitavat, ehk kogu liikumine, on algselt põhjustatud esimese liikumise poolt, mille käivitab kõige välimises sfääris paiknev meile mõistetamatu intellektiga olend ja mida vahenab maapealsetele uurijatele põhimõtteliselt kättesaamatu element, eeter.

Aristotelese suur teooria ühendab kogu maailmapildi alates meie mõistes keemiast ja kosmoloogiast kuni bioloogia, maateaduse ja isegi geograafiani. Sellist ühendteooriat on erakordselt raske ümber lükata, sest seal on kõik kõigega seotud.

Aristarchus (c. 310 – c. 230 BC) proovis seda siiski, väites, et tegelikult tiirleb maakera ümber statsionaarse päikese. Ta uskus ka, et

kinnistähed on teised päikesed, et universum on palju suurem kui arvati (ehkki kaasaegne seisukoht oli, et universumi mastaabis ei ole maakera suurem kui liivatera) ning, et maakera pöörleb ümber oma telje. Paraku ei suutnud Aristarchuse geotsentriline teooria toetajaid leida, kuna see ei pidanud vastu vaatluslikule testile. Geotsentrilisest teooriast tuleneb nimelt, et tähtedel esineb maalt vaadates parallaks, mis tähendab, et kui maakera koos astronoomiga teeb poolringi ümber päikese, siis kinnistähe näiv asukoht taevavõlvil muutub, sest astronoom vaatleb teda teise nurga alt. Pange oma nimetissõrm näost u 10 cm kaugusele, sulgege parem silm, seejärel avage see ning sulgege vasak silm ja te näete oma sõrme parallaksi selle näiva asukoha muutusena. Mõõtmised ei näidanud aga parallaksi olemasolu (sest maa trajektoori diameeter on palju lühem maa kaugusest tähtedest). Parallaksi suudeti esimest korda mõõta alles 1838, siis kui juba iga koolijüts uskus, et maakera tiirleb ümber päikese!

Ühte Aristotelese kosmoloogia olulist puudust nähti siiski kohe. Nimelt ei suuda see seletada, miks osad planeedid teavavõlvil suunda muudavad ja mõnda aega lausa vastupidises suunas liiguvad (retrogressioon). Kuna astronoomiat kasutasid põhiliselt astroloogid, siis pöörati planeetide liikumisele suurt tähelepanu. Lahenduseks ei olnud aga mitte suure teooria ümbertegemine või ümberlükkamine, vaid uue teaduse nõudmine, mis "päästaks fenomenid". Siin tuli appi Ptolemaios (c. AD 100 – c. 170), kes lõi matemaatilise mudeli, kus planeedid mitte lihtsalt ei liigu ringtrajektoori mõõda, vaid samal ajal teevad ka väiksemaid ringe ümber esimese suure ringjoone. Neid väiksemaid ringe kutsutakse epitsükliteks. See mudel suutis planeetide liikumist taevavõlvil piisavalt hästi ennustada, et astroloogide seltskond maha rahustada.

Ptolemaiosel ja tema järgijatel oli tegelikult mitu erinevat mudelit, millest osad ei sisaldanud epitsükleid. Olulise erinevusena Aristotelesest, kelle maailmapildis oli selline asi välistatud, ei asunud maakera ptolemailistes mudelites universumi keskpunktis, nii et päike ei teinud ringe ümber maakera vaid ümber tühja punkti.

Kuna leidus epitsüklitega mudel ja ilma epitsükliteta mudel, mis andsid identseid ennustusi, on selge, et Aristotelese teooria ja fenomenide päästmise mudelid on põhimõtteliselt erinevad. Kui Aritoteles **seletas** maailma põhiolemust põhjuslike seoste jadana (mitte matemaatiliselt), siis Ptolemaios **kirjeldas/ennustas** sellesama maailma käitumist matemaatiliste (mitte põhjuslike) struktuuride abil.

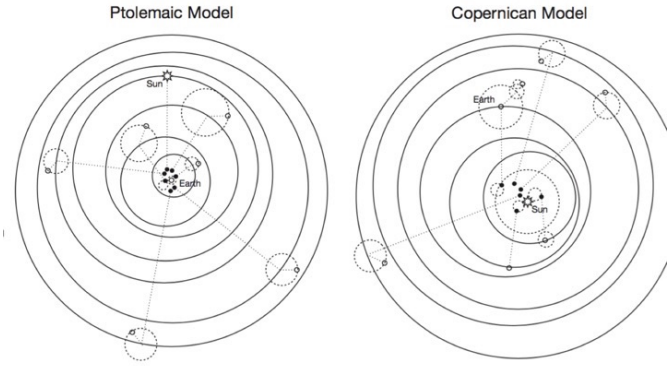
Nii tekkis olukord, kus maailma mõistmiseks kasutati Aristotelese ühendteooriat, aga selle kirjeldamiseks ja tuleviku ennustamiseks hoopis ptolemailisi mudeleid, mida keegi päriselt tõeks ei pidanud ja mida hinnati selle järgi, kui hästi need päästsid fenomene.

See toob meid Koperniku (1473 – 1543) juurde, kes ajaloolaste

arvates vallandas teadusliku revolutsiooni asetades päikese universumi keskele. Kas Kopernik tõrjus sellega kõrvale Aristotelese, Ptolemaiose või mõlemad? Tundub, et ta soovis kolmandat, suutis esimest ning tolleaegsete lugejate arvates üritas teist — ehk välja pakkuda alternatiivi ptolemailistele mudelitele, mis olid muutunud väga keerukaks, aga ka ennustustäpseks. Kuna Koperniku raamat läks trükki, kui autor oli juba oma surivoodil, kirjutas sellele eessõna üks tema vaimulikust sõber, kes püüdis oodatavat kiriku pahameelt leevendada vihjates, et päikese keskele viimine on vaid mudeldamise trikk, millest ei tasu järeldada, et maakera ka tegelikult ümber päikese tiirleb.[4] Kuna eessõna oli anonüümne, eeldasid lugejad, et selle kirjutas autor. Lisaks, kuigi Kopernik tõstis päikese keskele, jäi ta planeetide ringikujuliste trajektooride juurde, mis tähendas, et selleks, et mitte fenomenide päästmisel hätta jääda, oli ta sunnitud liigutama maakera ja planeete ümber päikese mõõda epitsükleid. Kokkuvõttes oli Koperniku mudel peaaegu sama keeruline kui ptolemailikud mudelid ja selle abil tehtud ennustused planeetide liikumise kohta olid väiksema täpsusega. Seega, ennustava mudelina ei olnud sel suuri eeliseid.[5]

[4] Piibel räägib, kuidas jumal peatas taevavõlvil päikese, mitte maa.

[5] Koperniku mudel suutis siiski ennustada nähtusi (planeetide näiv heledus jõuab maksimumi nende lähimas asukohas maale), mida Ptolemaios ei ennustanud. See ei tähenda, et need fenomenid oleksid olnud vastuolus Ptolemaiose mudeliga. Lihtsalt, nende Ptolemaiose mudelisse sobitamiseks oli vaja osad mudeli parameetrid fikseerida kindlatele väärtustele. Seega Koperniku mudel töötas sellisel kujul nagu see esitati, samas kui Ptolemaiose mudel vajas *post hoc* tuunimimst.



Kui aga vaadata Koperniku produkti teooriana, siis oli sellel selgeid eeliseid Aristotelese maailmateooria ees. Juba ammu oli nähtud komeete üle taevavõlvi lendamas (mis Aristotelese järgi asusid kinnistähtede muutumatus sfääris), nagu ka supernoova tekkimist ja kadu, ning enam ei olnud kaugel aeg, mil Galileo joonistas oma teleskoobist kraatreid kuu pinnal, näidates, et kuu ei saanud koosneda täiuslikust viiendast elemendist ja et sellel toimusid ilmselt sarnased füüsikalised protsessid kui maal.

Lausearvutuslik loogika

Enne, kui siirdume tõenäosusteooria juurde, teeme lühikese sissejuhatuse klassikalisse loogikasse, sest tõenäosusteooria ei ole lõppude-lõpuks midagi muud, kui loogika laiendus juhule, kus me ei ole kindlad selles, mida räägime. Niisiis, loogika ülesanne on modelleerida inimkeelseid lauseid.[6] Nõnda mudeldame me ühtlasi mõtlemist, kaasa arvatud teaduslik mõtlemine. Nagu ikka, ei eelda me ka siin, et mudel vastaks täpselt tegelikkusele.

[6] või nende sisu ehk propositsioone.

Matemaatiliselt on loogika tihedalt seotud hulgateooriaga:

- *eksperiment (defineerituna laialt, ja nõnda kasutatud ainult selles peatükis): iga protsess, mida me saame vaadelda ja mille käitumine tulevikus on osaliselt ennustamatu.*
- *sample space (sündmusteruum, hüpoteesiruum): Eksperimendi kõikide võimalike tulemuste hulk. Sündmusteruumi elemendid on katsepunktid (sample points).*
- *sündmus on sündmusteruumi alamhulk.*
- *Sündmus A sisaldub sündmuses B siis, kui iga A element on ka B element, ehk $A \subset B$ või ekvivalentselt $B \supset A$, ehk A implitseerib B-d.*
- *A ja B on võrdsed (A = B) siis, kui $A \subset B$ ja $B \subset A$. A ja B on võrdsed siis kui nad koosnevad samadest sündmusteruumi elementidest.*
- *Ilma elementideta hulk on tühi hulk ja sellele vastav sündmus on nullsündmus ehk võimatu sündmus.*
- *Hulk, mis sisaldab kõiki sündmusteruumi elemente, mis ei kuulu sündmusesse A, on mitte-A, ehk A komplement ehk $\neg A$. Sündmus $\neg A$ toimub kui A ei toimu.*
- *Hulk, mis sisaldab kõiki sündmusteruumi elemente, mis kuuluvad sündmusesse A või sündmusesse B (või mõlemasse) on A ja B ühisosa, ehk $A \cup B$ ehk $A \vee B$ ehk A või B. Sündmus $A \cup B$ toimub, kui toimub A või toimub B või toimuvad nii A kui B.*

- *Hulk, mis sisaldab kõiki sündmusteruumi elemente, mis kuuluvad nii sündmusesse A kui sündmusesse B, on A ja B intersektsioon, ehk $A \cap B$ ehk $A \wedge B$ ehk A ja B. Sündmus $A \cap B$ toimub, kui toimuvad nii A kui B.*

- *Kui $A \cap B$ on tühi hulk, siis on A ja B üksteist välistavad sündmused (disjoint events).*

Meie keelemudeli baas-süntaks koosneb sõnadest nagu "ja", "või", "mitte", "kui ... siis", mida kutsume *konnektiivideks*.

Suured tähed A, B, C, ... tähistavad *atomaarseid lauseid*.[7] Iga atomaarne lause tähistab ühte või mitut inimkeelset lauset. Loogiku jaoks pole atomaarsete lausete sisemine struktuur oluline, sest sellest ei sõltu mudelkeele lausete valiidsus.

Mudelkeele esimene tase koosneb atomaarsetest lausetest, mis on ühendatud konnektiividega. Need on seega juba *liitlaused* ehk *komposiidid*. Teise taseme laused koosnevad konnektiividega ühendatud 1. taseme lausetest, ja nii edasi. Näiteks $((A \to B) \vee (B \wedge A)) \to C$ on 3-tasemeline lause, kus sulud näitavad, milliseid komponentlauseid mingi konnektiiv parasjagu ühendab.

Lisaks konnektiividele sisaldab meie keelemudel tõeväärtusi: T(rue) ja F(alse). Me eeldame, et mitmetasemeliste lausete tõeväärtused sõltuvad ainult nende aluseks olevate atomaarsete lausete tõeväärtustest.[8]

Konnektiivid koos neid tähistava sümboliga:
not ($\neg$) – negatsioon
and ($\wedge$) – konjunktsioon
or ($\vee$) – disjunktsioon
if ... then ($\to$) – implikatsioon e konditsionaal (if – *antecent*, then – *consequent*).

[7] Me tähistame edaspidi suuri tähti A, B, ... - lause, propositsioon, sündmus, hüpotees, tõendusmaterjal, andmed jms. Loogika matemaatilise vormi seisukohast pole vahet, kuidas me neid nimetame. Me kasutame erinevaid tähistusi, sest need seovad loogika matemaatilise struktuuri loogika rakendustega erinevates teadus- ja filosoofia harudes.

[8] Seda eeldust kutsutakse tõetabeli printsiibiks.

Tõetabel

Tõetabel formaliseerib konnektiivide kasutamise, kehtesdades reeglid, mis defineerivad mudelkeele niiviisi, et selle laused sarnaneksid võimalikult palju tavakeelele. Kui me tähistame suvalist liitlauset X-ga, siis tõetabel näeb välja nii:

A	B	X
T	F	
F	T	
T	T	
F	F	

Tõetabel annab kõik võimalikud kombinatsioonid atomaarsete lausete tõeväärtustest ja ütleb iga sellise kombinatsiooni kohta, kas X on tõene või väär. Tabeli iga rida annab kombinatsiooni atomaarsete lausete tõeväärtustest, mis omakorda määrab X-i tõeväärtuse sellel real.

Negatsioon

Kõigepealt anname tõetabeli negatsioonile, mis on unaarne konnektiiv, ehk töötab ühe lause piires

A	not A
T	F
F	T

Sõnadega: $\neg A$ on tõene siis, kui A on väär, ja vastupidi. Negatsioon ei tee muud, kui põõrab lause tõeväärtuse vastupidiseks.

Konjunktsiooni võib kasutada näiteks nii:
P1 (Premiss 1): Seda ja teist
J1 (Järeldus 1): Seda
J2: teist
Või:
P1: ei ole külm ega tuuline
J1: ei ole külm
J2: ei ole tuuline
Aga lausest $\neg(A \wedge B)$ ei saa midagi järeldada:
P1: Ma ei ole praegu Pariisis ega Tallinnas
J1: —

Konjunktsioon

Nüüd tõetabel konjunktsioonile, mis on binaarne konnektiiv, ühendades kahte lauset.

A	B	A and B
T	T	T
T	F	F
F	T	F
F	F	F

Ehk sõnadega: $A \wedge B$ on tõene siis ja ainult siis kui A ja B on mõlemad tõesed.

Disjunktsiooni saab kasutada näiteks nii:
P1: Ei seda ega teist ($\neg(A \vee B)$)
J1: Ei seda ($\neg A$)
J2: ei teist ($\neg B$)
Või
P1: ei mitte-A ega mitte-B ($\neg(\neg A \vee \neg B)$)
J1: A
J2: B
Samas, lausest $A \vee B$ ei saa midagi järeldada:
P1: Ma olen kas Pariisis või Tallinnas
J1: —-

Disjunktsioon

Nüüd disjunktsioon. See on inklusiivne *või*, mis kehtib ka siis, kui A ja B mõlemad kehtivad.

A	B	A or B
T	T	T
T	F	T
F	T	T
F	F	F

$A \vee B$ on väär siis ja ainult siis, kui A ja B on mõlemad väärad.

Kuidas aga oleks lood ekslusiivse disjunktsiooniga (*xor*), kus A = T ja B= T viivad väärale disjunktsioonile? Me ei vaja xor jaoks tingimata eraldi konnektiivi (sümbolit), sest selle tõetabel langeb kokku lause

$$(A \vee B) \wedge \neg(A \wedge B)$$

tõetabeliga.

Selle arvutamiseks evalueerime kõigepealt sisemised disjunktiooni $A \vee B$ ja $A \wedge B$, seejärel negatsiooni $\neg(A \wedge B)$ ning lõpuks kogu lause (see on tabelis keskel olev "and").

A	B	A or B (I)	and (III)	not (II)	A and B (I)
T	T	T	F	F	T
T	F	T	T	T	F
F	T	T	T	T	F
F	F	F	F	T	F

Tabeli evalueerimise järjekord on antud rooma numbritega tabeli veergude päistes.

Ja võrdluseks A xor B tõetabel

A	B	A xor B
T	T	F
T	F	T
F	T	T
F	F	F

Kuna nende tabelite läbiarvutamisel saadud tõeväärtused on identsed, on meil tegu loogilise ekvivalentsusega [9].

$$(A \vee B) \wedge \neg(A \wedge B) \Leftrightarrow A \; xor \; B$$

[9] Ekvivalentsed võivad olla ka laused, mis ei koosne samadest atomaarlausetest, senikaua kui nende tõetabeli kõik read on sama tõeväärtusega (näiteks A ja $A \wedge (B \vee \neg B)$).

Disjunktsiooni ja konjunktsiooni on võimalik avaldada teineteise kaudu:

$$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$$

$$\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$$

Konditsionaal

Ja lõpuks konditsionaali $A \to B$ tõetabel

A	B	if A then B
T	T	T
T	F	F
F	T	T
F	F	T

Konditsionaal on väär siis ja ainult siis kui A on tõene ja B on väär. Vahest kipuvad inimesed nägema konditsionaali põhjusliku seose

Konditsionaali saab kasutada näiteks nii
P1: $\neg(A \to B)$
J1: A
J2: $\neg B$
Jällegi, lausest $A \to B$ ei saa midagi järeldada A ega B kohta.
P1: $A \to B$
J1: —

mudelina. See ei ole hea mõte, sest loogilised tehted eeldavad ainult koos või eraldi esinemist, mitte põhjuslikke ega ajalisi suhteid.

Konditsionaali $A \rightarrow B$ osaline vaste tõenäosusteoorias on tingimuslik tõenäosus $P(B \mid A)$, mis ütleb "B tõenäosus tingimusel, et A on tõene".

Veel üks oluline samasus:

$$A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$$

Lisaks võime konditsionaali avaldada ka läbi disjunktsiooni või konjunktsiooni:

$$A \rightarrow B \Leftrightarrow \neg A \vee B \Leftrightarrow \neg(A \wedge B)$$

Ainus põhjus, miks meil on eraldi konnektiiv nimega konditsionaal, on selle järelduslikku vormi sage kasutamine. Seega on konditsionaal loogikas sisuliselt vähetähtis mugavussümbol, mitte põhjusliku seose sügavmõtteline mudel.

Tautoloogia ja kontradiktsioon

Tautoloogiad on laused, mille tõetabelis on X alati tõene.

A	A or notA
F	T
T	T

Tautoloogiad väljendavad loogikas vankumatuid tõdesid, seega annavad nad meile loogika tuletusreeglid. Järgnevalt mõned näited:

- $(A \vee \neg A)$-st tuleb välistatud kolmanda seadus, mille kohaselt iga propositsioon on kas tõene või väär (ja mitte kunagi mõlemat korraga).
- $(A \rightarrow A)$-st tuleb samasusseadus (ühegi lause sisu ei muutu arutluse käigus)
- $(\neg(A \wedge \neg A))$-st tuleb vasturäkivusseadus (ükski lause ei saa olla iseendaga vastuolus)
- $(A \rightarrow (B \rightarrow A))$-st tuletub implikatsioon.

Seevastu $A \wedge \neg A$ on kontradiktsioon ehk vasturäkivus, sest selle tõetabelis on X alati väär. Vasturäkivused lähtuvad alati võimatutest eeldustest, mis tähendab, et nende olemasolu lükkab ümber eelduste süsteemi, mis nad sünnitas.

A	A and notA
F	F
T	F

Enamus lauseid ei ole ei tautoloogiad ega kontradiktsioonid, vaid kontingentsed. Kontingentsed laused võivad olla nii tõesed kui väärad.

Nagu juba eespool mainitud, kui tõetabelis leidub rida, kus kõik atomaarsed laused on tõesed ja X on väär, siis ja ainult siis on tegu kontradiktsiooniga. Antud juhul on selline tabeli 2. rida.

loogiline argument ja valiidne järeldamine

Loogiline argument koosneb premissidest ja järeldustest. Premissid on laused, mille kohta me eeldame, et need on tõesed, ja järelduse me dedutseerime premissidest lähtuvalt sellest eeldusest. Näiteks: (i) maa on kerakujuline või kuu on juustust. Lisaks eeldame, (ii) et maa ei ole kerakujuline. Siit tuleb loogiliselt valiidne järeldus: kuu on juustust, ehk

---

P1: $A \vee B$
P2: $\neg A$
J: B

Siin on meil tegemist loogilise **argumendiga**, mis koosneb kahest **premissist** (P1 ja P2) ja **järeldusest** (J).

Definitsioon: premissidest Γ järeldub loogiliselt lause p siis ja ainult siis, kui $\Gamma \wedge \neg p$ on vasturääkiv

Mis juhtub, kui me eeldame, et järeldus B on hoopiski väär, aga premissid $\neg A$ ja $A \vee B$ on mõlemad tõesed? Sellisel juhul on meil tegu loogilise vasturääkivuse e kontradiktsiooniga. Seega on premissidest dedutseeritud järeldus loogiliselt tõsikindel; iga deduktiivne järeldus on juba peidus premissides ja ei sisalda endas uut informatsiooni.

Järelduse loogiline valiidsus (loogiline kehtivus) ei taga selle korrektsust, ehk kehtivust päris maailmas (kui tagaks, siis me elaksime vaid matemaatikast koosnevas maailmas, mille mõistmiseks poleks vaja teha empiirilisi uuringuid). Me võime sama hästi eeldada, et (P1) maa on kerakujuline või kuu on juustust e $A \vee B$, (P2) et kuu ei ole juustust e $\neg B$, ja siit järeldub, et maa on kerakujuline:

P1: $A \vee B$
P2: $\neg B$
J: A

Argument on **valiidne** siis ja ainult siis, kui olukord, kus kõik premissid oleksid tõesed ja järeldus oleks väär, on loogiliselt vasturääkiv. Argument on **korrektne** (*sound*) siis ja ainult siis, kui see on valiidne ja kõik premissid on tõesed.

See järeldus on valiidne ja korrektne, aga empiiriliselt mitte väga huvitav.

Argumendi valiidsus tähistab pelgalt selle semantilist struktuuri (ehk loogilist vormi). Argumendi korrektsus tähendab, et argumendi järeldus on ka sisuliselt kehtiv ehk tõene. Valiidne järeldamine eeldab, et premissid ja järeldus on ehitatud atomaarsetest

lausetest nii, et ei esine atomaarsete lausete tõeväärtuste kombinatsiooni, mis muudaks kõik premissid tõeseks ja järelduse vääraks. Kui siiski esineb selline kombinatsioon, siis oleme leidnud loogilise vasturääkivus ehk kontradiktsiooni ja meie järeldamismehhanism ei saa olla valiidne.

Selle näitlikustamiseks kontrollime argumendi

P1: $A \rightarrow B$
P2: $\neg A$
J: $\neg B$

---

valiidsust tõetabeli abil:

A	B	P1: if A then B	P2: notA	J: notB
T	T	T	T	T
T	F	F	F	T
F	T	T	T	F
F	F	T	T	T

Tõetabelist on näha, et see argument ei ole valiidne, sest tabeli 3. reas on tõesed premissid ja väär järeldus. Nii lihtne see ongi. Pane tähele, et sellises tõetabelis on huvitavad ainult sellised read, kus ükski premiss pole väär ja järeldus on väär. Kõiki teisi ridu võib ignoreerida. Kuna tabeli ridade arv võrdub kaks astmes atomaarsete lausete arv, tasub seda meeles pidada.

Modus Ponens ja Modus Tollens

Bertrand Russelile, kellel on suured teened formaalse loogika arendamisel 20. sajandi alguses, kuulub väike nali teadusliku meetodi kohta, nagu seda nägid paljud 20. sajandi teadusfilosoofid (Russell, 1945):

> If p, then q; now q is true; therefore p is true. E.g. if pigs had wings then some winged animals are good to eat; therefore pigs have wings. This form of inference is called `scientific method`.

See inglise huumor näitlikustab induktiivset teadusliku mõtlemise mudelit, mis ekslikult kasutab deduktiivse lausearvutusliku süllogismi mitte-valiidset vormi. Tegemist on sedavõrd levinud eksitusega, et sellel on lausa oma ladinakeelne nimi, mida võib tõlkida kui "peale seda, järelikult selle pärast" (*Post hoc ergo propter hoc*).

Selle süllogismi vähem naljakas rakendus oleks:
P1: Kui patsiendil on gripp, siis on tal (tõenäoliselt) palavik
[$A \rightarrow B$]
P2: palavik [B]
J1: gripp [A]
J2: tõenäoliselt gripp
Paraku kumbki järeldus ei kehti.

Kavalam katse lausearvutusliku loogika abil teaduslikku mõtlemist mudeldada kuulub teadusfilosoof Karl Popperile (ca 1930). Et Popperi mudelit tutvustada, alustame valiidsest deduktiivsest argumendist ladinakeelse nimega *Modus Ponens*

P1: $A \to B$
P2: A
J: B
Ehk,
P1: kõik mehed on sead (kui mees, siis siga)
P2: Aristoteles on mees
J: Aristoteles on siga

Modelleerimaks üldist ja alati kehtivat loodusseadust, mis oli Popperi jaoks teaduslik teooria *par excellence*, seondub selle argumendiga probleem, millest oli teadlik juba Aristoteles. Kui me tahame tõsikindlalt näidata, et kõik mehed on sead, siis peame minema induktiivset rada pidi ja testima tõepoolest kõiki mehi – praegusi, eilasi ja homseid – selles osas, kui palju nad sigu meenutavad. See ei ole paraku teostatav.

Popper püüdis probleemi lahendada, tuues sisse valiidse deduktiivse argumendi vormis *Modus Tollens*:

P1: $A \to B$
P2: $\neg B$
J: $\neg A$
ehk:
P1: kõik mehed on sead
P2: Aristoteles ei ole siga
J: Aristoteles ei ole mees

Aga seda võib vaadata ka nii: Kui me eeldame, et Aristoteles siiski on mees, ja et Aristoteles ei ole siga, siis argumendi valiidsuse päästmiseks teeme järelduse, et P1 on väär (st kõik mehed ei ole teps mitte sead). Sellisel viisil loogilise vasturääkivuse lahendamine on täiesti lubatud ja soositud tegevus.

Seega oli Popperi retsept teadlastele (loe: füüsikutele)

1. postuleeri üldine teooria kujul kõik X-d on Y.

2. Dedutseeri sellest testitav alamteooria vormis x_i on Y.

3. Juhul kui me suudame empiiriliselt näidata, et x_i on väär, oleme sellega deduktiivselt ümber lükanud ka X kehtimise.

Seda skeemi illustreerib hästi Enrico Fermi tsitaat:

> If your experiments succeed in proving the hypothesis, you have made a measurement; if they fail to prove the hypothesis, you have made a discovery.

Teooria falsifitseerimine

Sellist suure teooria ümber lükkamist kitsama haardega alamteooria testimise läbi nimetatakse teooria falsifitseerimiseks. Siit tuleneb ka Popperi ettepanek teaduse ja mitte-teaduse eristamiseks: kõik teaduslikud teooriad peavad olema vähemalt põhimõtteliselt falsifitseeritavad, sest muidu ei saaks neid Popperi teadusliku mõtlemise mudeli abil ümber lükata. [10] Igal juhul lõi Popper kõigepealt teadusliku mõtlemise formaalse mudeli ning teatas seejärel, et kuna tema mudel töötab ainult teatud struktuuriga teooriate peal, siis peaksid teadlased ajama oma teooriad just sellesse vormi, või leppima sildiga "mitte-teaduslik".

Falsifikatsionismi teine puudus seisneb selles, et alamteooria ümberlükkamiseks *Modus Tollensi* abil, peame olema täiesti kindlad, et meie katseaparatuur teeb seda, mida me tahame, et mõõtmisviga ei vii meid ekslikele järeldustele jne. Muidu me ei saaks kasutada lausearvutuslikku loogikat, mille premissid saavad omada vaid kahte tõeväärtust ja seega ei saa olla tõenäosuslikud ega võib-olla kehtivad. Popperi vastus sellele vastuväitele oli, et isegi kui me kasutame olude sunnil eeldusi, mille kehtivuses ei saa kindel olla, peame olema valmis vajaduse korral iga sellise eelduse lähemaks uurimiseks avama ja läbi vaatama. [11] Senikaua kui meie vaimne tasakaal vastab eelmises lauses kirjeldatule, on see, mida me teeme, Popperi mõistes teadus. Paraku, teaduse ja mitte-teaduse vahelist piiri ei määra enam mitte see, mida me teeme, vaid see, mida me põhimõtteliselt oleksime nõus tegema. [12]

Falsifitseerimise kui teadusliku mõtlemise mudeli põhiline puudus on, et see töötab lausearvutusliku loogika raames, mis tähendab, et see jääb paratamtult hätta teooriatega, mis ennustavad millegi juhtumist tõenäosuslikult. Näiteks teooria, mille kohaselt suitsetamine põhjustab kopsuvähki, aga mitte igal suitsetajal. [13] Lausearvutuses ei ole ühtegi mehhanismi tõenäosuslike propositsioonidega töötamiseks ja Popperi, kes oli mõnede arvates oma põlvkonna nutikaim filosoof, 70 aastat kestnud pingutused selline mehhanism luua, jooksid liiva.

[10] Sellest tuleneb omakorda, et mida lihtsam on teooriat falsifitseerida, seda "teaduslikum" on see teooria. Näiteks teooria, mille kohaselt igal kolmapäeval kell 14:00 sajab Ilmatsalu ilmajaamas 3 mm õllevihma, on suurepäraselt falsifitseeritav ja seega super-teaduslik.

[11] loomulikut "võimaluse korral ja rahaliste vahendite olemasolul".

[12] Seega on tavapärastele mudeli-eeldustele (mudel peegeldab reaalsust) lisatud eeldused laia maailma kohta (et me teooria testimisel suudame tuvastada õiged maailma-eeldused, mida teaduse jaoks avada). Kogu see kompott tundub kahtlane.

[13] Suitsetamise põhiline suremust tõstev mõju on läbi südamehaiguste, mitte vähi.

Lausearvutusest tõenäosuste loogikasse

Mis juhtub, kui meie premiss ei ole mitte "kuu on tehtud juustust" vaid "kuu võib olla tehtud juustust"? Sellisel juhul ei tule loogiline järeldus kujul "A", vaid "tõenäoliselt A". Lausearvutuse reeglid eeldavad, et premissid on kas tõesed või väärad, mis tähendab, et premisside ja järelduste tõenäosused tohivad omada vaid kahte väärtust, 1 ja 0. Seega vajame siin teistsugust loogikat, mis lubaks ebakindlate premisside põhjal teha parimaid võimalikke ebakindlaid järeldusi. Me vajame tõenäosusteooriat.

Kui lausearvutus töötab must-valges tõene-väär maailmas, siis tõenäosusteooria opereerib halli varjunditega. Tõenäosusteoorialt ootame, et see annaks meile järeldused kujul "A tõenäosus" - $P(A)$ - või "A tõenäosus, juhul kui kehtib B" - $P(A|B)$. [14] Lisaks ootame, et alati, kui tõenäosused on fikseeritud ühe ja nulliga, annaks tõenäosusteooria välja samad järeldused kui lausearvutus. Üldiselt ootame me mõlemalt loogikalt sama: konverteerida premissid parimateks võimalikeks järeldusteks, mida saaksime (küll mööndustega) mudeli maailmast päris maailma üle kanda.

[14] $P(A|B)$ on *tinglik tõenäosus* ja $P(A)$ on *marginaalne tõenäosus*.

Lausearvutus on deduktiivne süsteem, kus järelduse tõesus sisaldub juba premissides. Kui loogik on valiidse järelduseni jõudnud, siis see järeldus on igavene – seda ei saa muuta uusi premisse või andmeid lisades. Seda omadust nimetatakse monotoonilisuseks. Teisisõnu, lausearvutuslik loogika on mõtlemise mudel, mis ei sisalda kahtlusi ega isegi võimalust kahtlusteks. Selline mudel ei ole ilmselgelt see, mida otsib teadlane, kes peab oma järeldusi tegema mittetäieliku informatsiooni tingimustes.

Tõenäosusteooria on matemaatika haruna deduktiivne aksiomaatiline süsteem, aga mõtlemise mudelina kasutatakse seda hoopiski induktiivsel moel. See tähendab, et me püüame piiratud andmete põhjal jõuda ebakindlatele järeldustele, aga seejuures seda ebakindlust numbriliste tõenäosuste abil kvantifitseerides. Uusi andmeid lisades saame oma episteemilise ebakindluse määra muuta, aga ainus viis saavutada tõsikindlust (ja monotoonilisust), on tuues arvutusse sisse null- ja ühiktõenäosused [15]. Seega on tõenäosusteooriat mõtlemise mudelina rakendades teaduslikus praktikas üsna võimatu jõuda tõsikindlatele järeldustele. Ja inimesed, kes teaduses opereerivad lausearvutusliku loogikaga (ja seega ei mõtle tõenäosuslikult), eeldavad vaikimisi, et nende jaoks on teadus matemaatikat meenutav tõsikindel süsteem, mis oma sisendites (katseskeemid, andmed, nende analüüs) ei sisalda ebakindlust. Samas, ka meie, kes me kasutame tõenäosusteooriat, peame eeldama, et see on vaid mõtlemise mudel, mitte teaduslik mõtlemine ise oma ehedal kujul. Kohe, kui keegi mõtleb välja parema mudeli, hakkame kõik kasu-

[15] "Episteemiline ebakindlus" tähendab, et segadus asub meie peas, mitte maailma ülesehituses

tama seda. Aga senikaua peame õppima tõenäosusteooriat ja selle rakendust, mida kutsume Bayesi statistikaks.

Tõenäosusteooria

Et mõista, kuidas tõenäosusteooria töötab, läheme tagasi aastasse 1654, mil sai lahenduse juba ca 200 aastat euroopa parimaid päid vaevanud küsimus. Küsimus, mis sisuliselt pani aluse kaasaegsele loodusteaduslikule mõtlemisele ja mille lahendusest sõltus lisaks paljude tavaliste inimeste hingerahu, oli järgmine: kaks mängurit panevad lauale võrdse summa raha, mis on määratud mängu võitjale. Oletame, et nad heidavad täringuid ja et iga heide, mil üks neist saab rohkem silmi kui teine, läheb sellele mängijale kirja punktina. Kes iganes kogub esimesena n punkti, võidab mängu. Siiamaani on kõik selge, võitja võtab kõik. Aga nüüd tuleb konks. Nimelt peavad mängijad olude sunnil mängu lõpetama, enne kui keegi on n punkti kogunud, ja neil on vaja otsustada juba kogutud punktide põhjal, kuidas võiduraha õiglasel viisil omavahel ära jagada.

Esimese hooga võib see tunduda lihtne ülesanne – miks mitte jagada raha vastavalt juba kogutud punktide suhtele? [16] Kuid veidi järele mõeldnuna hakkab meid närima kahtluseuss. Mis siis, kui mängijad tahtsid mängida kuni 100 punktini? Siis peaks 1-punktiline vahe mängu alguses olema palju väiksema kaaluga kui olukorras, kus plaan oli koguda näiteks 4 punkti. Seega, kuigi me teame minevikku ja ei saa kunagi teada, mis oleks juhtunud tulevikus, mis kõigele lisaks jääb igavesti sündimata, oleme olukorras, kus ei ole võimalik teha õiglasi tulevikku suunatud otsuseid!

[16] Kui ühel mängijal on 2 punkti ja teisel 1 punkt, siis esimene saaks 2/3 ja teine 1/3 võidusummast

See ülimalt pessimistlik järeldus kehtis sajandeid, kuni Blaise Pascal ja Pierre Fermat võidusumma jagamise küsimuse uuesti üles võtsid ja selle lahendasid (selleks andis ajendi Pascali sõbra Chevaliere de Mere praktiline vajadus). Selgus, et õiglane lahendus on üllatavalt lihtne, kuid tuleb meile kätte kõrge hinnaga, milleks on loomuliku mõtlemise pea peale keeramine. Fermat ja Pascali lahenduse võti seisnes kardinaalses vaatepunkti nihutamises: selle asemel, et projitseerida minevikku tulevikku, peame vaatlema kõiki loogiliselt võimalikke tulevikke, ja seda isegi olukorras, kus me teame, et meie maailmas ei realiseeru neist ükski.

Oletame, et mängu katkestamise hetkel on mängijal A-l on 2 punkti, B-l on 3 punkt ning et plaan oli mängida, kuni emb-kumb

kogub 5 punkti. Me teame kindlalt, et meie maailmas ei ole sellel mängul tulevikku, aga püüdkem siiski hetkeks ette kujutada olukorda, kus mäng siiski jätkub. [17] Nüüd on loogiliselt võimalikud järgmised tulevikud:

1) A2B3 –> A3B3 –> A4B3 –> A5B3

2) A2B3 –> A3B3 –> A4B3 –> A4B4 –> A5B4

...

16) A2B3 –> A2B4 –> A2B5

Siin lahkneb iga maailmaliin võrdse tõenäosusega kaheks uueks, kus järgmise punkti kogub üks kahest mängijast. Võidusumma õiglaseks jagamiseks peame teadma, mitmes tulevikus võitnuks mängu A (5) ja mitmes B (11). Seega saab A 5/16 võidusummast ja B saab ülejäänu, ehk 11/16.

[17] Kujutelge, et me elame universumis, mis on osa multiversumist, kus iga sündmus, mis saab toimuda k-l erineval viisil, viib antud universumi lahknemisele k-ks uueks universumiks. Selles metafüüsikas leidub lõpmata palju universumeid, millest igaühes viib iga tolmukübeme lend uutele lahknemistele.

Nüüd jõuame tõenäosuse mõiste juurde. Oletame, et me heidame oma koduses universumis münti 1000 korda ja saame 500 kulli. Seega on kullide suhteline sagedus meie katses 500/1000 = 0.5 ja omades piisavalt suurt katseseeriat võime öelda, et tõenäosus, et iga järgmine veel mittetoimunud mündivise annab kulli, on ligikaudu 0.5 ehk 50% [18]. Sellise tõenäosuse laiendamine mänguritele on lihtne, ainus erinevus on, et me ei võta sagedust enam oma maailmaliinist vaid üle kõikide võimalike maailmade. Seega on tõenäosus, et mängu oleks võitnud mängija A 5/16 ehk 0.3125 ja mängijale B on see 1 - 0.3125 = 11/16 = 0.6875.

[18] Varjatud eeldustena usume, et me tulevikus viskame samasugust münti sarnasel viisil ja et kui me viskaksime uue 1000-se seeria, siis tulemus oleks üsna sarnane sellega, mida nägime.

Tõenäosusteooria mõistes on need alternatiivsed ja mitte kunagi tõeks saavad (sest mäng ei lähe neist üheski kunagi lõpuni) maailmad sama reaalsed kui juba toimunud mündivisete seeria meie oma maailmas. Aga tõenäosus ei kajasta enam mitte fakte, vaid meie teadmiste baasil loodud mudelit, mille kitsastes piirides loeme ülesse kõik loogiliselt võimalikud tulevikud, sõltumata sellest, kas me usume, et mõni neist iial realiseerub. Meie jaoks ei ole tähtis, kas need paralleeluniversumid ka tegelikult kuskil olemas on – piisab sellest, et suudame neid oma vaimusilmas ette kujutada. Seega elavad tõenäosused mudeli maailmas, sõltudes mudeli eeldustest – antud juhul täringuvisete tulemuste sõltumatus ja see, et mõlemal mängijal on võrdne võimalus võita igal täringuviskel – ja need kajastavad meie uskumusi maailma kohta, sest mudel põhineb just nendel uskumustel. Huvitaval kombel eeldab tõenäosusteoorial põhinev ratsionaalsus midagi, mis tavamõtlemisele tundub täiesti ebaratsionaalsena: nimelt lähtumist sellest, mis oleks võinud juhtuda, kuid

mille kohta me teame, et see kunagi ei juhtu. Sellisel mõtlemise väänamisel on oma hind, teadusliku mõtlemise ebaintuitiivsus, aga ka omad kasud.

Kui piirdume tõenäosuste arvutamisel vaid oma maailmaliinist võetud sagedustega, siis saame tõenäosuste arvutamiseks kasutada ainult juba toimunud sündmusi, millel on stabiilne toimumissagedus. Tõenäosusteooria laiendab tõenäosuste kasutust ka ühekordsetele sündmustele nagu tuumasõja toimumine järgmise aasta jooksul või teadusliku teooria tõelevastavus, milledel meie maailmaliinis ei ole sagedust, ja isegi sellistele sündmustele, mis ei saagi meie maailmas toimuda (nagu tuumasõja toimumine 1965 aastal, kui Kennedyt poleks tapetud). Tõenäosusteooria abil saab küsida ka juba toimunud üskiksündmuste kohta – näiteks milline on tõenäosus, et 1. aprilli kevadtormi Saaremaal põhjustas globaalne kliimasoenemine [19]. Tõenäosusteooria abil teeme formaalselt järeldusi kõikide võimalike maailmade kohta, aga kasutame neid oma maailma mõistmiseks. Selline äraspidine mõtlemine maailmadest, mida me ei saa kunagi külastada, ongi kaasaegse teadusliku meetodi tuksuv süda.

[19] Selleks võrdleme sellest tormist tugevamate tormide esinemissagedust alternatiivsetel Saaremaadel, kus ei toimu kliimasoenemist, nende maailmadega, kus see toimub sarnaselt meie omale. Põhjuslik mõju on selles vaates tõenäosuslik, mitte deterministlik, ja selle mõju tugevus tuleb välja counterfaktuaalses analüüsis: kui me jätaksime ära põhjuse, kui suure tõenäosusega jääks siis ära ka tagajärg

---

Veel üks võimalus andmetest üksiksündmuse tõenäosusele jõuda on andmeid mudeldades. Kui me mudeldame mõõtmisviga normaaljaotusega ja sobitame selle jaotuse kuju oma andmetega, siis saame me normaaljaotuse funktsiooni matemaatiliste omaduste kaudu arvutada, millise tõenäosusega jääb üksik tulevikumõõtmine mingisse ette antud vahemikku. Sama kehtib loomulikult ka teiste jaotusmudelite (lognormaal, binoom jne) kohta. Põhimõtteliselt saame sama tulemuse sooritades n mõõtmist ja küsides n+1 mõõtmise kohta, milline osakaal algsetest mõõtmistest on sellest väiksemad või suuremad (eeldades, et n on suur). Sellisel matemaatiliselt ülilihtsal moel töötame hiljem posterioorsete valimitega

---

On aeg laskuda teooria kõrgustest maa peale ja sukelduda järeldavasse statistikasse, mis oma olemuselt on ümberpööratud tõenäosusteooria. Kui tõenäosusteoorias me fikseerime mudeli parameetrid (täringuvisete sõltumatus ja silmade võrdsed tõenäosused) ja neist lähtuvalt arvutame andmed (erinevate täringusilmade kombinatsioonide osakaalud), siis järeldavas statistikas me vastupidi lähtume andmetest (täringuvistest), et arvutada mudeli parameetrid (näiteks, kui palju erineb meie täringu puhul ühe silma saamise tõenäosus 1/6-st). Kui tõenäosusteoorias me eeldame, et teame, kuidas süsteem on üles ehitatud, ja ennustame sellest lähtuvalt võimalike (hüpoteetiliste) andmete tõenäosusi, siis statistikas me kontrollime neid

eeldusi päriselt olemasolevate andmete põhjal. Seega annab tõenäosusteooria matemaatiliselt tõsikindlaid vastuseid ideaalmaailmade kohta, samas kui statistika püüab andmete põhjal teha järeldusi päris maailma kohta.[20]

Järeldav statistika arvutab vastused kahele küsimusele: 1. mis on kõige usutavam parameetriväärtus? ja 2. kui suur ebakindlus seda hinnangut ümbritseb? Kuna andmed tulevad meile lõpliku suurusega valimina koos mõõtmisvea ja bioloogilise varieeruvusega, on ebakindlus hinnagusse sisse ehitatud. Hea protseduur kvantifitseerib selle ebakindluse ausalt. Eesmärk ei ole mitte ebakindlust vähendada (seda teeme eelkõige katse planeerimise tasemel), vaid seda võimalikult täpselt kirjeldada. Järeldav statistika püüab teha andmete põhjal järeldusi looduse kohta.

Süstemaatilist viga ei saa kunagi välistada – see on loogiline paratamatus, mis tuleneb asjaolust, et andmeid on vähem kui võimalikke veaallikaid, mille vahel saab aga vahet teha vaid neidsamaseid andmeid kasutades. Seega veaallikad, millest igaüks on vaadeldav eraldiseisva alternatiivse teadusliku hüpoteesina, on andmete poolt alamääratud, mistõttu kindel teadmine teaduses on võimatu. Teaduslikud hüpoteesid ulatuvad alati teistpoole andmeid, ja ütlevad rohkem kui oleks võimalik andmete põhjal õigustada. Seega sisaldab teadus endas alati usku, mistõttu me söandame riskida isegi mudelite kasutamisega oma hüpoteeside kinnitamisel.

See õpik õpetab Bayesi statistikat, mis põhineb tõenäosusteoorial ja tänu sellele moodustab sidusa terviku. Bayesi statistika põhineb Bayesi teoreemil, mis on triviaalne tuletis tõenäosusteooria aksioomidest. Tänu Cox-i teoreemile (1961) teame, et klassikaline lausearvutuslik loogika on tõenäosusteooria erijuht ning, et Bayesi teoreem on teoreetiliselt parim viis tõenäosustega töötamiseks. Seega, kui te ei saa oma järeldustes päris kindel olla, siis on teoreetiliselt parim lahendus tõenäosusteooria ja Bayesi teoreem.

[20] Kui tõenäosusteooria tegeleb deduktiivse järeldamisega, sest fikseeritud mudeli puhul on ka andmete sagedused rangelt fikseeritud ja täpselt määratavad, siis järeldav statistika on induktiivne selles mõttes, et andmed ei fikseeri mudeli struktuuri üheselt. Me saame samu andmeid sobitada mitme erineva mudeliga ja pole võimatu, et need sobituvad võrdselt hästi paljudesse täiesti erinevatesse mudelitesse. See tähendab, et statistika ja sellel põhinev teadus ei saa meile anda matemaatilise kindlusega kehtivaid järeldusi, ja et lõppude lõpuks sõltub meie teaduslike järelduste kvaliteet mudelite kvaliteedist.

Ebakindluse allikad on (i) mõõtmisviga, mis on sageli normaaljaotusega, (ii) bioloogiline varieeruvus, mis on sageli lognormaaljaotusega, (iii) mudeli viga, kus matemaatiline jaotusfunktsioon ei vasta looduses toimuvale, (iv) algoritmi viga, kus algoritm ei tee seda, mida kasutaja tahab ja (v) süstemaatiline viga, mis juhtub, kui te saate valesti aru oma katsesüsteemist, harrastate teaduslikku pettust või teete kõike muud, mis kallutab teie andmeid mingis kindlas suunas.

Tõenäosusteooria aksioomid on tuletatavad järgmistest eeldustest:

- hüpoteesi usutavuse määra saab kirjeldada reaalarvuga 0 ja 1 vahel
- ratsionaalne mõtlemine vastab kvalitatiivselt tervele mõistusele: tõendusmaterjal hüpoteesi toetuseks tõstab selle hüpoteesi usutavust.
- mõtlemine peab olema konsistentne: kui me saame järeldusi teha rohkem kui ühel viisil, peame lõpuks ikkagi alati samale lõppjäreldusele jõudma

Tõenäosusteooria määrab kõikide võimalike sündmuste esinemise tõenäosused, eeldades, et hüpotees H kehtib (H on siin lihtsalt teine nimi "eeldusele").

Statistika hindab H-i kehtimise tõenäosuse lähtuvalt kogutud andmetest, matemaatilistest mudelitest ning taustateadmistest.

- kogu kättesaadav relevantne informatsioon tuleb järelduste tegemisel arvesse võtta (totaalse informatsiooni printsiip)
- ekvivalentsed teadmised on representeeritud ekvivalentsete numbritega.

Tõenäosusteooria aksioomid, mida on neli tükki, ütlevad tõlkes inimkeelde, et

(1) iga sündmuse/hüpoteesi/atomaarlause (edaspidi "sündmuse") tõenäosus on suurem või võrdne nulliga - $P(A) \geq 0$,

(2) loogiliselt paratamatu sündmuse tõenäosus on üks - $P(\Omega) = 1$,[21]

(3) üksteist välistavate sündmuse puhul võrdub tõenäosus, et toimub üks või teine sündmus, nende sündmuste tõenäosuste summaga - $P(A \vee B) = P(A) + P(B)$. See on *finiitse additiivsuse printsiip*.

(4) et sündmuse A tõenäosus, juhul kui me eeldame sündmuse B kehtimist, võrdub nende kahe sündmuse koosesinemise tõenäosuse jagatisega sündmuse B tõenäosusest - $P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$. [22]

[21] Ω on *sample space*, mis koosneb üksteist välistavatest ja ammendavatest hüpoteesidest. Nende hüpoteeside tõenäosuste summa on 1, mis tähendab, et täpselt üks neist kehtib paratamatult.

[22] $P(A \mid B)$ on tinglik tõenäosus, kus me ei väida mitte, et B päriselt kehtib, vaid küsime: "Kui peaks juhtuma, et B kehtib, siis milline oleks sellisel juhul A tõenäosus?".

Need aksioomid, mis postuleeriti Andrei Kolmogorovi poolt ca 1933, peaksid olema iseenesestmõistetavad ja ainult neist on tuletatud kogu tõenäosusteooria.[23] Tõenäosusteooria on matemaatika haru, mis tähendab, et sümbolitel P, A, B, jms ei ole muud fikseeritud tähendust, kui et need käituvad vastavalt tõenäosusteooria aksioomidele ja neist dedutseeritud teoreemidele. Nendes piirides võime anda neile sümbolitele ükskõik millise tähenduse, mis seoks tõenäosusteooria matemaatilise struktuuri päris maailmaga. Näiteks $P(A \mid B)$ võib tähendada "hüpoteesi A tõenäosust tingimusel, et meil on andmed B", aga sama hästi ka "andmete A tõenäosust tingimusel, et kehtib hüpotees B", või ka midagi muud. $P(A)$ võib meie jaoks tähistada "hüpoteesi tõenäosust", "andmete tõenäosust", "tõendusmaterjali tõenäosust" ja "sündmuse tõenäosust", aga ka "homse vihma tõenäosust" või "tõenäosust, et parameetri väärtus > 2").

[23] Kui 4. aksioom ei tundu teile iseenesestmõistetavana, siis lahendage järgmine ülessanne: urnis on 3 kera ja 3 kuupi, millest kaks kera on sinised ja üks on punane. Millise tõenäosusega tõmbame kastist punase kera? Lahendus:
$P(kera) = 1/2$; $P(punane \mid kera) = 1/3$;
$P(kera \wedge punane) = P(kera)P(punane \mid kera)$
$= 1/2 \times 1/3 = 1/6$
Siit on lihtne avaldada 4. aksioom.

Kui tõenäosused on 0 või 1, siis taandub tõenäosusteooria matemaatliselt oma erijuhule, milleks on lausearvutuslik loogika. Lausearvutusel on huvitav omadus, monotoonilisus, mille kohaselt kui juba on saavutatud loogiliselt valiidne tulemus, siis uute andmete lisandumisel ei saa me seda muuta. Seevastu tõenäosusteoorias ja statistikas muudavad uued andmed hüpoteesi kehtimise tõenäosust. Selles mõttes ei saa tõenäosuslik teadus kunagi valmis ja kui inimene on 100% veendunud mingi hüpoteesi/sündmuse tõesuses või vääruses, siis seisavad tema uskumused väljaspool teadust selles mõttes, et neid ei ole võimalik teaduslike argumentidega mõjutada.

Mõned tuletised tõenäosusteooria aksioomidest

Me anname siin tuletused ilma tõestuskäikudeta, mis on aga lihtsad. Tinglike tõenäosuste puhul eeldame, et nii A kui B tõenäosus > 0.

Tõenäosusteooria põhituletised:

5. $0 \leq P(A) \leq 1$ - tõenäosused jäävad 0 ja 1 vahele

6. $P(\neg A) = 1 - P(A)$, üksteist välistavate ammendavate hüpoteeside tõenäosused summeeruvad ühiktõenäosusele.

7. Monotoonsus: Kui A sisaldub B-s ($A \subset B$), siis $P(A) \leq P(B)$. Kui B tuleneb deduktiivselt A-st $[A \vee (B \wedge \neg A) \Leftrightarrow B]$, siis $P(B) \leq P(A)$.

8. $P(A \ \& \ B) \leq P(A)$; $P(B) \leq P(A \vee B)$

9. Kui B tuleneb deduktiivselt A-st ja $P(A) > 0$, siis $P(B \mid A) = 1$ ja $P(\neg B \mid A) = 0$.[24]

10. Loogiliselt ekvivalentsed propositsioonid/hüpoteesid on sama tõenäosusega – kui $A \Leftrightarrow B$, siis $P(A) = P(B)$

11. Definitsioon: A ja B on üksteisest sõltumatud siis ja ainult siis kui $P(A \mid B) = P(A)$

12. Kui A ja B on üksteisest sõltumatud, siis

$$P(A \wedge B) = P(A)P(B)$$

13. Kui A ja B ei ole üksteisest sõltumatud, siis

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

ja kolmele sündmusele:

$$P(A \vee B \vee C) = P(A) + P(B) + P(C) - \\ P(A \wedge B) - P(B \wedge C) - P(A \wedge C) + \\ P(A \wedge B \wedge C)$$

14. Totaalne ehk marginaalne tõenäosus:

$$P(A) = P(A \mid B)P(B) + P(A \mid \neg B)P(\neg B)$$

ehk

$$P(A) = P(A \mid B_1)P(B_1) + P(A \mid B_2)P(B_2) + ...$$

üksteist välistavatele B-dele.

[24] Kui tõendusmaterjal e tuleneb deduktiivselt hüpoteesist H (H ennustab e-d) ja kui $P(H) > 0$ ning $P(e) < 1$, siis $P(H \mid e) > P(H)$, ehk e tõstab H tõenäosust.

15. Bayesi teoreem:

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

kus vastavalt 15. punktile

$$P(B) = P(A)P(B \mid A) + P(\neg A)P(B \mid \neg A)$$

või

$$P(B) = P(A_1)P(B \mid A_1) + P(A_2)P(B \mid A_2) + ...$$

Bayesi teoreemi kasutatakse määramaks hüpoteesi tõenäosuse pärast uute faktide (andmete) lisandumist olemasolevatele teadmistele. Selleks peab hüpoteesiruum olema jagatud vähemalt kaheks ammendavaks ja üksteist välistavaks hüpoteesiks. Kui A on H_1 ning mitte-A on ammendav ja välistav H_2 ja B tähistab andmeid (data), saame Bayesi teoreemi ümber kirjutada

$P(H_1 \mid data)$ on H_1 kehtimise tõenäosus meie andmete ja eelteadmiste korral, ehk posteerior,
$P(H_1)$ on H_1 kehtimise eelnev, meie andmetest sõltumatu tõenäosus, ehk prior,
$P(data \mid H_1)$ on andmete esinemise tõenäosus tingimusel, et H_1 kehtib, ehk tõepära.

$$P(H_1 \mid data) = \frac{P(H_1)P(data \mid H_1)}{P(H_1)P(data \mid H_1) + P(H_2)P(data \mid H_2)}$$

Jagamistehe normaliseerib ühele kõikide hüpoteeside tõenäosuste summa meie andmete korral ja seeläbi viib posteeriori vastavusse tõenäosusteooria aksioomidega – kui meil on i ammendavat üksteist välistavat hüpoteesi, siis murrujoone alla läheb $\sum P(data \mid H_i)P(H_i)$.

Bayesi teoreem on triviaalne tuletus tõenäosusteooria aksioomidest, milles pole midagi maagilist. See ei ole automaatne meetod, mis tagaks inimkonna teadmiste kasvu, vaid lihtsalt parim võimalik viis andmemudeli ja taustateadmiste mudeli ühendamiseks ja normaliseerimiseks tinglikuks tõenäosuseks (hüpoteesi tõenäosus meie andmete ja taustateadmiste korral). Edasi sõltub kõik mudelite, andmete ja taustateadmiste kvaliteedist.

Näited tõenäosusteooria tuletiste rakendamisest

Järgnevatel näidetel on ühist kaks asja: need on matemaatiliselt triviaalselt lihtsad, aga intuitiivselt lootusetult keerulised. Kõigi nende puhul on inimestel tugev intuitsioon, mis on vale – ja tõenäosusteooria tundmine ei anna meile paremat intuitsiooni. Seega, ainus, mis üle jääb, on iga probleemi taandamine tõenäosusteooria valemitele ja selle tuimalt läbi arvutamine.

8. Punkt Linda on 31 aastane, vallaline, sõnakas ja väga nutikas. Ta õppis ülikoolis filosoofiat ja muretses sel ajal sügavalt diskrimineerimise ja sotsiaalse õigluse pärast ning osales tuumarelva vastastel meeleavaldustel. Kumb on tõenäolisem? Linda on pangateller. Või

Linda on pangateller, kes osaleb feministlikus liikumises. Kuigi enamus vastajatest eelistab 2. varianti, on see sõna otses mõttes loogikavastane.

12. Punkt Kui me viskame täringut 3 korda, kui suure tõenäosusega saame vähemalt ühe kuue? Naiivselt võiks arvata, et see tõenäosus on 50%. Kuid rakendades tõenäosusteooriat saame teistsuguse vastuse. Lihtsuse huvides defineerime küsimuse ümber: kui suure tõenäosusega ei saa me 3-l viskel ühtegi kuute? Vastus: kui igal viskel on 0 kuue tõenäosus 5/6, siis $(5/6) * (5/6) * (5/6) = 0.58$ ja $1 - 0.58 = 0.42$, mis tähendab, et vähemalt 1 kuue (või ükskõik mis numbri ühest kuueni) saame 42% tõenäosusega. Teine näide (NYT 03-12-2017): te ostate maja Texases Hustonis, millele müüja annab garantii, et üleujutuse tõenäosus on 1% aastas. Seadus nimetab seda näidikut "100 aasta suurvee-tasemeks". 1% näidu puhul ei pea te seaduse järgi ostma üleujutusekindlustust. Kui suure tõenäosusega tabab teie maja üleujutus pangalaenu perioodi vältel (30 aastat)? Vastus: $1 - (99/100)^{30} = 0.26$.

13. Punkt Kui tõenäosus, et homme sajab pussnuge on 0.1 ja et ülehomme sajab pussnuge on 0.1, siis millise tõenäosusega sajab vähemalt ühel neist päevadest? Eeldades sündmuste sõltumatust:

$$P(homme\ sajab \vee lehomme\ sajab) = \\ 0.1 + 0.1 - 0.1 \times 0.1 = 0.19$$

Kui me aga teame, et sadu erinevatel päevadel on korreleeritud näiteks nii:

$$P(lehomme\ sajab \mid homme\ sajab) = 0.2$$

$$P(lehomme\ sajab \mid \neg homme\ sajab) = 0.15$$

siis

$$P(homme\ sajab \vee lehomme\ sajab) = \\ P(homme\ sajab) + P(lehomme\ sajab) - \\ P(homme\ sajab \quad \& \quad lehomme\ sajab)$$

Nüüd peame arvutama $P(lehomme\ sajab)$, kasutades 15. punkti (marginaliseerimist), misjärel saame valemi

$$P(A \vee B) = P(A) + P(B) - P(A)P(B \mid A)$$

Kui vihm on korreleeritud, siis väheneb tõenäosus, et sajab vähemalt ühel päeval.

4. Punkt Meil on kolm münti, millest 1. on aus, 2. on mõlemal küljel kull ja 3. on mõlemal küljel kiri. Te valite juhulikult ühe neist müntidest, viskate seda ja saate kirja. Millise tõenäosusega on teisel küljel kull? Vastus ei ole 50%. Lahendus: vastavalt 4. aksioomile

$$P(kull|kiri) = \frac{P(kull \wedge kiri)}{P(kiri)}$$

$P(kull \wedge kiri) = 1/3$, sest ühel mündil 3st on erinevad küljed ja tõenäosus, et juhuslikult valitud mündi alumisel küljel on kull, on keskmine kolmest tõenäosusest, millega me saame kolmel mündil kulli: mean(c(1, 1/2, 0)) = 1/2. Seega, vastus on $(1/3)/(1/2) = 2/3$. Kui me saame mündiviskel kirja, siis on tõenäosusega 2/3 on teisel küljel kull!

Selles näites mõõdab tõenäosus selgelt episteemilist ebakindlust, sest pärast mündiviset pole ülesandes midagi juhuslikku. Teie kaaslane võib münti kergitades õige vastuse teada saada, misjärel tema tõenäosus on 0 või 1, aga senikaua kui ta teile selle kohta vihjet ei anna, jääb teie jaoks tõenäosus endiseks. See tähendab, et tõenäosus on pelgalt teie isikliku teadmatuse mõõt. Kuna me selle ülesande lahendamiseks kasutasime tingliku tõenäosuse definitsiooni, siis igaüks, kes piirab enda jaoks tõenäosuse mõiste juhuslike protsessidega, kasutab tõenäosust viisil, mis keelab ära tingliku tõenäosuse kui sellise.

14. Punkt Kui A tähistab sündmust "ma sooritan eksami edukalt" ja B tähistab sündmust "ma õpin eksamiks", ning meil on dihhotoomne valik: õpin / ei õpi, siis

$$P(hea\ hinne) = P(pin)P(hea\ hinne \mid pin) + P(ei\ pi)P(hea\ hinne \mid ei\ pi)$$

Ehk sõnadega kirjutatult: Hea hinde tõenäosus võrdub korrutisega kahest tõenäosusest – tõenäosus, et ma eksamiks õpin, ja tõenäosus, et ma saan hea hinde siis kui ma õpin –, millele tuleb liita teine korrutis kahest tõenäosustest – tõenäosus, et ma ei õpi, ja tõenäosus, et ma saan hea hinde ka ilma õppimata. Siit saad ise enda jaoks välja arvutada ennustuse, millise tõenäosusega just sina selle kursuse edukalt läbid.

15. Punkt 1) Bayesi teoreemi rakendamine diskreetsetele hüpoteesidele: Oletame, et 45 aastane naine saab rinnavähi sõeluuringus mammograafias positiivse tulemuse. Millise tõenäosusega on tal rinnavähk? Kõigepealt jagame hüpoteesiruumi kahe diskreetse hüpoteesi vahel: H_1 - vähk ja H_2 - mitte vähk. Edasi omistame numbrilised väärtused järgmistele parameetritele:

1. H_1 tõepära, ehk tõenäosus saada positiivne mammogramm juhul, kui patsiendil on rinnavähk (testi sensitiivsus): $P(+ \mid H_1) = 0.9$
2. H_2 tõepära, ehk tõenäosus saada positiivne mammogramm juhul, kui patsiendil ei ole rinnavähki (1 - testi spetsiifilisus): $P(+ \mid H_2) = 0.08$. Pane tähele, et 0.9 + 0.08 ei võrdu ühega, mis tähendab, et tõepära pole tõenäosusteooria mõttes päris tõenäosus.
3. Eelnev tõenäosus, et patsiendil on rinnavähk $P(H_1) = 0.01$ (see on rinnavähi sagedus 45 a naiste populatsioonis; kui me teame pat-

siendi genoomi järjestust või rinnavähijuhte tema lähisugulastel, võib P(H_1) tulla väga erinev).

4. $P(H_2) = 1 - P(H_1) = 0.99$

Nüüd arvutame posterioorse tõenäosuse $P(H_1 \mid +)$

```
likelihood_H1 <- 0.9
likelihood_H2 <- 0.08
prior_H1 <- 0.01
prior_H2 <- 1 - prior_H1
posterior1 <- likelihood_H1 * prior_H1/(likelihood_H1 * prior_H1 + likelihood_H2 *
    prior_H2)
posterior1
```

```
## [1] 0.1020408
```

Nagu näha, postiivne tulemus rinnavähi sõeluuringus annab 10% tõenäosuse, et teil on vähk (ja 90% tõenäosuse, et olete terve). Selle mudeli parameetriväärtused vastavad enam-vähem tegelikele mammograafia veasagedustele ja tegelikule populatsiooni vähisagedusele.

Mis juhtub, kui me teeme positiivsele patsiendile kordustesti? Nüüd on esimese testi posteerior meile prioriks, sest see kajastab definitsiooni järgi kogu teadmist, mis meil selle patsiendi vähiseisundist on (muidugi eeldusel, et me esimese mudeli kohusetundlikult koostasime).

```
likelihood_H1 <- 0.9
likelihood_H2 <- 0.08
prior_H1 <- posterior1
prior_H2 <- 1 - prior_H1
posterior2 <- likelihood_H1 * prior_H1/(likelihood_H1 * prior_H1 + likelihood_H2 *
    prior_H2)
posterior2
```

```
## [1] 0.5610973
```

Patsiendile võib pärast kordustesti positiivset tulemust öelda, et ta on 44% tõenäosusega vähivaba. Eelduseks on, et me ei tea midagi selle patsiendi geneetikast ega keskkonnast põhjustatud vastuvõtlikusest vähile ning, et testi ja kordustesti vead on üksteisest sõltumatud (mitte korreleeritud).

2) Bayesi teoreemi kasutamine kohtus. Olgu meil 2 hüpoteesi: syydi ja syytu, ning 2 ühikut tõendusmaterjali: DNA ja tunnistaja ütlus. Lisaks on syydistataval nõrk motiiv ja nõrk alibi. Mida me teada

tahame, on kui palju tõenäolisem on syyaluse syy võrreldes syytusega.

$$\frac{P(syydi|tqendid)}{P(syytu|tqendid)} = \frac{P(DNA|syydi)}{P(DNA|syytu)} \times \frac{P(ytlus|syydi)}{P(ytlus|syytu)} \times \frac{P(motiiv|syydi)}{P(motiiv|syytu)} \times \frac{P(alibi|syydi)}{P(alibi|syytu)}$$

Siin annab meile alibi ja motiivi tõepärasuhete korrutis prioriite suhte ja ülejäänud kaks liiget lähevad kirjal tõepärasuhetena. Iga juurdetulev iseseisev tõend läheb uue liikmena korrutamistehtesse sisse, ja olles arvesse võtnud kogu relevantse tõendusmaterjali saame teada, mitu korda on süüaluse süü tõenäolisem kui süütus. Sõltuvalt sellest, kas see number ületab 50, 100, 500 või kasvõi 10000, võime võtta vastu otsuse inimene kas süüdi või õigeks mõista. Naljakal kombel on anglo-ameerika kohtusüsteemis lubatud kohtule esitada tõepärasuhteid, näiteks DNA-tõendite puhul, aga mitte tõepärasuhteid omavahel läbi korrutada. Seda ilmselt seepärast, et peale sellist tehet kaoks vajadus vandemeeste järele ning kohtuotsus muutuks algoritmiliseks. Kohtuniku roll oleks siis otsustada, millist tõendusmaterjali milliste tõepäranumbritega arvutusse sisse panna ja millist mitte. Tõenäosusteooria ütleb paraku, et kui me välistame mõne olulise osa tõendusmaterjalist näiteks sellepärast, et see omandati süüaluse põhiõigusi rikkudes, siis on ratsionaalne kohtuotsus loogiliselt võimatu (totaalse informatsiooni printsiip).

Bayesi teoreemi kasutamine pideva suuruse (näiteks keskväärtuse või standardhälbe) hindamiseks on põhimõtteliselt samasugune, ainult et nüüd on meil lõpmata suur arv hüpoteese (iga teoreetiliselt võimalik parameetri väärtus on siin "hüpotees"), mis tähendab, et vastavalt Bayesi teoreemile on meil vaja ka lõpmata hulka tõepärasid ja lõpmata hulka prioreid. Lõpmata hulk tõepärasid ja prioreid tähendab lihtsalt, et me avaldame need kahe pideva funktsioonina, misjärel saame neist kahest funktsioonist arvutada kolmanda pideva funktsiooni, posteeriori. Posteeriorist saab omakorda arvutada iga mõeldava parameetriväärtuste vahemiku tõenäosuse või usalduspiirid, milles mingi meie poolt etteantud tõenäosusega paikneb parameetri tegelik väärtus (vt ptk 10). Ja posterioorse funktsiooni tipp (mood) vastab kõige tõenäolisemale parameetriväärtusele.

Mida kitsam on posteerior, seda kitsamad tulevad sellest arvutatud usalduspiirid. Seega peaksime püüdma panna oma mudelitesse parameetreid (statistikuid), mille posteeriorid tulevad võimalikult kitsad (vt allpool ptk "ajalooline vahepala" selle kohta, kuidas aritmeetiline keskmine on selline statistik).

Tõenäosuse episteemiline tõlgendus

Tõenäosus P ei ole matemaatiliselt midagi enamat, kui reaalarv, mis rahuldab Kolmogorovi aksioomide poolt seatud tingimusi. Tõenäosuse mõiste, mis rahuldaks teaduse vajadusi, on pigem filosoofiline kui matemaatiline probleem. Tõenäosusteooria õpetab meid tõenäosustega matemaatiliselt ümber käima, kuid ei anna meile seost matemaatiliste tõenäosuste ja päris maailma vahel, ega ei ütle, mida tõenäosus teaduses tähendab. Kaasajal kasutatkse kahte põhilist tõenäosuse tõlgendust, episteemilist (Bayesiaanlikku) ja objektiivset (sageduslikku), millest me siin käsitleme esimest. Sageduslikku tõlgenduse kohta vt lisa 1.

Bayesiaanlik statistika opereerib episteemilise tõenäosusega. See tähendab, et tõenäosus annab numbrilise mõõdu meie ebakindluse määrale mõne hüpoteesi ehk parameetriväärtuse kehtimise kohta. Seega mõõdab tõenäosus meie teadmiste kindlust (või ebakindlust). Näiteks, kui arvutus näitab, et homme on vihma tõenäosus 60%, siis me oleme 60% kindlad, et homme tuleb vihma. Aga hoolimata sellest, mida me vihma kohta usume, homme kas sajab vihma või mitte, ja seega on homse vihma objektiivne tõenäosus meie akna taga 0 või 1 – ja mitte kunagi 0.6.

Tõenäosuse formaalne tõlgendus tuleb otse kihlveokontorist. Kui sa arvutasid, et vihma tõenäosus homme on 60%, siis see tähendab, et sa oled ratsionaalse olendina nõus maksma mitte rohkem kui 60 senti kihlveo eest, mis võidu korral toob sulle sisse 1 EUR – ehk 40 senti kasumit. Seega on "ausa kihlveo shansid" (fair betting odds) sinu jaoks 60:40 ehk 3:2 vihma kasuks, mis tähendab, et sa usud, et nende kihlveoshansidega oled sa enda jaoks tasakaalustanud riski nii võidu kui kaotuse korral ja usud, et pikas perspektiiviis jääd sa nii mängides nulli. Seega, ausa kihlveo shansid a:b annavad episteemilise tõenäosuse valemiga a/(a + b), ja tõenäosusest a saab sansid valemiga b = 1 - a.

Selles mõttes on Bayesi tõenäosus subjektiivne. Kui me teaksime täpselt, mis homme juhtub, siis ei oleks meil selliseid tõenäosusi vaja. Seega, kui te hoolimata kõigest, mida ma selle kohta eelpool kirjutanud olen, ikkagi usute, et teadus suudab tõestada väiteid maailma kohta samamoodi, nagu seda teeb matemaatika formaalsete struktuuride kohta, siis pääsete sellega statistika õppimisest ja kasutamisest. Aga kui te siiski arvutate Bayesi (või sageduslikke) tõenäosusi, siis ei ütle teie tõenäosuse tõlgenduse valik iseenesest mitte midagi selle kohta, kas maailm ise on tõenäosuslik või deterministlik.[25]

Kui mõõdame pidevat suurust, näiteks inimeste pikkusi, siis saame arvutuse tagajärjel tõenäosused kõigi võimalike parameetriväärtuste kohta, ehk igale mõeldavale pikkuse väärtusele. Kuna pideval

Kuna kõik tõenäosuse tõlgendused alluvad samadele aksioomidele, siis peavad kõik valiidsed tõenäosust sisaldavad argumendid olema tõlgitavad erinevate tõlgenduste vahel. Võtame näiteks olukorra, kus hüpotees h deduktiivselt ennustab tõendusmaterjali e-d ja lisaks oleks e väga ebatõenäoline juhul, kui h ei kehti. Paraku, isegi kui katse tulemus on e, ei saa ilma h-i eeltõenäosust P(h) kasutamata ikkagi öelda midagi h-i kehtimise tõenäosuse kohta, sest Bayesi teoreem kehtib kõikide tõenäosuse tõlgenduste korral. Aga just seda viga teevad pahatihiti teadlased, kes kasutavad traditsioonilist sageduslikku statistikat, mis ei sisalda formaalseid meetodeid eeltõenäosuste arvesse võtmiseks.

[25] Kvantmehhaanika kanooniline nn Kopenhaageni tõlgendus, mis vaatleb kvantolekuid reaalses maailmas "päriselt" tõenäosuslikena, kasutab hoopiski tõenäosust, mis ei põhine lausearvutuslikul loogikal, samas kui alternatiivne QBism e kvant-bayesiaanlik suund tõlgendab kvantmehhaanika valemeid läbi episteemiliste tõenäosuste ja muudab sellega kvantmehhaanika suures osas klassikalise tõenäosusteooria laienduseks. Need kaks alternatiivset lähenemist annavad kvantmehhanika matemaatilisele apararatuurile kardinaalselt erinevad füüsikalised tõlgendused.

suurusel on lõpmata hulk võimalikke väärtusi, avaldame me sellised tõenäosused pideva tõenäosusfunktsioonina, e järeljaotusena e posteeriorina. Posteerior näeb sageli välja nagu normaaljaotus ja me võime selle põhjal arvutada, kui suur osa summaarsest tõenäosusest, mis on 1, jääb meid huvitavasse pikkuste vahemikku. Kui näiteks 67% posteeriori pindalast jääb pikkuste vahemikku 178 kuni 180 cm, siis me usume, et 0.67-se (67%-se) tõenäosusega asub tegelik keskmine pikkus kuskil selles vahemikus.

Kõrvalepõige - tõenäosused kui kihlveosansid hipodroomil

USA ja Aasia hipodroomidel kehtestavad kihlveosansid mängijad. Oletame lihtsuse huvides, et hipodroom ei võta kihlvedudelt vaheltkasu. Oletame ka, et 1/6 kihlveorahast pannakse hobusele A ja et A võidab. Sellisel juhul saavad A-le panustajad tagasi kuuekordse kihlveosumma ja kihlveosansid on tahvli peale kirjutatud kui 5:1. Seega saab iga A-le panustatud euro eest tagasi 6 eurot, millest 5 EUR-i on kasum. Mida teeb sellises olukorras mängur, kes usub, et tal on võiduajamise kohta siseinfot, mis ei kajastu teiste mängijate panustes (ja seega avaldatud kihlveosanssides)? Juhul kui ta on oma salainfos kindel, panustab ta kogu oma raha, aga jagab oma kihlveo kõikide hobuste vahel nii, et tema kihlveod igale hobusele on proportsionaalsed vastava hobuse võidu tõenäosusega. Kuna mingi hobune võidab igal juhul, võidab ka mängur kindlasti ühe kihlveo. Seega ei kaota ta kunagi kogu oma raha. Ja mis veelgi parem, senikaua kui salainfo töötab (kas või tõenäosuslikult), on see kiireim viis oma rahakoti sisu kasvatada. [26]

Kuidas aga toimida tegelikus olukorras, kus hipodroom võtab endale kuni veerand kihlveosummast? Selle küsimuse lahendas John Kelly 1956 aastal. Nimelt peaks mängija mängu panema sellise osa oma kogurahast: *edge/odds*, kus *edge* ütleb kui palju oleks su oodatav keskmine võidusumma, kui sa saaksid sama kihlvedu palju kordi üha uuesti mängida ja *odds* on avalikud kihlveoshansid. Kui hobuse A avalikud shansid on 8:1 (odds = 8/1) ja siseinfo kohaselt on võidushanssid 1:3, ehk 1/3, siis peaksid panema *(1/3)/8 = 4%* oma rahast A-le [27].

Sellisel viisil mängides on pikas plaanis maksimeeritud varanduse protsentuaalne kasv. Ja varandus ei lange kunagi nulli (kuigi ta võib kahaneda ükskõik kui pisikeseks). Seega on tegu maksimaalse riskitasemega strateegiaga, mis küll maksimeerib pikas perspektiivis kasumi, aga realistlikus ajaaknas võib viia pisarateni. Maksimaalne kasum protsentides on siin võrdne siseinfo vooga (bittides ajaühikus), mis muudab selle tõenäosusteooria rakenduse ühtlasi Shannoni informatsiooniteooria rakenduseks ja seob selle entroopiaga.

[26] Kui sellises mängus panustada igale hobusele vastavalt ametlikele kihlveosanssidele, mis vastavad kõigi mängurite agregeeritud eeltõenäosusele, jääb mängija pikas plaanis nulli (sest kogu raha jagatakse kõigi mängijate vahel vastavalt neile kihlveosuhetele).

[27] Kui siseinfo puudub, siis edge = 0 ja *edge/odds = 0*. Kui *edge = odds*, siis on tegu kindlalt fikseeritud tulemusega. Sellisel juhul annab 8:1 shansid kindla peale 8-kordse kasumi ja *odds/edge = 1/1 = 1*, mis tähendab, et sa paned mängu kogu oma raha.

1738 kirjutas Daniel Bernoulli artikli, kus ta kirjeldab järgmist õnnemängu: Peeter viskab kulli ja kirja senikaua, kuni saab kulli. Ta lubab anda Paulile 1 dukati siis, kui kull tuleb 1. viskel, 2 dukatit, kui see juhtub 2. viskel, 4 dukatit 3. viske korral jne. Kui palju peaks Paul olema nõus ratsionaalse olendina maksma selles mängu osalemise eest? Ehk, mis on Pauli oodatud kasu sellest mängust. Matemaatilise ootuse (expectation) saamiseks tuleb korrutada võidu tõenäosus võidusummaga. Meil on 1/2 tõenäosus võita 1 dukat, 1/2 x 1/2 = 1/4 tõenäosus võita 2 dukatit ja nii edasi lõpmatusse. Pauli ootus võidusummale võrdub kõikide nende liikmete summaga, mis paraku on lõpmata suur. Seega näitaks tõenäosusteooria, nagu oleks Paulist ratsionaalne osta lõpmatult kallis pilet mängule, mis tõenäoliselt toob talle tagasi vaid mõne dukati; ja sama absurdne loogika kehtiks nagu uue bõrsile tuleva aktsia hinna määramisel!

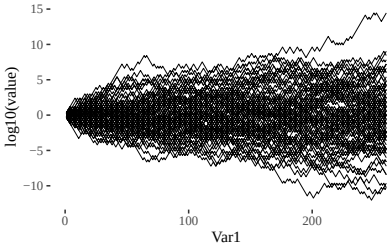
Kuidas lahendada keeruline olukord? Kui meil on võrdsete võidushanssidega mäng, siis eelneva matemaatilise ootuse arvutmine taandub aritmeetilisele keskmisele. Näiteks, kui me viskame münti ja sa paned mängu 10 EUR-i, mille kulli korral kahekordistad ja kirja korral kaotad, siis sinu ootus = (20 + 0)/2 = 10. Seega on 10 EUR-i Bernoulli järgi selle mängu "aus" väärtus. Seega peaks sulle olema ükskõik, kas 10 EUR-i eest mängu astuda või mitte. Aga ometigi ükski mõistlik inimene ei vahetaks 10 EUR-i oma pihus 10-EURse võidulootuse vastu, mis realiseerub 50% tõenäosusega. Bernoulli pakkus lahenduseks riskantsete pakkumiste korral kasutada matemaatilise ootusena aritmeetilise keskmise asemel geomeetrilist keskmist. Geomeetriline keskmine on alati väiksem kui aritmeetiline keskmine, ja kui mõni mängu väljund annab tulemuseks 0-i (kogu raha kaotamise), siis tuleb ka geom keskmine null, mis tähendab, et ratsionaalne inimene väldib täringumänge, vähemalt niikaua kui täringud on ausad. Geomeetrilise keskmise kui matemaatilise ootuse teine tulem on, et ostsus, kas näiteks kindlustust osta, sõltub ostja varandusest. Suhteliselt vaene kaupmees võib kindlustust ostes oma geom keskmist tõsta isegi siis, kui sellesama kindlustuspoliisi müümine on kasulik ka rikkale kindlustusfirmale, mis samuti tõstab oma geomeetrilist keskmist.

Selgub, et Kelly poolt Bernoullist sõltumata välja töötatud edge/odds valemi võib ümber sõnastada: paiguta oma raha viisil, mis maksimeerib kõikide võimalike tulemuste geomeetrilise keskmise. Seda reeglit, mis ühtlasi maksimeerib ka oodatud tulu mediaani, kutsutakse Kelly kriteeriumiks. Kui tulemused ei ole võrdselt tõenäolised, siis tuleb need läbi korrutada vastavate tõenäosustega. Selline tõenäosustega kaalumine on omakorda ekvivalentne oodatud rikkuse logaritmi maksimeerimisega. Siin tuleb välja oluline point: mängija,kes paneb iga kord mängu sama summa, peaks oma ootuse määramisel

lähtuma aritmeetilisest keskmisest, samas kui mängija, kes paneb mängu (taasinvesteerib) mingi kindla protsendi oma varandusest, arvestab geomeetrilise keskmisega.

Eriti karmi mängu leiutas 1960-ndate lõpus Claude Shannon (sama mees, kes lõi entroopial põhineva informatsiooniteooria): Oletame, et teil on võimalus investeerida aktsiasse, mille hind kõigub juhuslikult nii, et hinnamuutuse määrab mündivise. Kui tuleb kiri, siis aktsia hind tõuseb 2 korda, kui tuleb kull, siis hind langeb 2 korda. See tähendab, et teie investeering annab teile kahekordse tulu, kui teil õnnestub aktsiahinna muutuse suund ära arvata (mida te saate teha 50% tõenäosusega), ja vastasel juhul kotate poole oma investeeringust. Selles mängus puuduvad tehingukulud ja te võite iga sammuga oma raha välja võtta või juurde panna, niipalju kui soovite. Te astute mängu 100 EUR-iga ja hiljem saate opereerida selle summaga +/- võidud/kaotused.

100 iseseisvat mängu, millest igaüks koosneb 26ost juhuslikust sammust. Tõenäosus, et teie mäng lõpetab madalamal, kui see alustab, on 50%, tõenäosus, et algsest panusest (1 ühik, ehk log-skaalal 0) jääb alles alla kümnendiku on 10%, jne. NB! log-skaalas normaalsed deviatsioonid, tähendavad, et mängude rahalised väärtused on lognormaaljaotusega, mille matemaatilist ootust kirjeldab geomeetriline keskmine.




```
set.seed(2343)
f <- function(stock) {
    cash <- stock
    a <- rbinom(1, 1, prob = 0.5)
    b <- c(1/2, 2)
    step <- b[a + 1] * stock
    sum(step, cash)/2
}

stock <- 50
for (i in 1:100) {
    stock[i + 1] <- f(stock[i])
}

d <- as.data.frame(stock)
d$index <- 1:nrow(d)
d %>% ggplot(aes(index, log10(stock))) + geom_line() +
ggthemes::theme_tufte()
```

Shannoni strateegia tüüpiline näide: rikkus kasvab eksponentsiaalselt (y-telg on log-skaalas) ja 100 sammuga on lihtne seda 1000-kordseks kasvatada!

Shannon näitas, et ka sellisest mängust on võimalik kasu lõi-

gata, ja mitte vähe. Optimaalne strateegia on oma algse rahasumma jagamine kahte võrdsesse ossa, millest 50 EUR-i eest ostad aktsiaid ja 50 EUR-i säilitad sularahana. Peale igat juhuslikku sammu tuleb teie kogusumma ümber jagada nii, et enne järgmise sammu astumist on sellest pool aktsiates ja pool sularahas. Sellisel kombel müüte oma aktsiat iga kord, kui selle väärtus tõuseb, ja ostate seda juurde iga kord, kui selle väärtus langeb. Kui eelmine joonis kajastab geomeetrilist juhukõndi, siis Shannoni meetod annab sellest palju kiirema kasvu, mis 260 sammu korral võib küündida miljonitesse ühe mängu pandud ühiku kohta. Seega saate niiviisi parima riskile adjusteeritud kasumi. Selle mängu trikk on lihtne – kuna mängu tulemuste geomeetriline keskmine (ehk matemaatiline ootus) on null, siis ootus aritmeetilise keskmisena on positiivne ja meie kasumiootus on seega >0. Et seda mängu võita, tuleb lihtsalt igas sammus kasum välja võtta, et see sularahas kõrvale panna, ja kaotuse korral tuleb juurde investeerida. Seda mängu võiks mängida ka pärisbõrsil, aga ainult siis kui juhuslikult sammuva aktsia volatiilsus on piisavalt suur, et õigustada tehingukulusid. Kui me modifitseerime oma strateegiat nii, et hoiame 1/4 oma varast aktsias ja 3/4 sularahas (ehk pool Kelly kriteeriumit), siis langeb kasumiootus suhteliselt palju vähem (1/4 võrra) kui teie portfelli volatiilsus [28], ja kui me eeldame, et aktsia hind ajas tõuseb, siis vastavalt selle tõusu-usu suurusele paneme mängu ka suurema osakaalu oma rahast.

[28] Kui tavapärase Kelly strateegiaga eksisteerib 1/3 tõenäosus, et te kaotate vähemalt poole oma rahast enne, kui te suudate selle kahekordistada, siis nüüd on see tõenäosus “vaid” 1/9.

Tõenäosusteooriast tulenevad statistika põhiprintsiibid

1. statistilise analüüsi kvaliteet sõltub mudeli eeldustest & struktuurist. Kuna maailm ei koosne matemaatikast, teevad matemaatilised mudelid alati eeldusi maailma kohta, mis ei ole päris tõesed ja mida ei saa tingimata empiiriliselt kontrollida. Mündiviske näites eeldasime, et mündivisked olid üksteisest sõltumatud. Kui me sellest eeldusest loobume, läheb meie mudel keerulisemaks, sest me peame mudelisse lisama teavet visetevahelise korrelatsiooni kohta. Aga see keerulisem mudel toob sisse uued eeldused. Üldiselt peaks mudeli struktuur kajastama katse struktuuri, mis kaasaegses statistikas tähendab sageli hierarhilisi mudeleid.

2. Ebakindluse määr statistilise hinnangule andmete keskväärtusele kahaneb võrdeliselt ruutjuurega andmete hulgast. Kui kahe mündiviske asemel teeksime kakskümmend, siis saaksime samade eelduste põhjal teha oluliselt väiksema ebakindluse määraga järeldusi mündi aususe kohta. Ruutjuure seos tähendab, et mida suuremaks valim läheb, seda vähem tõstab valimi suurendamine n andmepunkti võrra hinnangu täpsust. Siit tuleneb ka soovitus

pigem tõsta katset disainides efekti suurust kui valimi suurust – juhul kui see on praktiliselt teostatav, võib tegu olla odavama lahendusega.

3. statistilise analüüsi kvaliteet sõltub andmete kvaliteedist. Kui münt on aus, aga me viskame seda ebaausalt, siis, mida rohkem arv kordi me seda teeme, seda tugevamalt usub teadusüldsus millessegi, mis pole tõsi.

4. statistilise analüüsi kvaliteet sõltub taustateadmiste kvaliteedist. Napid taustateadmised ei võimalda parandada andmete põhjal tehtud järeldusi juhul, kui andmed mingil põhjusel ei vasta tegelikkusele. Adekvaatsete taustateadmiste lisamine mudelisse aitab vältida mudelite üle-fittimist.

5. Järeldused ühe hüpoteesi kohta mõjutavad järeldusi ka kõikide alternatiivsete hüpoteeside kohta. Relevantsete hüpoteeside eiramine viib ekslikele järeldustele kõigi teiste hüpoteeside kohta. Me ei saa põhimõtteliselt rääkida tõendusmaterjali tugevusest ühe hüpoteesi kontekstis – tõendusmaterjal on suhteline ja selle tugevust mõõdab tõepärade suhe $P(andmed \mid H_1)/P(andmed \mid H_2)$.

Tõenäosusjaotuste kirjeldamine

- Muutuja on iga omadus või deskriptor, mis võib omada rohkem kui üht väärtust.
- Pideval muutujal on lõpmatu palju võimalikke erinevaid väärtusi, diskreetsel e kategoorilisel muutujal on neid lõplik hulk.
- Muutuja X tõenäosusjaotus on tõenäosuste hulk, mis omistatakse kõikidele võimalikele X-i väärtustele.
- need tõenäosused ei tohi olla negatiivsed ja nende summa peab olema 1.

Pideva muutuja korral on tulemuseks pidev tõenäosuste jaotus ehk tihedusfunktsioon f. Kui joonistame f-i 2D koordinaatsüsteemis, siis tõenäosus, et muutuja X väärtus jääb a ja b vahele, võrdub tihedusfunktsiooni pindalaga, mis jääb a ja b vahele. Kogu tihedusfunktsiooni pindala on 1. Ka mitmele muutujale vastab tõenäosusjaotus - joint probability distribution - mis omistab tõenäosuse igale mõeldavale muutujate väärtuse kombinatsioonile ja mis samuti summeerub ühele.

Oodatud väärtus (expected value; E(X)) aitab meil muutujat summeerida ühe arvuga. E(X) kujutab endast muutuja paljukordsete sõltumatute mõõtmiste keskväärtust. E(X) saamiseks korrutame

muutuja iga võimaliku väärtuse selle väärtuse tõenäosusega ja liidame saadud korrutised.

$$E(X) = \sum xP(X = x)$$

Näiteks kui täringul on kõikide tahkudel võrdne esinemistõenäous, siis E(X) = 3.5.

```
1/6 * 1 + 1/6 * 2 + 1/6 * 3 + 1/6 * 4 + 1/6 * 5 + 1/6 * 6
```

```
## [1] 3.5
```

See on arvutuslikult sama, mis aritmeetiline keskmine:

```
(1 + 2 + 3 + 4 + 5 + 6)/6
```

```
## [1] 3.5
```

Kui aga täring on kallutatud ja "6" saamise tõenäosus on $1/3$, siis eeldusel, et teistel numbritel on võrdne tõenäosus, siis E(X) = 4

```
(2/15) * 1 + (2/15) * 2 + (2/15) * 3 + (2/15) * 4 + (2/15) * 5 + (1/3) * 6
```

```
## [1] 4
```

Pane tähele, et tõenäosused summeeruvad ühele, st mingi täringu külg tuleb igal juhul ülesse.

```
(2/15) * 5 + (1/3)
```

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

Kui meil on vaja arvutada Y-i oodatud väärtus tingimusel, et X = x, siis korrutame Y iga võimaliku väärtuse y tõenäosusega $P(X = x|Y = y)$ ja seejärel summeerime need korrutised

$$E(Y|X = x) = \sum yP(Y = y|X = x)$$

Lisaks:

- konstandi oodatud väärtus võrdub selle konstandiga $E(c) = c$
- konstandi võib tuua operaatori ette $E(cX) = cE(X)$
- summa oodatud väärtus on oodatud väärtuste summa $E(X + Y) = E(X) + E(Y)$
- sõltumatute juhuslike suuruste korrutise oodatud väärtus on nende oodatud väärtuste korrutis $E(XY) = E(X)E(Y)$.

E(X) on vaid üks võimalus paljudest arvata midagi kõige "parema" või "õigema" X-i väärtuse kohta. See lähenemine minimeerib oodatud ruutvea ja töötab eeskätt sümmeetriliste X-i andmejaotuste korral. Seega on E(X) e aritmeetiline keskmine piirjuht lineaarsest regressioonist vähimruutude meetodil (piirjuht, kus regressioonimudelis puuduvad prediktorid, ehk intercept-only regressioonimudel).

Seevastu mediaan on vähemtundlik outlieritele, töötab paremini mitte-sümmeetriliste andmejaotuste korral ja minimeerib oodatud absoluutvea.

Andmete varieeruvust kirjeldab dispersioon (*variance*) ehk Var, mis on defineeritud kui muutuja keskmine ruuterinevus oma keskväärtusest

$$Var(X) = E((X - \mu)^2)$$

Standardhälve on ruutjuur dispersioonist, mis on sama ühikuga, kui algsed andmed.

- Kui keskväärtus mu on valitud nii, et see ruutude summa ehk $Var(X)$ on minimaalne, siis on see keskväärtus ühtlasi oodatud väärtus E, ehk teisisõnu on see jaotuse *centraltendency*. Seega on aritmeetiline keskmine vähimruutude meetodi erijuht.
- Kui me aga valime mu väärtuse selliselt, et minimeeritud oleks erinevuse $(X - \mu)$ absoluutväärtus, mitte selle ruut, siis on mu sama, mis mediaan.

Veel üks viis jaotuse summeerimiseks on selle *Highest Density Interval* ehk HDI. Näiteks 90% HDI hõlmab endast 90% jaotuse pindalast sellisel viisil, et ükski jaotuse punkt, mis jääb sellest intervallist välja, ei näita suuremat tõenäosust (st ei ole kõrgemal) ühegist punktist, mis on selle intervalli sees. Kui jaoutsel on mitu küüru, siis võib HDI koosneda mitmest lahus seisvast osast. Seega erineb 90% HDI jaotuse 5% ja 95% kvantiilide poolt piiritletud intervallist, mis jätavad jaotuse mõlemast servast välja 5% jaotuse pindalast. Sümmeetriliste jaotuste nagu normaaljaotus korral need intervallid kattuvad.

Kahe muutuja (X ja Y) kovariatsioon on defineeritud kui

$$\sigma_{XY} = E[(X - E(X))E(Y - E(Y))]$$

ja see määrab, mil määral Y ja X lineaarselt koos-varieeruvad. Mittelineaarset koosvarieeruvust ei saa tüüpiliselt ühe numbriga iseloomustada – seda iseloomustab täielik konditsionaalne tõenäosus.

Normaliseeritud kovariatsioon annab korrelatsioonikoefitsiendi, mis jääb -1 ja 1 vahele:

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

Lognormaalse jaotuse korral töötab järgmine skeem:

1) logaritmi andmed
2) arvuta aritmeetiline keskmine ja standardhälve log skaalas
3) anti-logaritm aritmeetilisest keskmisest annab geomeetrilise keskmise algses skaalas (see tuleb alati väiksem kui aritmeetiline keskmine, aga iseloomustab paremini oodatud väärtust) ja anti-logaritm standardhälbest annab nn multiplikatiivse SD, mis tuleb geom keskmisega läbi jagada ja läbi korrutada. Niiviisi saadud SD piirid defineerivad 68 protsenti lognormaaljaotusest, samamoodi nagu tavaline SD defineerib 68 protsenti normaaljaotusest.

Korrelatsioonikordaja on sümmeetriline: $\rho_{XY} = \rho_{YX}$.

Regressioonimudeli y = a + bx tõus b avaldub kui

$$b = R_{YX} = \frac{\sigma_{XY}}{\sigma_Y^2}$$

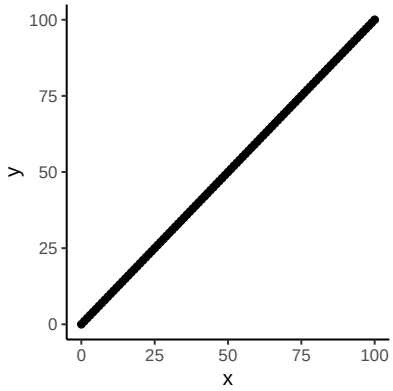
$R_{YX} = R_{XY}$ ainult siis, kui $Var(X) = Var(Y)$, mis täendab, et tavaliselt on Y ja X vaheline tõus erinev X-i ja Y-i vahelisest tõusust. Teisisõnu ei ole regressioon, erinevalt korrelatsioonist, sümmeetriline.

Sissejuhatus regressiooni: lineaarsed mudelid

Sirge võrrand

Oletame, et me mõõtsime N inimese pikkuse cm-s ja me tegime seda 2 korda ja meie mõõtmised on absoluutselt täpsed. Kui me tähistame iga inimese 1. mõõtmise x-ga ja 2. mõõtmise y-ga, siis Y = X ja me saame X-i teades ennustada Y täpse väärtuse (ja vastupidi). Seega, kui me teame X-i väärtust, ei anna Y-i väärtus meile mingit lisainformatsiooni (ja vastupidi).

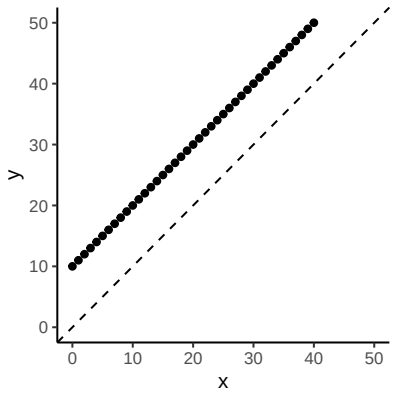
```
library(tidyverse)
x <- 0:100
y <- x

ggplot(data = NULL, aes(x = x, y = y)) +
geom_point() +
theme_classic()
```



Y=X on sirge võrrand, mis ütleb, et sirge lõikepunkt y-teljega (ehk intercept ehk a) on null (y=0) ja et sirge tõus on 1, st X-i väärtuse muutus 1 ühiku võrra toob kaasa ka Y väärtuse samasuunalise 1 ühikulise muutuse

Mis juhtub, kui teine mõõtmine on süstemaatilises nihkes alati täpselt 10 ühiku võrra? Seda kirjeldab sirge võrrand Y = 10 + X:

```
x <- 0:100
a <- 10
y <- a + x

ggplot(data = NULL, aes(x = x, y = y)) +
geom_point() +
geom_abline(linetype = 2) +
ylim(0, 50) + xlim(0, 50) +
theme_classic()
```

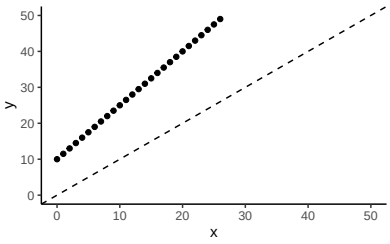


Mis juhtub, kui Y mõõtmine on küll algselt 10 ühiku võrra nihkes, aga selle nihke suurus kasvab võrdeliselt X-i väärtuse kasvuga,

näiteks 0.5 ühiku võrra iga X-i ühiku kohta. Seda olukorda kirjeldab võrrand Y = 10 + 1.5X. Siin on sirge tõus mitte 1, nagu kahes eelmises näites, vaid 1.5. Kui tähistame tõusu b-ga, siis

```
x <- 0:100
a <- 10
b <- 1.5
y <- a + b * x
```

```
ggplot(data = NULL, aes(x = x, y = y)) +
geom_point() +
geom_abline(linetype = 2) +
ylim(0, 50) + xlim(0, 50) +
theme_classic()
```



Muutes a ja b väärtusi, saame sirget 2D ruumis suvaliselt paigutada. Kui tahame lineaarset mudelit 3D ruumis, tuleb võrrandisse juurde veel üks liige: $Y = a + b_1X_1 + b_2X_2$. Kui 2D ruumis positsioneerime 2 parameetri – a ja b – abil sirge, siis 3D mudelis on 3 parameetrit (a, b_1 ja b_2), mille väärtused määravad üheselt tasapinna.

Siiamaani oleme ennustanud täpset Y-i väärtust, kasutades selleks täpset X-i väärtust. Nüüd ehitame mudeli, kus Y-i väärtus ei sõltu X-i väärtusest mitte täpselt (deterministlikult), vaid umbkaudu (stohhastiliselt). See vastab olukorrale, kus X-i mõõtmised on täpsed, aga Y-mõõtmised toimuvad juhusliku mõõtevea tingimustes, kusjuures see viga on sümmetriliselt jaotunud õige väärtuse ümber. Ehk teisisõnu, iga individuaalne mõõtmistulemus on kõige suurema tõenäosusega õige Y väärtus ja tema nihe õigest väärusest, ehk residuaal, toimub võrdse tõenäosusega mõlemas suunas ja selle nihke tõenäosus langeb järsult sedamõõda, kuidas kasvab nihke suurus. Ehk teisisõnu, me mudeldame mõõtmistulemusi ja nende nihet tõesest Y väärtusest normaaljaotusega.

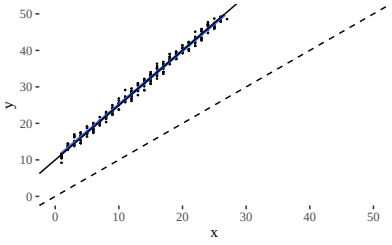
$$Y = a + bX + \epsilon$$

on selline mudel, kus ϵ on veakomponent. Veakomponent on oma olemuselt varieeruvuse mudel (jah, see tähendab: mudel mudelis), mis tavalise lineaarse regressiooni korral on normaaljaotuse kujul ja mis sisaldab endas seda osa Y-i varieeruvusest, mida ei seleta mudelisse pandud regressorid $X_1 \ldots X_n$. Seega koosneb veakomponent nii mõõtmisveast kui bioloogilisest varieeruvusest, mis on kõike muud kui "viga". Hiljem õpime seda veakomponenti mitmetasemelisetes mudelites osadeks jaotama.

Seda mudeldame nii:

```
x <- 1:30
a <- 10
```

```
b <- 1.5

# igale diskreetsele x-i väärtusele vahemikus 0-20

# vastab nüüd 10 stohhastilist y väärtust
y <- a + b * x + replicate(10, rnorm(n = x, mean = 0, sd = 1)) %>%
as_tibble()

# ekvivalentselt y <- replicate(10, rnorm(10, mean = a + b * x, sd=1))

df1 <- tibble(x = x) %>%
bind_cols(y) %>%
pivot_longer(-x) %>%
rename(y = value)

ggplot(data = df1, aes(x, y)) +
geom_point(size = 0.2) +
geom_abline(linetype = 2) +
geom_smooth(method = lm, se = FALSE) +
geom_abline(aes(intercept = 10, slope = 1.5)) +
ylim(0, 50) + xlim(0, 50) +
ggthemes::theme_tufte()
```



Must pidevjoon on meie poolt defineeritud nn protsessimudel, mille intercept = 10 ja tõus = 1.5 ja sinine joon on simuleeritud andmete põhjal fititud joon. Selle interpcept ja tõus on väga lähedal "tegelikule".

```
coef(lm(data = df1, y ~ x))
```

```
## (Intercept)           x
##   10.127292    1.497809
```

Sellega oleme ehitanud lineaarse regressioonimudeli, mis ristub y teljega 10 juures (intercept ehk alfa ehk a = 10) ja mille tõus ehk beta ehk b = 1.5 ja juhuslik viga on normaaljaotusega, mille standardhälve on 1. Sellised mudelid on regressiooni tööhobused – ainult, et tavapäraselt me ei tea a, b, ja veakomponendi sd väärtusi ja peame neid andmete põhjal ennustama. Seda kutsutakse mudeli fittimiseks. Sinist joont joonisel kutsume mudeli "parimaks fitiks" ja see annab meile ennustuse Y-i keskväärtusele igal X-i väärtusel. Statistilises praktikas on meil vaja määrata selle sirge asukoht ruumis fittides mudeli koefitsiendid ja lisaks arvutada seda hinnangut ümbritsev ebakindluse määr (ehk usaldusintervallid mudeli koefitsientidele, millest omakorda saab usaldusväärse piirkonna sirge paiknemisele ruumis).

Sellise mudeli korral on osa informatsiooni Y-i väärtuse kohta olemas X-i väärtuses, ja osa ei ole. See puuduv osa on mudeldatud veakomponendis.

Regressioonanalüüsi abil ei analüüsi me tavaliselt mitte mõõtmine - kordusmõõtmine olukorda, vaid olukorda, kus samadel mõõteobjektidel (olgu selleks inimindiviidid või ensüümi preparatsioonid) mõõdetakse mitut erinevat tunnust (näiteks pikkust ja kaalu). Me võime näiteks küsida, kui hästi me suudame ennustada inimese keskmist kaalu, teades tema pikkust. Sellisel juhul on regressioonimudel $kaal = a + b \times pikkus + \epsilon$.

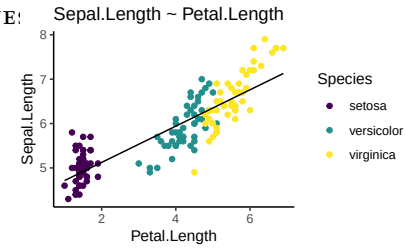
Lineaarse regressioonimudeli matemaatilisest struktuurist aru saamine võimaldab tuua välja mõned seda tüüpi mudelite eeldused.

- me ennustame x-i täpse väärtuse pealt y-i keskmist väärtust (mitte y täpset väärtust)
- kogu varieeruvus on y-i suunaline. Me eeldame, et me teame x-i täpset väärtust. Selle eelduse rikkumine viib mudeli tõusu alahindamisele. Seega on regressioon ebasümmeetriline. y-i ennustamine x-i põhjal ei ole sama, mis x-i ennustamine y-i põhjal.
- me eeldame, et y-i varieeruvus on sama kõikidel x-i väärtustel
- me eeldame, et y-i varieeruvus on normaaljaotusega.
- me eeldame, et y ja x vahelist seost saab kirjeldada sirgega – lineaarsuse eeldus.

Need eeldused kehtivad lihtsa vähimruutude meetodiga mudeli fittimisel. Bayesiaanlik mudeldamine seevastu võimaldab lihtsa vaevaga muuta viimaseid kolme eeldust. Me võime mudeldada erinevaid y-i varieeruvusi erinevatel x-i väärtustel. Me võime kasutada ükskõik millist varieeruvusmudelit normaaljaotuse asemel, olgu selleks pidevad jaotused nagu eksponentsiaalne jaotus või diskreetsed jaotused nagu binoomjaotus. Me võime muuta mudeli mittelineaarseks muutes mudeli baasosa struktuuri või muutes veamudeli jaotusfunktsiooni. Seda kõike õpime allpool.

Kahe parameetriga sirge mudeli fittimine kahedimensiooniliste andmetega käib R-s niimoodi (kasutame R-i “iris” andmesetti):

```
m <- lm(Sepal.Length ~ Petal.Length, data = iris)

augment(m, iris) %>% ggplot(aes(Petal.Length, Sepal.Length, color = Species)) +
geom_point() +
geom_line(aes(y = .fitted), color = 1) +
labs(title = "Sepal.Length ~ Petal.Length") +
scale_color_viridis(discrete = TRUE) +
theme_classic()
```



Mudeli fittimine tähendab siin lihtsalt, et sirge on 2D ruumi asetatud nii, et see oleks võimalikult lähedal kõikidele punktidele.

Mudeli koefitsientide väärtused saame kasutades funktsiooni coef():

```
coef(m)
```

```
##  (Intercept) Petal.Length
##    4.3066034    0.4089223
```

```
coef(m)[1]
```

```
## (Intercept)
##    4.306603
```

```
coef(m)["Petal.Length"]
```

```
## Petal.Length
##    0.4089223
```

Siin a = (Intercept) ja b = Petal.Length ehk 0.41.

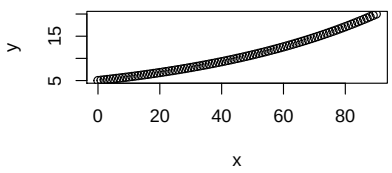
Kõrvalepõige: sirge võrrandist eksponentsiaalse kasvu ja kahanemise võrrandini

kui y = a + bx defineerib sirge, siis $y = Ae^{bx}$, kus $A = e^a$, annab eksponentsiaalse kasvu juhul, kui b>0 ja eksponentsiaalse kahanemise kui b<0. A annab y-i väärtuse juhul kui $x = 0$ ja b annab y-i kasvu või kahanemise kiiruse. Oletame, et meil on kultuur, kus on katse nullpunktis 5 bakterirakku, mis kahekordistuvad 45 minuti tagant. siis $y = 5 \times 2^{(x/45)} = 5 \times e^{(log(2)/45)x}$, kus $e^{(log(2)/45)x}$ on kasvukiirus kordades. Kui x=100 min, siis selle aja jooksul suureneb rakkude arv 4.7 korda.

```
2^(100/45)
```

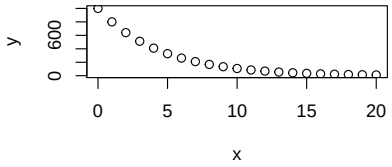
```
## [1] 4.666116
```

```
x <- 0:90
y <- 5 * 2^((x/45))
plot(y ~ x)
```



Nüüd mängime, et meil on katseklaasis 1000 valgumolekuli, millest iga tunniga lagundatakse 20%. Siis $y = 1000 \times (1 - 0.2)^x = 1000 \times e^{log(0.8)x}$.

```
x <- 0:20
y <- 1000 * (1 - 0.2)^x
plot(y ~ x)
```



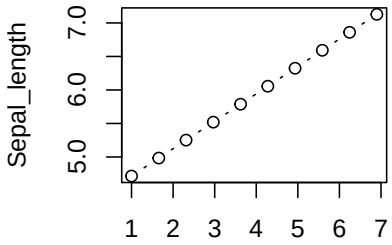
log-log suhe on määratud valemiga $log(y) = a + b \times log(x)$, mis annab nn power law $y = Ax^b$, kus $A = e^a$. A on y-i väärtus kui x=1.

Ennustus lineaarsest mudelist

Anname x-le rea väärtusi, et ennustada y keskmisi väärtusi nendel x-i väärtustel. Siin me ennustame y (Sepal_length) keskväärtusi erinevatel x-i (Petal_length) väärtustel, mitte individuaalseid Sepal_length väärtusi. Me kasutame selleks deterministlikku mudelit kujul $Sepal\ length = a + b \times Petal\ length$. Hiljem õpime ka bayesiaanlike meetoditega individuaalseid Sepal_length-e ennustama.

Järgnev kood on sisuliselt sama, millega me üle-eelmisel plotil joonistasime mudeli $y = a + bx$. Me fikseerime mudeli koefitsiendid fititud irise mudeli omadega ja anname Petal_length muutujale 10 erinevat väärtust originaalses mõõtmisvahemikus. Aga sama hästi võiksime ekstrapoleerida ja küsida, mis on oodatav Sepal_length, kui Petal_length on 100 cm. Sellele küsimusele on ebareaalne vastus, aga mudel ei tea seda.

```
## Genereerime uued andmed Petal.Length vahemikus
Petal_length <- seq(min(iris$Petal.Length), max(iris$Petal.Length), length.out = 10)
## Võtame mudeli koefitsendid
a <- coef(m)[1]
b <- coef(m)[2]
## genereerime Sepal_length väärtused
Sepal_length <- a + b * Petal_length

plot(Sepal_length ~ Petal_length, type = "b")
```



Mudelist saab kahte tüüpi ennustusi: (1) saame ennustada Y keskmist väärtust X-i konkreetsel väärtusel ja (2) saame ennustada individuaalseid Y väärtusi X-i konkreetsel väärtusel.

Neli mõistet

Mudelis $y = a + bx$ on x ja y muutujad, ning a ja b on parameetrid. Muutujate väärtused fikseeritakse andmetega, parameetrid fititakse andmete põhjal. Fititud mudel valib kõikide võimalike seda tüüpi mudelite hulgast välja täpselt ühe unikaalse mudeli ja ennustab igale x-i väärtusele vastava kõige tõenäolisema y väärtuse (y keskväärtuse sellel x-i väärtusel).

- Y – mida me ennustame (*dependent variable, predicted variable*).
- X – mille põhjal me ennustame (*independent variable, predictor*).

- Muutuja (variable) — iga asi, mida me valimis mõõdame (X ja Y on kaks muutujat). Muutujal on sama palju fikseeritud väärtusi kui meil on selle muutuja kohta mõõtmisandmeid.
- Parameeter (parameter) — mudeli koefitsient, millele võib omistada suvalisi väärtusi. Parameetreid tuunides fitime mudeli võimalikult hästi sobituma andmetega.

Mudel on matemaatilise formalism, mis püüab kirjeldada füüsikalist protsessi. Statistilise mudeli struktuuris on komponent, mis kirjeldab ideaalseid ennustusi (nn protsessi mudel) ja eraldi veakomponent (ehk veamudel), mis kirjeldab looduse varieeruvust nende ideaalsete ennustuste ümber. Mudeli koostisosad on (i) muutuja, mille väärtusi ennustatakse, (ii), muutuja(d), mille väärtuste põhjal ennustatakse, (iii) parameetrid, mille väärtused fititakse ii põhjal ja (iv) konstandid.

Mudeli fittimine

Mudelid sisaldavad nii (1) matemaatilisi struktuure, mis määravad mudeli tüübi, kui (2) parameetreid, mida saab andmete põhjal tuunida, niiviisi täpsustades mudeli kuju ehk paiknemist matemaatlises ruumis. Näiteks võrrand $y = a + bx$ määrab mudeli, kus $y = x$ on see struktuur, mis tagab, et mudeli tüüp on sirge, ning a ja b on parameetrid, mis määravad sirge asendi 2D ruumis. Seevastu struktuur $y = x + x^2$ tagab, et mudeli $y = a + b_1x + b_2x^2$ tüüp on parabool, ning parameetrite a, b_1 ja b_2 väärtused määravad selle parabooli täpse kuju. Ja nii edasi.

Mudeli parameetrite tuunimist nimetatakse mudeli fittimiseks. Mudelit fittides on eesmärk saavutada antud tüüpi mudeli maksimaalne sobivus andmetega (kus "andmed" hõlmavad nii valimiandmeid kui taustateadmisi). Sellele tegevusele annab mõtte meie lootus, et mudeli tüüp kajastab mingit looduses toimuvat protsessi, mis meile teaduslikku huvi pakub. Ning, kuigi mudeli fit maksimeeritakse mudeli tüübi kohta, püüab see andmete vaatenurgast vaadatuna olla optimaalne, mitte maksimaalne (vt järgmine peatükk mudeli üle- ja alafittimisest). Kahjuks ei ole selline optimaalsus kuigi hästi matemaatilisse vormi valatav. Siin on tegu pigem teadlase sooviga, mille filosoofiline eeldus on, et meie andmetes on peidus nii andmeid genereeriva loodusliku protsessi üldine olemus (essents), kui juhuslik müra ehk valimiviga, ning et mudeli üldine kuju (sirge, parabool, jms) on sobiv selleks, et neid kahte omavahel lahku ajada.

Lihtsa ühe prediktoriga lineaarse mudeli $y = \alpha + \beta x$ korral tähendab fittimine andmepunktide ja sirge poolt antud mudelien-

nustuse vaheliste kauguste ruutude minimeerimist (kaugusi mõõdetakse paralleelselt y-teljega). Seega minimeeritakse residuaalide ruutude summaks e *Residual Sum of Squares*: $RSS = \sum(y_i - \hat{y}_i)^2$, kus $\hat{y}_i = \alpha + \beta x_i$ on ennustus i-ndale andmepunktile. Me võime i-nda vaatluse avaldada kui $y_i = \alpha + \beta x_i + \epsilon_i$ kus ϵ_i on i-nda vaatluse residuaal või viga (ehk mudeli ennustuse $\hat{y}_i$ ja tegeliku andmepunkti y_i erinevus). Definitsiooni kohaselt võrdub residuaalide keskväärtus nulliga ja nad on normaaljaotusega $\epsilon_i \sim N(0, \sigma)$. Kui $X \sim N(\mu, \sigma)$ ja $Y \sim N(0, \sigma)$, siis $X = \mu + Y$.

Lisaks RSS-ile iseloomustab mudelit ka totaalne ruutude summa ehk $TSS = \sum(Y_i - \bar{Y})^2$ ehk Y-i andmepunktide kauguse ruutude summa Y_i keskväärtusest. TSS = RSS + mudeli poolt seletatud varieeruvus. Mudeli poolt seletatud varieeruvuse osakaal $R^2 = (TSS - RSS)/TSS$ kujutab endast seega lihtsa Y-i keskväärtuse (ehk ilma prediktoriteta mudeli ehk TSS-i) ja X-prediktoritega mudelist arvutatud Y-i keskväärtuse (RSS) ennustuste võrdlus. Mida lähemal on andmepunktid regressioonijoonele, seda väiksem on RSS ja seda suurem tuleb R^2. Kõrge R^2 ei tähenda seega, et sirge tõusunurk β peab olema suur. Kui teie mudeldamise eesmärk on ennustamine, siis on kõrge R^2 teile tähtis. Kui aga mudeldamise eesmärk on testida põhjuslikke seoseid, siis pole R^2 väärtusel teie jaoks suuremat tähtsust. Siis on tähtis pigem see, kuidas X-i muutus viib Y-i muutusele sõltumata teistest võimalikest prediktoritest (siin on eesmärk välistada alternatiivseid seletusi Y-i muutumisele, mis ei eelda X-i põhjuslikku mõju).

Kuna ka juhusliku müra muutuja lisamine mudelile tõstab R^2, on välja tõõtatud statistik, adjusteeritud R^2, mis tõstab uue muutuja lisades R^2 ainult siis, kui uue muutuja mõju R^2-le on suurem kui oleks juhuslikust mürast koosneva muutuja mõju. Adjusteeritud R^2 kasutatakse enamasti selleks, et näha, kas muutuja lisamine regressioonimudelisse parandab mudeli ennustusjõudu.

Regeressioonijoone tõus (β) on korrelatsioon y-i ja x-i väärtuste vahel, mis on läbi korrutatud nende standardhälvete suhtega. Ühtlasi peab regressioonijoon läbima punkti (*mean(x), mean(y)*), mis tähendab, et teades selle punkti asukohta ja β väärtust saame joont pikendades leida selle lõikepunkti y-teljega, mis ongi mudeli intercept $\alpha = \bar{y} - \beta\bar{x}$. Kui meil on mitu x-prediktorit, siis on mudeli koefitsientide fittimine keerulisem.

Bayesiaanlik lähenemine asendab RSS-i minimeerimise andmete tõepära maksimeerimisega, mis on matemaatiliselt sama asi. Tõepära on kõigi y_i andmepunktide tõenäosuste korrutis

$$likelihood = p(y_1|x_1, \theta) \times p(y_2|x_2, \theta) ... \times p(y_n|x_n, \theta)$$

kus θ tähistab vektorit koefitsientidest α, $\beta_1 \ldots \beta_k$. Meie eesmärk

on leida θ väärtused, mis maksimeeriksid eelneva korrutise, ehk maksimeeriksid tõepära. See käib niimoodi: kui andmepunktile y_i vastav mudeli fit on $(\theta^T x_i, \sigma + \epsilon_i)$ ja $\epsilon_i \sim N(0, \sigma)$, siis see tähendab, et $y_i \sim N(\theta^T x_i, \sigma)$. Nii vastab igale y_i väärtusele normaaljaotus, mis kirjeldab selle andmepunkti tõepära. (Matemaatiliselt võime sama hästi maksimeerida ka $log(y_i)$ tõepära, sest seda tehes maksimeerime ühtlasi y_i tõepära.) Seega on meie andmekogumi tõepära individuaalsetele andmepunktidele vastavate normaaljaotuste korrutis.

Hea mudel on

1. Võimalikult lihtsa struktuuriga, mille põhjal on veel võimalik teha järeldusi protsessi kohta, mis genereeris mudeli fittimiseks kasutatud andmeid;

2. Sobitub piisavalt hästi andmetega (eriti uute andmetega, mida ei kasutatud selle mudeli fittimiseks), et olla relevantne andmeid genereeriva protsessi kirjeldus;

3. Genereerib usutavaid simuleeritud andmeid.

Sageli fititakse samade andmetega mitu erinevat tüüpi mudelit ja püütakse otsustada, milline neist vastab kõige paremini eeltoodud tingimustele. Näiteks, kui sirge suudab kaalu järgi pikkust ennustada paremini kui parabool, siis on sirge mudel paremas kooskõlas teadusliku hüpoteesiga, mis annaks mehhanismi protsessile, mille käigus kilode lisandumine viiks laias kaaluvahemikus inimeste pikkuse kasvule ilma, et pikkuse kasvu tempo kaalu tõustes langeks. Samas, see et me oleme oma andmeid fittinud n mudeliga ja otsustanud, et mõned neist on paremad kui teised, ei tähenda, et mõni meie mudelitest oleks hea ka võrdluses tegeliku looduses valitseva olukorraga. Mudelid on pelgalt matemmatilised formalismid, mis võivad, aga kindlasti ei pea kajastama füüsikalist maailma, ja meie mudelitevalik sõltub meile jõukohasest matemaatikast. Siinkohal ei tasu unustada, et matemaatika kirjeldab eelkõige abstraktseid mustreid, mitte otse füüsikalist maailma.

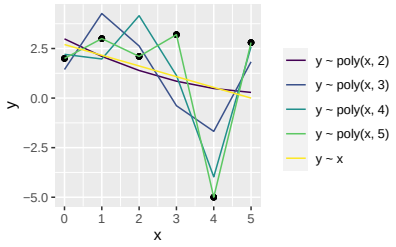
See, et teie andmed sobivad hästi mingi mudeliga, ei tähenda automaatselt, et see fakt oleks teaduslikult huvitav. Mudeli parameetrid on mõtekad mudeli matemaatilise kirjelduse kontekstis, aga mitte tingimata suure maailma põhjusliku seletamise kontekstis. Siiski, kui mudeli matemaatiline struktuur loodi andmeid genereeeriva loodusliku protsessi olemust silmas pidades, võib mudeli koefitsientide uurimisest selguda olulisi tõsiasju suure maailma kohta.

Mudeli fittimine: X ja Y saavad oma väärtused otse andmetest; parameetrid võivad omandada ükskõik millise väärtuse. Fititud mudelist ennustamine: X-le saab omistada ükskõik millise väärtuse; parameetrite väärtused on fikseeritud; Y väärtus arvutatakse mudelist

Üle- ja alafittimine

Osad mudelite tüübid on vähem paindlikud kui teised (parameetreid tuunides on neil vähem liikumisruumi). Kuigi sellised mudelid sobituvad halvemini andmetega, võivad need ikkagi paremini kui mõni paindlikum mudel välja tuua andmete peidetud olemuse. Statistiline mudeldamine eeldab, et me usume, et meie andmetes leidub nii müra (mida mudel võiks ignoreerida), kui signaal (mida mudel püüab tabada). Ilma signaalita süsteemi poleks arusaadavatel põhjustel mõtekas mudeldada ja ilma mürata süsteemi mudel tuleks ilma varieeruvuse (vea) komponendita, ehk deterministlik. Kuna mudeli jaoks näeb müra samamoodi välja kui signaal, on iga mudel kompromiss üle- ja alafittimise vahel. Me lihtsalt loodame, et meie mudel on piisavalt jäik, et mitte liiga palju müra modelleerida ja samas piisavalt paindlik, et piisaval määral signaali tabada.

Üks kõige jäigemaid mudeleid on sirge, mis tähendab, et sirge mudel on suure tõenäosusega alafittitud. Teises äärmuses on polünoomsed mudelid, mis on väga paindlikud, mida on väga raske tõlgendada ja mille puhul esineb suur mudeli ülefittimise oht. Ülefititud mudel järgib nii täpselt valimiandmeid, et sobitub hästi valimis leiduva juhusliku müraga ning seetõttu sobitub halvasti järgmise valimiga samast populatsioonist (igal valimil on oma juhuslik müra). Üldiselt, mida rohkem on mudelis tuunitavaid parameetreid, seda paindlikum on mudel, ja seda kergem on seda valimiandmetega sobitada. Veelgi enam, alati on võimalik konstrueerida mudel, mis sobitub täiuslikult kõikide andmepunktidega (selle mudeli parameetrite arv on n-1). Selline mudel on täpselt sama informatiivne kui andmed, mille põhjal see fititi – ja täiesti kasutu.



Vähimruutude meetodil fititud mudeleid saame võrrelda AIC-i näitaja järgi. AIC - Akaike Informatsiooni Kriteerium - vaatab mudeli sobivust andmetega ja mudeli parameetrite arvu. Väikseim AIC tähitab parimat fitti väikseima parameetrite arvu juures (kompromissi) ja väikseima AIC-ga mudel on eelistatuim mudel. Aga seda ainult võrreldud mudelite hulgas. AIC-i absoluutväärtus ei loe - see on suhteline näitaja.

model_formula	aic
y ~ x	35.04317
y ~ poly(x, 2)	37.00603
y ~ poly(x, 3)	36.05250
y ~ poly(x, 4)	32.46881
y ~ poly(x, 5)	-Inf

AIC näitab, et parim mudel on mod_e4. Aga kas see on ka kõige kasulikum mudel? Mis siis, kui 3-s andmepunkt on andmesisestaja näpuviga?

Ülefittimise vältimiseks kasutavad Bayesi mudelid informatiivseid prioreid, mis välistavad ekstreemsed parameetriväärtused.

Lineaarse regressiooni eeldused

Matemaatilised eeldused, mille kehtimisel annab regressioonimudel mitte-kallutatud hinnangu "tõelistele" populatsiooni parameetritele (regresiooni koefitsientidele ja standardvigadele). Neid eeldusi võib vaadata kui tingimusi, mille kehtimisel on lubatud regressioonimudelist teha kehtivaid järeldusi.

1. Lineaarsus: ennustus Y-muutujale on lineaarne funktsioon prediktoritest $Y = \beta_1 X_1 + \beta_2 X_2 + \beta_n X_n$, ehk ekvivalentselt: kõikidel X-i väärtustel keksmine residuaal = 0. See keskmise residuaali "eeldus" on automaatselt tagatud fittimismeetodiga, mis adjusteerib intercepti sellele vastavaks. See on sama, mis öelda, et lineaarne regressioon mõõdab adekvaatselt ainult lineaarseid seoseid.

2. Normaalsus - residuaalid on normaaljaotusega. NB! Me ei räägi siin mitte Y-muutuja, vaid residuaalide normaaljaotusest. Selle eelduse rikkumine ei mõjuta niivõrd koefitsientide punkthinnanguid, kui usalduspiire. Ja tsentraalne piirteoreem kehtestab selle eelduse pea alati, juhul kui y-muutuja on enam-vähem pidev ja andmepunktide arv piisavalt suur (>15-50). Seega ei tasu enamasti normaalsuse pärast põdeda, välja arvatud siis, kui X on kategooriline muutuja, millel on vähe diskreetseid tasemeid. Residuaalide normaalsus on siiski oluline, kui meie eesmärk on fititud mudelist üksikute andmepunktide ennustamine.

3. Sõltumatus - residuaalide väärtused on iseseisvad ja identse jaotusega (*i.i.d.*), ehk residuaalid ei ole omavahel korreleeritud. Selle eelduse rikkumine mõjutab nii standardvigu kui koefitsientide punktväärtusi.

4. homoskedastilisus ehk konstantne Y-muutuja suunaline varieeruvus kõigil X-i väärtustel. See tähendab ühtlasi residuaalide konstantset varieeruvust. Selle eelduse rikkumine kallutab standardvigu, mitte koefitsientide punktväärtusi.

5. Kui meil on mitmene regressioon, kus osad prediktorid, nn. võtmemuutujad, pakuvad meile otsest huvi ja teised prediktorid on nn. kontrollmuutujad, mille taset me tahame mudeli fiti tõlgendamisel konstantsena hoida, siis võtmemuutujate (aga mitte kontrollmuutujate) väärtused ei tohi olla korreleeritud vealiikmega. See on sama, mis öelda, et vealiikmes ei tohi peituda lisaprediktoreid, mis põhjuslikult mõjutavad Y väärtusi ja on samas

Lineaarsed mudelid on proportsionaalsed: tõstes X-i väärtuse kahekordseks tõuseb oodatav Y-i väärtus alati sama kiirusega (sõltumata X-i numbrilisest väärtusest). Mittelineaarsed mudelid käituvad teistmoodi. Näiteks auto kiiruse sõltuvus kütusekulust on mittelineaarne, sest kihutav auto vajab kiiruse tõstmiseks ühe ühiku võrra rohkem kütust kui normaalkiirusel sõitev auto. 2+ prediktori kombineeritud mõju võib olla lineaarne ka siis, kui ühekaupa mõjud on mitte-lineaarsed. Aga kui prediktorite ühekaupa mõjud on lineaarsed, siis on ka kombineeritud mõju alati lineaare.

korreleeritud meie mudelis oleva võtmemuutujaga. See eeldus on tähtis põhjuslike mõjude uurimisel regressiooni abil, aga mitte ennustamisel ega mudeli fiti jaoks mõjukate prediktorite määramisel. Näide: Y - sissetulek, võtme-X - kooliharidus aastates, kontroll-muutja - IQ on määramata ja läheb otse vealiikmesse. Sellisel juhul mõjutab IQ mudeli koefitsiente ja me võime koolihariduse mõju valesti tõlgendada. Samas, see ei mõjuta mudeli ennustusjõudu.

6. X-muutuja suunal puudub mõõtmisviga ja ebakindlus, mis tähendab, et x-i väärtused on täpselt teada (me ei ennusta neid). See eeldus on samuti tähtis ainult siis, kui püüame anda oma mudelile põhjusliku tõlgenduse. Mudelis $Y = \alpha + \beta_1 X_1 + \beta_2 X_2$ viib mõõtmisviga X_2-s tavaliselt β_2 koefitsiendi nulli suunas ja ühtlasi vähendab $\beta_2 X_2$ liikme mõju β_1 fittimisel [29]. Mõõtmisviga ühes prediktoris võib kallutada ka teiste, ilma mõõtmisveata, prediktorite koefitsientide hinnanguid. Mõõtmisviga on ohtlik võtme-X-muutujate korral, aga mitte niivõrd kontroll-muutujate osas.

[29] Kui on täidetud vähemalt üks kahest eeldusest - X-i mõõtmisviga sõltub X-i väärtusest või X1 mõõtmisviga on korreleeritud X2 muutujaga - siis võib mõõtmisvea mõju X-i koefitsientidele olla mõlemas suunas.

7. Mitme prediktoriga lineaarse regressiooni puhul tuleb sisse veel kollineaarsuse eeldus: me eeldame, et ükski prediktorite paar pole täiuslikult lineaarselt korreleeritud (pole lineaarne funktsioon üksteisest). [30] Täieliku kollineaarsuse korral mudel ei lahendu, aga sellisel juhul on põhjuseks enamasti viga mudeli spetsifitseerimisel. Kui meil on probleem osalise kollinneaarsusega, siis juhtub alati sama asi – mudeli koefitsientide usalduspiirid lähevad absurdselt laiaks. Seega, kui teie mudelis on mõistlikud usalduspiirid, siis ei pea te kollineaarsuse pärast muretsema. Samuti, kollineaarsus ei kahjusta kuidagi teie mudeli ennustusvõimet – see raskendab vaid mudeli põhjusliku tõlgendamist.

[30] Multikollineaarsus tähendab X1 ja X2 omavahelist seotust mudelis, mis on konditsioneeritud teistele muutujatele.

*Meeldetuletus: regressioon hindab prediktorite **keskmist** mõju Y-i väärtusele.*

Keskmine mõju tähendab, et ravim, mis kahjustab pooli patsiente ja teeb teisele poolele sama palju head, omab mudelis null-mõju ja vastavaid koefitsiente. Samuti, kui mudel ütleb, et laste õppeedukust mõjutab kaks korda enam kodune taust kui õpetaja kvaliteet, siis see grupi tasemel hinnang ei tähenda, nagu me saaksime järeldada, et Juhani ja veel tuhandete laste õppeedukus ei sõltuks eelkõige tema õpetajast. Ja peale selle, interventsioonina on palju lihtsam parandada õpetajate kvaliteeti kui lapsevanemate oma. Ja peale selle, õpetajate roll mudelis on samuti keskmistatud, mis tähendab, et kui paljud õpetajad on piisavalt kehvad, et mitte omada suuremat mõju oma õpilastele, siis vähemus tõeliselt häid õpetajaid võivad omada palju suuremat mõju kui kodune taust.

Küll aga saame residuaale vaadates hinnata, millistele patsientidele oli mõju positiivne ja millistele negatiivne; ja millistele oli mõju suurem ja millistele väiksem.

Eeldused praktilise tähtsuse järjekorras:

1. Valiidsus – te mõõdate asju, mis on relevantsed teadusliku küsimuse seisukohast. Näiteks, kui soovite mõõta kolesterooli alandava ravimi mõju, on mõistlik mõõta suremust, mitte pelgalt kolesterooli taset veres.

2. Esinduslikkus – andmed peaksid olema esinduslikud laiema populatsiooni suhtes. Väikesed ja kallutatud valimid ei ole sageli esinduslikud.

3. Lineaarsus ja sellest tulenev mudeli additiivsus. Väga tähtis on, et lineaarse regressiooniga mõõdetavad seosed oleks ka tõesti lineaarsed. Kui lineaarsusega on probleeme, võib aidata prediktorite transformeerimine (log(x) või 1/x) või uute prediktorite mudelisse lisamine. Samuti on võimalik prediktoritena samasse mudelisse panna nii x kui x^2. Näiteks kui me paneme mudelisse nii muutuja *vanus* kui ka *vanus*^{2}, saame modelleerida seost, kus y vanuse kasvades alguses kasvab ja siis kahaneb (aga ka U kujulist seost vanusega). Sellisel juhul võib olla ka mõistlik rekodeerida vanus kategooriliseks muutujaks (näit 4 vanuseklassi), mille tasemeid saab siis ükshaaval vaadata.

4. Residuaalide sõltumatus. Selle eelduse rikkumine viib liiga kitsastele usalduspiiridele.

5. Homoskedastilisus ja residuaalide normaalsus on vähemtähtsad. Log-normaalsete vigadega võiks lineaarsel regresioonil mudeldada log(Y) skaalas (vähimruutude meetodil) või Bayesi regressioonil mittelineaarset lognormaalset tõepäramudelit kasutades (vt. ptk 13).

Lineaarse mudeli laiendused

Baasmudelit kujul $y = \alpha + \beta x + \epsilon$ laiendused

1) rohkem additiivseid (sõltumatuid) prediktoreid ($y = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon$)

2) mitte-lineaarne mudel (näit $log(y) = \alpha + \beta log(x) + \epsilon$)

3) mitte-additiivsed nn interaktsioonimudelid ($y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$)

4) Üldistatud lineaarsed mudelid e GLM, mis näiteks võimaldavad mudeldada diskreetseid y-muutujaid. (vt allpool).

5) segatud mudelid, näiteks zero-inflated Poissoni mudel või hurdle lognormaalne mudel. Need mudeldavad nn segatud andmestikke, kus iga andmepunkt võib olla genereeritud kahe erineva protsessi poolt, millest üks genereerib ainult nulle. Siin fititakse korraga kaks mudelit, üks kummagi protsessi kohta.

6) mitte-parameetrilised mudelid, millel on nii palju koefitsiente, et nende abil saab joonistada ükskõik kui keerulisi x *vs*. y kurve. Näiteks splainid.

7) mitmetasemelised mudelid, kus gruppidesse jagatud koefitsiendid varieeruvad grupiviisiliselt.

7.1) mõõteviga inkorporeerivad mitmetasemelised mudelid (neid mudeleid me selles raamatus ei käsitle)

7.2) puuduvate andmete imputatsioonimudelid, kus y- ja/või x-muutujad sisaldavad segamini nii fikseeritud andmeid kui koefitsiente (puuduvad andmed mudeldadakse mitmetasemelises mudelis nii, et igale puuduvale andmepunktile vastab oma koefitsient, mis fititakse ühes suures regressioonimudelis)

7.3) nn gaussian process regressiooni mitmetasemelised mudelid, kus grupid ei ole mitte diskreetsed vaid pidevad kaugusmõõdud mingis matemaatilises ruumis. (neid mudeleid me selles raamatus ei käsitle)

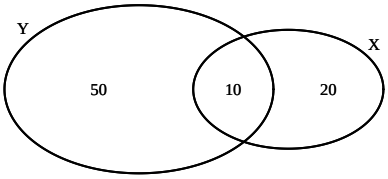
8) kõikvõimalikud keerulised masinõppemeetodid (random forest, boosted gradient trees jne jne), mida iseloomustab läbipaistmatu mudelistruktuur ja mida ei kasutata seega põhjuslike hüpoteeside testimisel. (neid puhtennustuslikke mudeleid me selles raamatus ei käsitle)

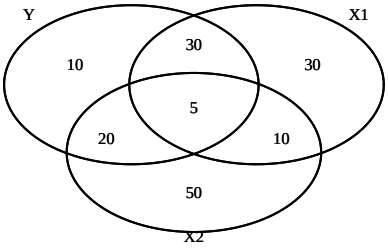
Mitme sõltumatu prediktoriga mudel

Esiteks vaatame mudelit, kus on mitu prediktorit x_1, x_2, … x_n, mis on aditiivse mõjuga. See tähendab, et me liidame nende mõjud, mis omakorda tähendab, et me usume, et x_1 … x_n mõjud y-i väärtusele on üksteisest sõltumatud. Mudel on siis kujul

$$y = a + b_1 x_1 \ + b_2 x_2 \ + \ ... + \ b_n x_n$$

Mitme prediktoriga mudeli iga prediktori tõus (beta koefitsient) ütleb, mitme ühiku võrra ennustab mudel y muutumist juhul kui see prediktor muutub ühe ühiku võrra ja kõik teised prediktorid ei muutu üldse (Yule, 1899).





Veel üks võimalus mitme prediktoriga mudelit mõista on läbi Venni diagrammi. Kõigepealt lihtne mudel $Y = b_0 + b_1 X_1$. Kui Venni

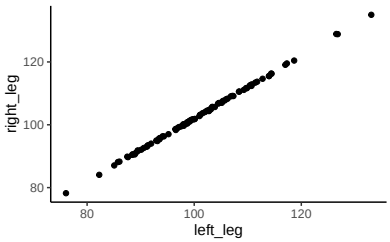
diagrammil ring a tähistab Y-i varieeruvust ja ring b tähistab X_1 varieeruvust, siis nende ühisosa suurus näitab seda osa Y varieeruvust, mis on mudelis seletatav X_1 varieeruvusega – ehk R^2, mis antud juhul on 10/(10 + 50). Ehki R^2 ei anna meile regressioonijoone tõusu, on just Y ja X-i ühine varieeruvus see, ms määrab b_1 koefitsiendi. Järgmine Venni diagramm illustreerib mitmest regressiooni $Y = b_0 + b_1X_1 + b_2X_2$, kus Venni diagrammi ring Y vastab Y varieeruvusele, X_1 vastab X_1 varieeruvusele ja X_2 vastab X_2 varieeruvusele. Nüüd lisandub R2 määramisse X_2 ühisosa Y-ga, mis tõstab R^2 ($R^2 = (20 + 5 + 30)/(20 + 5 + 30 + 10)$). X_1 ja X_2 ühist varieeruvust Y-ga (5), mis küll sisaldub R^2-s, ei kasutata siiski mudeli koefitsientide arvutamises, sest ei ole selge, kumb prediktor selles osas Y mõjutab. Seega kasutatakse mudeli koefitsientide fittimisel vaid osa kättesaadavast informatsioonist ("30" ja "20"), mistõttu laienevad veapiirid sedamõõda, kuidas mudelisse uusi liikmeid lisatakse. Kui $X_1 - X_2 - Y$ ühine varieeruvus läheb väga suureks, siis on meil tegemist kollineaarsusega, millisel juhul on meil alles liiga väha $X_1 - Y$ ja $X_2 - Y$ kovarieeruvust, et mõistlike usalduspiiridega koefitsiente arvutada.

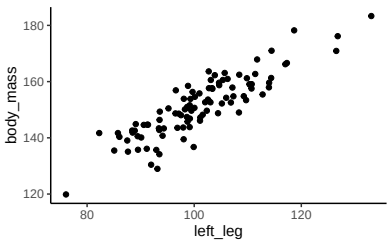
Venni diagramm annab meile veel ühe viisi tõlgendamaks mitmest regressiooni: millist lisaväärtust omab X1 ennustamaks Y-i väärtust peale seda, kui me teame, kuidas X2 mõjutab Y-t (ja ekvivalentselt: millist lisaväärtust omab X2 ennustamaks Y-i väärtust peale seda, kui me teame, kuidas X1 mõjutab Y-t). Näiteks, me teame, et jalapikkuse (X) järgi on võimalik ennustada inimese kaalu (Y). Aga mis juhtub kui fitime mudeli, kus ennustame kaalu vasaku jala pikkuse + parema jala pikkuse järgi. Kuna jalgade pikkused on tugevalt korreleeritud, siis mudelis kaotavad mõlemad jalad eraldi võetuna oma ennustusvõime (mõlema jala tõusukoefitsiendid tulevad laiade usalduspiiridega, mis hõlmavad nulli), kuigi mudeli fit (R^2) ega selle ennustusvõime tervikuna sellest ei muutu. Seda nähtust kutsutakse kollineaarsuseks. Ekstreemsel juhul, kus $X_1 = X_2 = X$ ja valitseb täielik kollineaarsus, taandub regressioonivõrrand $Y = \alpha + \beta_1X_1 + \beta_2X_2$ võrrandile $Y = \alpha + (\beta_1 + \beta_2)X$-le. Siit on näha, et β1 ja β_2 mõjusid X-le ei ole sellisel juhul võimalik lahku ajada. Ehk Bayesi keeles, β_1 ja β_2 posteeriorite summa tuleb sama, mis β posteerior lihtsas võrrandis $Y = \alpha + \beta X$.

Näide kollineaarsusest: me ennustame jalapikkuse järgi kehamassi simuleeritud andmetel

```
left_leg <- rnorm(100, 100, 10)
right_leg <- left_leg + rnorm(100, 2, 0.2)
body_mass <- left_leg + rnorm(100, 50, 5)
ggplot(data = NULL, aes(left_leg, right_leg)) +
```

```
geom_point() + theme_classic()


ggplot(data = NULL, aes(left_leg, body_mass)) +
geom_point() + theme_classic()


summary(lm(body_mass ~ left_leg))

##
## Call:
## lm(formula = body_mass ~ left_leg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7225  -3.4310   0.2962   3.9403  10.7063
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 53.95333    5.43294   9.931   <2e-16 ***
## left_leg     0.96355    0.05357  17.985   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.206 on 98 degrees of freedom
## Multiple R-squared:  0.7675, Adjusted R-squared:  0.7651
## F-statistic: 323.5 on 1 and 98 DF,  p-value: < 2.2e-16

summary(lm(body_mass ~ left_leg + right_leg))

##
## Call:
## lm(formula = body_mass ~ left_leg + right_leg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.7023  -3.3816   0.3567   3.9545   9.9995
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   60.477      8.214   7.363 5.91e-11 ***
## left_leg       3.880      2.756   1.408    0.162
## right_leg     -2.923      2.762  -1.058    0.293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```





```
## Residual standard error: 5.203 on 97 degrees of freedom
## Multiple R-squared:  0.7701, Adjusted R-squared:  0.7654
## F-statistic: 162.5 on 2 and 97 DF,  p-value: < 2.2e-16
```

Nagu näha, kahe mudeli fitid on praktiliselt võrdsed (adj R-squared), aga tõusukoefitsiendid ja nende standardvead (ja seega CI-d) erinevad kardinaalselt. See on kollineaarsus pähklikoores.

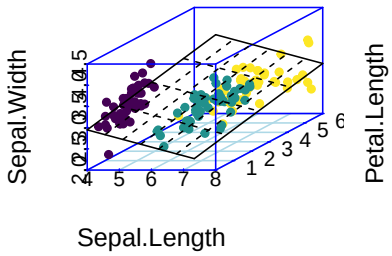
Milliseid muutujaid (regressoreid) peaks üks hea lineaarne mudel sisaldama, milliseid peaks me mudelist välja viskama ja milliseid igal juhul sisse panema? Matemaatiliselt põhjustab regressorite eemaldamine ülejäänud regressorite koefitsientide ebakonsistentsust, välja arvatud siis, kui (i) välja visatud regressorid ei ole korreleeritud sisse jäetud regressoritega või (ii) välja vistatud regressorite koefitsiendid võrduvad nulliga, mis muudab nad ebarelevantseteks. Kuidas sa tead, et kõik vajalikud regressorid on sul üldse olemas (olematuid andmeid ei saa ka mudelisse lisada)? Loomulikult ei teagi, mis tähendab lihtsalt, et mudeldamine on keeruline protsess, nagu teaduski. Pane ka tähele, et koefitsiendi "mitte-oluline" p väärtus ei tähenda iseenesest, et koefitsient tõenäoliselt võrdub nulliga või on nulli lähedal, vaid seda, et meil pole piisavalt andmeid, et vastupidist kinnitada. Koefitsiendi hinnangu usalduspiirid on selles osas palju parem töövahend.

Kui meie andmed on kolmedimensionaalsed (me mõõdame igal mõõteobjektil kolme muutujat) ja me tahame ennnustada ühe muutuja väärtust kahe teise muutuja väärtuste põhjal (meil on kaks prediktorit), siis tuleb meie kolme parameetriga lineaarne regressioonimudel tasapinna kujul. Kui meil on kolme prediktoriga mudel, siis me liigume juba neljamõõtmelisse ruumi.

Regressioonitasand 3D andmetele. Kahe prediktoriga mudel, kus Sepal.Length ja Petal.Length on prediktorid ja Sepal.Width ennustatav muutuja.

Seda mudelit saab kaeda 2D ruumis, kui kollapseerida kolmas mõõde konstandile.

2D-le kollapseeritud graafiline kujutus 3D andmete põhjal fititud mudelist. Vasemal, muutuja Petal.Length on kollapseeritud konstandile. Siin on regressioonijoon hoopis teises kohas, kui lihtsas ühe prediktoriga mudelis (paremal).




```
p2 <- p + geom_abline(intercept = coef(m1)[1], slope = coef(m1)[2]) + labs(title = deparse(formula(m1)))
devtools::source_gist("8b4d6ab6a333ef1cd14e8067c3badbae", filename = "grid_arrange_shared_legend.R")
grid_arrange_shared_legend(p1, p2)
```

Võrreldes mudelite m1 (üks prediktor) ja m2 (kaks prediktorit) Sepal.Length (b_1) koefitsienti on näha, et need erinevad oluliselt.

```
coef(m1)
```

```
##  (Intercept) Sepal.Length
##    3.4189468   -0.0618848
```

```
coef(m2)
```

```
##  (Intercept) Sepal.Length Petal.Length
##    1.0380691    0.5611860   -0.3352667
```

Kumb mudel on siis parem? AIC-i järgi on m2 kõvasti parem kui m1, lisakoefitsendi (Petal.Length) kaasamisel mudelisse paranes oluliselt selle ennustusvõime.

```
AIC(m1, m2)
```

```
##    df       AIC
## m1  3 179.46442
## m2  4  92.11691
```

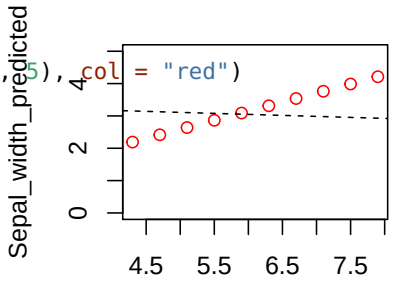
Ennustused sõltumatute prediktoritega mudelist

Siin on idee kasutada fititud mudeli struktuuri ennustamaks y keskmisi väärtusi erinevatel x_1 ja x_2 väärtustel. Kuna mudel on fititud, on parameetrite väärtused fikseeritud.

```
## New sepal length values
Sepal_length <- seq(min(iris$Sepal.Length), max(iris$Sepal.Length), length.out = 10)
## Keep new petal length constant
Petal_length <- mean(iris$Petal.Length)
## Extract model coeficents
a <- coef(m2)["(Intercept)"]
b1 <- coef(m2)["Sepal.Length"]
b2 <- coef(m2)["Petal.Length"]
## Predict new sepal width values
Sepal_width_predicted <- a + b1 * Sepal_length + b2 * Petal_length
```

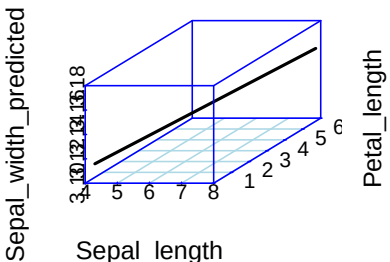
Ennustatud y väärtused erinevatel x_1 väärtustel kui x_2 on konstantne, punane joon. Katkendjoon, ühe prediktoriga mudeli ennustus.

```
plot(Sepal_width_predicted ~ Sepal_length, type = "b", ylim = c(0,5), col = "red")
# Prediction from the single predictor model
abline(m1, lty = "dashed")
```



Nüüd joonistame 3D pildi olukorrast, kus nii x_1 kui x_2 omandavad rea väärtusi. Mudeli ennustus on ikkagi sirge kujul – mis sest, et 3D ruumis.

Kahe prediktoriga mudeli ennustus 3D ruumis.



Interaktsioonimudel

Interaktsioonimudelis sõltub ühe prediktori mõju teise prediktori väärtusest:

$$y = a + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2$$

Ekvivalntne viis interaktsiooni spetsifitseerida on läbi võrrandisüsteemi:

$$y = a + \gamma x_1 + b_2 x_2$$

$$\gamma = b_1 + b_3 x_2$$

Siit on hästi näha, et me teeme kaks lineaarset regressiooni, millest teine modelleerib x_1 muutuja koefitsiendi sõltuvust x_2 muutuja väärtusest.

Samamoodi kehtib ka ümberkirjutus

$$y = a + \gamma x_2 + b_1 x_1$$

$$\gamma = b_2 + b_3 x_1$$

mis tähendab, et ühtlasi modelleerime me ka x_2 koefitsiendi sõltuvust x_1-st. Mudeli koefitsientide tõlgendamise teeb keeruliseks, et gamma tõlgendamisel tuleb arvesse võtta kolm asja - b_2, b_3 ja x_1.

Kaks muutujat võivad interakteeruda sõltumata sellest, kas nad on omavahel korreleeritud või mitte. Interaktsioon ei implitseeri korrelatsiooni, ega vastupidi.

Sageli on nii, et prediktoreid, mille mõju y-le on suur, tasub mudeldada ka interaktsioonimudelis (näiteks suitsetamise mõju vähimudelites kipub olema interaktsiooniga). Interaktsioonimudelis on b_1 koefitsient otse tõlgendatav ainult siis, kui $x_2 = 0$ (ja b_2 ainult siis, kui $x_1 = 0$). Kui interaktsioonimudel fititakse tsentreeritud x-muutujatel, mille keskväärtus on null (või standardiseeritud muutujatel), siis muutub koefitsientide tõlgendamine lihtsamaks:

- b_1 annab y tõusu, kui x_1 tõuseb 1 ühiku võrra ja x_2 on fikseeritud oma keskväärtusel
- b_2 annab y tõusu, kui x_2 tõuseb 1 ühiku võrra ja x_1 on fikseeritud oma keskväärtusel).
- b_3 ütleb, kui palju muutub x_1 mõju y-le, kui x_2 muutub ühe ühiku võrra. Samamoodi, b_3 ütleb, kui palju muutub x_2 mõju y-le, kui x_1 muutub ühe ühiku võrra.

NB! Ärge standardiseerige faktormuutujaid ehk *dummy*-regressoreid kujul 1, 0 - neid on lihtsam tõlgendada algsel kujul 0/1 skaalas.

Interaktsioonimudelite koefitsientide tõlgendamine ei ole eriti lõbus tegevus, mistõttu võib tekkida kiusatus ühe keerulise mudeli asemel ehitada mitu lihtsat, kus me fitime igale interaktsioonimuutuja tasemele oma mudeli, jagades oma andmed iseseisvate mudelite vahel. See on iseenesest mõistlik mõte, mille puuduseks on, et vähemate andmete peal fititud mudelitel tulevad laiemad usalduspiirid.

Edaspidi õpime selliseid mudeleid graafiliselt tõlgendama, kuna koefitsientide otse tõlgendamine ei ole siin sageli perspektiivikas.

Kui meil on mudelis interaktsiooniliige $x_1 x_2$, siis on enamasti mõistlik ka lisada eraldi liikmetena ka x_1 ja x_2.

Interaktsioonimudelis sõltub x_1 mõju tugevus y-le x_2 väärtusest. Selle sõltuvuse määra kirjeldab b_3 (x_1 ja x_2 interaktsiooni tugevus). Samamoodi ja sümmeetriliselt erineb ka x_2 mõju erinevatel x_1 väärtustel. Ainult siis, kui

Näiteks mudel, milles on pidev y-muutuja, pidev prediktor "education" ja binaarne prediktor "sex_male" (1 ja 0):

$$score = a + b_1 \times education + b_2 \times sex_{male} + b_3 \times education \times sex_{male}$$

Kui me tahame interaktsioonimudelile $y = a + b_1x_1 + b_2x_2 + b_3x_1x_2$ lisada kontrolli counfounder z-le, mis võiks olla korreleeritud y-ga ja korreleeritud x_1-ga, siis tasub ka sellele modelleerida interaktsioon kujul $y = a + b_1x_1 + b_2x_2 + b_3x_1x_2 + b_4z + b_5x_2z$ [31].

[31] Kuna iga interaktsiooniliikme lisamine kasvatab koefitsientide veapiire, tasub võrdluseks fittida ka lihtsam, ilma b_5x_2z liikmeta mudel

Kui meil on kaks faktor-prediktorit, siis mudel kujul

$$y = 0 + bx_1x_2$$

Mudeldab eraldi nende faktorite tasemete kõikvõimalud kombinatsioonid (võttes eelduseks võrdse Y varieeruvuse kõigil faktorite tasemetel).

Oletame, et meil on lisaks pidevale prediktorile x1 ka faktor-prediktor x2. Diskreetsed e faktor-prediktorid rekodeeritakse automaatselt nn dummy-muutujateks. Kahevalentse muutuja sex = c("male", "female") korral ilmub regressiooni uus dummy-muutuja, sex_female, kus emased on 1-d ja isased 0-d. Üldine intercept vastab siis isaste mõjule ja sex_female intercept annab emaste erinevuse isastest. Kui meil on n-tasemega diskreetne muutuja, rekodeerime selle n-1 dummy-muutujana, millest igaüks on 0/1 kodeeringus ja millest igaühe interceptid annavad erinevuse null-taseme (selle taseme, mis ei ole rekodeeritud dummy-muutujana) interceptist. Mudeli seisukohast pole oluline, millise faktortunnuse taseme me nulltasemeks võtame. Terminoloogiliselt on meie n-tasemega faktortunnus seletav muutuja (explanatory variable), millest tehakse n-1 regressorit. Seega tehniliselt on mudeli liikmed regressorid, mitte seletavad muutujad. Üks seletav muutuja võib anda välja mitu regressorit (nagu eelmises näites) ja üks regressor võib põhineda mitmel muutujal (näit x_1x_2 interaktsiooniterm).

Interaktsioonimudeli 2D avaldus on kurvatuuriga tasapind, kusjuures kurvatuuri määrab b_3.

Interaktsiooniga mudel on AIC-i järgi pisut vähem eelistatud võrreldes kahe prediktoriga mudeliga m2. Seega, eriti lihtsuse huvides, eelistame m2-e.

```
m3 <- lm(Sepal.Width ~ Sepal.Length + Petal.Length + Sepal.Length * Petal.Length,
    data = iris)
AIC(m1, m2, m3)
```

```
##    df       AIC
## m1  3 179.46442
## m2  4  92.11691
## m3  5  93.42623
```

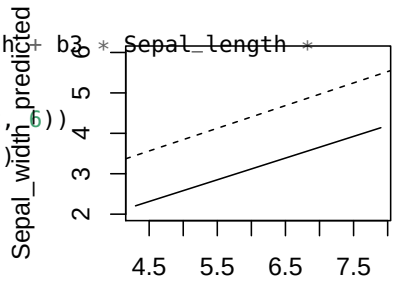
Interaktsiooniliikme (b_3) määramine on tunduvalt väiksema täpsusega (suuremate veapiiridega) kui põhiliikmete oma. Selleks, et realistlikes oludes tagada interaktsiooniliikmele sama täpsus, tuleks valimi suurust tõsta 10-20 korda. See asjaolu on relevantne näiteks randomiseeritud katse-kontroll skeemis. Juhul, kui kõik patsiendid reageerivad ravimile ja platseebole samamoodi, siis puuduvad selles skeemis ravi interaktsioonid teiste muutujatega ja see tähendab, et meil ei pea olema juhuvalim. Sel juhul piisab ka näiteks sellest, kui me võtame uuringusse need patsiendid, kes seda ise tahavad ja randomiseerime nad kahte gruppi. Kui aga ravim (või platseebo) mõjub osadele patsientidele teistmoodi kui teistele, siis on meil kas (1) vaja juhuvalimit, mida randomiseerida, või (2) võime kasutada ka nn mugavusvalimit, aga ainult siis kui oskame modelleerida ravimi (või platseebo) interaktsioone ja kui meil on piisavalt suur valim, et see mudeldamine oleks piisavalt täpne.

Ennustused interaktsioonimudelist

Kõigepealt anname rea väärtusi x_1-le ja hoiame x_2 konstantsena.

(ref:ennustus-interaktsioonimudelist) Ennustus interaktsioonimudelist, kus x_1 (Sepal_Length) on antud rida väärtusi ja x_2 (Petal_length) hoitakse konstantsena (pidevjoon). Interaktsioonimudeli regressioonijoon on paraleelne ilma interaktsioonita mudeli ennustusele (katkendjoon).

```
Petal_length <- mean(iris$Petal.Length)
a <- coef(m3)["(Intercept)"]
b1 <- coef(m3)["Sepal.Length"]
b2 <- coef(m3)["Petal.Length"]
b3 <- coef(m3)["Sepal.Length:Petal.Length"]
Sepal_width_predicted <- a + b1 * Sepal_length + b2 * Petal_length + b3 * Sepal_length *
    Petal_length
plot(Sepal_width_predicted ~ Sepal_length, type = "l", ylim = c(2, 6))
abline(coef(m2)[c("(Intercept)", "Sepal.Length")], lty = "dashed")
```

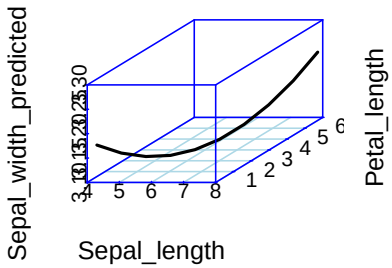


Nagu näha viib korrutamistehe selleni, et interaktsioonimudeli tõus erineb ilma interaktsioonita mudeli tõusust.

Kui aga interaktsioonimudel plottida välja 3D-s üle paljude x_1 ja x_2 väärtuste, saame me regressioonikurvi (mitte sirge), kus b_3 annab kurvatuuri.

Ennustused 3D interaktsioonimudelist üle paljude x_1 (Sepal_Length) ja x_2 (Petal_length) väärtuste.

Vau! See on alles ennustus!



Mida teha kui X-i ja Y-i suhe ei ole sirge?

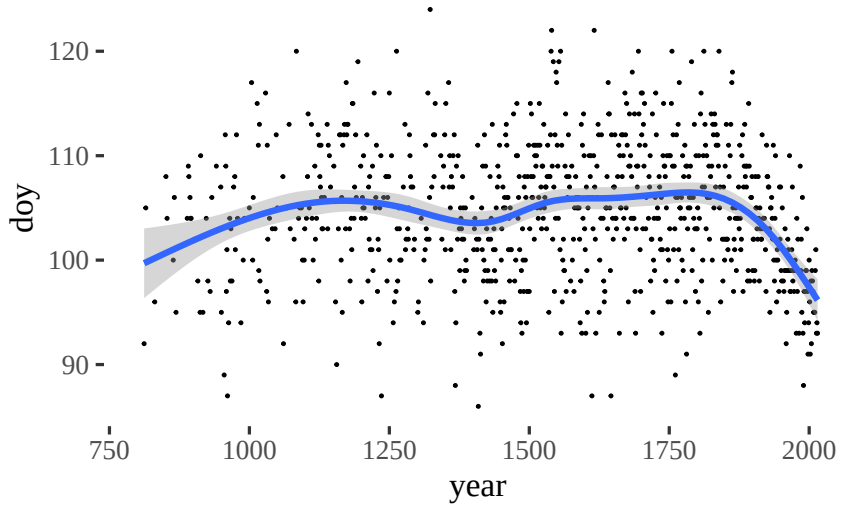
Kui me tahame sirge asemel tõmmata läbi andmepunktide kurvi, siis lihtsaim võimalus on polünoomne regressioonimudel kujul $Y = \alpha + \beta_1 X + \beta_2 X^2 + ...$, kus iga järgmise polünoomse liikme X^3, X^4 jne lisamine lisab kõverale ühe kurvi, kus kõver muudab suunda. Sellisel viisl on võimalik mudeldada üskkõik kui keerulist seost X-i ja Y-i vahel, aga mudeli koefitsientidel ei ole siin mingit mudelivälist tõlgendust, polünoomsed mudelid kipuvad sageli andmeid üle-fittima ja nende mudelite usalduspiirid kipuvad X-i jaotuse servades väga laiaks minema. NB! polünoomsed mudelid piirduvad enamasti poly(2) või poly(3)-ga (kõrgemad ploynoomid tahavad pahasti andmeid ülefittida) ja nendega töötamisel pole tõesti muud valikut, kui mudeli ennustused välja plottida (koos veapiiridega muidugi).

```
library(rethinking)
data(cherry_blossoms)
d <- cherry_blossoms
```

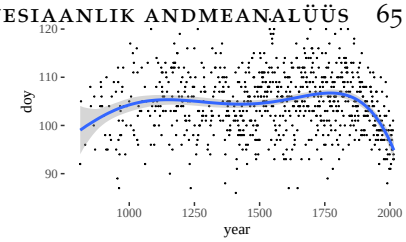
doy on kirsside õitselemineku päev (aasta algusest).

```
ggplot(d, aes(year, doy)) + geom_point(size = 0.1) + geom_smooth() + ggthemes::theme_tufte()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
#b_poly5 <- brm(doy~poly(year, 5), data=d, cores = 3, chains = 3)
#write_rds(b_poly5, "b_poly5.rds")
b_poly5 <- read_rds("b_poly5.rds")
plot(conditional_effects(b_poly5),

     points=TRUE, point_args= list(size=0.1),

     theme=ggthemes::theme_tufte())
```



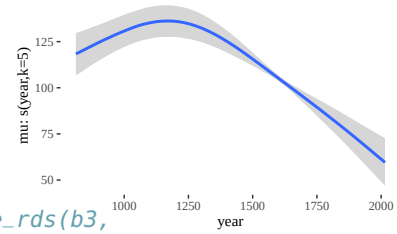
Kaasaegsem meetod on splainid (splines), mis tähendab mitme sirge fittimist kasutaja poolt ette antud katkestuskohtadega. Spline on sujuv funktsioon, mis koosneb väiksematest osafunktsioonidest. Splaine on mitmesuguseid, kuid kõige lihtsamad on nn B-splainid, kus B tähendab "basis" ehk osafunktsiooni. Kui tavaline polünoomne regressioon transformeerib prediktori võttes selle ruutu, ja seega määrab üks funktsioon kogu kurvi kuju, siis splainides on X-telg (X-muutuja) jagatud osadeks, kusjuures iga punkt X-teljel on fititud kasutades kahte lähimat osafunktsiooni.

$$\mu = \alpha + w_1 B_{i,1} + w_2 B_{i,2} + w_3 B_{i,3} + \dots$$

kus iga $B_{i,n}$ on n-da osafunktsiooni väärtus real i ja w-d on osafunktsioonide kaalud. w-d töötavad nagu sirge tõusud, adjusteerides iga osafuntsiooni mõju mudeli ennustusele vastaval X_i väärtusel. Seega on splainid üks lineaarse regressiooni alamliike. B-splainide kasutamisel peame käsitsi määrama murdepunktid ehk sõlmed X-teljel. Mida lähemal on mingi X-telje punkt sõlmele n, seda suuremat mõju omab Y-i hinnangule sellel X-i väärtusel n-is osafunktsioon ja vastavalt väiksemat mõju omab n-1s osafunktsioon (ja kõigi ülejäänud osafunktsioonide mõju on null). Seega muutub X-teljel liikudes sujuvalt 2 osafunktsiooni mõju osakaal, ja sõlmest läbi liikudes muutub ka see, millised osafunktsioonid üldse mõju avaldavad.

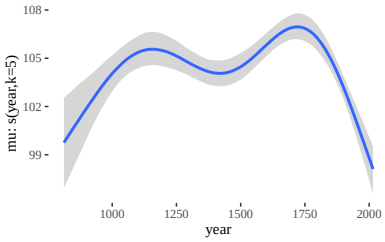
Kui palju sõlmi teha (k = 5) ja kuhu need paigutada, saame brms-is ise valida:

```
# b2 <- brm(doy~s(year, k = 5), knots = list(year = c(760, 1200, 1500, 1800,
# 2000)), data=d, cores=3, chains =3) write_rds(b2, 'raamat/data/b2.rds')
b2 <- read_rds("raamat/data/b2.rds")
# summary(b2)
plot(conditional_smooths(b2), theme = ggthemes::theme_tufte())
```



default paigutus annab erineva pildi!

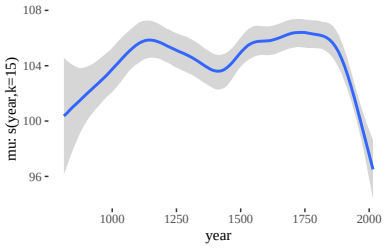
```
# b3 <- brm(doy~s(year, k = 5), data=d, cores=3, chains =3) write_rds(b3,
# 'raamat/data/b3.rds')
b3 <- read_rds("raamat/data/b3.rds")
plot(conditional_smooths(b3), theme = ggthemes::theme_tufte())
```



Järgnevalt proovime kasutada 15 sõlme ja paigutame need 15-sse X-muutuja kvantiili. See tagab, et kahe sõlme vahele jääb alati sama palju andmepunkte.

```
# d2 <- d[ complete.cases(d$doy) , ] # complete cases on doy knot_list <-
# quantile( d2$year , probs=seq(0,1,length.out=15) ) b4 <- brm(doy~s(year, k =
```

```
# 15), knots = list(knot_list), data=d2, cores=3, chains =3) write_rds(b4,
# 'raamat/data/b4.rds')
b4 <- read_rds("raamat/data/b4.rds")
plot(conditional_smooths(b4), theme = ggthemes::theme_tufte())
```



See meetod paistab olevat ka brms-i default. Kood `brm(doy~s(year, k = 15))` töötab samamoodi.

Regressioon kui kirjeldus ja kui põhjuslik hüpotees

Regressioonanalüüsi võib vaadelda 1) empiirilise kirjeldusena y ja x-i koos-varieerumisest või 2) muutujate vaheliste põhjuslike suhete analüüsina. Esimesel juhul ei tõlgenda me x ja y suhet x-i mõjuna y-le ja senikaua kui mudeli fit väljaspool andmeid, mida kasutati selle mudeli fittimiseks, on piisavalt hea, ei ole võimalik, et me fitime vale struktuuriga mudeli. Kui me fitime 2 mudelit (i) $Y = \alpha + \beta_1 X_1$ ja (ii) $Y = \alpha + \beta_1 X_1 + \beta_2 X_2$, siis eeldame, et kahe mudeli β_1 koefitsiendid tulevad erinevad. Aga sellest pole midagi, sest need kirjeldavad mõlemal juhul vaid empiirilisi seoseid.

Teisel, põhjuslikul juhul on kõik teisiti. Eeldades et X_2 on üks Y-i põhjustest, on nüüd esimese mudeli veakomponendis peidus ka $\beta_2 X_2$. Kui X_1 ja X_2 on omavahel korreleeritud, siis tekib meil seetõttu ka korrelatsioon X_1 ja veakomponendi vahel, mis rikub mudeli eeldusi, kallutades mudeli fittimisel meie hinnangut β_1-le, misläbi osa X_2 mõjust Y-le omistatakse ekslikult X_1-le. See kõik juhtub siis, kui teise mudeli β_2 ei ole null ja esineb X_1 ja X_2 vaheline põhjuslik seos.

Selle kallutatuse tõlgendamine sõltub omakorda X_1 ja X_2 vahelise põhjusliku seose struktuurist. Oluline on mõista, et regressioon-imudeli enda struktuuris ei ole põhjuslikku infot - mudel ei tea isegi põhjuslikkuse olemasolust. Seega on meil lisaks regressioonimudelile vaja iseseisvat põhjuslike mõjude mudelit, mille formuleerime puhtalt teaduslikest asjaoludest lähtuvalt. Ja lisaks on meil vaja algoritmilist meetodit, mis võimaldaks meil põhjusliku mudeli struktuuri järgi otsustada, milline peab olema regressioonimudeli struktuur.

Kõige lihtsam põhjuslik mudel vastab randomiseeritud eksperimendile, kus me võrdleme katse- ja kontrolltingimust. Me usume, et kui katsetingimus (ravimi manustamine) mõjutab võrreldes kontrolltingimusega (platseeboga) katse väljundit (suremust), siis me oleme näidanud, et vastav ravim vähendab suremust. Ning me usume, et välja arvatud katsetöötlus (ravimi manustamine) on katse ja kontrlollgrupid keskeltläbi piisavalt sarnased, et nende juhuslikud erinevused ei mõjuta oluliselt katsetulemust – see on randomiseerimise roll katseskeemis. Seega on meie põhjuslik skeem *ravim* $\implies$ *suremus* ja regressioonimudel suremus~ravim, mis

sisuliselt taandub kahe grupi keskmiste suremuste võrdlusele.

Kuidas on aga asjalood, kui meil ei lubata katset teha? Näiteks, kuidas määrata suitsetamise mõju kopsuvähile? Siin ei ole meil tegemist randomiseeritud katsega (me ei tohi jagada populatsiooni juhuslikult kahte gruppi ja sundida neist ühte suitsetama). Seega peame kasutama statistilisi meetodeid, et kontrollida oma tulemust nn confounderite vastu. Siin on lihtsaim regressioonimudel $vhk \sim suitsetamine + muutuja_1 + ...$

Aga muutujaid on maailmas palju ja meil peab olema mingi reegel, mille järgi otsustada, millised neist additiivsesse mudelisse sisse panna ja millised välja jätta. Mudeli ennustusjõu maksimeerimine siin ei aita. Selle asemel peame mõistma võimalike põhjuslike skeemide suhet mitmese regressioonimudelitega. Põhjuslikud skeemid on nagu legod, mis kombineerivad endis nelja loogiliselt võimalikku ehitusplokki.

1. toru: $x \Longrightarrow z \Longrightarrow y$

Kui me tahame teada, kas x (põhjus) mõjutab y-t (tagajärge), siis toru puhul mudel $tagajrg \sim phjus + z$ vähendab x-i mõju (sest see mõju käib läbi z-i). Samas, mudel $tagajrg \sim phjus$ näitab x-i mõju. Seega, kumba mudelit kasutada sõltub sellest, kas me tahame näidata x-i otsest või kaudset mõju y-ile.

Järgnev skeem illustreerib olukorda, kus me tahame testida rohelist otseteed põhjuse ja tagajärje vahel ja selleks peame sulgema kõrvaltee, ehk tagaukse. Tagaukse sulgemiseks konditsioneerime oma regressioonimudeli z-l, ehk lisame z-i mudelisse $tagajrg \sim phjus + z$.

2. kahvel: $x \Longleftarrow z \Longrightarrow y$

Kahvli puhul regressioonimudel $y \sim x$ näitab x-i mõju y-le (ehkki meie põhjuslikkuse mudelis puudub x ja y vaheline põhjuslik seos), aga mudel $y \sim x + z$ välistab selle mõju. Seega peaksime sellise põhjusliku hüpoteesi korral mudelisse z-i sisse panema, sest see aitab kontrollida z konfounding mõju vastu. Jällegi, kahvlist tagaukse sulgemiseks lisame z-i oma mudelisse $tagajrg \sim phjus + z$.

3. laupkokkupõrge: $x \Longrightarrow z \Longleftarrow y$

Laupkokkupõrke korral on olukord eelnevaga vastupidine. Nüüd avab mudel y $x + z$ tagaukse ja laseb z-i segava mõju mudelisse sisse, mis tekitab meile võlts-põhjusliku suhte x ja y vahel. Laupkokkupõrke korral on tagauks suletud senikaua, kui me ei lisa z-i oma mudelisse.

4. järglane: see on toru, kus z-i juurest hargneb veel üks nool D-le. Siin saame me mudelisse A lisades ligikaudu sama tulemuse,

mis z-i lisades. Seega, kui z-i väärtused pole meile teada, võime hädaga ka A-d kasutada.

Järglasest tagaukse võib sulgeda nii z-i (eelistatult) kui D (kui z on küll maailmas olemas, kuid puudub meie andmetest) mudelisse lisamise kaudu, aga ehk pole mõtet lisada mõlemat, sest z ja D võivad olla tugevalt kollineaarsed ehk omavahel korreleeritud.

Näiteks võib meil tekkid olukord, kus testime suitsetamise mõju vähile, aga ühtlsi usume, et inimeste elukoht mõjutab iseseisvalt nii vähki kui suitsetamist (tööstuslinnades haigestuvad ka mittesuitsetajad enam vähki, aga neis elav töölisklass ka suitsetab rohkem). Selles põhjuslikus skeemis töötab vanus kahvlina, millega arvestamiseks tuleb see regressioonimudelisse muutujana sisse panna: $vhk \sim suits + vanus$.

Ja lõpuks tähtis tähelepanek, et meie põhjuslikes skeemides mängivad oma rolli ka need muutujad, mida me mõõtnud ei ole. Me usume, et isade (ja emade) haridustase on põhjuslikus seoses nende laste haridustasemega ($isa \implies laps$). Oletame, et tahame uurida põhjuslikku hüpoteesi, mille kohaselt ka vanaisa haridustase mõjutab otseselt nende lapselaste haridustaset ($vanaisa \implies laps$). Aga loomulikult usume me ka, et vanaisa haridustase mõjutab tema laste haridustaset ($vanaisa \implies isa$). Seega saame loogilise skeemi torukujulise tagauksega, mille sulgemiseks peame fittima mudeli $laps \sim vanaisa + isa$

mitmese regressiooni üldised printsiibid

1. võta sisse kõik teaduslikku huvi pakkuvad muutujad ja viska välja muutujad, mille kohta sul pole põhust arvata, et nad võiksid y väärtusi mõjutada. Mõtle selle peale, milline võiks olla mudeli struktuur enne, kui sa andmed mudelisse paned.

2. muutujad, mida sa reaalselt mõõtsid, ei pruugi olla need muutujad, mis mudelisse lähevad (isegi siis kui nad on teaduslikult relevantsed) – näiteks arvuta kehamassiindeks mõõdetud muutujate põhjal.

3. Kui muutujate vahel esineb väga tugevaid korrelatsioone (kollineaarsus), siis kombineeri kollineaarsed muutujad üheks või transformeeri neid või viska mõni kollineaarne muutujatest välja.

4. muutujad, mis ei varieeru, ei oma ka regressioonis mõju.

5. tugeva mõjuga muutujate puhul võib olla vajalik sisse tuua nende muutujate interkatsioond.

6. Milliseid muutujaid sisse panna sõltub nii teaduslikust küsimusest, katsedisainist, taustateadmistest, kui andmetest (ka N-st) ja beta koefitsientidest. Kui prediktori beta-koefitsient on kitsaste veapiiridega, tasub see mudelisse jätta. Kui beta tuleb väga laiade veapiiridega, siis vastava regressori eemaldamine mudelist võib tõsta ülejäänud beta-de hinnangute täpsust. Samas, kui regressor on mudelis selgelt teaduslikel põhjustel (näiteks katse disain tingib selle koha mudeli struktuuris) siis tuleks see muutuja ikkagi sisse jätta. Kui beta väärtus tundub teile teaduslikult absurdne, siis vaadake kõigepealt veapiire. Kui need on kitsad, alles siis tasub hakata mõtlema, mis teie mudelil või prediktoril viga võiks olla, et te saite sellise beta (näiteks muutuja väärtused ei kajasta populatsiooni vaid on kallutatud). On väga tähtis, et te kirjeldate oma käsikirjas, milliseid valikuid ja miks te tegite muutujaid lisades/välja visates. See aitab vältida vale-avastuste kirjandusse sattumist.

7. Logaritmitud x-muutujate korral mudeldate te prediktorite multiplikatiivseid suhteid ka additiivses mudelis $y = a + b_1 log(x_1) + b_2 log(x_2)$. See on sageli hea mõte. Alati tasub kaaluga nullist suuremate y ja x muutujate logaritmimist (vt allpool).

Bayesiaanlik lähenemine regressioonile: Eesmärk on mudeldada protsessi, mis meie andmeid genereeris. Näiteks, kui meil on juhuvalim ja me tahame võrrelda katse ja kontrollgruppi, siis mudel `y ~ x`, kus x on binaarne indikaatormuutuja "katse/kontroll", võiks töötada. Kui aga katsed on tehtud mitme erineva ensüümibatchiga, siis oleks parem mudel võib-olla `y ~ x + (x|batch)`. Bayesiaani eesmärk on kanda edasi taustinfos ja andmetes peituvat ebakindlust järeldustesse (vastavalt priorite ja tõepärafunktsiooni kaudu), nii et järelduste ebakindlus oleks aus ja samas nii väike kui loogiliselt võimalik. Posteeriori laius (ebakindluse mõõduna) on alati väiksem kui priori või likelihoodi oma (eeldades, et prior kannab mudelisse lisainfot, st et prior ei ole tasane).

4 põhilist regressiooni strateegiat

Kõigile 4 starteegiale on ühine, et need nurjab nn pööratud põhjuslikkus, kus Y-muutuja varieeruvus põhjustab X-muutuja varieeruvust (k.a. olukord kus X põhjustab Y ja samal ajal Y põhjustab X-i. Näiteks olukord, kus alkoholi liigtarbimine viib alla üliõpilase õppeedukuse ja samal ajal halvad hinded ajavad ta jooma.)

1) Põhjusliku mudeli testimine – see on kõige keerulisem juht, mida käsitleb eelmine peatükk.

2) Ennustamine – pane mudelisse sisse kõik muutujad, mis mudeli adjusteeritud R^2 parandavad ja kogu moos. Välja tasub visata tugevalt kollineaarsed muutudad ja muutujad, mille tõusukoefitsient tuleb usaldusväärselt null (kitsaste veapiiridega; p väärtus on siin väärtusetu). Muidu on nii, et senikaua kuni muutujate lisamine tõstab mudeli ennustusjõudu, ei ole tähtis, kas ja kuidas selle mudeli regressorid üksteise koefitsientide väärtusi mõjutavad (need küsimused muutuvad oluliseks siis, kui me anname koefitsientidele põhjuslikke tõlgendusi). Ennustuslike mudelite puhul on põhiline oht ülefittimine, kus fititakse rohkem müra kui signaali. Seega on oluline mudeli ennustusjõu testimine andmetel, mida ei ole kasutatud selle mudeli fittimisel.

3) Prediktorite määramine, mis ennustavad Y-muutujat. Siin me tahame teada (i) kas mingi muutuja X mõjutab Y-it, (ii) kui palju X mõjutab Y-t ja (iii) millised võiksid olla parimad prediktorid Y ennustamiseks. Seega on 3) lihtsam versioon 2)-st, mis tähendab, et me eelistame lihtsaid 1-2 muutujaga mudeleid ja tõlgendame neid ettevaatlikult. Enamasti võidab muutuja, millele on teistest selgelt kõrgem R^2.

4) Individuaalsete mõõteobjektide uurimine võrreldes mudeliennustusega. Siin on kaks võimalust. Kui iga mõõteobjekti kohta on üks mõõtmine, siis tuleb fittida adjusteeritud regressioonimudel ja järjestada residuaalid. Kui me tahame näiteks anda auhinna klassi parimale õpilasele, aga samas tahame, et see hinnang ei sõltuks tema perekonna jõukusest (sest usume, et rikaste vanemate lapsed saavad kunstliku eelise), siis fitime mudeli *keskmine_hinne = a + b x vanemate_sissetulek* ja suurima positiivse residuaaliga laps ongi võtja. Kui meil on aga igale lapsele mitu mõõtmist, mis on ka piisavalt suure varieeruvusega, siis võime teha K lapse kohta K - 1 nn *dummy* muutujat ja fittida mudel *hinne = a + dummy_muutujad + vanemate_sissetulek* ja me saame tulemuseks iga lapse võrdluse referentslapsega (igale *dummy* muutujale). Nüüd leiame suurima positiivse tõusuga tõusu-koefitsiendi ja laps, kellele see kuulub, saabki auhinna.

Vaatame nüüd veidi keerulisemat juhtu, me tahame auhinda anda parimale õpetajale ja meie Y-muutuja on õpetaja evaluatsioonihinne (üks iga klassi kohta, mida see õpetaja õpetab). Iga klassi kohta me teame selle õpilaste keskmist hinnet (X_1) ja selle klassi suurust (X_2). Seega tuleb meil iga õpetaja kohta mitu andmepunkti. Nüüd teeme K õpetaja kohta K-1 dummy muutujat ja meie mudel tuleb Y = *dummy_muutujad* + $X1$ + $X2$. Siin ei saa me paraku residuaale ($Y_i - \bar{Y}_i$) kasutada, sest residuaal eemaldaks õpetaja efekti

(iga õpetaja keskmine residuaal võrdub nulliga). Selle asemel kasutame *hinnang petajale* $i = Y_i - (\hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i})$. Kus mütsikesega betad on fititud koefitsiendid ja X1i tähistab sellele õpetaja vastavale klassile vastavat keskmist hinnet. Seega saame õpetajale nii palju adjusteeritud hindeid, kui palju on sellel õpetajal klasse. Selles mudelis ei määra me klassi suuruse ja klassi keskmise hinde mõju igale õpetajale eraldi vaid eeldame, et need mõjud on õpetajate vahel piisavalt sarnased (mis võib olla vale). Samas on meil nii rohkem andmeid, mille pealt β_1 ja β_2 fittida. Hiljem õpime sellisel kohal mitmetasemelisi mudeleid kasutama.

Vähimruutude meetodiga fititud mudelite töövoog – lm()

Kuna lm() funktsiooniga ja bayesi meetodil fititud mudeliobjektidega töötamine on mõnevõrra erinev, õpetame seda eraldi. Siinkohal anname põhilise töövoo lm() mudelobjektide inspekteerimiseks.

Töötame m3 mudeliobjektiga, mis on interaktsioonimudel:
$Sepal.Width \sim Sepal.Length \times Species$
ehk

$$Speal.Width = a + b_1 \times Sepal.Length + b_2 \times Species + b_3 \times Sepal.Length \times Species$$

```
library(ggeffects)
m3 <- lm(Sepal.Width ~ Sepal.Length * Species, data = iris)
```

- **Vaatame mudeli koefitsiente**

```
tidy(m3)
```

```
## # A tibble: 6 x 5
##   term                            estimate std.error statistic  p.value
##   <chr>                              <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                       -0.569     0.554     -1.03 3.06e- 1
## 2 Sepal.Length                       0.799     0.110      7.23 2.55e-11
## 3 Speciesversicolor                  1.44      0.713      2.02 4.51e- 2
## 4 Speciesvirginica                   2.02      0.686      2.94 3.85e- 3
## 5 Sepal.Length:Speciesversicolor    -0.479     0.134     -3.58 4.65e- 4
## 6 Sepal.Length:Speciesvirginica     -0.567     0.126     -4.49 1.45e- 5
```

Interaktsioonimudeli koefitsientide jõllitamine on sageli tühi töö ja vaimu närimine. Õnneks on meil muid meeotodeid, kuidas lm() mudelitega töötada.

Võrdluseks - nii fitime eraldi mudeli igale irise liigile. Tulemus on sarnane interaktsioonimudeliga kategoorilisele muutujale (Species), kuid siin ei ole tavapärast eeldust, et iga liigi varieeruvused on

võrdsed. Samas, interaktsioonimudelit saab fittida ka pidevale muutujale!

```
iris %>% split(.$Species) %>% map(~lm(Sepal.Width ~ Sepal.Length, data = .)) %>%
    map(summary) %>% map_dfr(~broom::tidy(.), .id = "Species")
```

Adjusteeritud r2 tasub eraldi üle vaadata.

```
summary(m3)$r.squared
```

```
## [1] 0.6227084
```

0.62 tähendab, et mudel suudab seletada mitte rohkem kui 62% y-muutuja (Sepal.Width) varieeruvusest.

- **Testime mudeli eeldusi**

NB! Praktikas ei pruugi need testid olla väga vajalikud ja tegelikes töövoogudes kasutatakse neid pigem harva. Me eelistame grrafilisi lahendusi formaalsetele testidele, mille p väärtused kipuvad olema ebastabiilsed väikestel valimitel ja põsivalt madalad suurtel valimitel.

`broom::augment()` annab meile tabelina fititud väärtused (.fitted), residuaalid (.resid), fittitud väätruste standardvead (.se.fit). Residuaal = y data value - fitted value. Seega positiivne residuaal näitab, et mudeli ennustus keskmisele y väärtusele mingil x-muutujate väärtusel on madalam kui juhutb olema tegelik y-i andmepunkti väärtus. See võib olla tingitud y-muutuja normaalsest bioloogilisest varieeruvusest, aga ka sellest, et mudel ei kirjelda täiuslikult x-ide ja y tegelikku seost.

```
(a_m3 <- broom::augment(m3))
```

```
## # A tibble: 150 x 10
##    Sepal.Width Sepal.Length Species .fitted .se.fit   .resid   .hat .sigma
##          <dbl>        <dbl> <fct>     <dbl>   <dbl>    <dbl>  <dbl>  <dbl>
##  1         3.5          5.1 setosa     3.50  0.0399 -0.00306 0.0215  0.273
##  2         3            4.9 setosa     3.34  0.0403 -0.343   0.0218  0.272
##  3         3.2          4.7 setosa     3.18  0.0512  0.0163  0.0354  0.273
##  4         3.1          4.6 setosa     3.10  0.0591 -0.00380 0.0471  0.273
##  5         3.6          5   setosa     3.42  0.0385  0.177   0.0200  0.273
##  6         3.9          5.4 setosa     3.74  0.0581  0.157   0.0455  0.273
##  7         3.4          4.6 setosa     3.10  0.0591  0.296   0.0471  0.272
##  8         3.4          5   setosa     3.42  0.0385 -0.0232  0.0200  0.273
##  9         2.9          4.4 setosa     2.94  0.0772 -0.0441  0.0803  0.273
## 10         3.1          4.9 setosa     3.34  0.0403 -0.243   0.0218  0.273
## # ... with 140 more rows, and 2 more variables: .cooksd <dbl>, .std.resid <dbl>
```

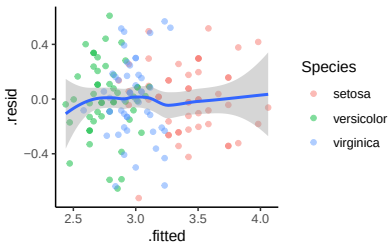
.hat >1 sugereerib high leverage andmepunkte
.std.resid on studentiseeritud residuaal, mis on sd ühikutes (.resid/sd(.resid))

- *Lineaarsus - residuaalid~fitted plot*

Residuals vs fitted plot testib lineaarsuse eeldust - kui .resid punktid jaotuvad ühtlaselt nulli ümber, siis mudel püüab kinni kogu süstemaatilise varieeruvuse teie andmetest ja see mis üle jääb on juhuslik varieeruvus.

```
ggplot(a_m3, aes(.fitted, .resid)) +
geom_point(aes(color = Species), alpha = 0.5) +
geom_smooth() + theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



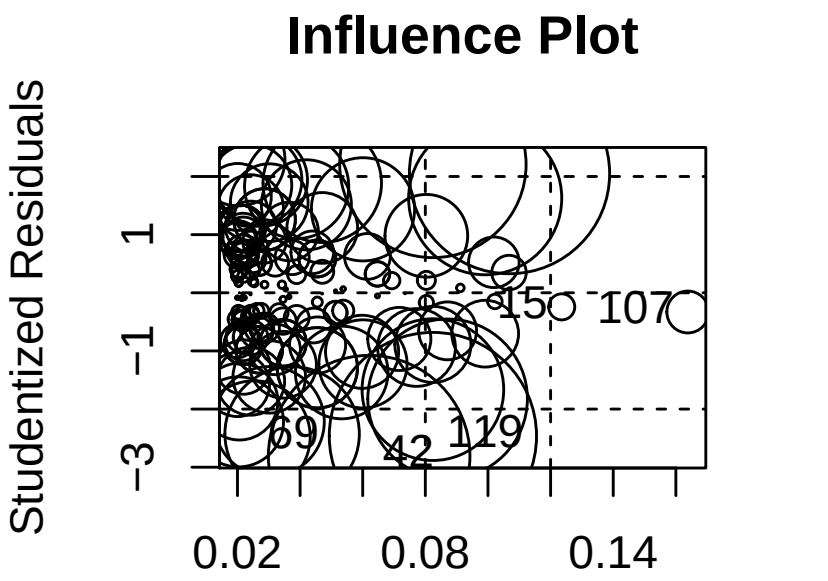
- *Mõjukuse plot*

- *outlier*id – studentideeritud residuaalid > 2 või < -2. Studentiseeritud residuaali saab (ligikaudu) jagades vaatluse residuaali residuaalide standardhälbega. See protseduur võimaldab paremini võrrelda erinevate vaatluste residuaale.

Standardiseeritud residuaali arvutamine: Kui E_i on i-s residuaal, k on mudeli regressorite arv ja n on vaatluste arv, siis $h_i = 1/n + E_i / \sum E^2$, $S_E = (E^2/(n-k-1))^{1/2}$ ja $E_{st} = E_i/(S_E(1-h_i)^{1/2})$ kus E~st on standardiseeritud residuaal, mis suurtel valimitel on väga sarnane studentiseeritud residuaaliga (mis erineb selle poolest, et välistab iga residuaali S_E-st seda residuaali genereerinud vaatluse). Kui n on suur, siis kehtib enam-vähem seos $E_{st} = E_i/sd(E)$, kus E_st on nii standardiseeritud kui studentiseeritud residuaal.

- *high leverage* vaatlused - hat > 1 - sugereerib *high leverage* vaatlust. Keskmine hat value $= (k+1)/n$, kus k on regressorite arv (mitte arvestades intercepti) ja n on vaatluste arv. NB! Kuna *high leverage* vaatlused tõmbavad regressioonijoon enda suunas, siis on neil sageli madalad residuaalid (erinevalt outlieritest, mis ei ole *high leverage* vaatlused)

```
library(car)
influencePlot(m3, id.method="identify",

              main="Influence Plot",

              sub="Circle size is proportional to Cook's distance")
```




```
##        StudRes        Hat       CookD
## 15  -0.2425051 0.12355047 0.001390772
## 42  -2.8097561 0.06205447 0.083074901
## 69  -2.4769993 0.02533856 0.025669068
## 107 -0.3305319 0.16381329 0.003589354
## 119 -2.4644123 0.08241137 0.087816707
```

Horisontaalsed referentsjooned näitavad 0, 2 ja -2 studentiseeritud residuaale. Vertikaalsed referentsjooned näitavad hat-väärtusi 2h ja 3h.

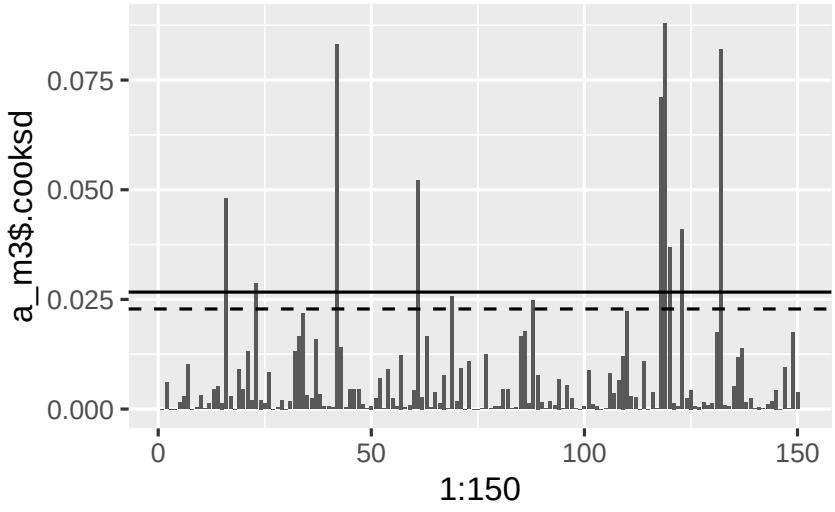
> Regressiooni *outlier* on vaatlus, mille y-muutja väärtus on ebatavaline X-muujutuja väärtuse kontekstis. Seega annab *outlier* mudeli fittimisel kõrge residuaaliga punkti. Lihtsalt (mitte-konditsionaalselt) ebatavalised Y-i või X-i väärtused ei pruugi olla *outlier*id. Kui peaks juhuma, et *outlier* langeb kokku ebatavalise X-i väärtusega, siis selle punkti eemaldamine muudab märkimisväärselt mudeli koefitsiente. Selline *outlier* on ühtlasi ka *high leverage* vaatlus. Siit jõuame mõjukate vaatluste (*Influential observations*) defineerimisele — Mõjukus mudeli koefitsientidele = *Leverage* x *"outlierness"*. *High leverage* andmepunktid on x-muutujate ekstreemsed punktid, mille lähedal ei ole n-mõõtmelises ruumis (kui teil on n x-muutujat) teisi punkte. Seetõttu läheb fititud mudel just nende punktide lähedalt mõõda. Mõjukad punktid on tüüpiliselt ka *high leverage* punktid, kuid vastupidine ei kehti!

- *Cooki kaugus - mõjukus*

.cooksd on Cook-i kaugus, mis näitab mõjukust. Rusikareeglina tähendab *cooksd* $>$ 3 cooksd keskväärtust, et tegu võiks olla mõjuka vaatlusega. Teine võimalus on pidada mõjukaks igat punkti, mis on kõrgem kui 4/n. Kolmanadad arvavad jälle, et *.cooksd* $>$ 1 v *.cooksd* $>$ 0.5 viitab mõjukale vaatlusele. Üldiselt on kõigi mudeli eelduste kontrollidega nii, et vastava statistiku jaotuse jõllitamine on sageli kasulikum kui automaatselt mingi *cut-offi* järgi talitamine.

Cooki D andmepunktile saame valemist $D_i = \frac{E_i'}{k+1} + \frac{h_i}{1-h_i}$, kus D_i on i-ndale vaatlusele vastav Cooki kaugus ja E_i' on sellele vaatlusele vastav studentiseeritud residuaal.

```
ggplot(data = NULL, aes(x = 1:150, y = a_m3$.cooksd)) +
geom_col() +
geom_hline(yintercept = 4/150) +
geom_hline(yintercept = 3 * mean(a_m3$.cooksd), lty = 2)
```

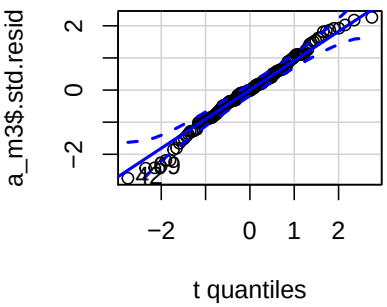


- *Residuaalide normaalsus - qq plot*

Kas residuaalid on normaaljaotusega? NB! studentiseeritud residuaalid on studenti t jaotusega ja üldiselt on targem vaadata neid, kui tavalisi residuaale. Studenti t jaotusele pean ette andma ka vabadusastmete arvu e df-i.

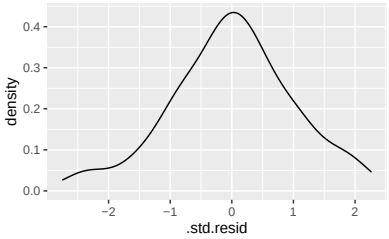
```
car::qqPlot(a_m3$.std.resid, distribution = "t", df = 149)
```

```
## [1] 42 69
```



QQ-plot näitab erinevust normaaljaotusest (t jaotusest) eelkõige residuaalide jaotuse sabades. Antud juhul on kõik hästi. Samuti tasub meeles pidada, et välja arvatud väikesetel valimitel hoolitseb tsentraalne piirteoreem selle eest, et residuaalide normaalsus ei oleks vajalik eeldus.

Olulisem on vaadata, et residuaalide jaotus ei oleks mitmetipuline. Kui on, siis võib see olla märgiks, et mudelist on puudu mõni faktormuutuja, mis andmetes olevad diskreetsed loomulikud alampopulatsioonid lahku ajaks.

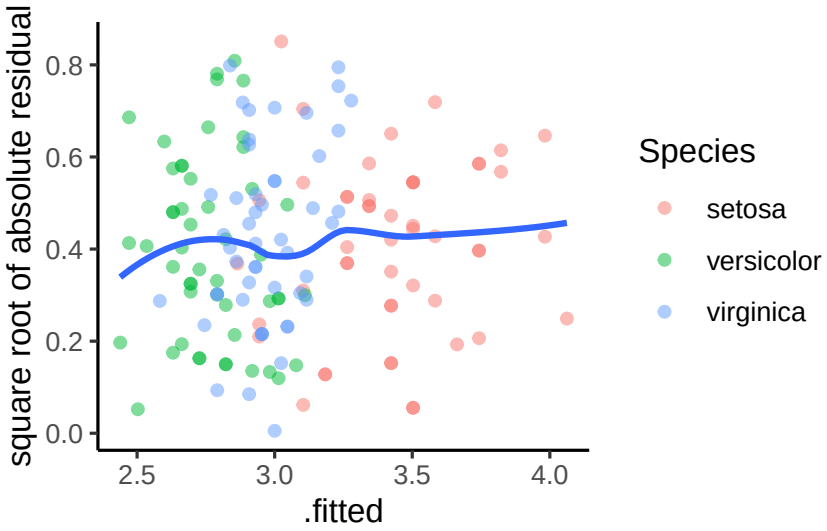



```
ggplot(a_m3, aes(.std.resid)) + geom_density()
```

- *Homoskedastilisus - Scale-location plot*

Scale-location plot - homoskedastilisuse eeldust ehk seda, et varieeruvus ei sõltuks prediktormuutuja väärtusest. Y-teljel on ruutjuur studentiseeritud residuaalide absoluutväärtusest.

```
ggplot(a_m3, aes(.fitted, .resid %>% abs %>% sqrt)) +
geom_point(aes(color = Species), alpha = 0.5) +
ylab("square root of absolute residual") +
geom_smooth(se = FALSE) + theme_classic()
```



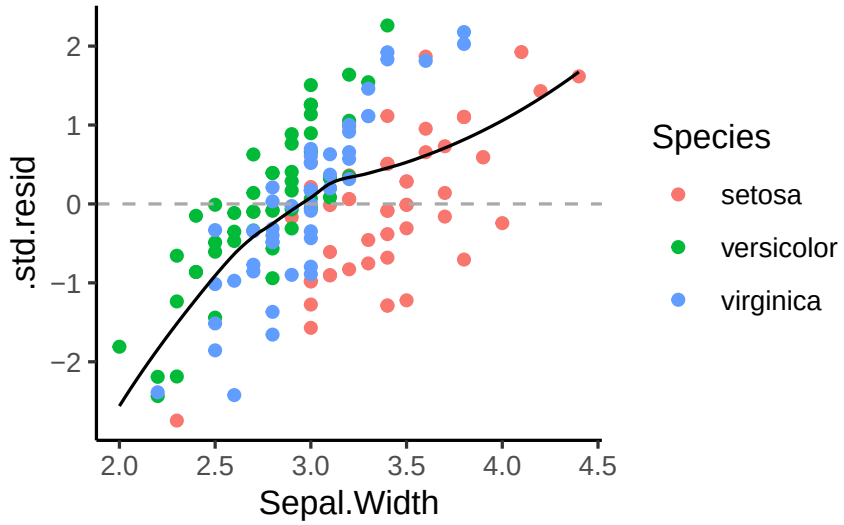
Sagedusliku statistika rohi heteroskedastilisuse vastu on robustsne standardviga (Humer-White'i estimaator), mis võimaldab erinevaid standardvigu erinevatel X-i väärtustel. Robustsed standardvead on enamasti suuremad kui tavalised.

- *Residuaalid y ja x muutujate vastu*

Kõigepealt residuaalid y-muutja vastu

```
ggplot(a_m3, aes(Sepal.Width, .std.resid)) +
geom_point(aes(color = Species)) +
geom_hline(yintercept = 0, lty = 2, color = "darkgrey") +
geom_smooth(se = F, color = "black", size = 0.5) +
theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Mudel paistab süstemaatiliselt alahindama Sepal Width-i tema madalatel väärtustel ja vastupidi. Horisontaalne punktiirjoon näitab, kus mudel vastab täpselt andmetele. Studentiseeritud residuaalid on sd ühikutes

- *Teeme mudeli põhjal ennustusi (marginal plots)*

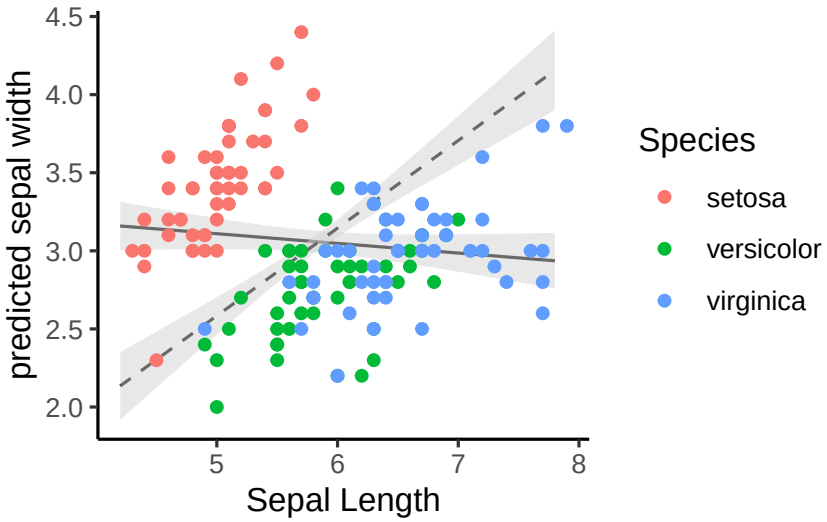
Me ennustame y-i keskmisi väärtuseid etteantud x-i väärtustel.

ggpredict() ennustab y-muutuja väärtusi ühe x-muutuja väärtuste järgi, hoides kõiki teisi x-muutujaid konstantsena.

Kõigepealt võrdleme lihtsa 1 prediktoriga mudeli ennustust kahe prediktoriga mudeli ennustusega

```
lm1 <- lm(Sepal.Width ~ Sepal.Length, data = iris)
lm2 <- lm(Sepal.Width ~ Sepal.Length + Petal.Length, data = iris)

mydf <- ggpredict(lm1, terms = "Sepal.Length")
mydf2 <- ggpredict(lm2, terms = "Sepal.Length")
```

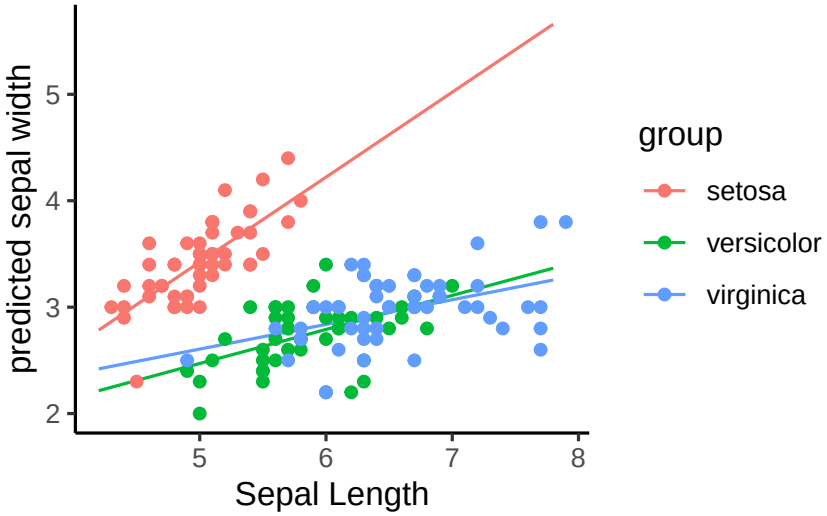
```
# terms võtab kuni 3 muutujat, millest 2 peavad olema faktormuutujad.

ggplot(mydf, aes(x, predicted)) +
  geom_line() +

  geom_ribbon(data = mydf, aes(ymin = conf.low, ymax = conf.high),

              alpha = 0.5, fill="lightgrey") +

  geom_line(data = mydf2, aes(x, predicted), lty=2)+

  geom_ribbon(data = mydf2, aes(ymin = conf.low, ymax = conf.high),

              alpha = 0.5, fill="lightgrey") +

  geom_point(data=iris, aes(Sepal.Length, Sepal.Width, color=Species)) +

  xlab("Sepal Length") +

  ylab("predicted sepal width")+

  theme_classic()
```



Katkendjoonega fit on additiivsele mudelile, mis arvestab Irise liikidega ja punktiirjoon on interaktsioonimudelile.

```
mydf <- ggpredict(m3, terms = c("Sepal.Length", "Species"))
ggplot(mydf, aes(x, predicted)) +
geom_line(aes(color = group)) +
geom_point(data = iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
xlab("Sepal Length") +
```

```
ylab("predicted sepal width") +
theme_classic()
```



Nii saab sisestada üksikuid parameetriväärtusi ja neile ennustusi teha:

```
(mydf1 <- ggpredict(m3, terms = c("Sepal.Length [5, 22]", "Species [setosa, versicolor]")))
```

```
##
## # Predicted values of Sepal.Width
## # x = Sepal.Length
##
## # Species = setosa
##
##  x | Predicted |   SE |         95% CI
## ------------------------------------
##  5 |      3.42 | 0.04 | [ 3.35,  3.50]
## 22 |     17.00 | 1.88 | [13.32, 20.68]
##
## # Species = versicolor
##
##  x | Predicted |   SE |        95% CI
## -----------------------------------
##  5 |      2.47 | 0.08 | [2.31,  2.63]
## 22 |      7.91 | 1.21 | [5.53, 10.28]
```

- *Võrdleme mudeleid*

---

1. Eeldus - kõik võrreldavad mudelid on fititud täpselt samade andmete peal.

2. Eeldus (ei ole vajalik AIC meetodi puhul) - tegemist on nn nested mudelitega. Nested mudel tähendab, et kõik väiksema mudeli liikmed on olemas ka suuremas mudelis.

   Mudelite võrdlus ANOVA-ga (ainult nested mudelid)

```
tidy(anova(lm1, lm2, m3))
```

```
## Warning: Unknown or uninitialised column: 'term'.
```

```
## # A tibble: 3 x 6
##   res.df   rss    df sumsq statistic   p.value
##    <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
## 1    148  27.9    NA NA         NA   NA
## 2    147  15.4     1 12.5      169.   4.83e-26
## 3    144  10.7     3  4.71      21.2  2.06e-11
```

   Mudelite võrdlus AIC-ga

```
AIC(lm1, lm2, m3)
```

```
##     df       AIC
## lm1  3 179.46442
## lm2  4  92.11691
## m3   7  43.34175
```

AIC (Akaike Informatsiooni Kriteerium) on number, mis püüab tabada mõistlikku tasakaalu mudeli fiti valimiandmetega ja parsinoomia vahel. Väiksema AIC-ga mudel on eelistatud suurema AIC-ga mudeli ees (samas, AIC-l kui ühel arvul puudub tõlgendus).

Probleem AIC-i taga on selles, et parem fit valimiandmetega võib tähendada mudeli ülefittimist (ja seega halvemat mudelit). Kuna ülefittimise tõenäosus kasvab koos mudeli keerukusega (parameetrite arvuga), eelistame võimalikult lihtsat mudelit, mis samas seletaks võimalikult suure osa valimiandmete varieeruvusest.

Andmete transformeerimine

Lineaarsed transformatsioonid võivad hõlbustada mudeli koefitsientide tõlgendamist (näit. skaala millimeetritest meetritessse, tsentreerimine, standardiseerimine). Mittelineaarsed transformatsioonid (logaritmimine, jms) muudavad mudeli fitti ja võivad olla kasulikud mudeli aditiivsuse/lineaarsuse parandamisel. Oluline on mõista, et transformeeritud andmetega mudeleid tuleb tõlgendada transformeeritud skaalas. Seega, kui algsel skaalal pole muud tõlgendust, kui et väärtused on monotoonilised (näiteks suurem number on alati tähtsam kui väiksem number), siis sobib meile sama hästi iga lineaarne transformatsioon sellest skaalast (näiteks ruutjuure võtmine vms). Bioloogias enamasti asjad nii lihtsad ei ole ja seetõttu keskendume siin paremini tõlgendatavatele transformatsioonidele.

Logaritmimine

Kui muutujal saavad olla ainult positiivsed väärtused, siis on logaritmimine sageli hea mõte. Enne logaritmima asumist peab andmetest kaotama ka nullid, näiteks asendades need mingi väikese positiivse arvuga. Logaritmilises skaalas andmetele fititud mudelite β koefitsiendid peaaegu alati < 1.

Miks ja millal muutujaid logaritmida?

1. Muutuja(te) logaritmimine muudab muutujate vahelised suhted mitte-lineaarseks, samas säilitades mudeli lineaarsuse. Ja vastupidi, kui tavalises meetrilises skaalas juhtuvad additiivse mudeli muutujate vahelised seosed olema mitte-lineaarsed, siis x-muutuja(te) logaritmimine võib need muuta lineaarseks, ja sellega päästa ühe olulisema lineaarse regressiooni eelduse.

2. Logarimimine on hea, kui soovite y ja x muutuja omavahelist sõltuvust tõlgendada üle suhtelise muutuse ehk muutuse protsendi. Kui algses skaalas on β koefitsiendi loomulik tõlgendus additiivne: x-i kasv 1 ühiku võrra ennustab y-i kasvu β ühiku võrra, siis naturaallogaritmitud x-i korral on üks loomulikest multiplikatiivsetest tõlgendustest: x-i kasv 1 ühiku võrra ennustab y-i muutust

... protsendi võrra. Additiivsel juhul me liidame ja lahutame, multiplikatiivsel juhul aga korrutame ja jagame.

3. Muutuja logaritmimine võib viia selle muutuja lähemale normaaljaotusele (lognormaaljaotuse logaritm on normaaljaotusega). Algselt paremale kaldu jaotuse äärmuspunktid võivad regressioonile liiga suurt kaalu omada, milline probleem sageli kaob logaritmimisel.

4. Kui mudeli residuaalid on ümber nulli tugevalt paremale poole kiivas jaotusega, siis andmete logaritmimine võib need normaliseerida. Samuti siis, kui residuaalide sd on proportsionaalne fititud väärtusega (st CV, mitte SD, on konstantne) ja siis, kui te ususte, et residuaalid peegeldavad multiplikatiivseid vigu.

5. Kui y ja x-i vaheline sõltuvus on eksponentsiaalne.

6. Y-muutuja logaritmimine võib aidata hetoroskedastilisuse vastu.

7. Teaduslik teooria võib indikeerida logaritmimist. Näit pH on log skaalas.

8. Logaritmimine võib lihtsustada mudelit, vähendades interaktsiooniliikmete arvu.

Mudeli fiti kvaliteedi koha pealt pole vahet, millist logaritmi te kasutate – erinevused on "pelgalt" mudeli koefitsientide tõlgendustes. Naturaallogaritmitud log(x) andmete peal fititud mudeli korral on algses lineaarses skaalas tõlgendatav logaritmitud andmete peal fititud β, aga log-skaalas muutujate väärtused ei tähenda peale vaadates suurt midagi. Vastupidiselt on kümnendlogaritmitud log10(x) või kahendlogaritmitud log2(x) andmed log skaalas tõlgendatavad, aga mitte neil fititud β lineaarses skaalas. Igal juhul eelistavad loodusteadlased kasutada log2 ja log10 skaalasid, mida on mugavam otse log-skaalas tõlgendada. log2 skaalas vastab üheühikuline muutus kahekordsele muutusele algses skaalas ja anti-logaritm on $2^{log2(x)}$. Log10 skaalas vastab üheühikuline muutus 10-kordsele muutusele algses skaalas ja anti-logaritm on $10^{log10(x)}$.

Naturaallogaritmitud andmetega töötamine

Järgnevalt õpetame naturaallogaritmitud andmetega fititud mudelite β koefitsiendite tõlgendamist algses, meetrilises skaalas ja suhtelises protsendiskaalas.

Naturaallogaritmi alus on $e \approx 2.71828$ ja sellel on järgmised matemaatilised omadused:

1. $log(e) = 1$

2. $log(1) = 0$
3. $log(A^r) = r * log(A)$
4. $e^{log(A)} = A$ ehk $exp(log(A)) = A$ ehk $2.72^{log(A)} \approx A$
5. $log(AB) = logA + logB$
6. $log(A/B) = logA - logB$
7. $exp(AB) = exp(A)^B$
8. $exp(A + B) = exp(A)exp(B)$
9. $exp(A - B) = exp(A)/exp(B)$

Lineaarsel regressioonil saab log-transformatsiooni kasutada kolmel erineval viisil:

- y = α + βx – lineaarne mudel (transformeerimata)
- y = α + β * log(x) – lineaar-log mudel (transformeeritud on prediktor(id))
- log(y) = α + βx – log-lineaar mudel (transformeeritud on y-muutuja)
- log(y) = α + β * log(x) – log-log mudel (transformeeritud on y ja x muutujad)

Lineaarses mudelis y = α + βx, annab β selle, mitu ühikut muutub Y keskväärtus, kui X muutub ühe ühiku võrra.

Lineaar-log mudelis jääb kehtima sama β tõlgendus, mis ülalpool, ainult et log-ühikutes. Seega viib logx-i muutus ühe log ühiku võrra y keskväärtuse muutusele β ühiku võrra (see kehtib muidugi ka log2 ja log10 skaalades). Naturaal-logaritmi korral on tõusukoefitsiendi tõlgendus "Y muutus juhul kui X tõuseb ühe protsendi võrra". Kui me juba kasutasime naturaallogaritmimist, siis tahame ilmselt tõlgendust pigem muutuse protsendina ja/või algsetes meetrilistes skaalas (log2 ja log10 ei võimalda selliseid mugavaid tõlgendusi):

- β on oodatud y muutus, kui x kasvab ex korda.
- Kui β on väike, siis saab seda tõlgendada kui suhtelist erinevust. Näiteks, kui β = 0.06, siis 1 ühikuline x-i muutus viib u 6%-sele y muutusele. Sedamõõda kuidas β kaugeneb nullist (näiteks β = 0.4), hakkab selline hinnang tõsiselt alahindama tegelikku x-i mõju y väärtusele.
- Oodatud y muutus kui x kasvab p protsenti on $\beta \times log([100 + p]/100)$. Näit, kui x kasvab 10% võrra (ehk kui korrutame x-i 1.1-ga), siis log(110/100) = 0.095 ja 0.095β on oodatud y muutus.

Log-lineaarse mudeli korral,

- kui x kasvab 1 ühiku võrra, siis oodatud y väärtus kasvab exp(β) korda.
- kui X kasvab 1 ühiku võrra, siis y kasvab β protsenti.
- Kui x kasvab c ühiku võrra, siis oodatud y väärtus kasvab exp(cβ) korda.
- Kui β on väike, siis 100β vastab y protsentuaalsele muutusele juhul kui x muutub 1 ühiku võrra (kui $\beta = 0.06$, siis x-i muutus 1 ühiku võrra viib y-i 6% tõusule).

Log-log mudeli korral on tõlgendus oodatud y-i muutus protsentides kui x muutub mingi protsendi võrra. Sellist suhet kotsutakse ökonomeetrias elastiliseks ja log x-i β koefitsient on "elastilisus."

- Kui me korrutame x-i e-ga, siis korrutame oodatud y-i väärtuse $exp(\beta)$-ga.
- Et saada y suhtelist muutust, kui x kasvab p protsenti, arvuta $a = log([100 + p]/100)$ ja siis võta $exp(a\beta)$.
- Kui X kasvab ühe protsendipunkti võrra, siis Y kasvab β protsenti.

Andmete normaliseerimine põõratud normaaltransformatsiooniga

Kui tahate tingimata suvalise jaotusega andmeid normaaljaotusesse transformeerida ja logaritmimine ei kõlba selleks, siis võite kasutada järgmist meetodit.

1) Järjesta pidevad andmed suuruse järgi ja asenda iga andmepunkti väärtus tema järjekorranumbriga 1 ... N
2) teisenda need järjekorranumbrid z-skoorideks kasutades põõratud normaaljaotuse kumulatiivset tihedusfunktsiooni: $1/(2N), 3/(2N), ...(2N - 1)/(2N)$.
3) mudelda neid andmeid klassikalise lineaarse regressiooniga, mis eeldab normaalsust.

R-i kood on siin.

```
library(RNOmni)
# Sample from the chi-1 square distribution
y = rchisq(n = 1000, df = 1)
# Rank-normalize
z = rankNorm(y)
```

Sellel meetodil on kaks väga olulist puudust: see kaotab informatsiooni, mida kannavad algse andmejaotuse pikkades sabades olevad andmepunktid ja sel viisil transformeeritud andmete ühikud on täiesti mittemidagiütlevad. Sellest hoolimata kasutatakse seda laialt geneetilistes assosatsiooniuuringutes pidevate tunnuste puhul. Seda on kasutatud kehamassiindeksi, veres leiduvate metaboliitide, geenikespressiooninäitude jms puhul.

Standardiseerimine

Kui prediktor x_1 on mõõdetud näiteks eurodes ja prediktor x_2 aastates, siis on meil fititud koefitsientidele b_1 ja b_2 peale vaadates raske öelda, kumb mõjutab y-muutuja väärtust rohkem. Kuna euro ühik on palju granuleeritum kui aasta, siis võib ka väga väike nullist erinev b_1 omada mudeli seisukohast suuremat tähtsust kui suhteliselt suur b_2.

$x.z = (x - mean(x))/sd(x)$

Sellisel viisil standardiseeritud andmete keskväärtus on 0 ja sd = 1. Seega on kõik predikorid samas skaalas ja me mõõdame efekte sd ühikutes. See lubab võrrelda algselt erinevas skaalas prediktoreid. Intecept tähendab nüüd keskmist ennustust, juhul kui kõik prediktorid on fikseeritud oma keskväärtustel.

Kui mudel sisaldab lisaks pidevatele prediktoritele ka binaarseid prediktoreid, siis on kasulikum standardiseerida üle 2xSD, jättes binaarsed muutujad muutmata.

$x.z2 = (x - mean(x))/(2 * sd(x))$

Nüüd tähendab 1 ühikuline muutus efekti -1 SD-st kuni 1 SD-ni üle keskväärtuse. Selline standardiseerimine muudab pidevate ja binaarsete regressorite beta-koefitsiendid omavahel enam-vähem võrreldavaks selles mõttes, et nullist kaugemal asuv koefitsient viitab, et vastav regressor on mudelis suurema mõjuga.

Korrelatsioon üle regressiooni ja regressioon keskmisele

Kui standardiseerime nii y kui x-i

```
x <- (x - mean(x)) / sd(x) y <- (y  -mean(y)) / sd(y)
```

siis y~x regressiooni intercept = 0 ja tõus on sama, mis x ja y vaheline korrelatsioonikoefitsient r. Seega jääb tõus alati -1 ja 1 vahele.

Siit tuleb ka seletus nähtusele, mida kutsutakse regressiooniks keskmisele (*regression to the mean*). Fakti, et y on sellises mudelis alati 0-le lähemal kui x, kutsutaksegi regressiooniks keskmisele. Näiteks, kui olete 20 cm keskmisest pikem ja pikkuse päritavus on 0.5, siis on oodata, et teie järglased on keskeltläbi 10 cm võrra keskmisest pikemad (ja teist lühemad). Selle pseudo-põhjusliku nähtuse avastas Francis Galton.

Tsentreerimine

x.c1 = x - mean(x) annab keskväärtuseks nulli aga jätab varieeruvused algsesse skaalasse. Näiteks interaktsioonimudelite koefitsiendid on otse tõlgendatavad tsentreeritud prediktorite korral.

Teine võimalus on tsentreerida mõnele teaduslikult mõistlikule väärtusele. Näiteks IQ-d saab tsentreerida 100-le (x - 100).

Mudeli koefitsientide transformeerimine

Ilma interaktsioonideta mudeli korral saab sama tulemuse, mis prediktoreid tsentreerides, kui me reskaleerime tavalises skaalas fititud mudeli koefitsiendid, korrutades iga β oma prediktori kahekordse sd-ga ($\beta_x = \beta \times 2 \times sd(x)$). Nende β_x-de pealt näeb iga muutuja suhtelist tähtsust mudelis.

Teine võimalus beta koefitsientide transformeerimiseks, mis ei eelda prediktorite normaalsust, on korrutada betad läbi vastavate muutujate interkvartiilsete range-dega (IQR).

Hoitatus: standardiseeritud koefitsiente ei tohi kasutada, et võrrelda samade prediktorite mõju erinevate andmete peal fititud sama struktuuriga mudelile.

Bootstrap

Sissejuhatus: andmed ei ole sama, mis tegelikkus

Alustuseks simuleerime juhuvalimi n = 3 lõpmata suurest normaaljaotusega populatsioonist, mille keskmine on 100 ja sd on 20.

```
set.seed(1)  # makes random number generation reproducible
Sample <- rnorm(n = 3, mean = 100, sd = 20)
Sample
```

```
## [1]  87.47092 103.67287  83.28743
```

```
mean(Sample)
```

```
## [1] 91.47707
```

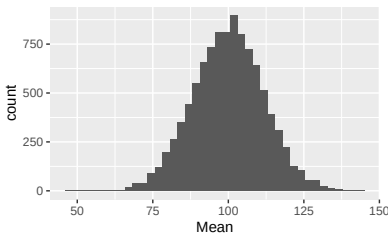
```
sd(Sample)
```

```
## [1] 10.76701
```

Nagu näha on meie konkreetse valimi keskmine 10% väiksem kui peaks ja valimi sd lausa kaks korda väiksem. Seega peegeldab meie valim halvasti populatsiooni — aga me teame seda ainult tänu sellele, et tegu on simulatsiooniga.

Kui juba simuleerida, siis robinal: tõmbame ühe juhuvalimi asemel 10 000, arvutame seejärel 10 000 keskmist ja 10 000 sd-d ning vaatame nende statistikute jaotusi ja keskväärtusi. Simulatsioon on nagu tselluliit — see on nii odav, et igaüks võib seda endale lubada.

Meie lootus on, et kui meil on palju valimeid, millel kõigil on juhuslik viga, mis neid populatsiooni suhtes ühele või teisele poole kallutab, siis rohkem on valimeid, mis asuvad tõelisele populatsioonile pigem lähemal kui kaugemal. Samuti, kui valimiviga on juhuslik, siis satub umbkaudu sama palju valimeid tõelisest populatsiooniväärtusest ühele poole kui teisele poole ja vigade jaotus tuleb sümmeetriline.



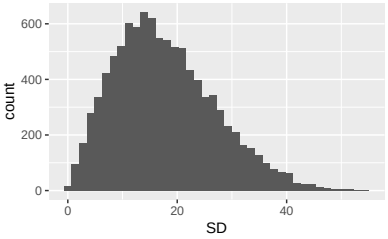

```
mean(Summary$Mean)
```

```
## [1] 99.98043
```

```
mean(Summary$SD)
```

```
## [1] 17.76452
```

Oh-hooo. Paljude valimite keskmiste keskmine ennustab väga täpselt populatsiooni keskmist aga sd-de keskmise keskmine alahindab populatsiooni sd-d. Valem, millega sd-d arvutatakse, töötab lihtsalt kallutatult, kui n on väike (<10). Kes ei usu, kordab simulatsiooni valimiga, mille N=30.



Ja nüüd 10 000 SD keskväärtused:

```
# funktsioon jaotuse moodi määramiseks

mode <- function(x, adjust = 1) {

    x <- na.omit(x)

    dx <- density(x, adjust = adjust)

    dx$x[which.max(dx$y)]
}

mode(Summary$SD)
```

```
## [1] 14.07554
```

SD-de jaotus on ebasümmeetriline ja mood ehk kõige tõenäolisem valimi sd väärtus, mida võiksime oodata, on u 14, samal ajal kui populatsiooni sd = 20. Lisaks alahinnatud keskmisele sd-le on sd-de jaotusel paks saba, mis tagab, et teisest küljest pole ka vähetõenäoline, et meie valimi sd populatsiooni sd-d kõvasti üle hindab.

Arvutame, millise sagedusega on valimite standardhälbed > 25

```
mean(Summary$SD > 25)
```

```
## [1] 0.2114
```

Me saame >20% tõenäosusega pahasti ülehinnatud SD.

```
mean(Summary$SD < 15)
```

```
## [1] 0.4344
```

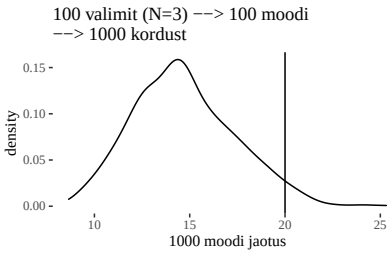
Ja me saame >40% tõenäosusega pahasti alahinnatud sd. Selline on väikeste valimite traagika.

Aga vähemalt populatsiooni keskmise saame me palju valimeid tõmmates ilusasti kätte — ka väga väikeste valimitega.

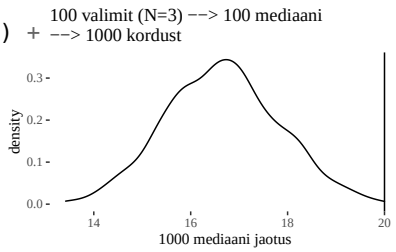
See, et valimite keskmine SD tuleb väiksem kui populatsiooni SD tuleb SD arvutamise algoritmist, mis töötab halvasti väikestel valimitel. Näitame seda simulatsiooniga kus arvutame moodi või mediaani simulatsioonist, kus on 100 väikest valimit (N=3) ja seejärel kordame seda 100 korda, et saada 100 moodi või mediaani.

```
fun1 <- function(N, N_simulations) {

  df <- tibble(a = rnorm(N * N_simulations, 100, 20),

               b = rep(1:N_simulations, each = N)) %>%

    group_by(b) %>%

    summarise(Mean = mean(a), SD = sd(a))

  mode(df$SD)
}

# N - valimi suurus

# N_simulations - valimite arv

# fun1 väljund on valimite standardhälvete jaotuse mood

fun2 <- function(n, N, N_simulations)

  replicate(n, fun1(N=N, N_simulations = N_simulations))

# fun2 annab SD-de moodid (n tükki)

b <- fun2(n=1000, N=3, N_simulations=100)

#tekitame 1000 SD-de jaotuse, millest igaühes 100 SD-d, moodid

#ja plotime need

ggplot(data=NULL, aes(b)) +

  geom_density() +

  geom_vline(xintercept = 20)+
```

```
  xlab("1000 moodi jaotus")+
  ggtitle("100 valimit (N=3) --> 100 moodi \n--> 1000 kordust") +
  ggthemes::theme_tufte()
```




```
mean(b > 20)
## [1] 0.033
fun1_1 <- function(N, N_simulations) {
  df <- tibble(a = rnorm(N * N_simulations, 100, 20),
               b = rep(1:N_simulations, each = N)) %>%
    group_by(b) %>%
    summarise(Mean = mean(a), SD = sd(a))
  median(df$SD)
}
#siin sama, mis fun1, ainult et saame mediaanid (mitte moodid)
fun2_1 <- function(n, N, N_simulations)
  replicate(n, fun1_1(N=N, N_simulations = N_simulations))
c <- fun2_1(n=1000, N=3, N_simulations=100)
ggplot(data=NULL, aes(c)) +
  geom_density() +
  geom_vline(xintercept = 20) +
  xlab("1000 mediaani jaotus")+
  ggtitle("100 valimit (N=3) --> 100 mediaani \n--> 1000 kordust") +
  ggthemes::theme_tufte()
mean(c > 20)
```

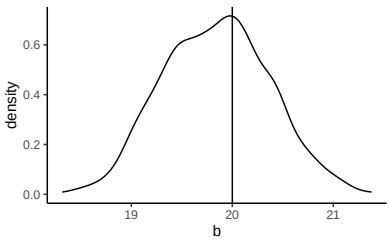


```
## [1] 0
```

Ainult tühine osa meie 100 valimi keskmistest on suuremad kui populatsiooni tõene väärtus.

Ja nüüd teeme seda uuesti veidi suurema valimiga N=60

```
b <- fun2(n = 1000, N = 60, N_simulations = 100)
ggplot(data = NULL, aes(b)) +
geom_density() +
geom_vline(xintercept = 20) +
theme_classic()
```




```
mean(b > 20)
```

```
## [1] 0.394
```

Nüüd on u pooled väärtustest suuremad kui populatsiooni väärtus.

Kahjuks pole meil ei vahendeid ega kannatust loodusest 10 000 valimi kogumiseks. Enamasti on meil üksainus valim. Õnneks pole sellest väga hullu, sest meil on olemas analoogne meetod, mis töötab üsna hästi ka ühe valimiga. Me teeme lihtsalt ühest valimist mitu, mis meenutab pisut mittemillegist midagi tegemist, aga veidi üllatuslikult töötab selles kontekstis üsna hästi. Seda metoodikat kutsutakse *bootstrappimiseks* ja selle võttis esimesena kasutusele parun von Münchausen. Too jutukas parun nimelt suutis end soomülkast iseenda patsi pidi välja tõmmata (koos hobusega), mis ongi bootstrappimise põhimõte. (Inglise kultuuriruumis tõmbab bootstrappija ennast mülkast välja oma saapaserva pidi – siit ka meetodi nimi.) Statistika tõmbas oma saapaid pidi mülkast välja Brad Efron 1979. aastal.

Bootstrappimine, ehk valimid valimist.

Populatsioon on valimile sama, mis on valim bootstrappitud valimile.

Nüüd alustame ühestainsast empiirilisest valimist ja genereerime sellest 2000 virtuaalset valimit. Selleks tõmbame me oma valimist virtuaalselt 2000 uut juhuvalimit (bootstrap valimit), millest igaüks on sama suur kui algne valim. Saladus seisneb selles, et bootstrap valimite tõmbamine käib asendusega, st iga empiirilise valimi element, mis bootstrap valimisse tõmmatakse, pannakse kohe algsesse valimisse tagasi. Seega saab seda elementi kohe uuesti samasse bootstrap valimisse tõmmata (kui juhus nii tahab). Seega sisaldab tüüpiline bootstrap valim osasid algse valimi numbreid mitmes korduses ja teisi

üldse mitte. Iga bootstrap valimi põhjal arvutatakse meid huvitav statistik (näiteks keskväärtus) ja kõik need 2000 bootstrapitud statistikut plotitakse samamoodi, nagu me ennist tegime valimitega lõpmata suurest populatsioonist. Ainsad erinevused on, et bootstrapis võrdub andmekogu suurus, millest bootstrap valimeid tõmmatakse, algse valimi suurusega ning, et iga bootstrapi valim on sama suur kui algne valim (sest meie poolt arvutatud statistiku varieeruvus, me tahame oma bootstrap valimiga tabada, sõltub valimi suurusest). Tüüpiliselt kasutatakse bootstrapitud statistikuid selleks, et arvutada usaldusintervall statistiku väärtusele.

Bootstrap ei muuda meie hinnangut statistiku punktväärtusele. Ta annab hinnangu ebakindluse määrale, mida me oma valimi põhjal peaksime tundma selle punkthinnangu kohta.

Bootstrappimine on üldiselt väga hea meetod, mis sõltub väiksemast arvust eeldustest kui statistikas tavaks. Bootstrap ei eelda, et andmed on normaaljaotusega või mõne muu matemaatiliselt lihtsa jaotusega. Tema põhiline eeldus on, et valim peegeldab populatsiooni – mis ei pruugi kehtida väikeste valimite korral ja kallutatud (mitte-juhuslike) valimite korral. Lisaks, tavaline bootstrap ei sobi hierarhiliste andmestruktuuride analüüsiks ega näiteks aegridade analüüsiks. Bootstrappida saab edukalt enamusi statistikuid, mida te võiksite elu jooksul arvutada, aga on siiski erandeid: näiteks valimi maksimum ja miinimumväärtused.

Bootstrap empiirilisele valimile suurusega n töötab nii:

1. tõmba (asendusega) empiirilisest valimist B uut virtuaalset valimit (B bootstrap valimit), igaüks suurusega n.
2. arvuta keskmine, sd või mistahes muu statistik igale bootstrapi valimile. Tee seda B korda.
3. joonista oma statistiku väärtustest histogramm või density plot

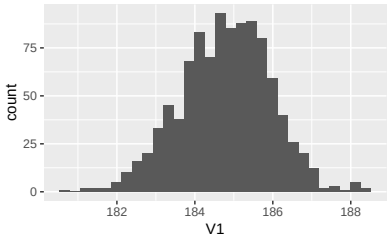
Nende andmete põhjal saab küsida palju toreidaid küsimusi — vt allpool.

Mis on USA presidentide keskmine pikkus? Meil on viimase 11 presidendi pikkused.

(ref:bootpost) Bootstrapitud posteerior USA presidentide keskmisele pikkusele. Järgnevas koodis ütleme me kõigepealt, et B = 1000 (et me võtame 1000 bootstrap valimit)

```
heights <- tibble(value = c(183, 192, 182, 183,

                            177, 185, 188, 188, 182, 185, 188))
n <- nrow(heights) #empirical sample size
B <- 1000 #nr of bootstrap samples
boot1 <-  replicate(B, sample_n(heights, n, replace = TRUE))

df22 <- boot1 %>% #boot1 object is a list
```

```
  as.data.frame() %>% #convert this list into a data frame

  summarise_all(mean) %>% t() %>% as_tibble()

ggplot(df22, aes(V1)) + geom_histogram()
```



Mida selline keskväärtuste jaotus tähendab? Me võime seda vaadelda posterioorse tõenäosusjaotusena. Selle tõlgenduse kohaselt iseloomustab see jaotus täpselt meie usku presidentide keskmise pikkuse kohta, niipalju kui see usk põhineb bootstrappimises kasutatud andmetel. Senikaua, kui meil pole muud relevantset teavet, on kõik, mida me usume teadvat USA presidentide keskmise pikkuse kohta, peidus selles jaotuses. Need pikkused, mille kohal jaotus on kõrgem, sisaldavad meie jaoks tõenäolisemalt tegelikku USA presidentide keskmist pikkust kui need pikkused, mille kohal posterioorne jaotus on madalam.

Kuidas selle jaotusega edasi töötada? See on lihtne: meil on 2000 arvu (2000 bootstrapitud statistiku väärtust) ja me teeme nendega kõike seda, mida parasjagu tahame.

Miks just 92% usaldusinterval? Vastus on, et miks mitte? Meil pole ühtegi head põhjust eelistada üht usaldusvahemiku suurust teisele. Olgu meil usaldusinteval 90%, 92% või 95% — tõlgendus on ikka sama. Nimelt, et me usume, et suure tõenäosusega jääb tegelik keskväärtus meie poolt arvutatud vahemikku. Mudeli ja maailma erinevused tingivad niikuinii selle, et konkreetne number ei kandu mudelist otse üle pärismaailma. Eelnevalt mainitud kihlveokontor töötab mudeli maailmas, mitte teie kodu lähedasel hipodroomil.

92% usaldusintervalli arvutamiseks on kaks meetodit, mis enamasti annavad vaid veidi erinevaid tulemusi.

1. HPDI — Highest Density Probability Interval — alustab jaotuse tipust (tippudest) ja katab 92% jaotuse kõrgema(te) osa(de) pindalast

```
HPDI(heights$value, prob = 0.92)
```

```
## |0.92 0.92|
##   177   192
```

2. PI — Probability Interval — alustab jaotuse servadest ja katab kummagist servast 4% jaotuse pindalast. PI 90%-le on sama, mis arvutada 5% ja 95% kvantiilid (jne).

```
PI(heights$value, prob = 0.9)
```

```
##    5%   95%
## 179.5 190.0
```

```
# quantile(heights$value, probs = c(0.05, 0.95)) teeb sama asja
```

HPDI on üldiselt parem mõõdik kui PI, aga teatud juhtudel on seda raskem arvutada. Kui HPDI ja PI tugevalt erinevad, on hea mõte piirduda jaotuse enda avaldamisega — jaotus ise sisaldab kogu informatsiooni, mis meil on oma statistiku väärtuse kohta. Intervallid on lihtsalt summaarsed statistikud andmete kokkuvõtlikuks esitamiseks.

Kui suure tõenäosusega on USA presidentide keskmine pikkus suurem kui USA populatsiooni meeste keskmine pikkus (178.3 cm mediaan)?

```
mean(heights$value > 178.3)
```

```
## [1] 0.9090909
```

Ligikaudu 100% tõenäosusega (valimis on 1 mees alla 182 cm, ja tema on 177 cm). Lühikesed jupatsid ei saa Ameerikamaal presidendiks!

Kuidas lahendada bootstrap, kui mei tahame usaldusintervalle kahe ebavõrdse grupi erinevusele? Näiteks kui meil on katsegrupis N = 25 ja kontrollgrupis N = 20, ja me tahame arvutada statistikut ES = katsegrupi keskmine - kontrollgrupi keskmine.

1. tõbma katsegrupist N = 25 bootstrapvalim
2. tõmba kontrollgrupist N = 20 bootsrapvalim
3. lahuta kontrollgrupi bootstrapvalimi mediaan katsegrupi omast (või aritmeetriline keskmine või ükskõik mis muu keskmise näitaja, mida hing ihaldab)
4. korda punkte 1-3 B korda ja tööta edasi bootstrapjaptusega, nagu eespool näidatud.

Mõned tava-bootstrapi paketid

Professionaalid kasutavad boot paketti, mis on ebameeldiva süntaksiga, aga see-eest laialt rakendatav. Boot paketi peale on ehitatud tavainimesele hästi kasutatav pakett bootES (Kirby and Gelranc, 2013, Behav Res 45:905–927), mis teeb lihtsaks usalduspiiride leidmise erinevat tüüpi efekti suurustele, kaasa arvatud lihtsad hierarhilised ja ühefaktorilise ANOVA tüüpi katseskeemid. Nendes pakettides tasub üldjuhul kasutada meetodit nimega BCa (bias-corrected-and-accelerated) usalduspiiride arvutamiseks. See meetod püüab parandada bootstrap-valimite võimalikku kallutatust (esineb sedavõrd, kui

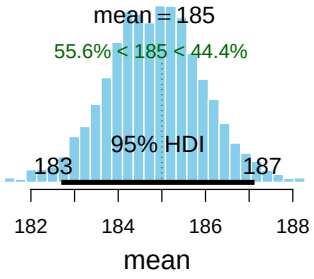
bootstrap-jaotuse tipp ei ole samas kohas kui oleks paljude päris-valimite pealt arvutatud statistikute jaotuse tipp) ja olukorda, kus statistiku väärtuse varieeruvuse määr sõltub statistiku väärtusest. BCa edukaks arvutamiseks peab bootstrap valimite arv kõvasti ületama valimi suurust. Simulatsioonidega on näidatud, et BCa (ja teisi) usalduspiire saab mõistlikult arvutada valimitelt, mille suurus on > 15. Sellest väiksemate valimite korral peate eeldama, et teie usaldusinetvallid valetavad. Aegridade, kus esineb järjestikuste ajapunktide vahelisi sõltuvusi, tuleks kasutada nn block bootstrappi, mida implementerrib näiteks boot::tsboot().

Bayesi bootstrap

Kui klassikalise bootstrap meetodi pakkus välja B. Efron aastal 1979, siis selle Bayesi versioon avaldati D.B. Rubini poolt 1981. a. Bayesi versioon bootstrapist on implementeeritud "bayesboot" paketis funktsioonis `bayesboot()`. Hea lihtsa seletuse Bayesi bootstrapi kohta saab siit `https://www.youtube.com/watch?v=WMAgzr99PKE` ja lihtsa r koodi selle meetodi rakendamiseks saab siit `https://www.r-bloggers.com/simple-bayesian-bootstrap/`.

Lühidalt, erinevalt eelkirjeldatud tava-bootstrapist simuleeritakse Bayesi bootstrapis posterioorjaotused, näiteks arvutatakse kaalutud keskmine, kus ühtlasest jaotusest pärit kaalud on prioriks.

Näited sellest, kuidas kasutada bayesbooti standardhälbe, korrelatsioonikoefitsiendi ja lineaarse mudeli koefitsientide usalduspiiride arvutamiseks leiate `?bayesboot` käsuga.




```
heights_bb <- bayesboot(heights$value, mean)
plot(heights_bb, compVal = 185)
```

```
HPDI(heights_bb$V1, prob = 0.95)
```

```
##    |0.95    0.95|
## 182.6958 187.1220
```

Vaikimisi pannakse `bayesboot()` funktsioonis statistiku arvutamisel kaalud (prior) valimi indeksile, mis annab erineva tulemuse kui näiteks kaalutud keskmise arvutamisel, kus kaalud (prior) pannakse valimi väärtustele.

Aritmeetilise keskmise Bayesi bootstrap väärtused kasutades kaalutud keskmise funktsiooni `weighted.mean` saab niimoodi:

```
heights_bb_w <- bayesboot(heights$value,
                          weighted.mean,
```

```
                        use.weights = TRUE)
```

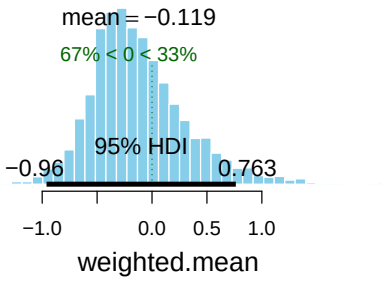
Tõenäosus, et keskmine on suurem kui 182 cm

```
mean(heights_bb[, 1] > 182)
```

```
## [1] 0.99125
```

Kahe keskväärtuse erinevus (ES = keskmine1 - keskmine2):

```
set.seed(1)

## Simulate two random normal distributions with mean 0.

## True difference is 0.
dfr <- tibble(a = rnorm(10, 0, 1), b = rnorm(10, 0, 1), c = a - b)
dfr_bb <- bayesboot(dfr$c, weighted.mean, use.weights = TRUE)
plot(dfr_bb, compVal = 0)
```



BayesianFirstAid raamatukogu funktsioon bayes.t.test() annab kasutades t-jaotuse tõepäramudelit üsna täpselt sama vastuse. See raamatukogu eeldab JAGS mcmc sämpleri installeerimist. Abi saab siit `https://github.com/rasmusab/bayesian_first_aid` ja siit `https://faculty.washington.edu/jmiyamot/p548/installing.jags.pdf`.

Parameetriline bootstrap

Kui me arvame, et me teame, mis jaotusega on meie andmed, ja meil on suhteliselt vähe andmepunkte, võib olla mõistlik lisada bootstrapile andmete jaotuse mudel. Näiteks, meie USA presidentide pikkused võiksid olla umbkaudu normaaljaotusega (sest me teame, et USA meeste pikkused on seda). Seega fitime kõigepealt presidentide pikkusandmetega normaaljaotuse ja seejärel tõmbame bootsrap valimid sellest normaaljaotuse mudelist. Normaaljaotuse mudelil on 2 parameetrit: keskmine (mu) ja standardhälve (sigma), mida saame fittida valimiandmete põhjal:

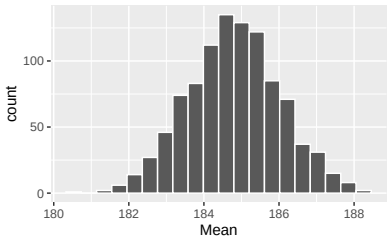
(ref:paramboot) Parameetrilise bootstrapi posteerior USA presidentide keskmisele pikkusele.

```
mu <- mean(heights$value)
sigma <- sd(heights$value)
N <- length(heights$value)
sample_means <- tibble(value = rnorm(N * 1000, mu, sigma),

                       indeks = rep(1:1000, each = N))
```

```
sample_means_sum <- sample_means %>%

  group_by(indeks) %>%

  summarise(Mean = mean(value))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(sample_means_sum, aes(x = Mean)) +

  geom_histogram(color = "white", bins = 20)
```




```
HPDI(sample_means_sum$Mean)
```

```
##    |0.89    0.89|
## 182.7899 186.7420
```

Üldiselt ei soovita me parameetrilist bootstrappi väga soojalt, sest täisbayesiaanlik alternatiiv, mida me kohe õppima asume, on sellest paindlikum.

Bootstrappimine ei ole kogu tõde

Bootstrappimine on võimas ja väga laia kasutusalaga meetodite kogum. Sellel on siiski üks oluline puudus. Nimelt ignoreerib see taustateadmisi (parameetriline bootstrap küll eeldab taustateadmisetele tuginevalt jaotusmudelit, kuid ignoreerib kogu muud taustateadmist). Miks on see probleem?

Mõtleme hetkeks sellele teadusliku meetodi osale, millel põhineb suuresti näiteks Darwini liikide tekkimise argument. See on nn *inference to the best explanation*, mille kohaselt on eelistatud see teooria, mis on parimas kooskõlas faktidega, ehk mille kehtimise korral on meie andmete esinemine kõige tõenäolisem. Kui mõni hüpotees omistab andmete esinemisele suure tõenäosuse, siis me ütleme tehnilises keeles, et see hüpotees on tõepärane (*has high likelihood*). Esmapilgul tundub see kõik igati mõistlik, kuid proovime lihtsat mõtteeksperimenti, kus loteriil võidab peaauhinna meile tundmatu kodanik Franz K. Meil on selle fakti seletamiseks kaks teooriat: 1. Franz K. võit oli juhuslik (loterii oli aus ja keegi peab ju võitma) ja 2. Franz K. noorem õde võltsis loterii tulemusi oma venna kasuks. Teine teooria sobib andmetega palju paremini kui esimene (sest kuigi keegi peab võitma, Franz K. võiduvõimalus oli ausal loteriil väga väike); aga ometi eelistab mõistlik inimene esimest teooriat, sest meil pole põhjust arvata, et Franz K.-l üldse on noorem õde, või et see õde omaks ligipääsu

loteriile. Ja kohe kui saame teada, et Franz K. on noorem õde, kes korraldab loteriid, leiame et asi on kahtlane.

Siit näeme, et lisaks tõepärale on selleks, et me usuksime mõne teooria kehtimisse, vaja veel, et see teooria oleks piisavalt tõenäoline meie taustateadmiste valguses. Bayesi teoreem ei tee muud, kui arvutab teooria kehtimise posterioorse tõenäosuse (järeltõenäosuse), kasutades selleks meie eelteadmiste ja tõepära kvantitatiivseid mudeleid. Seega, Bayesi paradigmas ei arvesta me mitte ainult andmetega, vaid ka taustateadmistega, sünteesides need kokku üheks posterioorseks jaotuseks ehk järeljaotuseks. Selle jaotuse arvutamine erineb bootstrapist, kuid tema tõlgendus ja praktiline töö sellega on sarnane. Erinevalt tavapärasest bootstrapist on Bayes parameetriline meetod, mis sõltub andmete modelleerimisest (normaaljaotus, t jaotus jne) ning sellest sõltumatust eelteadmise modelleerimisest.

Ehkki bootstrappimine ei arvesta taustateadmistega, ei tee seda olulisel määral ka paljud Bayesi mudelid (mudeldaja vaba valiku tõttu, mitte selle pärast, et mudel ei suudaks taustainfot inkorporeerida). Bayesi meetodite väljatöötajad ei tea sageli ette, milliste teaduslike probleemide lahendamiseks nende mudeleid hakatakse kasutama, ja seega ei kirjuta nad mudelisse ka väga ranget eelteadmist. Nende mudelite teadlastest kasutajad lepivad sageli selllega ja lasevad oma mudelite kaudu “andmetel kõneleda” enam-vähem sellistena, nagu need juhtuvad olema. Sellist lähenemist ei saa alati hukka mõista, sest vahest ei olegi meil palju eelteadmisi oma probleemi kohta, küll aga tuleb mainida, et sellistel juhtudel annab bootstrappimine sageli lihtsama vaevaga väga sarnase tulemuse kui Bayesi täismäng.

Bootstrapil on mõned imelikud formaalsed eeldused: 1. väärtused, mis ei esine valimiandmetes, on võimatud, 2. Väärtused, mis esinevad väljaspool valimi väärtuste vahemikku, on võimatud, 3. andmetes ei esine ajasõltuvusi ega hierarhilisi struktuure. Nendest puudustest hoolimata kasutatakse bootstrappimist laialt ja edukalt — eelkõige tema lihtsuse ja paindlikuse tõttu. Küll aga tähendab see, et bootstrap on harva parim võimalik meetod mingi ülesande lahendamiseks.

Bayesi põhimõte

Bayesi arvutuseks on meil vaja teada

1) milline on "*parameter space*" ehk parameetriruum? Parameetriruum koosneb kõikidest loogiliselt võimalikest parameetriväärtustest. Näiteks kui me viskame ühe korra münti, koosneb parameetriruum kahest elemendist: 0 ja 1, ehk null kulli ja üks kull. See ammendab võimalike sündmuste nimekirja. Kui me aga hindame mõnd pidevat suurust (keskmine pikkus, tõenäosus 0 ja 1 vahel jms), koosneb parameetriruum lõpmata paljudest arvudest.

2) milline on "*likelihood function*" ehk tõepärafunktsioon ehk andmemudel? Me omistame igale parameetriruumi elemendile (igale võimalikule parameetri väärtusele) tõepära. Tõepära parameetri väärtusel x on tõenäosus, millega me võiksime kohata oma andmeid, juhul kui x oleks see ainus päris õige parameetri väärtus. Teisisõnu, tõepära on kooskõla määr meie andmete ja parameetri väärtuse x vahel. Kuna meil on vaja modeleerida tõepära igal võimalikul parameetri väärtusel (mida pideva suuruse puhul on lõpmatu hulk), siis kujutame tõepära pideva funktsioonina, mis katab parameetriruumi. Tõepärafunktsioon ei summeeru/integreeru ühele – see ei ole tõenäosusfunktsioon. Sageli mudeldame tõepära normaaljaotusena (pidevate andmete korral) või binoomjaotusena (binaarsete andmete korral) või Poissoni jaotusena (countide korral) või lognormaaljaotusena (positiivsete pidevate ja parempoolse paksu õlaga andmete korral) või studenti t jotusena (robustne mudel pidevatele andmetele) või negatiivse binoomjaotusena (üledisperseeritud Poissoni protsessid) või eksponentsiaalse jaotusena (Poissoni tüüpi juhuslike sündmuste vaheline aeg).

3) milline on "*prior function*" ehk prior ehk eeljaotus? Igale tõepära väärtusele peab vastama priori väärtus. Seega, kui tõepära on modelleeritud pideva funktsioonina, siis on ka prior pidev funktsioon (aga prior ei pea olema sama tüüpi funktsioon, kui tõepära). Erinevus tõepära ja priori vahel seisneb selles, et kui tõepärafunktsioon annab just meie andmete keskväärtuse tõenäosuse

> igal parameetriväärtusel, siis prior annab iga parameetriväärtuse tõenäosuse, sõltumata meie andmetest. See-eest arvestab prior kõikide teiste relevantsete andmetega, sünteesides taustateadmised ühte tõenäousmudelisse. Me omistame igale parameetriruumi väärtusele eelneva tõenäosuse, et just see väärtus on üks ja ainus tõene väärtus. Priori jaotus summeerub 1-le. Prior kajastab meie arvamust, kui suure tõenäosusega on just see parameetri väärtus tõene; ehk seda, mida me usume enne oma andmete nägemist. Nendel parameetri väärtustel, kus prior (või tõepära) = 0, on ka posteerior garanteeritult 0. See tähendab, et kui te olete 100% kindel, et mingi sündmus on võimatu, siis ei saa ka mäekõrgune hunnik uusi andmeid seda uskumust muuta (eelduselt, et te olete ratsionaalne inimene). ("`http://optics.eee.nottingham.ac.uk/match/uncertainty.php` aitab praktikas priorit modelleerida (proovige *Roulette* meetodit).")

Edasi on lihtne. Arvuti võtab tõepärafunktsiooni ja priori, korrutab need üksteisega läbi ning seejärel normaliseerib saadud jaotuse nii, et jaotusealune pindala võrdub ühega. Saadud tõenäosusjaotus ongi posteeriorne jaotus ehk posteerior ehk järeljaotus. Kogu lugu.

Me teame juba pool sajandit, et Bayesi teoreem on sellisele ülesandele parim võimalik lahendus. Lihtsamad ülesanded lahendame selle abil täiuslikult. Kuna parameetrite arvu kasvuga mudelis muutub Bayesi teoreemi läbiarvutamine eksponentsiaalselt arvutusmahukamaks (sest läbi tuleb arvutada mudeli kõikide parameetrite kõikide väärtuste kõikvõimalikud kombinatsioonid), oleme sunnitud keerulisemad ülesanded lahendama umbkaudu, asendades Bayesi teoreemi MCMC algoritmiga, mis teie arvutis peituva propelleri Karlsoni kombel lendu saadab, et tõmmata valim "otse" posterioorsest jaotusest. Meie poolt kasutatava MCMC *Hamiltonian Monte Carlo* mootori nimi on Stan (www.mc-stan.org). See on eraldiseisev programm, millel on R-i liides R-i pakettide rstan, rethinking, rstanarm, brms jt kaudu. Meie töötame ka edaspidi R-s, mis automaatselt suunab meie mudelid ja muud andmed Stani, kus need läbi arvutatakse ja seejärel tulemused R-i tagasi saadetakse. Tulemuste töötlus ja graafiline esitus toimub jällegi R-is.

Alustame lihtsa näitega, mida saab käsitsi läbi arvutada.

Esimene näide

Me teame, et suremus haigusesse on 50% ja meil on palatis 3 patsienti, kes seda haigust põevad. Seega on meil kaks andmetükki (50% ja N=3). Küsimus: mitu meie patsienti oodatavalt hinge heidavad? Eeldusel, et patsiendid surevad üksteisest sõltumatult, on meil mün-

diviske olukord. Parameetriruum on neljaliikmeline: 0 surnud, 1 surnud, 2 surnud ja 3 surnud. Edasi loeme üles kõik võimalikud sündmusteahelad, mis loogiliselt saavad juhtuda.

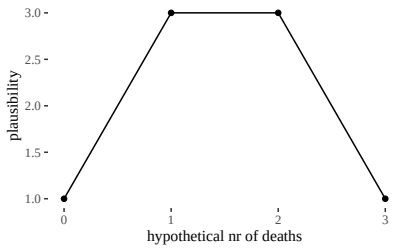
Me viskame kulli-kirja 3 korda: kiri = elus, kull = surnud. Võimalikud sündmused on: | kull kull kull | kull kiri kull | kiri kull kull | kull kull kiri | kull kiri kiri | kiri kiri kull | kiri kull kiri | kiri kiri kiri

Kui P(kull) = 0.5, siis lugedes kokku kõik võimalikud sündmused:

- 0 kulli ehk surnud - 1,
- 1 kulli ehk surnud - 3,
- 2 kulli ehk surnud - 3,
- 3 kulli ehk surnud - 1

Nüüd teame parameetriruumi iga liikme kohta, kui suure tõenäosusega me ootame selle realiseerumist. Näiteks, P(0 surnud) = 1/8, P(1 surnud) = 3/8, P(1 või 2 surnud) = 6/8 jne. Selle teadmise tõlgime tõepärafunktsiooniks.

```
# Parameter space: all possible futures
x <- seq(from = 0, to = 3)

# Likelihoods for each x value, or P(deaths I x)
y <- c(1, 3, 3, 1)

ggplot(data = NULL, aes(x, y)) +
geom_point() +
geom_line() +
xlab("hypothetical nr of deaths") +
ylab("plausibility") + ggthemes::theme_tufte()
```



Siit näeme, et üks surm ja kaks surma on sama tõenäolised ja üks surm on kolm korda tõenäolisem kui null surma (või kolm surma). Tõepära annab meile tõenäosuse Pr(mortality=0.5 & N=3) igale loogiliselt võimalikule surmade arvule (0 kuni 3).

Me saame sama tulemuse kasutades binoomjaotuse mudelit. Ainus erinevus on, et nüüd on meil y teljel surmade tõenäosus.

Sama asja saab põhjendada veidi matemaatilisemalt Bernoulli- ja binoomjaotus:

- katse γ võib omada kahte tulemust: 0 ja 1.
- parameeter θ on tõenäosus saada katse tulemusel 1. Seega jääb selle parameetri väärtus 0 ja 1 vahele.

- $P(\gamma = 1|\theta) = \theta$ (tõenäosus, et teeta teatud väärtusel on katse tulemus üks, on teeta.)
- $P(\gamma = 0|\theta) = 1 - \theta$ (vastastõenäosus katse tulemusele null)
- eelnevate valemite kombineerimine annab Bernoulli jaotuse: $P(\gamma|\theta) = \theta^{\gamma}(1-\theta)^{(1-\gamma)}$.

Selle jaotusega saab töötada kahel erineval viisil.

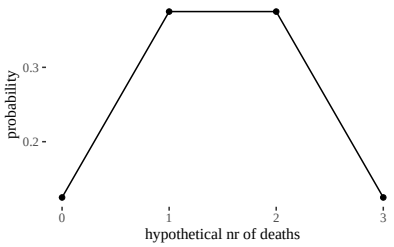
1) Me fikseerime teeta väärtuse andmepunktina ja küsime, millised on oodatavad gamma väärtused. Gamma on siin parameeter, mille saab fittida kahele erinevale väärtusele.

2) Me fikseerime gamma andmetena (näiteks gamma = 1) ja küsime, kui hästi on see gamma väärtus kooskõlas kõikide võimalike teeta väärtustega 0 ja 1 vahel. Nüüd on teeta parameeter, millel on lõpmata palju võimalikke väärtusi 0 ja 1 vahel (parameetriruum). Erinevad teeta väärtused annavad andmepunkt gammale erinevad tõenäosused, mis aga ei summeeru ühele. Selles vaates annab bernoulli jaotus meile Bernoulli tõepärafunktsiooni.

Bernoulli funktsiooni laiendus juhule, kus meil on z katsetulemust, mida tähistasime ühega, ja kokku N katset, on $\theta^{z}(1-\theta)^{N-z}$, mis on binoomjaotus. Binoomjaotus on lihtsalt tuletatav Bernoulli jaotusest.

Joonistame nüüd tõepära juhule, kus me fikeerime N-i ja teeta andmetena (vastavalt väärtustena 3 ja 0.5) ja käsitleme z-i parameetrina, millel on 4 võimalikku väärtust (prameetriruum: 0, 1, 2, 3 surma).

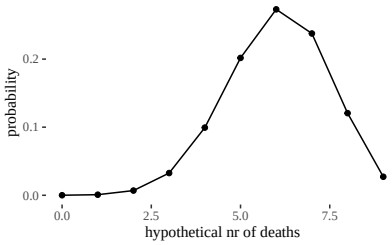
```
z <- seq(from = 0, to = 3)
y <- dbinom(x = z, 3, 0.5)

ggplot(data = NULL, aes(z, y)) +
geom_point() +
geom_line() +
xlab("hypothetical nr of deaths") +
ylab("probability") + ggthemes::theme_tufte()
```



Proovime seda koodi olukorras, kus meil on 9 patsienti ja suremus on 0.67:

```
z <- seq(from = 0, to = 9)
y <- dbinom(x = z, 9, 0.67)
```

```
ggplot(data = NULL, aes(z, y)) +
geom_point() +
geom_line() +
xlab("hypothetical nr of deaths") +
ylab("probability") + ggthemes::theme_tufte()
```

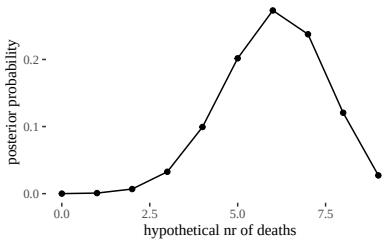


Lisame sellele tõepärafunktsioonile tasase priori (lihtsuse huvides) ja arvutame posterioorse jaotuse kasutades Bayesi teoreemi. Igale parameetri väärtusele on tõepära * prior proportsionaalne posterioorse tõenäosusega, et just see parameetri väärtus on see ainus tõene väärtus. Posterioorsed tõenäosused normaliseeritakse nii, et nad summeeruksid 1-le.

Me defineerime X telje kui rea 10-st arvust (0 kuni 9 surma) ja arvutame tõepära igale neist 10-st arvust. Sellega ammendame me kõik loogiliselt võimalikud parameetri väärtused.

(ref:posterior) Posteerior.

```
x <- seq(from = 0, to = 9)
# flat prior
prior <- rep(1, 10)

# Compute likelihood at each value in grid
likelihood <- dbinom(x, size = 9, prob = 0.67)

# Compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# Normalize the posterior, so that it sums to 1
posterior <- unstd.posterior/sum(unstd.posterior)

ggplot(data = NULL, aes(x, posterior)) +
geom_point() +
geom_line() +
xlab("hypothetical nr of deaths") +
ylab("posterior probability") + ggthemes::theme_tufte()
```



See on parim võimalik teadmine, mitu kirstu tasuks tellida, arvestades meie priori ja likelihoodi mudelitega. Näiteks, sedapalju, kui surmad ei ole üksteisest sõltumatud, on meie tõepäramudel (binoomjaotus) vale.

Teine näide: sõnastame oma probleemi ümber

Mis siis, kui me ei tea suremust ja tahaksime seda välja arvutada? Kõik, mida me teame on, et 6 patsienti 9st surid. Nüüd koosnevad

andmed 9 patsiendi mortaalsusinfost (parameeter, mille väärtust me eelmises näites arvutasime) ja parameeter, mille väärtust me ei tea, on surmade üldine sagedus (see parameeter oli eelmises näites fikseeritud, ja seega kuulus andmete hulka).

Seega on meil:

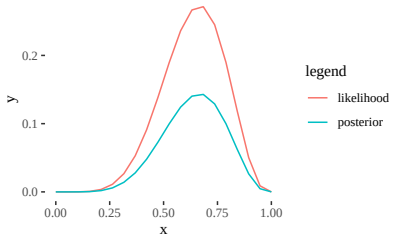
1. Parameetriruum 0% kuni 100% suremus (ost 1-ni), mis sisaldab lõpmata palju numbreid.
2. Kaks võimalikku sündmust (surnud, elus), seega binoomjaotusega modelleeritud tõepärafunktsioon. Nagu me juba teame, on r funktsioonis dbinom() kolm argumenti: surmade arv, patsientide koguarv ja surmade tõenäosus. Seekord oleme me fikseerinud esimesed kaks ja soovime arvutada kolmanda väärtuse.
3. Tasane prior, mis ulatub 0 ja 1 vahel. Me valisime selle priori selleks, et mitte muuta tõepärafunktsiooni kuju. See ei tähenda, et me arvaksime, et tasane prior on mitteinformatiivne. Tasane prior tähendab, et me usume, et suremuse kõik väärtused 0 ja 1 vahel on võrdselt tõenäolised. See on vägagi informatsioonirohke (ebatavaline) viis maailma näha, ükskõik mis haiguse puhul!

Tõepära parameetri väärtusel x on tõenäosus kohata meie andmeid juhul, kui x on juhtumisi parameetri tegelik väärtus. Meie näites koosneb tõepärafunktsioon tõenäosustest, et kuus üheksast patsiendist surid igal võimalikul suremuse väärtusel (0…1). Kuna see on lõpmatu rida, teeme natuke sohki ja arvutame tõepära 20-l valitud suremuse väärtusel.

Tehniliselt on sinu andmete tõepärafunktsioon agregeeritud iga üksiku andmepunkti tõepärafunktsioonist. Seega vaatab Bayes igat andmepunkti eraldi (andmete sisestamise järjekord ei loe).

```
# mortality at 20 evenly spaced probabilities from 0 to 1
theta <- seq(from = 0, to = 1, length.out = 20)

# Define prior
prior <- rep(1, 20)

# Compute likelihood at each value in grid
likelihood <- dbinom(6, size = 9, prob = theta)

# Compute product of likelihood and prior & standardize the posterior
posterior <- likelihood * prior/sum(likelihood * prior)

# put everithing into a tibble for plotting
a <- tibble(x = rep(x = theta, 2), y = c(likelihood, posterior), legend = rep(c("likelihood",
    "posterior"), each = 20))

ggplot(data = a) + geom_line(aes(x, y, color = legend)) +
ggthemes::theme_tufte()
```

Nüüd on meil posterioorne tõenäosusfunktsioon, mis summeerub 1-le ja mis sisaldab kogu meie teadmist suremuse kohta. Alati on kasulik plottida kõik kolm funktsiooni (tõepära, prior ja posteerior).



Kui n = 1

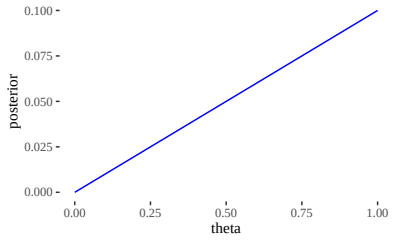
Bayes on lahe sest tema hinnangud väiksele N-le on loogiliselt sama pädevad kui suurele N-le. See ei ole nii klassikalises sageduslikus statistikas, kus paljud testid on välja töötatud N = Inf eeldusel ja töötavad halvasti väikeste valimitega.

Hea küll, me arvutame jälle suremust.

Bayes töötab andmepunkti kaupa (see et me talle ennist kõik andmed korraga ette andsime, on puhtalt mugavuse pärast).

N=1, esimene patsient suri.

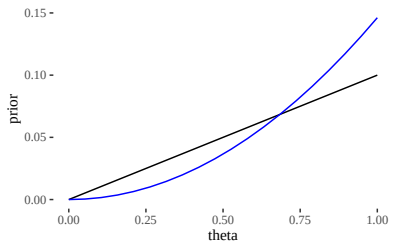
```
prior <- rep(1, 20)
likelihood <- dbinom(1, size = 1, prob = theta)
posterior <- likelihood * prior/sum(likelihood * prior)
ggplot(data = NULL) +
geom_line(aes(theta, posterior), color = "blue") +
ggthemes::theme_tufte()
```



Esimene patsient suri - 0 mortaalsus ei ole enam loogiliselt võimalik (välja arvatud siis kui prior selle koha peal = 0) ja mortaalsus 100% on andmetega (tegelikult andmega) parimini kooskõlas. Posteerior on nulli ja 100% vahel sirge sest vähene sissepandud informatsioon lihtsalt ei võimalda enamat.

N=2, teine patsient suri.

```
prior <- posterior
likelihood <- dbinom(1, size = 1, prob = theta)
posterior1 <- likelihood * prior/sum(likelihood * prior)
ggplot(data = NULL) +
geom_line(aes(theta, prior)) +
geom_line(aes(theta, posterior1), color = "blue") +
ggthemes::theme_tufte()
```

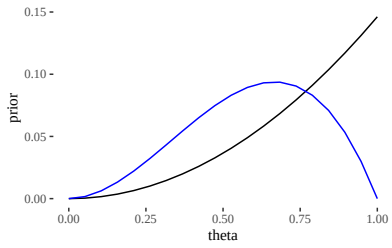


Teine patsient suri. Nüüd ei ole 0 ja 1 vahel enam sirge posteerior. Posteerior on kaldu 100 protsendi poole, mis on ikka kõige tõenäolisem väärtus.

N=3, kolmas patsient jäi ellu.

```
prior <- posterior1
likelihood <- dbinom(0, size = 1, prob = theta)
posterior2 <- likelihood * prior/sum(likelihood * prior)
```

```
ggplot(data = NULL) +
geom_line(aes(theta, prior)) +
geom_line(aes(theta, posterior2), color = "blue") +
ggthemes::theme_tufte()
```

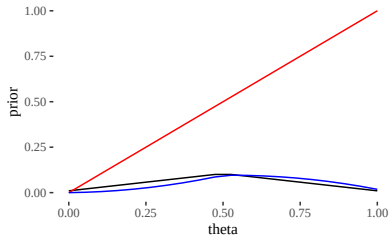


Kolmas patsient jäi ellu - 0 ja 100% mortaalsus on seega võimaluste nimekirjast maas ning suremus on ikka kaldu valimi keskmise poole (75%).

Teeme sedasama prioriga, mis ei ole tasane, illustreerimaks tõsiasja, et kui N on väike, siis omab prior suurt tähtsust posteeriori kuju määramisel.

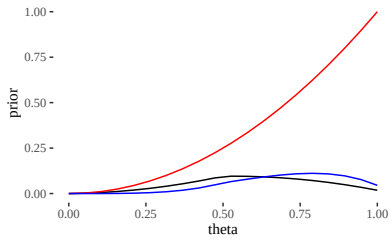
N=1 informatiivse prioriga (1. patsient suri).

```
prior <- c(seq(0.01, 0.1, length.out = 10), seq(0.1, 0.01, length.out = 10))
likelihood <- dbinom(1, size = 1, prob = theta)
posterior <- likelihood * prior/sum(likelihood * prior)
ggplot(data = NULL) +
geom_line(aes(theta, prior)) +
geom_line(aes(theta, likelihood), color = "red") +
geom_line(aes(theta, posterior), color = "blue") +
ggthemes::theme_tufte()
```



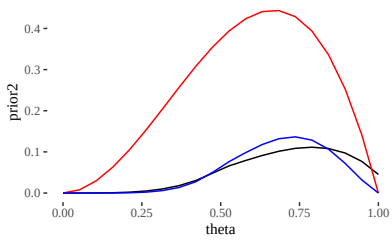
N=2 informatiivse prioriga (2. patsient suri).

```
prior <- posterior
likelihood <- dbinom(2, size = 2, prob = theta)
posterior1 <- likelihood * prior/sum(likelihood * prior)
ggplot(data = NULL) +
geom_line(aes(theta, prior)) +
geom_line(aes(theta, likelihood), color = "red") +
geom_line(aes(theta, posterior1), color = "blue") +
ggthemes::theme_tufte()
```



N=3 informatiivse prioriga (3. patsient jäi ellu). Nüüd on posteeriori tipp mitte 75% juures nagu ennist, vaid kuskil 50% juures — tänu priorile.

```
prior2 <- posterior1
likelihood <- dbinom(2, size = 3, prob = theta)
posterior2 <- likelihood * prior2/sum(likelihood * prior2)
ggplot(data = NULL) +
geom_line(aes(theta, prior2)) +
geom_line(aes(theta, likelihood), color = "red") +
geom_line(aes(theta, posterior2), color = "blue") +
ggthemes::theme_tufte()
```



Jaotusmudelid andmetele ja prioritele

Lihtne normaaljaotusega varieeruvuse mudel

Oletame, et me oleme mõõtnud nelja patsienti ja saanud tulemuseks 1.2, 2.12, 1.4 ja 8.34. Kuidas me oma valimit iseloomustame ja kas me peaksime 4. tulemuse, kui kahtlase, välja viskama? Arvatavasti tahaksime saada hinnangut kõige tõenäolisemale mõõtetulemusele patsientide populatsioonis, ehk siis keskmise või tüüpilise patsiendi väärtusele. Ja lisaks ka hinnangut patsientide vahelise varieeruvuse määrale. Meid võib ka huvitada võrrelda patsientide ja tervete inimeste varieeruvust. Esmapilgul tundub see lihtsa ülesandena, mis ei vaja mudeldamist – lihtsalt arvutame aritmeetilise keskmise ja standardhälbe ja meil on mõlemad hinnangud olemas. Aga tegelikult oleme probleemi ees, millele pole ühte õiget lahendust.

Kui me viskame 4. tulemuse välja, siis tuleb meie keskmine kuhugi 1.5 kanti, muidu aga piirkonda, mille lähedal meil ei ole ühtegi andmepunkti. Samuti annaks sd arvutus üsna erinevad tulemused. Kumb võimalus siis valida? Selleks peame ikkagi otsustama, kuidas modelleerida oma andmed. Sõltuvalt looduslikust protsessist, mis need andmed genereeris, võiks andmete mudel olla näiteks normaaljaotus, lognormaaljaotus vms. Kui valime normaaljaotuse, millise õlad langevad väga kiiresti, siis on vaid väike tõenäosus kohata tervelt veerandit oma andmepunktidest nõnda kaugel teistest, mis annab argumendi selle punkti eemaldamiseks. Aga lognormaaljaotuse korral, mille õlg laskub palju aeglasemalt, on tõenäosus 4. mõõtmisest isegi kaugemal olevaid andmeid kohata palju suurem ja seega peaksime selle andmepunkti sisse jätma. Erinevat tüüpi mudelitel on erinevad parameetrid, millele saab andmete põhjal väärtusi otsida. See, et normaaljaotuse parameetrit μ saab meie näites arvutada aritmeetilise keskmise kaudu, ei tähenda, et ka teiste mudelite korral peaksime sama lokatsiooniparameetrit fittima (või et neil mudelitel üldse oleks lokatsiooniparameeter). Sarnased lood on muidugi ka varieeruvust iseloomustava parameetriga.

Statistilist mudelit saab kasutada mitmel moel.

1. Mudel toob sisse lisainformatsiooni andmete jaotuse kuju kohta, mis tõstab järelduste kvaliteeti (või langetab seda, kui valisime kehva mudeli). Seega see tõstab kunstlikult andmemahtu (valimi suurust).

2. Võrreldes erinevat tüüpi mudelite sobivust andmetega ning omades aimu protsesside kohta, mida üks või teine mudel võiks adekvaatselt kirjeldada, on võimalik teha järeldusi mehhanismi kohta, mis genereeris andmed, mille põhjal mudelid fititi.

3. Me võime fititud mudeli põhjal teha ennustusi, ehk genereerida uusi andmeid *in silico*.

Selle mehhanismi alla mahuvad nii looduslikud protsessid, mida kirjeldavad meie teaduslikud teooriad, kui katsesüsteemi tehnilised eripärad nagu mõõtmistäpsus ja -kallutatus.

Niisiis lihtne mudel andmetele: μ ehk aritmeetiline keskmine kui hinnang kõige tõenäosemale väärtusele. See on deterministlik nn *protsessimudel*, kus samad valimiväärtused annavad alati sama ja ühese tulemuse. Statistiline mudel sisaldab endas nii protsessimudelit kui tõenäosuslikku nn *varieeruvuse mudelit* (ajaloolistel põhjustel kutsutakse seda sageli veamudeliks), mis tuleb sisse tõenäosusjaotuse kujul

$$dnorm(\mu, \sigma)$$

Selle mudeli on võimalik ümber sõnastada (seda seeläbi üldistades) lihtsa regressioonivõrrandina $y = b_0$, kusjuures $\mu = b_0$ ehk andmete keskväärtus võrdub regressioonisirge interceptiga. Asendades saame

$$y \sim dnorm(b_0, \sigma)$$

Tilde $\sim$ tähistab seose tõenäosuslikkust, ehk seda, et y muutuja ennustuslikd väärtused tõmmatakse juhuvalimina normaaljaotusest, mis omakorda on fititud empiiriliste väärtuste (ehk valimi) põhjal.

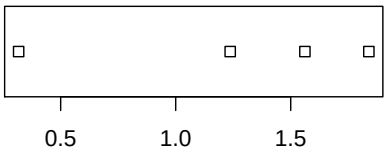
Seega on meil normaaljaotuse keskväärtus võimalik leida aritmeetilise keskmisena või samaväärselt vähimruutude meetodiga, mis paneb keskväärtuse kohta, kus keskväärtuse ja iga andmepunkti vahelise kauguste ruutude summa tuleb minimaalne. Vähimruutude meetod on üldisem, sest töötab ka juhul kui asendame μ regressioonivõrrandiga $\mu = b_0 + b_1x_1 + b_2x_2 + ... + b_ix_i$ (protsessimudel). Ja kui meie regressioonivõrrandid lähevad mittelineaarseks ja vähimruutude meetod nende fittimisel enam ei tööta, siis veelgi üldisem meetod, Bayesi teoreem, töötab ikka.

Kuigi aritmeetiline keskmine ja vähimruutude meetod annavad sama hinnangu lokatsiooniparameetrile, ei ütle need midagi sigma kohta. Samas Bayesi meetod annab hinnangu (koos usaldusintervalliga) mõlemale parameetrile.

```
Normaaljaotus mudeldab lokalisatsiooniparameetrit mu populatsiooni
tüüpilise või keskmise liikme hinnanguna ja varieeruvusparameetrit
sigma populatsiooni liikmete vaheliste erinevuste määra hinnanguna.
```

Arvutame lihtsa mudeli läbi vähimruutude meetodiga ja Bayesi meetodiga

```
set.seed(1234321)
andmed <- tibble(a = rnorm(4))
plot(andmed)
```




```
mean(andmed$a) %>% round(2)
```

```
## [1] 1.24
```

```
sd(andmed$a) %>% round(2)
```

```
## [1] 0.66
```

Vähimruutude meetodit rakendab lm() funktsioon

```
lm(a ~ 1, data = andmed)
```

term	estimate	std.error	statistic	p.value
(Intercept)	1.24	0.33	3.74	0.03

Ja Bayesi meetodil kasutades brms::brm() funktsiooni

```
brm(a ~ 1, data = andmed)
```

term	estimate	std.error	lower	upper
b_Intercept	1.24	0.68	0.20	2.25
sigma	1.27	1.24	0.46	3.02

Intercept asendab mu-d samamoodi nagu Juku asendab Juhanit.

Nagu näha, lm() fitib ainult mu parameetri, samas kui me Bayesi meetodit kasutades saame hinnangu (koos usalduspiiridega) kahele parameetrile: mu ehk intercept ja sigma ehk sd.

Meie poolt simuleeritud andmed tulevad normaaljaotusega populatsioonist, mille mu = 0 ja sd = 1. Kumbki meetod ei luba meile null-intercepti sest andmeid on vähe ja need on juhusliku valimivea tõttu kallutatud. See-eest sigma hinnang, mille Bayes meile annab on küll laiavõitu (ikka sellepärast, et meil on vähe andmeid), aga vähemalt hõlmab see endas õiget väärtust, ja mu hinnang ei jää tõest liiga kaugele.

Oletame, et me teame ette, et mu tegelik populatsiooniväärtus jääb suure tõenäosusega -0.2 ja 0.2 vahele, ja on kõige tõenäolisemalt 0. Ning, et sigma jääb tõenäoliselt 0.8 ja 1.2 vahele ning on kõige tõenäolisemalt 1. Seda teadmist saab inkorporeerida bayesi mudelisse nii

```
brm(a ~ 1, data = andmed, prior = c(prior(normal(0, 0.1), class = "Intercept"), prior(normal(1,
    0.1), class = "sigma")))
```

term	estimate	std.error	lower	upper
b_Intercept	0.05	0.10	-0.11	0.21
sigma	1.03	0.09	0.88	1.18

Nüüd domineerib eelnev informatsioon, mis on sisestatud kitsaste prioritena, nelja andmepunkti, mis nihutavad priorite väärtusi vaid õige pisut. Tabelis tähistavad "lower" ja "upper" 95% usalduspiire (credible intervals), mis mu ja sigma prioris vastaksid u kahele priori sd-le. Seega on andmed meie prioreid u kaks korda kitsendanud ja posteerior kirjeldab mu ja sigma tegelikku väärtust paremini kui prior. Sedamõõda kuidas andmeid juurde tuleb (või me priorid laiemaks teeme), hakkavad andmed domineerima priorite üle ja üsna varsti pole suurt vahet, milliseid prioreid me kasutame - posteeriorid tulevad ikka peaaegu samad.

Protsessimudel ja varieeruvuse mudel lineaarses regressioonis

Kui mudel $kaal = b_0 + b_1 \times pikkus$ ennustab, et 160 cm inimene kaalub keskmiselt 80 kg, siis protsessimudel ei ütle, kui suurt pikkusest sõltumatut kaalude varieeruvust võime oodata 160 cm-ste inimeste hulgas. Selle hinnangu andmiseks tuleb mudelile lisada varieeruvusekomponent, sageli normaaljaotuse kujul, mis modelleerib üksikute inimeste kaalude varieeruvust (mitte keskmise kaalu varieeruvust) igal mõeldaval ja mittemõeldaval pikkusel.

> Bioloogid, erinevalt füüsikutest, usuvad, et valimisisene andmete varieeruvus on tingitud pigem bioloogilisest varieeruvusest kui mõõtmisveast, aga loomulikult sisaldub selles ka mõõtmisviga. Neid varieeruvuskomponente saab omavahel lahutada mitmetasemelistes mudelites.

Kuidas varieeruvus lineaarsesse mudelisse sisse tuua? Ilma varieeruvuskomponendita mudel:

$$y = b_0 + bx$$

ennustab y-i keskväärtust erinevatel x-i väärtustel.
Varieeruvuskomponent:

$$y \sim dnorm(\mu,\ \sigma)$$

kus μ (*mu*) on mudeli poolt ennustatud keskväärtus ja σ (sigma) on mudeli poolt ennustatud standardhälve ehk varieeruvus andmepunktide tasemel. Varieeruvusmudelis on keskväärtuse ehk *mu*

ennustus endiselt deterministlik ja sigma töötab originaalsel andmetasemel, mitte keskväärtuste tasemel. See võimaldab protsessimudeli varieeruvusmudelisse sisse kirjutada lihtsalt *mu* ümber defineerides:

$$\mu = b_0 + bx$$

mis tähendab, et

$$y \sim dnorm(b_0 + b_1 x,\ \sigma)$$

See ongi sirge mudel koos varieeruvuskomponendiga. Seega on sellel lineaarsel regressioonimudelil kolm parameetrit: intercept b_0, tõus b_1 ja varieeruvusparameeter σ. Sellist mudelit on mõistlik fittida Bayesi teoreemi abil, milleks peame defineerima igale parameetrile priori.

Seega koosneb selle mudeli täiskirjeldus, mis sisaldab kogu vajalikku infot selle mudeli struktuuri ja eelduste kohta, 4-5st reast. Näiteks:

$$y \sim normal(\mu,\ \sigma)$$

$$\mu = b_0 + b_1 x$$

$$b_0 \sim normal(0, 100)$$

$$b_1 \sim normal(0, 1)$$

$$\sigma \sim student(3, 0, 3)$$

Bayesi meetodiga fititud mudel näitab, millised kombinatsioonid nendest kolmest parameetrist usutavalt koos esinevad, ja millised mitte. Seega on fititud 3 parameetriga bayesi mudel 3-dimensionaalne tõenäosusjaotus (3D posteerior). Muidugi saame ka ükshaaval välja plottida kolm 1D posteeriori, millest igaüks iseloomustab üht parameetrit ning on kollapseeritud üle kahe ülejäänud parameetri. Edaspidi õpime selliste mudelitega töötama.

Kõik statistilised mudelid on tõenäosusmudelid ning sisaldavad varieeruvuskomponenti.

Kuna erinevalt lokatsiooniparameetrist, ei aja me mudelis sigmat lahku vastavalt x-i väärtustele, siis enamasti meie varieeruvusmudel modelleerib igale x-i väärtusele (kaalule) sama suure y-i suunalise varieeruvuse (pikkuste sd).

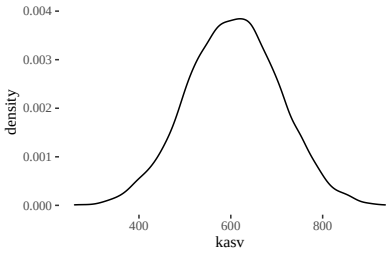
Enimkasutatud varieeruvusmudel on normaaljaotus

Alustuseks simuleerime lihtsate vahenditega looduslikku protsessi, mille tulemusel tekib normaaljaotus.

Oletame, et bakteri kasvukiirust mõjutavad 12 geeni, mille mõjud võivad olla väga erineva tugevusega, kuid mille mõjude suurused ei sõltu üksteisest. Seega nende 12 geeni mõjud kasvukiirusele liituvad. Järgnevas koodis võtame 12 juhuslikku arvu 1 ja 100 vahel (kasutades `runif()` funktsiooni). Need 12 arvu näitavad 12 erineva geeni individuaalsete mõjude suurusi bakteritüve kasvukiirusele. Meil on seega kuni 100-kordsed erinevused erinevate geenide mõjude suuruste vahel. Seejärel liidame need 12 arvu. Nüüd võtame uue 12-se valimi ja kordame eelnevat. Me teeme seda 10 000 korda järjest ja plotime saadud 10 000 arvu (10 000 liitmistehte tulemust) tihedusfuntksioonina.

Normaaljaotus tekib sõltumatutest efektidest. Kümne tuhande N = 12 suuruse juhuvalimi summa tihedusdiagramm.

```
kasv <- replicate(10000, sum(runif(12, 1, 100)))
p <- ggplot(tibble(kasv), aes(kasv)) +
geom_density() + ggthemes::theme_tufte()
p
```

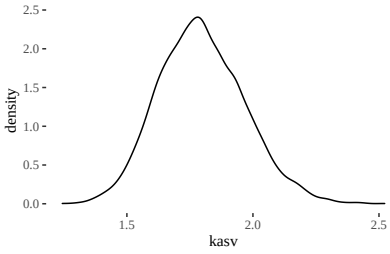


Selles näites võrdub iga andmepunkt 10 ooost ühe bakteritüve kasvukiiruse mõõtmisega. Seega, antud eelduste korral on bakteritüvede kasvukiirused normaaljaotusega.

Nüüd vaatame, mis juhtub, kui 12 geeni mõjud ei ole üksteisest sõltumatud. Kui 12 geeni on omavahel vastasmõjudes, siis nende geenide mõjud korrutuvad, mitte ei liitu. (Korrutamine pole ainus viis, kuidas vastasmõjusid modeleerida, küll aga kõige levinum.) Kõigepealt vaatleme juhtu, kus 12 geeni on kõik väikeste mõjudega ning seega mitte ühegi geeni mõju ei domineeri teiste üle. Seekord genreerime 12 juhuslikku arvu 1 ja 1.1 vahel. Siin tähendab arv 1.1 kasvu tõusu 10% võrra. Seejärel korrutame need 12 arvu, misjärel kordame eelnevat 10 000 korda.

Normaaljaotus tekib ka väikestest sõltuvatest efektidest. Kümne tuhande N = 12 suuruse juhuvalimi korrutiste tihedusdiagramm. Ühegi geeni mõju ei domineeri teiste üle.

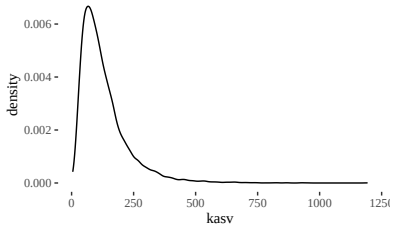
```
kasv <- replicate(10000, prod(runif(12, 1, 1.1)))
p %+% tibble(kasv)
```



Tulemuseks on jällegi normaaljaotus. Selles näites olid üksikud interakteeruvad geenid ükshaaval väikeste mõjudega ja ühegi geeni mõju ei domineerinud teiste üle. Mis juhtub, kui mõnel geenil on kuni 2 korda suurem mõju kui teisel?

Lognormaaljaotus tekib suurematest sõltuvatest efektidest. Kümne tuhande N = 12 suuruse juhuvalimi korrutiste tihedusdiagramm. Mõnel geenil on kuni 2 korda suurem mõju kui teisel.

```
kasv <- replicate(10000, prod(runif(12, 1, 2)))
p %+% tibble(kasv)
```

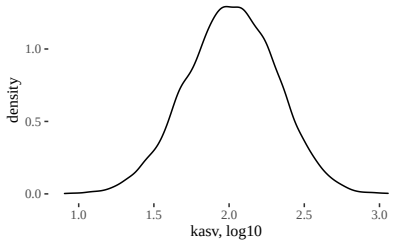


Nüüd on tulemuseks log-normaaljaotus. Mis teie arvate, kas teie poolt uuritavat tunnust mõjutavad faktorid, mis omavahel ei interakteeru või kui interakteeruvad, on kõik ühtlaselt väikeste efektidega? Või on tegu vastasmõjudes olevate faktoritega, millest osad on palju suuremate mõjudega, kui teised? Ühel juhul eelistate te normaaljaotust, teisel juhul peate õppima töötama ka lognormaaljaotusega.

Kui me vaatame samu andmeid logaritmilises skaalas, avastame, et need andmed on normaaljaotusega. See ongi andmete logaritmimise mõte.

Logaritmilises skaalas lognormaalsed efektid on normaaljaotusega. Kümne tuhande N = 12 suuruse juhuvalimi korrutiste tihedusdiagramm. Mõnel geenil on kuni 2 korda suurem mõju kui teisel.

```
kasv <- replicate(10000, log10(prod(runif(12, 1, 2))))
p %+% tibble(kasv) + labs(x = "kasv, log10")
```



Normaaljatuse avastas Gauss (1809), aga nime andis sellele Francis Galton (1860ndatel), kuna antropoloogilised mõõtmised "normaalselt" järgisid "vigade seadust", mille ta nimetas "Normaalseks jaotuste kurviks".

Normaaljaotuse mudel väikestel valimitel

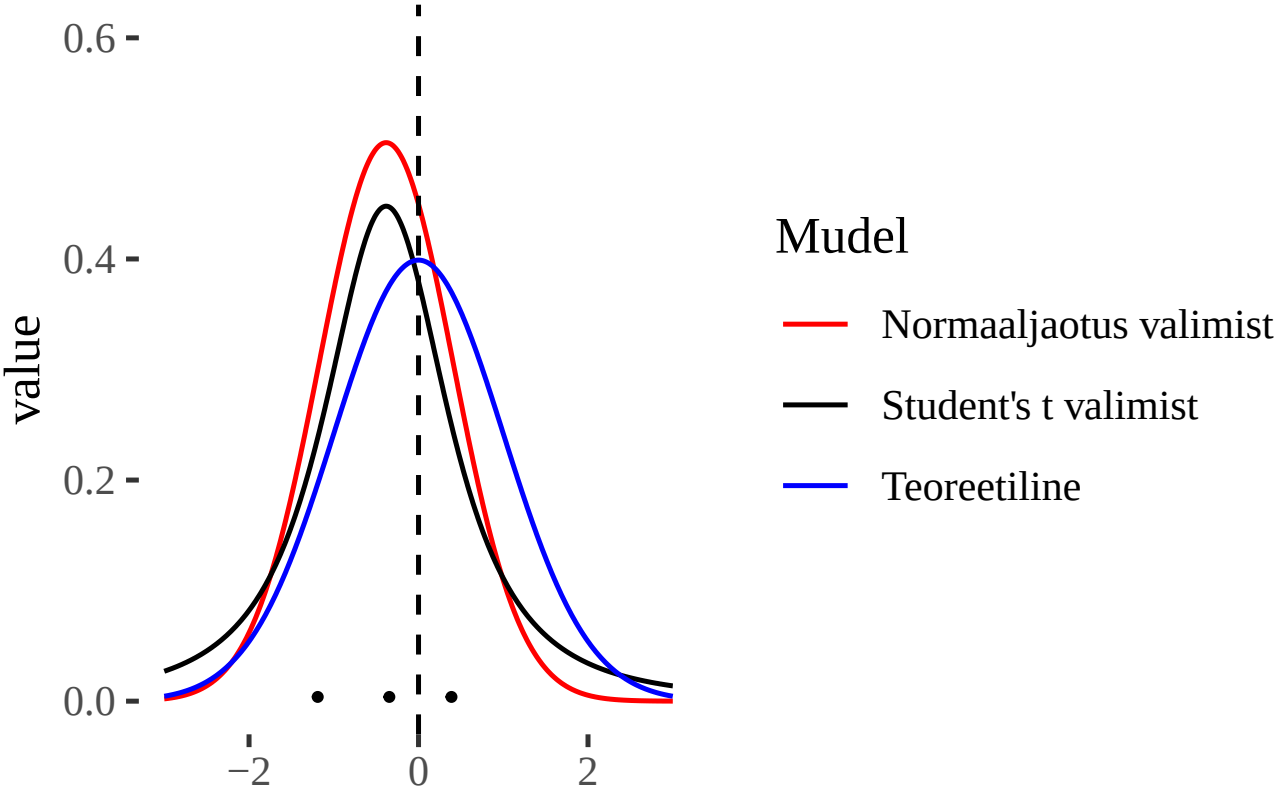
Oletame, et meil on kolm andmepunkti ning me usume, et need andmed on juhuslikult tõmmatud normaaljaotusest või sellele lähedasest jaotusest. Normaaljaotuse mudelit kasutades deklareerime, et me usume, et kui oleksime olnud vähem laisad ja 3 mõõtmise asemel sooritanuks 3000, siis need mõõtmised sobituksid piisavalt hästi meie 3 väärtuse peal fititud normaaljaotusega. Seega, me usume, et omades 3 andmepunkti me teame juba umbkaudu, millised tulemused me oleksime saanud korjates näiteks 3 miljonit andmepunkti. Oma mudelist võime simuleerida ükskõik kui palju andmepunkte.

Aga pidage meeles, et selle mudeli fittimiseks kasutame me ainult neid andmeid, mis meil päriselt on — ja kui meil on ainult 3 andmepunkti, on tõenäoline, et fititud mudel ei kajasta hästi tegelikkust.

Kuidas panna skeptik uskuma, et statistilised meetodid töötavad halvasti väikestel valimitel? Järgnevalt illustreerime seda ühe võimaliku valimiga paljudest, mis on tõmmatud imaginaarsest populatsioonist, mille parameetreid me teame. Me tõmbame 3-se valimi ning üritame selle valimi põhjal ennustada selleasama populatsiooni struktuuri. Kuna tegemist on simulatsiooniga, teame täpselt, et populatsioon, kust me tõmbame oma kolmese valimi, on normaaljaotusega,

et tema keskväärtus = 0 ja et tema sd = 1. Seega saame võrrelda oma ennustust populatsiooni tõeliste parameetriväärtustega. Me fitime oma valimiandmetega 2 erinevat mudelit: normaaljaotuse ja Studenti t jaotuse.

Juhuvalim normaaljaotusest, mille keskmine = 0 ja sd = 1 (n=3; andmepunktid on näidatud mustade munadena). Sinine joon - populatsioon, millest tõmmati valim; punane joon - normaaljaotuse mudel, mis on fititud valimi andmetel; must joon - Studenti t jaotuse mudel, mis on fititud samade andmetega. Mustad punktid, valim. Katkendjoon, populatsiooni keskmine, millest valim tõmmati.



Siin saame hinnata mudelite fitte jumala positsioonilt, võrreldes fititud mudelite jaotusi "tõese" sinise jaotusega. Mõlemad mudelid on süstemaatiliselt nihutatud väiksemate väärtuste poole ja alahindavad varieeruvust. t jaotuse mudel on oodatult paksemate sabadega ja ennustab o-st kaugele palju rohkem väärtusi kui normaaljaotuse mudel. Kuna me teame, et populatsioon on normaaljaotusega, pole väga üllatav, et t jaotus modeleerib seda halvemini kui normaaljaotus.

Igal juhul, mõni teine juhuvalim annaks meile hoopis teistsugused mudelid, mis rohkem või vähem erinevad algsest populatsioonist.

Mis juhtub kui me kasutame oma normaaljaotuse mudelit uute

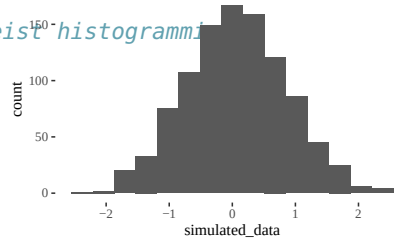
andmete simuleerimiseks? Kui lähedased on need simuleeritud andmed populatsiooni andmetega ja kui lähedased valimi andmetega, millega me normaaljaotuse mudeli fittisime?

Kasutame fititud mudeleid uute andmete simuleerimiseks.

```
# tõmbame 3 juhuslikku arvu normaalhaotusest, mille keskväärtus = 0 ja sd = 1.
dfr <- tibble(sample_data = rnorm(3))
dfr <- summarise_at(dfr, "sample_data", c("mean", "sd"))
dfr %>% round(2) %>% kable()
```

mean	sd
0.07	0.81

```
# simuleerime 1000 uut andmepunkti fititud mudelist
simulated_data <- rnorm(1000, dfr$mean, dfr$sd)
# arvutame simuleeritud andmete keskmise ja sd ning joonistame neist histogrammi
ggplot(tibble(simulated_data), aes(simulated_data)) +
geom_histogram(bins = 15) + ggthemes::theme_tufte()
```



Nagu näha, igati ootuspäraselt on uute (simuleeritud) andmete keskväärtus ja SD väga sarnased algsete andmete omale, mida kasutasime mudeli fittimisel. Kahjuks ei ole need aga kaugeltki nii sarnased algsele jaotusele, mille kuju me püüame oma andmete ja mudeli pealt ennustada. Seega on meie mudel üle-fittitud, mis tähendab, et ta kajastab liigselt neid valimi aspekte, mis ei peegelda algse populatsiooni omadusi. Loomulikult ei vasta ükski mudel päriselt tegelikkusele. Küsimus on pigem selles, kas mõni meie mudelitest on piisavalt hea, et olla kasulik. Vastus sellele sõltub, milleks plaanime oma mudelit kasutada.

```
mean(simulated_data > 0)
```

```
## [1] 0.535
```

```
mean(simulated_data > 1)
```

```
## [1] 0.116
```

Kui populatsioonivääртustest on 50% suuremad kui 0, siis mudeli järgi vaevalt 32%. Kui populatsioonivääртustest on 16% suuremad kui 1, siis mudeli järgi vaevalt 4%. See illustreerib hästi mudeli kvaliteeti.

```
sim_t <- rstudent_t(1000, 2, dfr$mean, dfr$sd)
mean(sim_t > 0)
```

```
## [1] 0.516
```

```
mean(sim_t > 1)
```

```
## [1] 0.189
```

Samad ennustused t jaotusest on isegi paremad! Aga kumb on ikkagi parem mudel populatsioonile?

Normaaljaotuse ja lognormaaljaotuse erilisus

Normaaljaotus ja lognormaaljaotus on erilised sest

(1) kesksest piirteoreemist (*central limit theorem*) tuleneb, et olgu teie valim ükskõik millise jaotusega, paljudest valimitest arvutatud **aritmeetilised keskmised** on alati enam-vähem normaaljaotusega. See kehtib enamuse andmejaotuste korral, kui n>30. Selle matemaatilise tõe peegeldus füüsikalisse maailma on "elementaarsete vigade hüpotees", mille kohaselt paljude väikeste üksteisest sõltumatute juhuslike efektide (vigade) summa annab tulemuseks normaaljaotuse.

Paraku enamus bioloogilisi mõõtmisi annavad tulemuseks eranditult mitte-negatiivseid väärtusi. Sageli on selliste väärtuste jaotused ebasümmeetrilised (v.a. siis, kui cv = sd/mean on väike), ja kui nii, siis on meil sageli tegu lognormaaljaotusega, mis tekkib log-normaalsete muutujate korrutamisest. Siit tuleb Keskne piirteoreem 2, mille kohaselt suvalise jaotusega muutujate **geomeetrilised keskmised** on enam-vähem lognormaaljaotusega, ning elementaarsete vigade hüpotees 2: Kui juhuslik varieeruvus tekib paljude juhuslike efektide korrutamisel, on tulemuseks lognormaaljaotus. Lognormaaljaotusega väärtuste logaritmimine annab normaaljaotuse.

(2) Nii normaal- kui lognormaaljaotus on maksimaalse entroopiaga jaotused. Entroopiat vaadeldakse siin informatsiooni & müra kaudu — maksimaalse entroopiaga süsteem sisaldab maksimaalselt müra ja minimaalselt informatsiooni (vastavalt Shannoni informatsiooniteooriale). See tähendab, et väljaspool oma parameetrite tuunitud väärtusi on normaal- ja lognormaaljaotused minimaalselt informatiivsed. Normaaljaotusel ja lognormaaljaotusel on kummagil kaks parameetrit, *mu* ja *sigma* (ehk keskmine ja standardhälve), mille väärtused fikseerides fikseerime üheselt jaotuse ehk mudeli kuju, lisades sinna minimaalselt muud (sooviamtut) informatsiooni. Teised maksimaalse entroopiaga jaotused on näiteks eksponentsiaalne jaotus, binoomjaotus, bernoulli jaotus, poissoni jaotus.

Maksimaalsel entroopial põhineb normaaljaotuse ja lognormaaljaotuse sage kasutamine Bayesi statistikas prioritena, sest me suudame paremini kontrollida, millist informatsiooni me neisse surume. Esimesel kesksel piirteoreemil seevastu põhineb kogu sageduslik statistika.

Kui meil on tegu nullist suuremate andmetega, siis on andmete logaritmimine sageli hea mõte. Logaritmitud andmete pealt arvutatud keskmise ja sd eksponentimine annab meile geomeetrilise keskmise ja multiplikatiivse sd. Kui me fitime lognormaaljaotust andmetega, siis fititud koefitsiendid mu ja sd tuleb eksponentida, et saada geomeetriline keskmine ja multiplikatiivne sd.

Normaaljaotuse ja lognormaaljaotuse võrdlus

Normaaljaotus

1. Normaaljaotusega ehk normaalsete juhuslike muutujate liitmine annab normaalse summa. Lineaarsed kombinatsioonid $Y = \alpha + \beta_1 X_1 + \beta_2 X_2$ jäävad normaalseks.
2. Normaalsete muutujate aritmeetilised keskmised on normaaljaotusega.
3. Keskne piirteoreem: mitte-normaalsete muutujate aritmeetilised keskmised on enam-vähem normaaljaotusega.
4. Elementaarsete vigade hüpotees: kui juhuslik varieeruvus on paljude juhuslike mõjude summa, on tulemuseks normaaljaotus.
5. Additiivne regressioonimudel (normaalne tõepära) viib additiivsetele vigadele (residuaalidele), mis omakorda viib konstantsele varieeruvusele (SD-le). Vead on normaaljaotusega.

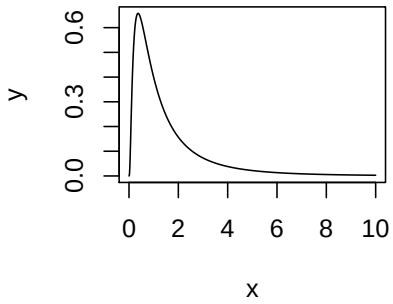
lognormaaljaotus

1. lognormaalsete juhuslike muutujate korrutamine annab lognormaalse korrutise.
2. Longnormaalsete muutujate geomeetrilised keskmised on lognormaaljaotusega.
3. Keskne piirteoreem: mitte-lognormaalsete muutujate geomeetrilised keskmised on enam-vähem lognormaaljaotusega
4. Elementaarsete vigade hüpotees: kui juhuslik varieeruvus on paljude juhuslike mõjude korrutis, on tulemuseks lognormaaljaotus
5. multiplikatiivne regressioonimudel (lognormaalne tõepära) viib multiplikatiivsete vigadeni ja konstantsele suhtelisele varieeruvusele (CV-le). Vigade jaotus on ebasümmeetriline.

Seega võime lognormaaljaotust kutsuda ka multiplikatiivseks normaaljaotuseks.

Lognormaaljaotus

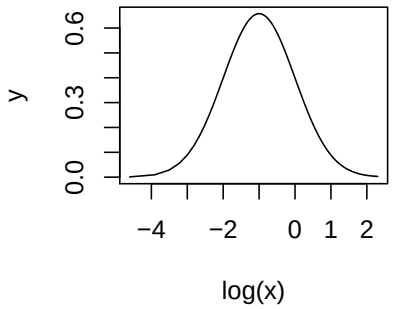
```
x <- seq(0, 10, length.out = 1000)
y <- dlnorm(x)
plot(x, y, typ = "l")
```



Seda jaotust, mis ei ulatu kunagi teisele poole nulli, iseloomustab, et x-i logaritmimine annab tulemuseks normaaljaotuse.

```
plot(log(x), y, type = "l")
```

Lognormaaljaotuse keskväärtus, standardhälve, mood ja mediaan:



$$keskväärtus = \exp(\mu + 1/2 \times \sigma^2)$$

$$sd = \exp(\mu + 1/2 \times \sigma^2) \times \sqrt{\exp(\sigma^2) - 1}$$

$$mood = e^{\mu - \sigma^2}$$

$$mediaan = e^{\mu}$$

Siin on siis μ ja σ arvutatud logaritmitud andmete pealt.

Mudelite arvutamine MCMC simulatsiooni abil

MCMC põhimõte

Me vajame MCMC protseduure puhtalt tehniistel põhjustel – kuna posteerior arvutatakse korraga üle kõikide mudeli parameetrite siis sisaldab see endas kõikide parameetrite kõikvõimalike väärtuste kombinatsioone. Ja seda ei oska keegi integreerimise teel või gridiga tuimalt läbi arvutades teha kui parameetrite arv kasvab u üle kümne. MCMC tõmbab juhuvalimi "otse" k-dimensionaalsest posterriorist, mis teeb meie jaoks sellega töötamise erakordsalt lihtsaks.

MCMC tähendab Markovi ahelate Monte Carlo meetodit. Sellel meetodil on siis kaks poolt: Markovi ahelad ja Monte Carlo. **Monte Carlo** on simulatsioonimeetod, mis kasutab juhuslike arvude genereerimist parameetriväärtuse umbkaudseks hindamiseks. Näiteks, et hinnata ebaregulaarse kujundi pindala, (i) joonista selle ümber ruut, mille pindala sa oskad mõõta, (ii) viska sellele ühendkujundile juhuslikult n palli, millest osad langevad algsele kujundile aga kõik langevad ruutu; (iii) nende pallide proportsioon korrutatuna ruudu pindalaga annab teile algse kujundi pindala.

Markovi ahel kujutab endast üksteisega tõenäosuslikult seotud sündmuste järjestust. Markovi ahel liigub sammhaaval mõõda fikseeritud hulka võimalikke sündmusi (parameetriruumi), kusjuures iga juba toimunud samm määrab järgmiste võimalike sammude tõenäosused. Igal sammul peatub ahel ühel sündmusel (parameetriväärtusel) ja seejärel viib järgmine samm uuele parameetriväärtusele, või mitte (ahel jääb paigale). See süsteem töötab ilma mäluta: järgmine samm sõltub sellest, millisel parameetriväärtusel on ahel praegu, mitte süsteemi ajaloost.

Lihtsaim ja ajalooliselt esimene Markovi ahela tüüp on Metropolise algoritm. Kujutage endale ette 1-mõõtmelist parameetriruumi, mis vastab ühe parameetriga mudelile, kus iga punkt kujutab endast ühte parameetriväärtust. Ahel maandub sellel 1-D sirgel juhuslikus punktis, misjärel on tal võrdne võimalus vaadata vasakule või paremale. Ta valib juhuslikult ühe parameetriväärtuse ehk prospekti, aga selle asemel, et sinna hüpata, hoopis kaalutleb meie andmetele

ja priorile toetudes, milline on selle prospekti tõenäosus võrreldes ahela praeguse positsiooniga. Ehk, kui palju kordi on prospekt andmete & priori poolt rohkem toetatud kui käesolev parameetriväärtus. Kui prospekt on paremini toetatud, siis liigub ahel alati uuele positsioonile. Kui aga prospekt on vähem toetatud kui ahela praegune positsioon, siis liigub ahel sellele proportsionaalselt väiksema tõenäosusega (alternatiiv liikumisele on paigale jäämine ja uue juhusliku prospekti valimine, millega kordub sama protseduur).

Sellisel viisil edasi-tagasi sammudes veedab ahel rohkem aega parameetriruumi piirkonnas, mis on andmetega paremas kooskõlas, ja pikas perspektiivis annavad ahela sammud juhuvalimi posteeriorist. Ahela esimesed tuhat sammu (200 kõlbab ka sageli) loetakse nn sissepõletamise perioodiks, mil ahel otsib katse-eksituse meetodil posteeriori tihedamat (tõenäolisemat) ala, ja neid samme ei salvestata posterioorse valimi hulka. Salvestatud ahela osa hõlmab tavaliselt 1000 sammu. Sageli jooksutame paraleelselt 3 või 4 iseseisvat ahelat ja vaatame, kas need konvergeeruvad samas parameetriväärtuste piirkonnas. See on tähtis kvaliteedinäitaja - kui mõni ahel liigub teistest eemal, siis ei saa me oma arvutust usaldada.

Seega võtab ahel posteeriorist juhuslikke arve, millega me saame hiljem otse töötada - näiteks joonistada posteeriori histogrammi ja leida sealt suurima tihedusega piirkonna, kuhu jääb 95% ahela sammudest (ehk 95% CI ehk 95% HDI [highest density interval]). Kui meie mudelis on n parameetrit, siis jookseb ahel n-dimensionaalses matemaatilises ruumis, kus osad parameetriväärtused on andmetega paremas kooskõlas kui teised. Tänapäeval jooksutatakse ka paarikümne tuhande parameetriga mudeleid, mille jaoks arvutil kulub tavaliselt kuni paar päeva.

Metropolise algoritm leiab peale paljude sammude astumist garanteeritult õige posteeriori ja võtab sellest juhuvalimi. Probleem on selles, kui palju samme selleks tegelikkuses kulub. Kuna algoritmil on võimalus ka paigale jääda, siis olukorras, kus posteerior asub ahela praegusest asukohast kaugel, või on väga kitsas (hõlmab vaid tühist osa parameetriruumist) võib ahelal kuluda liiga palju aega, et temast tegelikku kasu võiks tõusta. Selle pärast otsitakse, ja leitakse, sellele matemaatiliselt efektiivsemaid alternatiive, millest Stan kasutab nn Hamiltonian Monte Carlo-t.

Hamiltonian Monte Carlo lahendab probleemi jooksutades füüsikalise simulatsiooni käigus kuulikest n-dimnesionaalsel pinnal. Kuulike veedab kauem aega parameetriruumi piirkondades, mis on andmete poolt paremini kinnitatud. Igas punktis randomiseeritakse tema momentum. Pind, millel kuulike jookseb, on miinus log posteerior. See töötab hästi, aga vajab lisaks gradienti e log-posteeriori kurvatuuri, kuulikese massi, ühel trajektooril asuvate sammude hulka ja individ-

uaalsete sammude pikkust. Need kõik määratakse automaatselt, aga igaühega neist võivad seonduda veateated ja kvaliteediprobleemid.

MCMC ahelate kvaliteet

Kui Rhat on 1, siis see tähendab, et MCMC ahelad on ilusti jooksnud ja posteeriori sämplinud. Kui $Rhat > 1.1$, siis on probleem. Suur Rhat viitab, et ahel(ad) pole jõudnud konvergeeruda. Kui ahelad ei konvergeeru, siis võib karta, et nad ei sämpli ka sama posteeriori jaotust. Kontrolli, kas mudeli kood ei sisalda vigu. Kui ei, siis vahest aitab, kui pikendada warm-up perioodi. Vahest aitab mudeli re-parametriseerimine (siin on lihtne trikk tekitada priorid, mis ei erineks väga palju oma vahemiku poolest; sellega kaasneb sageli andmete tsentreerimine või standardiseerimine).

n_eff on efektiivne valimi suurus, mis hindab iseseisvalt sämplitud andmete arvu ning see ei tohi olla väga väike. n_eff on sammude arv, arvestades, et puudub autokorrelatsioon ahela järjestikuste sammude vahel.

Kui n_eff on palju väiksem kui jooksutatud markovi ahela pikkus (iga ahel on defaultina 1000 iteratsiooni pikk), on ahel jooksnud ebaefektiivselt. See ei tähenda tingimata, et posteerior vale oleks. Reegilina peaks $Neff/N > 0.1$

Ahelad peavad plotitud kujul välja nägema nagu karvased tõugud, mis on ilma paljaste laikudeta. plot(mudeliobjekt) näitab ahelaid.

Kui ahelad omavad pikki sirgeid lõike (n_eff tuleb siis väga madal), kus ahel ei ole töötanud, siis see rikub korralikult posteeriori. Tüüpiliselt aitavad nõrgalt informatiivsed priorid — priorite õige valik on sama palju arvutuslik vajadus kui taustainfo lisamine. Igal juhul tuleb vältida aladefineeritud tasaseid prioreid, mis võimaldavad ahelatel sämplida lõpmatust ja sel viisil õige tee kaotada. Peale selle, tasased priorid, mis ütlevad, et kõik parameetri väärtused on võrdselt tõenäolised, kajastavad harva meie tegelikke taustateadmisi.

Divergentsed transitsioonid. Näiteks hoiatus: There were 15 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.

Divergentne transitsioon tähendab, et Hamiltonian Monte Carlo füüsikaline simulatsioon ei tööta päris nii, nagu peaks (energia jäävuse seadus ei ole "kuulikese"' liikumises täidetud). See võib tähendada, et posteeriori sämplimine on väheefektiivne ja/või kallutatud. Õnneks võib üksikuid divergentseid transitsioone lihtsalt ignoreerida. Adapt delta määrab ahela sammu pikkuse. Kui see läheneb 1-le, läheb samm lühemaks ja vääheneb tõenäosus, et ahel astub posteeriorist kaugele välja ja "eksib ära". Paraku, lühem samm muudab posteeriori sämplimise aeglasemaks. Adapt delta on

vaikeväärtusega 0.8. Nii saab seda tõsta 0.99-le: brm(. . . , control = list(adapt_delta = 0.99)).

Maximum treedepth exceeded ei ole nii tõsine puudus kui divergentsed transitsioonid. See on pelgalt ahela jooksutamise efektiivsuse küsimus. Max treedepth on olemas selle pärast, et vältida ahelate igavesti jooksmist, selle vaikeväärtus on 10 ja kui oled nõus ahelaid pikemalt jooksutama, siis tee nii: brm(. . . , control = list(max_treedepth = 15)).

Täpsemad instruktsioonid Stani veateadete kohta leiad siit: `https://mc-stan.org/misc/warnings.html`

Paar sõna prioritest

Kui me arvutame lihtsaid mudeleid käsitsi, siis arvutuslikust seisukohast pole suurt vahet, millist priorit me kasutame - senikaua kui meil on nullist erinev priori väärtus kõigile teaduslikult mõtekaile parameetriväärtustele (igal parameetriväärtusel, kus priori väärtus on null, tuleb ka posteerior null). Keerulisemate mudelite korral, mida arvutame mcmc meetoditega, töötavad üldiselt paremini priorid, mis ei kata ühtlaselt kogu loogiliselt võimalikku parameetriväärtuste vahemikku - Inf . . . Inf. Prioreid, mis on võrreldes teaduslikult mõistlikke väärtustega väga laia küüruga, kutsutakse nõrgalt informatiivseteks ehk nõrgalt regulariseerivateks. Need aitavad mcmc ahelatel vältida parameetriruumi piirkondi, kus ei asu posterioorset tihedust, mida sämplida. Seega jooksevad ahelad selliste priorite korral kiiremini, mis aga ei tohiks mõjutada posteeriori kuju. Oleame, et meile huvipakkuva parameetri - mu - teaduslikut usutavad väärtused jäävad -1 ja 1 vahele. Siis nõrgalt regulariseeriv prior oleks näiteks normal(0, 10).

Kui me selliseid laiu prioreid kitsendame, siis varem või hiljem hakkavad need posteeriori kujule mõju avaldama. Nüüd ei ole me eesmärk enam suurendada ahelate efektiivsust, vaid lisada mudelisse tegelikku teaduslikku informatsiooni. See tähendab, et nn informatiivse priori jaotusfunktsiooni ja selle täpse kuju määrab meie teaduslik teadmine. Sellise priori konstrueerimine võib olla vähem või rohkem keerukas, aga mida rohkem on meil andmeid, seda vähem mõjutab prior posteeriori. Ja see tähendab omakorda, et kui andmeid juurde tuleb, ei tasu priorile täpse, teaduse poolt määratud kuju andmise vaev ennast ühel hetkel enam ära. Aga kui andmeid napib, siis sellevõrra on prioritega tehtud töö tulusam – ja sellesse tasub rohkem ressurssi panustada.

Olukorras, kus andmete kogumine on väga kallis (nagu ravimiuuringutel), tehakse väga tõsiseid pingutusi priorite spetsifitseerimisel.

Miks me ei kasuta priorite määramisel valimiandmeid (näiteks

ei tsentreeri oma prioreid valimi keskmisele)? Aga sellepärast, et valimid võivad peegeldada halvasti tegelikkust (juhuviga ja/või süstemaatiline viga) ja me tahame, et meie priorid vähendaksid antud probleemi, mitte ei võimendaks seda. Selle pärast peegeldavad priorid meie üldist, andmetest sõltumata teadmist võimalike parameetriväärtuste kohta. Nad on andmetest, mis lähevad tõepäramudelisse, sõltumatud.

brms

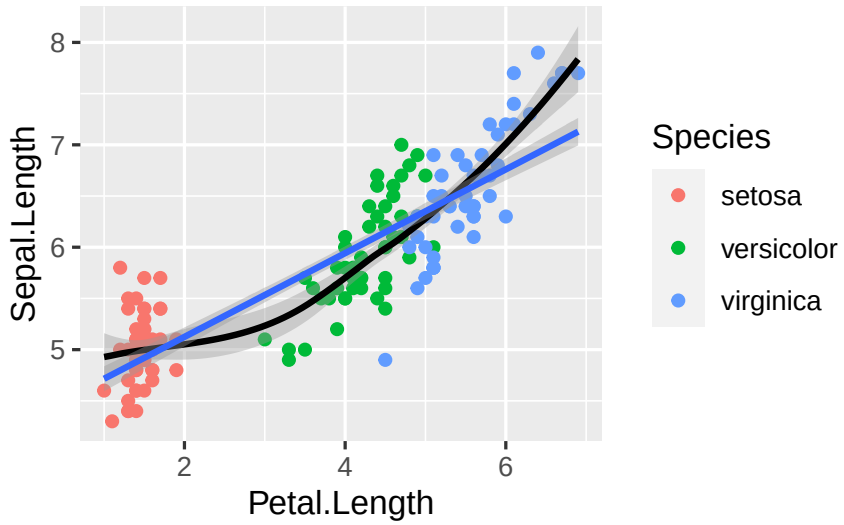
brms on pakett, mis võimaldab kirjutada lihtsat süntaksit kasutades ka üsna keerulisi mudeleid ja need Stan-is fittida. Brms on loodud Paul Bürkneri poolt (`https://github.com/paul-buerkner/brms`), ja see on oma kasutuslihtsuse tõttu jõudnud isegi ette Stani meeskonna arendatavast analoogsest paketist rstanarm (`https://github.com/stan-dev/rstanarm/blob/master/README.md`). rstanarm, mide me siin ei käsitle, püüab pakkuda tavalisete sageduslikele meetoditele (ANOVA, lineaarne regressioon jne) bayesi analooge, mille mudeli spetsifikatsioon ja väljund erineks võimalikult vähe tavalisest baas-R-i töövoost. brms on keskendunud mitmetasemelistele mudelitele ja kasutab põhimõtteliselt lme4 (`https://github.com/lme4/lme4/`) mudelite keelt. Loomulikult saab brms-is fittida ka lineaarseid ja mitte-lineaareid ühetasemelisi mudeleid.

brms-i töövoog

brms-iga modelleerimisel on mõned asjad, mida tuleks teha sõltumata sellest, millist mudelit te parajasti fitite. Kõigepealt peaksite kontrollima, et mcmc ahelad on korralikult jooksnud (divergentsed transitsioonid, rhat ja ahelate visuaalne inspekteerimine). Lisaks peaksite tegema posterioorse prediktiivse ploti ja vaatama, kui palju mudeli poolt genereeritud uued valimid meenutavad teie valimit. Samuti peaksite joonisel plottima residuaalid. Kui te inspekteerite fititud parameetrite väärtusi, siis tehke seda posteeriorite tasemel ja koos veapiiridega.

```
ggplot(iris, aes(Petal.Length, Sepal.Length)) +
geom_point(aes(color = Species)) +
geom_smooth(method = "loess", color = "black") +
geom_smooth(method = "lm")
```

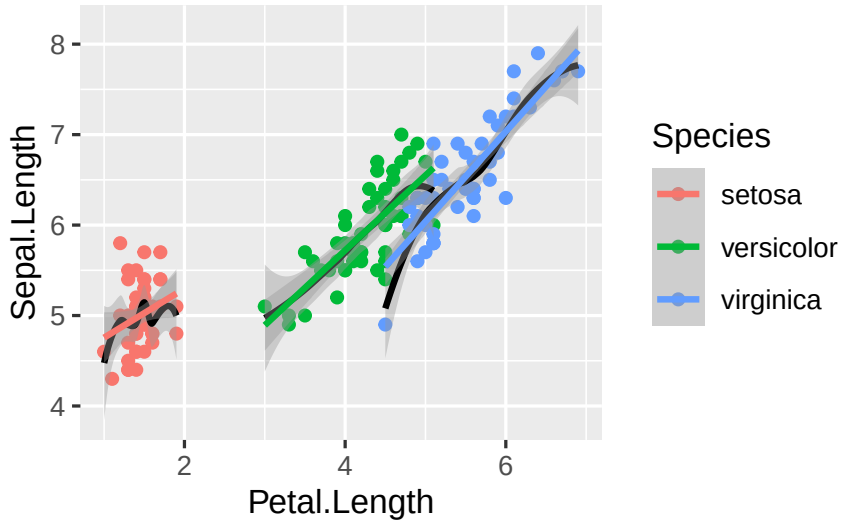
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Loess fit viitab, et 3 liiki ühe sirgega mudeldada pole võib-olla optimaalne lahendus.

```
ggplot(iris, aes(Petal.Length, Sepal.Length, color = Species)) +
geom_point() +
geom_smooth(method = "loess", aes(group = Species), color = "black") +
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Nüüd on loess ja lm heas kooskõlas - seos y~x vahel oleks nagu enam-vähem lineaarne. Siit tuleb ka välja, et kolme mudeli tõusud on sarnased, interceptid erinevad.

Kiire töövoog

Minimaalses töövoos anname ette võimalikult vähe parameetreid ja töötame mudeliga nii vähe kui võimalik. See on mõeldud ülevaatena Bayesi mudeli fittimise põhilistest etappidest

Mudeli fittimine:

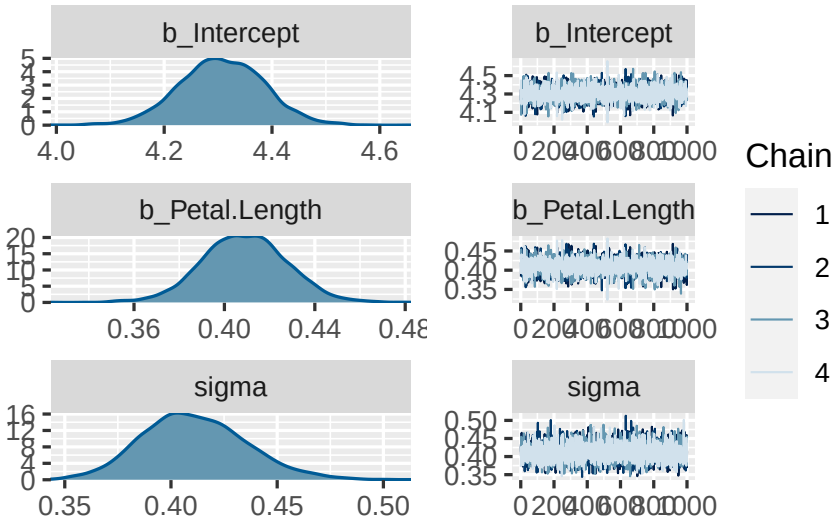
```
m_kiire <- brm(Sepal.Length ~ Petal.Length, data = iris)
```

Priorid on brms-i poolt ette antud ja loomulikult ei sisalda mingit teaduslikku informatsiooni. Nad on siiski "nõrgalt informatiivsed" selles mõttes, et kasutavad parametriseeringuid, mis enamasti võimaldavad mcmc ahelatel normaalselt joosta. Järgmises ptk-s õpime ise prioreid määrama.

Erandiks on siin tõusu ehk beta-koefitsiendi priorid, mis oma vaikeolekus on tasased ja ulatuvad lõpmatusse. See on selleks, et sundida kasutajat vähemalt neid prioreid käsitsi muutma. Siiski, lihtsamad mudelid töötavad hästi ka selliste jubedustega.

Posteeriorid ja mcmc ahelate konvergents

```
plot(m_kiire)
```



Fiti kokkuvõte - koefitsiendid ja nende fittimise edukust hindavad statistikud (Eff.Sample, Rhat)

```
tidy(m_kiire) %>% mutate_if(is.numeric, round, 2) %>% kable()
```

term	estimate	std.error	lower	upper
b_Intercept	4.31	0.08	4.18	4.43
b_Petal.Length	0.41	0.02	0.38	0.44
sigma	0.41	0.02	0.37	0.45
lp__	-85.34	1.22	-87.76	-84.02

Eff.Sample näitab efektiivset valimi suurust, mida ahelad on kasutanud. See on suht keeruline mõiste, aga piisab, kui aru saada, et see näitaja ei tohiks olla madalam kui paarkümmend.

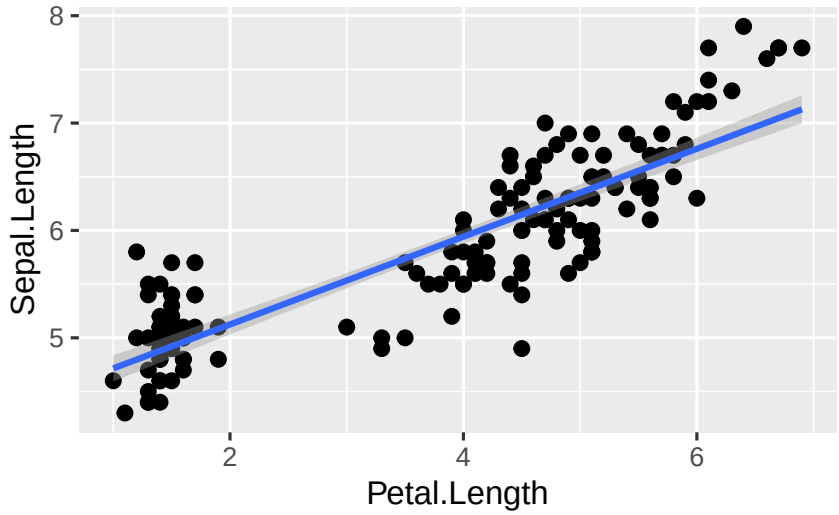
Rhat on statistik, mis vaatab ahelate konvergentsi. Kui Rhat > 1.1, siis on kuri karjas. Rhat 1.0 ei tähenda paraku, et võiks rahulikult

hingata – tegu on statistikuga, mida saab hästi tõlgendada häda kuulutajana, aga liiga sageli mitte vastupidi.

Ennustav plot ehk *marginal plot* – mudeli fit 95% CI-ga.

```
plot(marginal_effects(m_kiire), points = TRUE)
```

```
## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```



Põhjalikum töövoog

Põhiline erinevus eelmisega on suurem tähelepanu prioritele, mudeli fittimise diagnostikale ning tööle fititud mudeliga.

Spetsifitseerime mudeli, vaatame ja muudame vaikeprioreid

brms-i vaikepriorid on konstrueeritud olema üsna väheinformatiivsed ja need tuleks enamasti informatiivsematega asendada. Igasse priorisse tuleks panna nii palju informatsiooni, kui teil on vastava parameetri kohta. Kui te mõne parameetri kohta ei oska öelda, millised oleks selle mõistlikud oodatavad väärtused, siis saab piirduda brms-i antud vaikeväärtustega. Samas, kui keerulisemad mudelid ei taha hästi joosta (mida tuleb ikka ette), siis aitab sageli priorite kitsamaks muutmine.

```
get_prior(Sepal.Length ~ Petal.Length + (1 | Species), data = iris) %>% kable()
```

prior	class	coef	group	resp	dpar	nlpar	bound
	b						
	b	Petal.Length					
student_t(3, 6, 10)	Intercept						
student_t(3, 0, 10)	sd						
	sd		Species				
	sd	Intercept	Species				
student_t(3, 0, 10)	sigma						

Me fitime pedagoogilistel kaalutlustel shrinkage mudeli, mis tõmbab 3 liigi lõikepunkte natuke keskmise lõikepunkti suunas. On vaieldav, kas see on irise andmestiku juures mõistlik strateegia, aga teeme seda siin ikkagi.

Prioreid muudame nii:

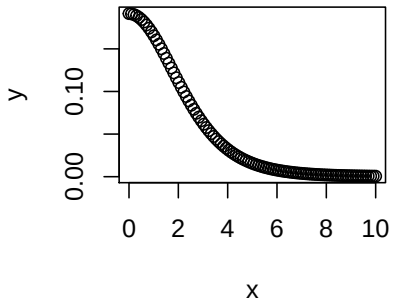
```
prior <- c(prior(normal(6, 3), class = "Intercept"), prior(normal(0, 1), class = "b"),
    prior(student_t(6, 0, 2), class = "sigma"))
```

Mitmetasemeline shrinkage mudel on abinõu ülefittimise vastu. Mudelite võrdlemisel otsitakse kompromissi - ehk mudeli, mille ennustused oleks andmepunktidele võimalikult lähedal ilma,et see mudel oleks liiga keeruliseks aetud (keerulisus on proportsionaalne mudeli parameetrite arvuga).

Me valime siin nn väheinformariivsed priorid, nii et regressiooni tulemus on suht hästi võrreldav lme4 sagedusliku mudeliga. “b” koefitsiendi priorile (aga mitte “sigma” ega “Intercept”-le) võib anda ka ülemise ja/või alumise piiri `prior(normal(0, 1), class = "b", lb = -1, ub = 10)` ütleb, et “b” prior on nullist erinev ainult -1 ja 10 vahel. “sigma” priorid on automaatselt lb = 0-ga, sest varieeruvus ei tohi olla negatiivne.

Alati tasub prioreid pildil vaadata, et veenduda nende mõistlikuses.

```
x <- seq(0, 10, length.out = 100)
y <- dstudent_t(x, df = 6, mu = 0, sigma = 2, log = FALSE)
plot(y ~ x)
```



Sigma prior, mida brms kasutab, on vaikimisi pool sümmeetrilisest jaotusest, mis lõigatakse nulli kohalt pooleks nii, et seal puuduvad < 0 väärtused (seega ei saa varieeruvuse posteerior minna alla nulli).

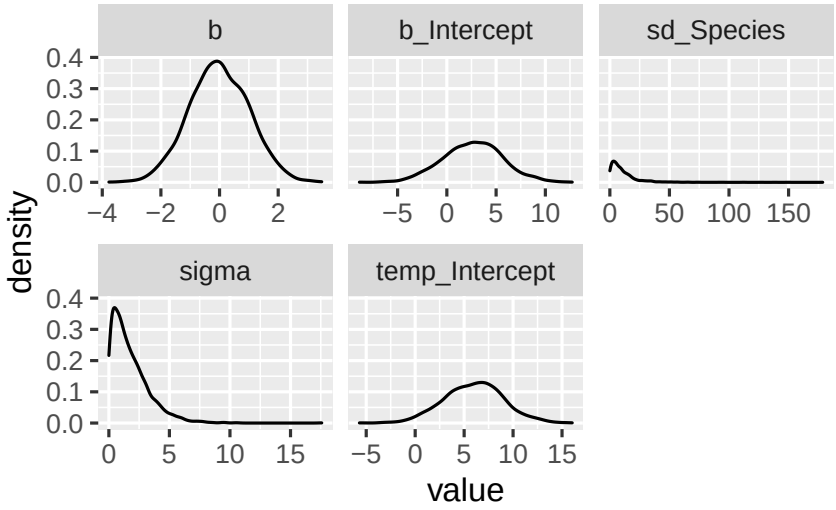
Me võime ka prioreid ilma likelihoodideta (tõepärafunktsioonideta) läbi mudeli lasta, misjärel tõmbame fititud mudelist priorite valimid (neid võiks kutsuda ka “priorite posteerioriteks”) ja plotime kõik priorid koos. Seda pilti saab siis võrrelda koos andmetega fititud mudeli posteerioritega. Selle võimaluse kasutamine on tõusuteel, sest keerulisemate mudelite puhul võib priorite ükshaaval plottimine osutuda eksitavaks.

Tekitame priorite valimid, et näha oma priorite mõistlikust (brm() argument on sample_prior = TRUE). Ühtlasi fitime ka oma mudeli koos andmete ja prioritega.

```
m1 <- brm(Sepal.Length ~ Petal.Length + (1 | Species), data = iris, prior = prior,
    family = gaussian, warmup = 1000, iter = 2000, chains = 3, cores = 3, sample_prior = TRUE)
```

Me fittisime mudeli m1 kaks korda: nii andmetega (selle juurde jõuame varsti), kui ka ilma andmeteta. Kui panna sisse `sample_prior = "only"`, siis jookseb mudel ilma andmeteta, ja selle võrra kiiremini. Vaikeväärtus on `sample_prior = "no"`, mis tähendab, et fititakse ainult üks mudel - koos andmetega. Ilma andmeteta (likelihoodita) fitist saame tõmmata priorite mcmc valimid, mille ka järgmiseks plotime.

```
prior_samples(m1) %>%
gather() %>%
ggplot() + geom_density(aes(value)) +
facet_wrap(~key, scales = "free_x")
```



Kui kasutame `sample_prior = "only"` varianti, siis on esimene koodirida erinev: `samples1 = as.data.frame(m1$fit)`.

brms-i Intercepti priorite spetsifitseerimisel tasub teada, et brms oma sisemuses tsentreerib kõik prediktorid nullile $x - mean(x)$, ja teie poolt ette antud prior peaks vastama neile tsentreeritud prediktoritele, kus kõikide prediktorite keskväärtus on null. Põhjus on, et tsentreeritud parametriseeringuga mudelid jooksevad sageli paremini. Alternatiiv on kasutada mudeli tavapärase süntaksi $y \sim 1 + x$ (või ekvivalentselt $y \sim x$) asemel süntaksit $y \sim 0 + intercept + x$. Sellisel juhul saab anda priorid tsentreerimata predikroritele. Lisaks on brms selle süntaksi puhul nõus b-le antud prioreid vaikimisi ka intercepti fittimisel kasutama.

`brm()` funktsiooni argumendid:

- family - tõepärafunktsiooni tüüp (modelleerib y muutuja jaotust e likelihoodi)
- warmup - mitu sammu mcmc ahel astub, enne kui ahelat salvestama hakatakse. tavaliselt on 500-1000 sammu piisav, et tagada

ahelate konvergents. Kui ei ole, tõstke 2000 sammuni.

- iter - ahelate sammude arv, mida salvestatakse peale warmup perioodi. Enamasti on 2000 piisav. Kui olete nõus piirduma posteeriori keskväärtuse arvutamisega ja ei soovi täpseid usaldusintervalle, siis võib piisata ka 200 sammust.
- chains - mitu sõltumatut mcmc ahelat jooksutada. 3 on hea selleks, et näha kas ahelad konvergeeruvad. Kui mitte, tuleks lisada informatiivsemaid prioreid ja/või warmupi pikkust.
- cores - mitu teie arvuti tuuma ahelaid jooksutama panna.
- adapt_delta - mida suurem number (max = 1), seda stabiilsemalt, ja aeglasemalt, ahelad jooksevad.
- thin - kui ahel on autokorreleeritud, st ahela eelmine samm suudab ennustada järgevaid (see on paha), siis saab salvestada näit ahela iga 5. sammu (thin = 5). Aga siis tuleks ka sammude arvu 5 korda tõsta. Vaikeväärtus on thin = 1. Autokorrelatsiooni graafilist määramist näitame allpool

Järgmine funktsioon trükib välja Stani koodi, mis spetsifitseerib mudeli, mida tegelikult Stanis fittima hakatakse. See on väga kasulik, aga ainult siis kui tahate õppida otse Stanis mudeleid kirjutama:

```
make_stancode(Sepal.Length ~ Petal.Length, data = iris, prior = prior)
```

Vaatame mudelite kokkuvõtet

Lihtne tabel mudeli m2 fititud koefitsientidest koos 95% usalduspiiridega

```
tidy(m2) %>% mutate_if(is.numeric, round, 2) %>% kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	1.04	0.29	3.60	0
Sepal.Length	0.56	0.07	8.59	0
Petal.Length	-0.34	0.03	-10.94	0

r prefiks tähendab, et antud koefitsient kuulub mudeli esimesele (madalamale) tasemele (Liigi tase) r - random - tähendab, et iga grupi (liigi) sees arvutatakse oma fit. b prefiks tähendab mudeli 2. taset (keskmistatud üle kõikide gruppide). 2. tasmel on meil intercept, b1 ja b2 tõusud ning standardhälve y muutuja ennustatud andempunktide tasemel. 1. tasemel on meil 3 liigi interceptide erinevus üldisest b_Intercepti väärtusest. Seega, selleks, et saada setosa liigi intercepti, peame tegema tehte 1.616 + 0.765.

tidy funktsiooni tööd saab kontrollida järgmiste parameetrite abil:

```
tidy(x, parameters = NA, par_type = c("all", "non-varying", "varying", "hierarchical"),
    robust = FALSE, intervals = TRUE, prob = 0.9, ...)
```

`par_type = "hierarchical"` kuvab grupi taseme parameetrite sd-d ja korrelatsioonid. “varying” kuvab grupi taseme interceptid ja tõusud (siis kui neid mudeldadakse). “non-varying” kuvab kõrgema taseme (grupi-ülesed) parameetrid. robust = TRUE annab estimate posteeriori mediaanina (vaikeväärtus FALSE annab selle aritmeetilise keskmisena posteeriorist).

Nüüd põhjalikum mudeli kokkuvõte:

```
m2
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length + Petal.Length, data = iris)
##
## Coefficients:
##  (Intercept)  Sepal.Length  Petal.Length
##       1.0381        0.5612       -0.3353
```

Siin on eraldi toodud grupi tasemel ja populatsiooni tasemel koefitsiendid ja gruppide vaheline sd (= 1.72). Pane tähele, et üldine varieeruvus sigma = 0.31 on palju väiksem kui gruppide vaheline varieeruvus sd(Intercept) = 1.72. Seega on grupid üksteisest tugevalt erinevad ja neid tuleks võib-olla tõesti eraldi modelleerida.

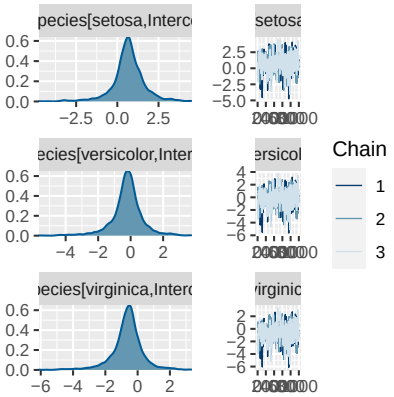
Divergentsed transitsioonid on halvad asjad - ahelad on läinud 17 korda metsa. Viisakas oleks adapt deltat tõsta või kitsamad priorid panna, aga 17 halba andmepunkti paarist tuhandest, mille mcmc ahelad meile tekitasid, pole ka mingi maailmalõpp. Nii et las praegu jääb nagu on. Need divergentsed transitsioonid on kerged tekkima just mitmetasemelistes mudelites.

Plotime posteeriorid ja ahelad

Siit on näha, et ahelad on ilusti konvergeerunud. Ühtlasi on pildil posterioorsed jaotused fititud koefitsientidele.

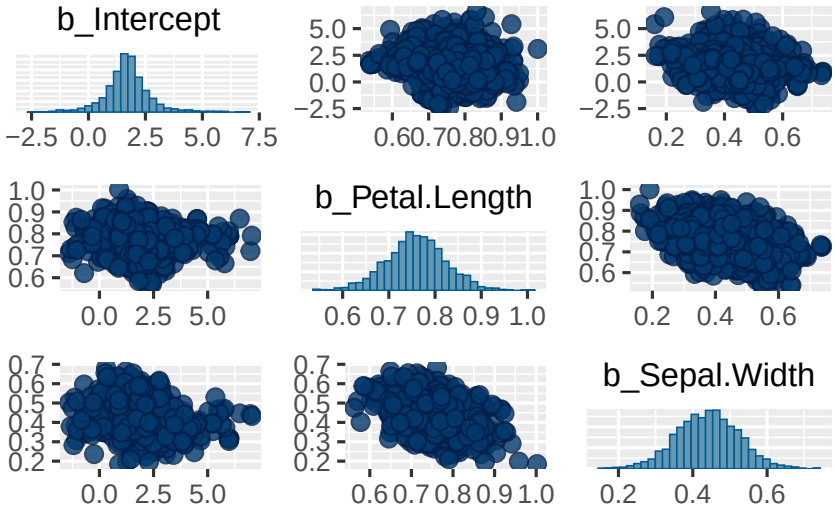
Regular expressioni abil saab plottida mudeli madalama taseme ahelaid & posteerioreid, mida plot() vaikimisi ei näita.

```
plot(m2, pars = "r_")
```



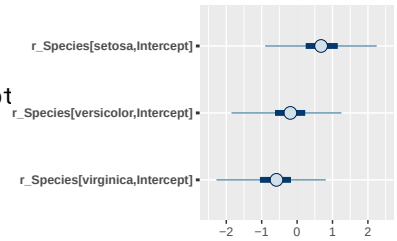
Vaatame korrelatsioone erinevate parameetrite posterioorsete valimite vahel. (Markovi ahelad jooksevad n-mõõtmelises ruumis, kus n on mudeli parameetrite arv, mille väärtusi hinnatakse.)

```
pairs(m2, pars = "b_")
```



Siin on posteeriorite põhjal arvutatud 50% ja 95% CI ja see plotitud.

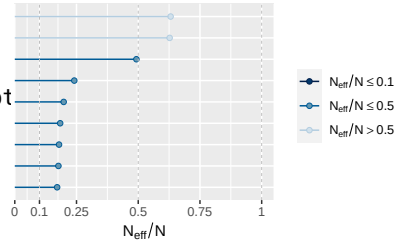
```
stanplot(m2, pars = "r_", type = "intervals")
```




```
## Warning: Method 'stanplot' is deprecated. Please use 'mcmc_plot
```

type = argument sisestamine võimaldab plottida erinevaid diagnostilisi näitajaid. Lubatud sisendid on “hist”, “dens”, “hist_by_chain”, “dens_overlay”, “violin”, “intervals”, “areas”, “acf”, “acf_bar”, “trace”, “trace_highlight”, “scatter”, “rhat”, “rhat_hist”, “neff”, “neff_hist” “nuts_acceptance”, “nuts_divergence”, “nuts_stepsize”, “nuts_treedepth” ja “nuts_energy”.

```
stanplot(m2, type = "neff")
```




```
## Warning: Method 'stanplot' is deprecated. Please use 'mcmc_plot
```

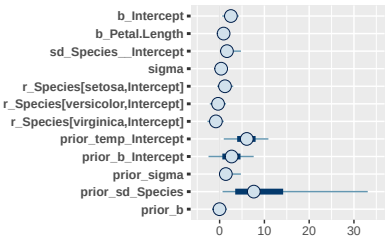
Neff on efektiivne valimi suurus ja senikaua kuni Neff/N suhe ei ole < 0.1, pole põhjust selle pärast muretseda.

Korjame ahelad andmeraami ja plotime fititud koefitsiendid CI-dega

```
model <- posterior_samples(m1)
```

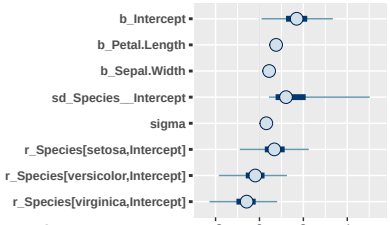
mcmc_intervals() on bayesplot paketi funktsioon. me plotime 50% ja 95% CI-d.

```
pars <- colnames(model)
mcmc_intervals(model, regex_pars = "[^(lp__)]")
```

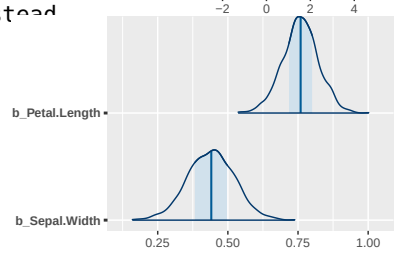


Näeme, et sigma hinnang on väga usaldusväärne, samas kui gruppide vahelise sd hinnang ei ole seda mitte (pane tähele posterioorse jaotuse ebasümmeetrilisust).

```
model2 <- posterior_samples(m2)
pars <- colnames(model2)
mcmc_intervals(model2, regex_pars = "[^(lp__)]")
```




```
mcmc_areas(model2, pars = c("b_Petal.Length", "b_Sepal.Width"))
```




```
## Warning: ‘expand_scale()‘ is deprecated; use ‘expansion()‘ instead
```

Bayesi versioon R-ruudust ja loo.

Kui suurt osa koguvarieeruvusest suudavad mudeli prediktorid seletada?

```
rstantools::bayes_R2(m2)
```

```
##      Estimate   Est.Error      Q2.5     Q97.5
## R2 0.8596074 0.008305745 0.8402698 0.8727995
```

`https://github.com/jgabry/bayes_R2/blob/master/bayes_R2.pdf` Annab põhjenduse sellele statistikule (mille arvutamine erineb tavalisest vähimruutudega arvutatud mudeli R^2-st). R^2 mõõdab mudeli fitti samade andmete peal, mida kasutati selle mudeli fittimiseks. Kui me lisame oma mudelile muutujaid, mis sisaldavad ainult müra ja ei oma mitte mingit seost y-muutujaga, siis R^2 tõuseb. Seega ei tee R^2 vahet heal fitil ja mudeli ülefittimisel, kus mudeli struktuur ennustab müra, mitte signaali.

Siin tuleb appi leave-one-out-cross-validation, ehk loo, kus me fitime sama mudelit üha uuesti ja uuesti oma andmestikul, kust on igal fitil eemaldatud üks andmepunkt. Siit saame fitinäitaja, mis hindab mudeli fitti out-of-sample ehk uutel andmetel. See arvutus tehakse tegelikult nutika *ad hoc* meetodiga, mis fitib reaalselt mudeli ainult ühel korral ja loo tulemused ei ole siiski ideaalsed out-of-sample fiti kirjeldused, aga see on parim, mis meil on. Loo on väga üldine mõõdik, mida saab kasutada kõikvõimalike erinevate mudelite võrdlemisel, mis on fititud täpselt samadel andmetel.

Võrdleme siin näiteks 4 mudelit. Eelmises mudelis (m1: Sepal.Length ~ Petal.Length + (1 | Species)) ennustame muutuja Sepal.Length väärtusi Petal.Length väärtuste põhjal shrinkage mudelis, kus iga irise liik on oma grupis.

Teine mudel, mis sisaldab veel üht ennustavat muutujat (Sepal.Width):

```
m2 <- brm(Sepal.Length ~ Petal.Length + Sepal.Width + (1 | Species), data = iris,
    prior = prior)
```

Kolmandaks ühetasemeline mudel, mis vaatab kolme irise liiki eraldi:

```
m3 <- brm(Sepal.Length ~ Sepal.Width + Petal.Length * Species, data = iris, prior = prior)
```

Ja lõpuks mudel, mis paneb kõik liigid ühte patta:

```
m4 <- brm(Sepal.Length ~ Petal.Length + Sepal.Width, data = iris, prior = prior)

loo1 <- loo(m1)
loo2 <- loo(m2)
loo3 <- loo(m3)
loo4 <- loo(m4)
loo_compare(loo1, loo2, loo3, loo4)

##    elpd_diff se_diff
## m3   0.0       0.0
## m2  -0.8       1.7
## m4 -10.3       5.0
## m1 -13.2       5.3
```

Siin võidab interaktsioonimudel m3, aga m2, kus lihtne interaktsioon on asendatud mitmetasemelise interaktsiooniga, on pea-aegu sama hea out-of-sample fitiga. Samas m4 ja m1 mudelid on omavahel väga sarnased. m1 on mitmetasemeline mudel, mis ei sisalda Sepal.Width muutjat. m4 küll sisaldab seda muutujat, aga ilma interaktsioonita. Nende mudelite erinevus parima fitiga mudelitest m3 & m2 on > 2 standardvea (se_diff), mida loetakse nõrgavõitu tõendusmatrjaliks, et nende mudelite out-of-sample fit tõesti erineb oluliselt (vähemalt 4-5 standardviga erinevus oleks tugev tõendusmaterjal). [32]

Mudelite võrdlus loo abil, kus osad mudelid on fititud logaritmitud prediktoritel ja teised algses lineaarses skaalas, ei ole otse võimalik (loo_compare annab veateate). Enne selle võrdluse tegemist tasub lugeda Gelman et al (2020) lk 202.

[32] Me ei süvene siin LOO või elpd statistilisse mõttesse, sest bayesi mudelite võrdlemine on kiiresti arenev ala, kus ühte parimat lahendust pole veel leitud. Keda see teema huvitab, siis Gelman, Hill, Vehtari (2020) "Regression and Other Stories" ptk 11 annab vastused.

```
loo_R2(m2)

##        R2
## 0.8537187
```

Me võime arvutada ka nn loo R-ruudu, mis annab hinnangu out-of-sample fitile, mis on otse võrreldav Bayesi R-ruuduga. Antud juhul on loo R-ruut 0.854 ja Bayesi R-ruudu hinnang jääb vahemikku 0.84 ja 0.87. Seega (1) ei ole meil siin põhjust arvata, et mudel oleks üle-fititud ja (2) mudeli ennustusjõud on erakordselt hea, mis tähendab, et vaid ca. 15% y-i varieeruvusest ei ole seletatud mudelisse pandud regressorite poolt (ja seegi võib olla ülehinnang, sest muutes mudelis olemasolevate regressorite omavahelisi interaktsioone võiksime vähemalt teoreetiliselt R-ruutu veelgi tõsta).

Bayesi R-ruut hindab mudeli fitti valimiga ja loo_R-ruut püüab hinnata mudeli fitti populatsiooniga, millest see valim tõmmati. Seega me usume, et loo_R-ruut on madalam kui Bayesi R_ruut. Küsimus on ihtsalt: kui palju madalam, ja millise absoluutväärtusega. Mida lugeda sobivalt suureks R-ruudu väärtuseks sõltub uurimisküsimusest – seega kontekstist. Ka 0.1 võib mõnes kontekstis olla ülisuur R-ruudu näit.

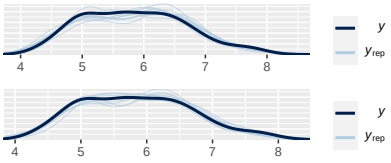
Oluline on siiski märkida, et loo abil parima mudeli välja selgitamine, et siis selle ühe mudeliga edasi minna, ei ole alati parim lahendus. Kui teie eesmärk on genereerida mudeli abil uusi andmeid, mis oleks võimalikult sarnased valimiandmetega, siis võib olla parem strateegia neid andmeid genereerida üle mitme mudeli (ennustuse üle mudelite keskmisetamine). Kui eesmärk on aga mudeli rakendamine uutel valimitel samast populatsioonist, siis tasuks kasutada näiteks post-strtifitseerimist, et viia valimi peal ehitatud mudel paremasse vastavusse populatsiooniga, millest see valim tõmmati.

Plotime mudeli poolt ennustatud valimeid – posterior predictive check

Kui mudel suudab genereerida simuleeritud valimeid, mis ei erine väga palju empiirilisest valimist, mille põhjal see mudel fititi, siis võib-olla ei ole see täiesti ebaõnnestunud mudeldamine. See on loogika posterioorse ennustava ploti taga.

Vaatame siin simultaanselt kõigi kolme eelnevalt fititud mudeli simuleeritud valimeid (y_rep) võrdluses algsete andmetega (y):

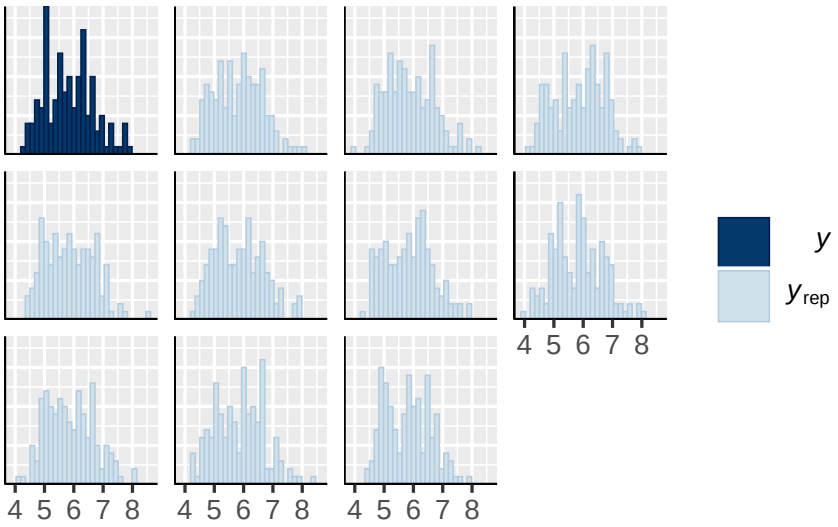
```
purrr::map(list(m2, m3), pp_check, nsamples = 10) %>%
grid.arrange(grobs = ., nrow = 3)
```



- y - tihedusplot empiirilistest andmetest
- y_rep – plotid mudeli poolt ennustatud iseseisvatest valimitest (igaüks sama suur kui empiiriline valim y) Jooniselt on näha, et m3 ennustused on võrreldes m1 ja m2-ga kõige kaugemal tegelikust valimist.

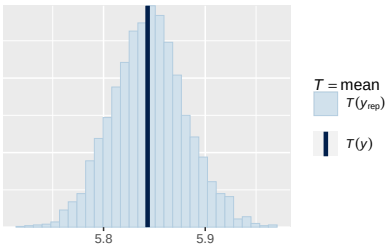
the same as before, but here we compare histograms of original data and 10 model-generated datasets of the same size as the original. For m3 only.

```
pp_check(m3, type = "hist")
```



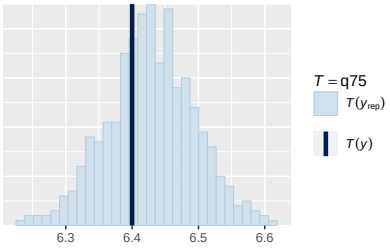
How well model-generated data captures the original data mean. we plot the histogram of data means from many model-generated datasets and compare this with the original data mean (single value shown as vertical line).

```
pp_check(m3, type = "stat", stat = "mean")
```



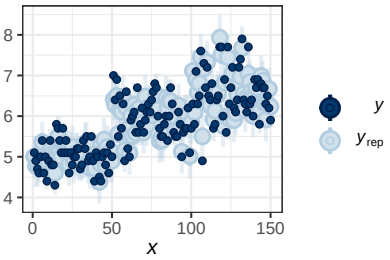
How well model-generated data captures the original data 75th quantile.

```
q75 <- function(y) quantile(y, 0.75)
pp_check(m3, type = "stat", stat = "q75", nsamples = 500)
```



For each datapoint (darkblue point), how the model (lightblue point with CI-s - darker lines are 50% CI, lihter lines are 90% CIs) predicts data that has the same x predictors values than this original datapoint. A good model would capture the original datapoints inside CIs 90% of the time. But then, a good model would also have short ranges for its CI-s.

```
pp_check(m3, type = "intervals") + theme_bw()
```



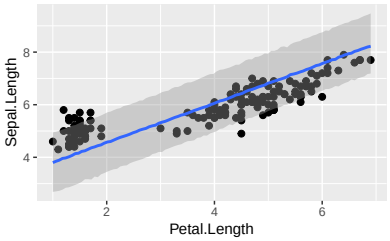
NB! Neid plotte saab kasutada ka prior predictive check-ides, ainus erinevus on, et `pp_check()`-i tuleb sisestada mudeliobjekt, mis on brms-iga `fittitud sample_prior = "only"` reziimis.

Plotime mudeli ennustusi - marginal effects plots

Teeme ennustused. Kõigepealt ennustame ühe keskmise mudeliga, mis ei arvesta mitmetasemelise mudeli madalamte tasemete koefitsientidega.

```
plot(marginal_effects(m2, effects = "Petal.Length",
                      method = "predict", probs = c(0.1, 0.9)),
     points = TRUE)
```

```
## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```

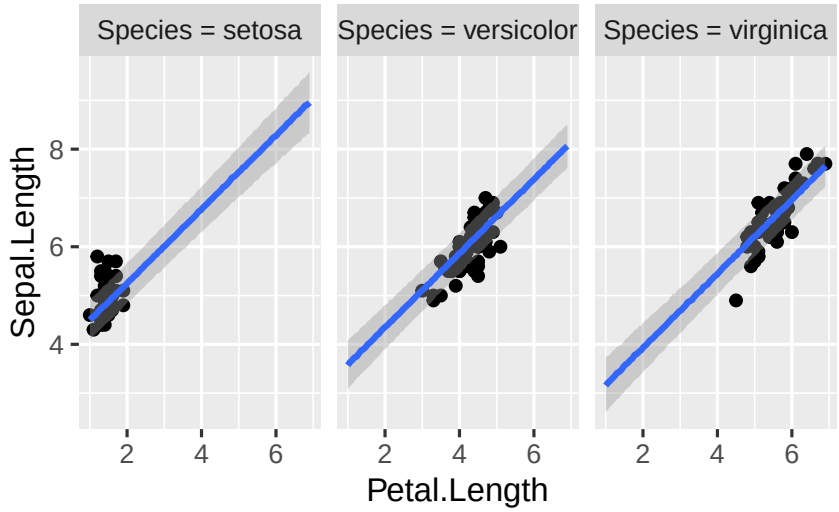


Ennustus on selles mõttes ok, et vaid väike osa punkte jääb sellest välja, aga laiavõitu teine!

Nüüd ennustame sama mudeli põhjal igale liigile eraldi. Seega kasutame mudeli madalama taseme koefitsiente. Peame andma lisaparameetri `re_formula = NULL`, mis tagab, et ennustuse tegemisel kasutatakse ka mudeli madalama taseme koefitsiente.

```
plot(marginal_effects(m2, effects = "Petal.Length", method = "predict", conditions = make_conditions(iris,
                      probs = c(0.1, 0.9), re_formula = NULL),
     points = TRUE)
```

```
## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```



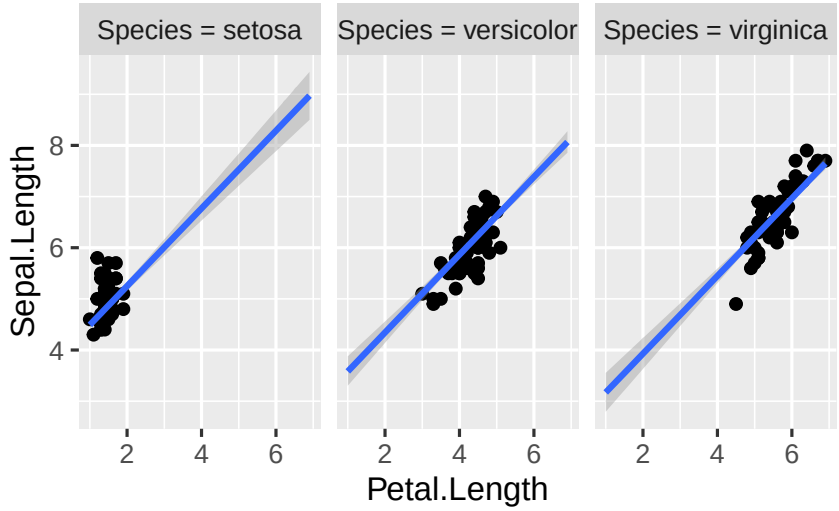
`method = "predict"` ennustab, millisesse vahemikku peaks mudeli järgi jääma 90% andmepunkte (k.a. uued andmepunktid, mida pole veel valimisse korjatud).

Tõesti, valdav enamus valimi punkte on intervallis sees, mis viitab et mudel töötab hästi. Seal, kus on rohkem punkte, on intervall kitsam ja mudel usaldusväärsem.

Järgneval pildil on `method = "fitted"`. Nüüd on enamus punkte väljaspool usaldusintervalle, mis sellel pildil mõõdavad meie usaldust regressioonijoone vastu.

```
plot(marginal_effects(m2, effects = "Petal.Length", method = "fitted", conditions = make_conditions(iris,

                      probs = c(0.1, 0.9), re_formula = NULL),

     points = TRUE)
```

```
## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```



`method = "fitted"` annab CI regressioonijoonele.

Argumendid:

- method – predict annab veapiirid (95% CI) mudeli ennustustustele andmepunkti tasemel. fitted annab veapiirid mudeli fitile endale (joonele, mis tähistab keskmist või kõige tõenäolisemat y muutuja väärtust igal x-i väärtusel)
- conditions - andmeraam, kus on kirjas mudeli nendele ennustatavatele (x) muutujatele omistatud väärtused, mida ei joonistata x teljele. Kuna meil on selleks mudeli madalama taseme muutuja Species, siis on lisaks vaja määrata argument *re_formula = NULL*, mis tagab, et ennustuste tegemisel kasutatakse mudeli kõikide tasemete fititud koefitsiente. re_formula = NA annab seevastu keskmise fiti üle kõigi gruppide (irise liikide)
- probs annab usaldusintervalli piirid.

Pane tähele, et argument `points` (ja muud lisaargumendid, nagu näiteks `theme`) kuuluvad `plot()`, mitte `marginal_effects()` funktsioonile.

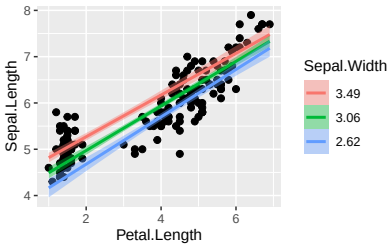
Tavaline interaktsioonimudel, aga pidevatele muutujatele.

```
m5 <- brm(Sepal.Length ~ Petal.Length + Sepal.Width + Petal.Length * Sepal.Width,
    data = iris, prior = prior, family = gaussian)
```

Kõigepealt plotime mudeli ennustused, kuidas Sepal Length sõltub Petal Length-ist kolmel erineval Sepal width väärtusel.

```
plot(marginal_effects(m5, effects = "Petal.Length:Sepal.Width"),
     points = TRUE)
```

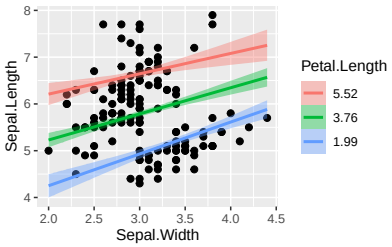
```
## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```



Ja siis sümmeetriliselt vastupidi.

```
plot(marginal_effects(m5, effects = "Sepal.Width:Petal.Length"),
     points = TRUE)
```

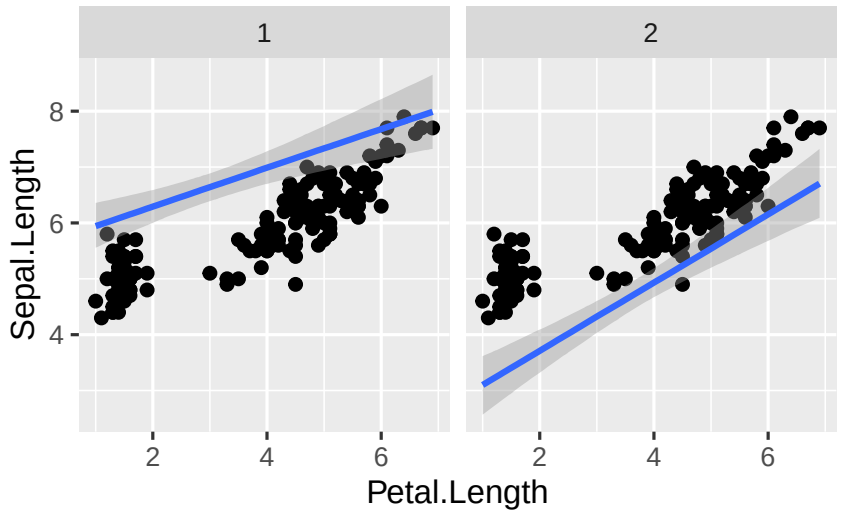
```
## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```



Siin lisame enda soovitud Sepal Width väärtused (5 ja 1.2), mis on väljaspool seda, mida loodus pakub. Pane tähele ennustuse laiemaid CI-e.

```
conditions <- data.frame(Sepal.Width = c(5, 1.2))
plot(marginal_effects(m5, effects = "Petal.Length",
                      conditions = conditions, re_formula = NULL),
     points = TRUE)
```

```
## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```



Alternatiivne tee

Alternatiivne millele? Teeme tabeli nende väärtustega, millele tahame mudeli ennustusi. Tabelis newx on spetsifitseeritud mudeli kõikide X muutujate väärtused! Me ennustame Y väärtusi paljudel meie poolt võrdse vahemaaga ette antud petal length väärtustel, kusjuures me hoiame sepal width väärtuse alati konstantsena tema valimi keskmisel väärtusel ja vaatame ennustusi eraldi kahele liigile kolmest. Liigid on mudeli madala taseme osad, seega kasutame ennustuste tegemisel mudeli kõikide tasemete koefitsiente.

```
newx <- expand.grid(Petal.Length = modelr::seq_range(iris$Petal.Length, n = 150),
    Sepal.Width = mean(iris$Sepal.Width), Species = c("setosa", "virginica"))
```

`expand.grid()` lõõb tabeli pikaks nii, et kõik võimalikud kombinatsioonid 3st muutujast on täidetud väärtustega.

`re_formula` NULL mudeldab eraldi liigid eraldi mudeli madalama taseme (liikide sees) koefitsiente kasutades

```
predict_interval_brms2 <- predict(m2, newdata = newx, re_formula = NULL) %>%
cbind(newx, .)
```

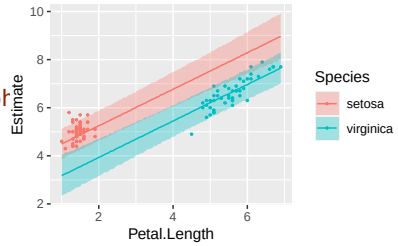
```
head(predict_interval_brms2)
```

```
##   Petal.Length Sepal.Width Species Estimate Est.Error     Q2.5    Q97.5
## 1     1.000000    3.057333  setosa 4.481417 0.3182085 3.831351 5.096920
## 2     1.039597    3.057333  setosa 4.523718 0.3152949 3.884371 5.147852
## 3     1.079195    3.057333  setosa 4.554434 0.3287716 3.922124 5.200628
## 4     1.118792    3.057333  setosa 4.578329 0.3187754 3.957441 5.226798
## 5     1.158389    3.057333  setosa 4.611136 0.3238057 3.987544 5.225735
## 6     1.197987    3.057333  setosa 4.641934 0.3181634 4.048390 5.275335
```

`predict()` ennustab uusi petal length väärtusi (Estimate veerg) koos usaldusinetrvalliga neile väärtustele

Siin siis eraldi ennustused kahele liigile kolmest, kaasa arvatud petal length väärtusvahemikule, kus selle liigi isendeid valimis ei ole (ja võib-olla ei saagi olla)

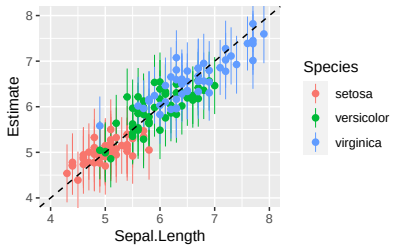
```
no_versicolor <- iris%>% dplyr::filter(Species != "versicolor")
ggplot(data = predict_interval_brms2, aes(x = Petal.Length, y = Estimate)) +

  geom_point(data = no_versicolor,

             aes(Petal.Length, Sepal.Length, color = Species), size=0.5) +

  geom_line(aes(color = Species)) +

  geom_ribbon(aes(ymin = Q2.5, ymax = Q97.5, fill = Species), alph
```



Ennustav plot - kuidas lähevad kokku mudeli ennustused reaalsete y-i andmepunktidega?

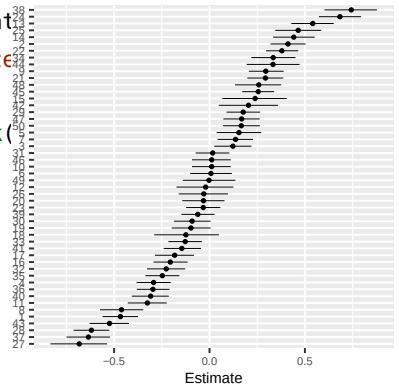
> NB! Tähtis on alati plottida mudeli ennustus y teljele. Vastupidi plottides ei jää ka ideaalsed ennustused intercept = 0, slope = 1 joonele.

```
pr <- predict(m2) %>% cbind(iris)
ggplot(pr, aes(Sepal.Length, Estimate, color = Species)) +
geom_pointrange(aes(ymin = Q2.5, ymax = Q97.5), size = 0.3) +
geom_abline(lty = 2) +
coord_cartesian(xlim = c(4, 8), ylim = c(4, 8))
```



Igale andmepunktile – kui palju erineb selle residuaal 0-st kui hästi ennustab mudel just seda andmepunkti. Ruumi kokkuhoiuks plotime välja ainult irise valiku 50-st andmepunktist.

```
set.seed(69)
as_data_frame(residuals(m2)) %>%
sample_n(50) %>%
ggplot(aes(x = reorder(seq_along(Estimate), Estimate), y = Estimat
geom_pointrange(aes(ymin = Q2.5, ymax = Q97.5), fatten = 0.1, size
coord_flip() +
theme(text = element_text(size = 8), axis.title.y = element_blank(
xlab("Residuals (95 CI)")
```



Ok, isendid nr 15 ja 44 paistavad olema vastavalt palju suurema ja väiksema Sepal Lengthiga kui mudel ennustab. Võib küsida, miks?

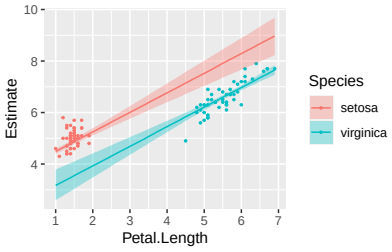
Nüüd plotime usaldusintervalli mudeli fitile ('keskmisele' Y väärtusele igal määratud X-i väärtusel), mitte Y-ennustusele andmepunkti kaupa. Selleks on hea fitted() funktsioon. Me ennustame m2 mudelist

vastavalt newdata parameetriväärtustele. Kui me newdata argumendi tühjaks jätame, siis võtab fitted() selleks automaatselt algse iris tabeli (ehk valimi väärtused).

```
predict_interval_brms2f <- fitted(m2, newdata = newx, re_formula = NULL) %>% cbind(newx,
    .) %>% mutate_if(is.numeric, round, 2)
head(predict_interval_brms2f) %>% kable()
```

Petal.Length	Sepal.Width	Species	Estimate	Est.Error	Q2.5	Q97.5
1.00	3.06	setosa	4.49	0.05	4.38	4.59
1.04	3.06	setosa	4.52	0.05	4.41	4.62
1.08	3.06	setosa	4.55	0.05	4.45	4.65
1.12	3.06	setosa	4.58	0.05	4.48	4.68
1.16	3.06	setosa	4.61	0.05	4.51	4.71
1.20	3.06	setosa	4.64	0.05	4.54	4.74

```
ggplot(data = predict_interval_brms2f,

       aes(x = Petal.Length, y = Estimate, color = Species)) +

  geom_point(data = no_versicolor,

             aes(Petal.Length, Sepal.Length, color = Species), size=0.5) +

  geom_line() +

  geom_ribbon(aes(ymin = Q2.5, ymax = Q97.5, fill = Species),

              alpha = 1/3, colour = NA) +

  scale_x_continuous(breaks = 0:10)
```



Mudeli genereeritud andmed ja valimiandmed mõõtmisobjekti (subjekti e taimeisendi) kaupa. See on sisuliselt posterior predictive plot (vt eespool).

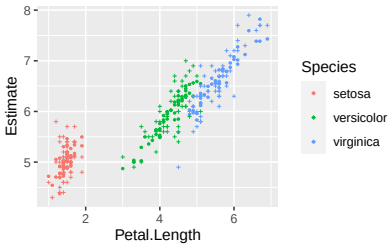
```
predicted_subjects_brms <- predict(m2) %>% cbind(iris, .)
```

`predict()` arvutab mudeli põhjal uusi Y muutuja andmepunkte. Võib kasutada ka väljamõeldud andmete pealt Y väärtuste ennustamiseks (selleks tuleb anda ette andmeraam kõigi X-muutujate väärtustega, mille pealt tahetakse ennustusi).

Punktid on ennustused ja ristikesed on valimiandmed

```
ggplot(data = predicted_subjects_brms,
```

```
      aes(x = Petal.Length, color = Species)) +
  geom_point(aes(y = Estimate), size=0.5) +
  geom_point(aes(y = Sepal.Length), shape = 3, size=0.5)
```



Alternatiiv – ansambliennustus

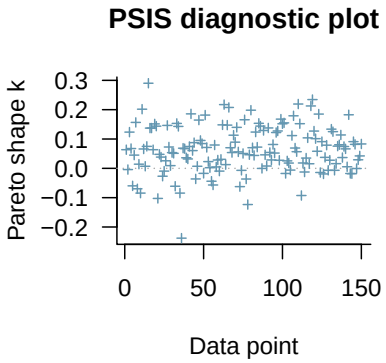
Kuna meil on 2 mudelit (m2 ja m3) mis on pea võrdselt eelistatud, siis genreerime ennustused mõlemast (mudelite ansamblist) proportsionaalselt nende waic skooridega. See ennustus kajastab meie mudeldamistööd tervikuna, mitte ühte "parimat" mudelit ja seega võib loota, et annab paremini edasi meie mudeldamises peituvat ebakindlust.

```
pp_a <- pp_average(m2, m3, weights = "waic", method = "predict") %>%
as_tibble() %>%
bind_cols(iris)
ggplot(data = pp_a, aes(x = Petal.Length, color = Species)) +
geom_point(aes(y = Estimate), size = 0.5) +
geom_point(aes(y = Sepal.Length), shape = 3, size = 0.5)
```

Mudeli eelduste kontroll

Pareto k otsib nn mõjukaid (influential) andmepunkte.

```
loo_m2 <- loo(m2)
plot(loo_m2)
```



Kui paljud andmepunktid on kahtlaselt mõjukad?

```
loo::pareto_k_table(loo_m2)
```

```
##
## All Pareto k estimates are good (k < 0.5).
```

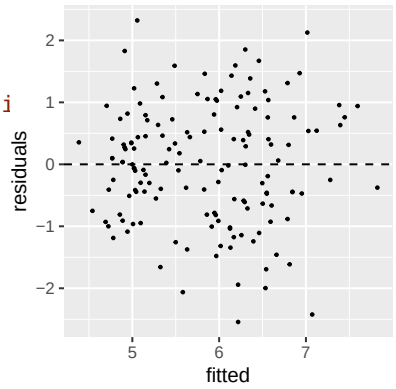
Plotime residuaalid

`resid()` annab residuaalid vektorina. Kõigepealt plotime residuaalid fititud (keskmiste) Y väärtuste vastu:

```
resid <- residuals(m2, type = "pearson")
```

```
## Warning: Type 'pearson' is deprecated and will be removed in the future.
```

```
fit <- fitted(m2)
ggplot() +
geom_point(aes(x = fit[, "Estimate"], y = resid[, "Estimate"]), si
geom_hline(yintercept = 0, lty = "dashed") +
labs(x = "fitted", y = "residuals")
```

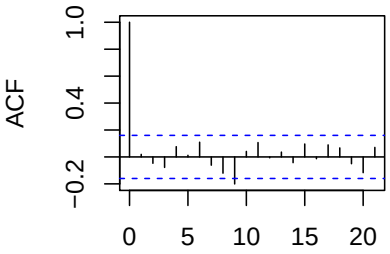


`type = "pearson"` annab standardiseeritud residuaalid $R = (Y - Y_p)/SD(Y)$, kus SD(Y) on hinnang Y-muutuja SD-le. alternatiiv on `type = "ordinary"`, mis annab tavalised residuaalid.

Residuals vs fitted plot testib lineaarsuse eeldust - kui .resid punktid jaotuvad ühtlaselt nulli ümber, siis mudel püüab kinni kogu süstemaatilise varieeruvuse teie andmetest ja see mis üle jääb on juhuslik varieeruvus.

Vaatame diagnostilist plotti autokorrelatsioonist residuaalide vahel.

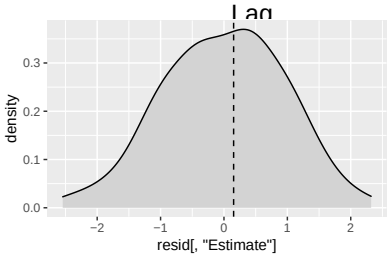
```
acf(resid[, 1])
```



Residuaalide autokorrelatsioonid on madalad - seega kõik paistab OK ja andmepunktide sõltumatus on tagatud.

Siin on residuaalide histogramm:

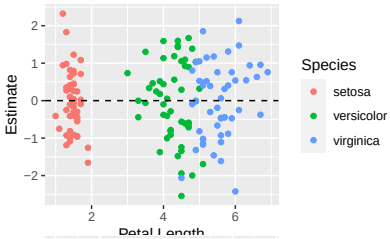
```
ggplot(data = NULL) +
geom_density(aes(x = resid[, "Estimate"]), fill = "lightgrey") +
geom_vline(xintercept = median(resid), linetype = "dashed")
```



Residuaalid on sümmeetrilise jaotusega ja meedian residuaal on peaaegu null. See on kõik hea.

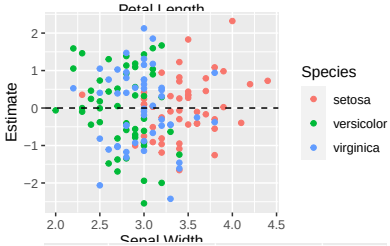
Ja lõpuks plotime standardiseeritud residuaalid kõigi x-muutujate vastu. Kõigepealt ühendame resid vektori irise tabeliga, et oleks mugavam plottida. residuaalid standardhälbe ühikutes saab ja ka tuleks plottida kõigi x-muutujate suhtes.

```
iris2 <- iris %>% cbind(resid)
ggplot(iris2, aes(Petal.Length, Estimate, color = Species)) +
geom_point() +
geom_hline(yintercept = 0, linetype = "dashed")
```

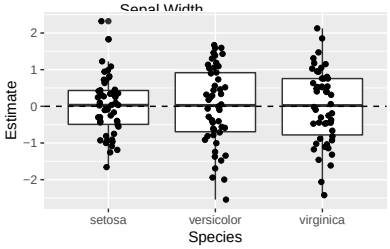


Tsiteerides klassikuid: “Pole paha!”. Mudel ennustab hästi, aga mõne punkti jaoks on ennustus 2 sd kaugusel.

```
ggplot(iris2, aes(Sepal.Width, Estimate, color = Species)) +
geom_point() +
geom_hline(yintercept = 0, linetype = "dashed")
```




```
ggplot(iris2, aes(Species, Estimate)) +
geom_boxplot() +
geom_hline(yintercept = 0, linetype = "dashed") +
geom_jitter(width = 0.1)
```



GLM - üldine lineaarne regressioon

GLM on tavapärase lineaarse mudeli laiendus, mis

(1) lubab kasutada ka teisi andmemudeleid (tõepärafunktsioone) peale normaalse ja studenti

(2) kasutab additiivseid lineaarseid protsessimudeleid, smamoodi nagu tavaline lineaarne regressioon

(3) lubab transformeerida lineaarse mudeli y - muutujat nn link funktsiooniga või, matemaatiliselt ekvivaletselt x- muutujaid nn inverse-link funktsiooniga.

Binoomjaotusega mudelid

Binoomjaotus

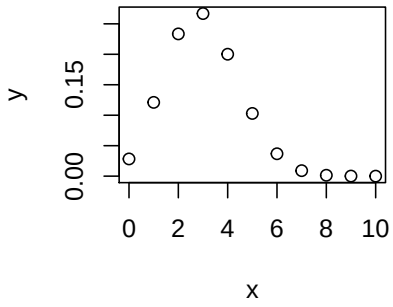
Kui teil on binaarne muutuja (sellel saab olla ainult kaks väärtust, näiteks sees/väljas, 1/0), mis kajastab sõltumatuid sündmusi, siis modelleerib seda binoomjaotus $y \sim Binomial(n, p)$, kus n on edukate sündmuste arv ja p on nende suhteline sagedus ($p = n/N$; N on kõikide sündmuste koguarv).[33] Tehniliselt on binoomjaotusel veel omadus, et valim võetakse asendustega, mis tähendab, et iga valimisse tõmmatud sündmus/mitte-sündmus pannakse populatsiooni tagasi, kus seda saab uuesti samasse valimisse tõmmata. Väljaspool abstraktset loogikat tähendab see, et populatsioon, millest valim tõmmatakse, peab olema palju suurem kui valim. N-i kasvades läheneb binoomjaotuse kuju normaaljaotusele.

[33] Sõltumatud sündmused on sellised, kus ühe sündmuse esinemise järgi ei saa ennustada teise sündmuse esinemist (st puudub korrelatsioon sündmuste esinemise vahel). Näiteks, kui me anname ravimit/platseebot N inimesele ja mõõdame surmasid, siis eeldusel, et ravimit saavate erinevate patsientide surmad on üksteisest sõltumatud ja sama tõenäosusega, saame kasutada binoomjaotust.

Binoomjatuse keksväärtus (keskmine sündmuste arv) on $N \times p$ ja standardhälve on $\sqrt{Np(1-p)}$. Kui Np võrdub täisarvuga, siis mediaan = mood = keskväärtus. Standardviga proportsioonile p on $\sqrt{\frac{p(1-p)}{N}}$ See standardviga (*standard error*) on teisisõnu standardhälve meie hinangule proportsiooni väärtusele. Kui n = 0 või N - n = 0, siis on selline SE arvutus eksitav.

```
n <- 10  # sündmuste koguarv
x <- seq(0, n)  # kõik võimalikud õnnestumiste arvud 10st sündmusest
```

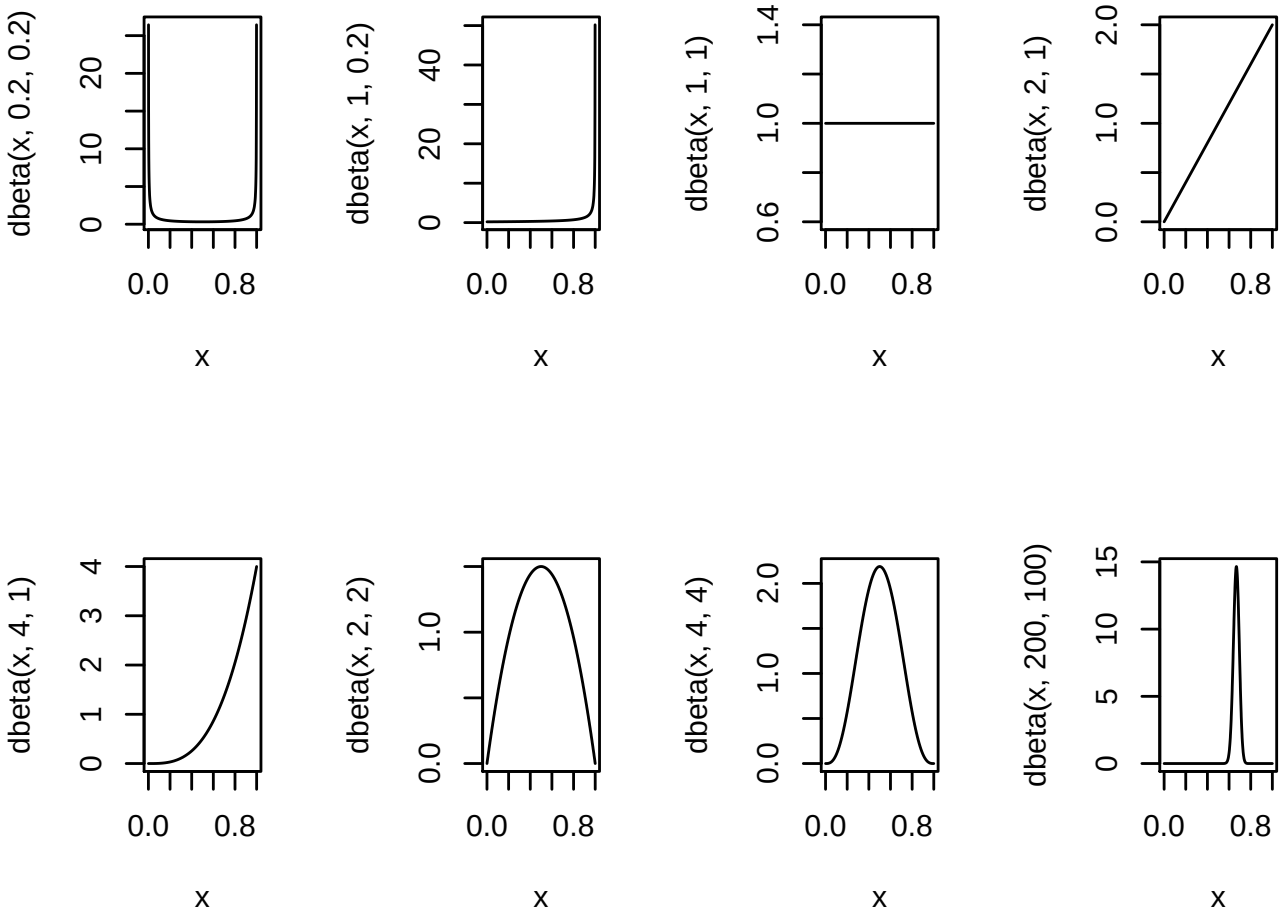
```
p <- 0.3  # 30% õnnestumisi (sagedus)
y <- dbinom(x, n, p)
plot(x, y)
```



Kui binoomjaotuse andmemudelile lisada betajaotuse prior, siis tuleb posteerior betajaotusega. Selle posterior tipp on määratud valemiga $\frac{\alpha+y}{\alpha+\beta+n}$ ning see jääb alati valimi proportsiooni y/n ja priori keskmise $\alpha/(\alpha+\beta)$ vahele. Posteeriori sd tuleb valemist $\sqrt{\frac{(\alpha+y)(\beta+n-y)}{(\alpha+\beta+n)^2(\alpha+\beta+n+1)}}$. See posterior läheneb n-i kasvades normaaljaotusele, milline lähenemine on kiirem, kui me avaldame tõenäosuse logit-skaalas logaritmitud kihlveoshansidena $log(p/(1-p))$.

Beta-prior katab vahemiku ost 1ni ja sellel on 2 parameetrit, a ja b.

Siin mõned näited erinevatest betajaotuse parametriseeringutest.



$beta(\theta \mid a, b)$ jaotuse keskväärtus on
$\mu = a/(a+b)$
ja mood on
$\omega = (a-1)/(a+b-2)$ (kui $a > 1$ ja $b > 1$).

Seega, kui a = b, siis on keskmine ja mood 0.5. Kui a > b, on keskmine ja mood > 0.5 ja kuid a < b, on mõlemad < 0.5.

Beta jaotuse "laiuse" annab "kontsentratsioon" $\kappa = a + b$. Mida suurem κ, seda kitsam jaotus.

$a = \mu\kappa$
$b = (1 - \mu)\kappa$
$a = \omega(\kappa - 2) + 1$
$b = (1 - \omega)(\kappa - 2) + 1$ kui $\kappa > 2$

Me võime κ-le omistada väärtuse nagu see oleks mündivisete arv, mis iseloomustab meie priori tugevust (juhul kui tõepära funktsioon tuleb andmetest, mis koosnevad selle sama mündi visetest). Kui meie jaoks piisaks ainult mõnest mündiviset, et priorist (eelnevast teadmisest) lahti ütelda, peaks meie prior sisaldama väikest kappat.

Näiteks, mu prior on, et münt on aus ($\mu = 0.5$; $a = b$), aga ma ei ole selles väga veendunud. Niisiis ma arvan, et selle eelteadmise kaal võrdub sellega, kui ma oleksin näinud 8 mündiviske tulemust. Seega $\kappa = 8$, mis tähendab, et $a = \mu\kappa = 4$ ja $b = (1 - \mu)\kappa = 4$. Aga mis siis kui me tahame beta priorit, mille mood $\omega = 0.8$ ja $\kappa = 12$? Siis saame valemist, et $a = 9$ ja $b = 3$. Selline psedo-andmeline olemus on ühine kõigile kojugaatsetele prioritele (vt `https://en.wikipedia.org/wiki/Conjugate_prior`).

nn dose-response kurviga hinnatakse LD50, ehk doosi, millega suremus saavutab 50%. Regressioonimudeli, y = a + bx korral, kus x on doos, LD50 = -a/b. Kui $\beta \leq 0$, siis ei ole LD50 kontseptuaalselt mõttekas. Seega tasub LD50 posteeriori arvutamisel võtta arvesse ainult need b väärtused, mis on nullist suuremad.

Logistiline regressioon kasutades benoulli v binoomjaotust

Tavalises lineaarses regressioonis on tavapärane, et kuigi me ennustame pidevat y-muutujat, on kas osad või kõik X-muutujad mittepidevad. Sellest pole kurja, meie mudelid jooksevad nii pidevate kui mittepidevate prediktoritega. Me eeldame küll, et y-muutuja on normaalne, aga ei eelda midagi sellist x-muutujate ehk prediktorite kohta. Samuti on lubatud prediktorite mitte-lineaarsed funktsioonid, nagu X_1X_2 või X^2, senikaua kui regressioonivõrrandi parameetrid (a, b_1, … b_n) on lineaarsetes additiivsetes suhetes. Aga kuidas käituda, kui meie poolt ennustataval Y-muutujal on vaid kaks võimalikku väärtust, 0 ja 1, ning ta on binoomjatusega?

Binoomjaotuse $y \sim Binomial(n, p)$ kuju määrab kaks parameetrit, n (katsete üldarv) ja p (sündmuse toimumise tõenäosus). Kuna n on enamasti teada ja me tahame lineaarselt ennustada p-d, siis vajame link-funktsiooni, mis painutaks lineaarse mudeli ennustusjoont nii, et see jääks alati vahemikku [0, 1]. Selleks kasutame logit linki $logit(p) = \alpha + \beta \times x$, kus $logit(p) = \log\left(\frac{p}{1-p}\right)$, mis tähendab, et logit(p) on naturaallogaritm kihlveoshanssidest. Siit saab omakorda tuletada, et igale x-i väärtusele vastab sündmuse toimumise tõenäosus p, kusjuures $p = \frac{exp(a+bx)}{1+exp(a+bx)}$. Nüüd oleme transformeerinud regressioonivõrrandi parempoolse osa, ja see nn logistiline pöörd-link

on matemaatiliselt ekvivalentle võrrandi parema poole transformeerimisega logit linki abil. Ehk teisisõnu, logistilise transformatsiooniga saame me võrrandi vasakul poolel olevad tõenäosused tagasi algsesse [0, 1] tõenäosuste skaalasse. Sellist lähenemist binoomjaotusele (või bernoulli jaotusele) kutsutakse logistiliseks regressiooniks.

Me teeme n katset ja kodeerime iga eduka katse 1-ga ja mitteeduka katse 0-ga. Kui $n = 1$, siis y on ühtedest ja nullidest koosnev vektor (muutuja) ja p on tõenäosus, et suvaline katse annab tulemuseks 1. Eeldades binoomjaotust (või bernoulli jaotust) ning logit linki on siin tegu logistilise regressiooniga. Kui $n > 1$, siis on tegu aggregeeritud binoomse logistilise regressiooniga. Me demonstreerime allpool mõlemaid.

Logistiline regressioon

Kui me püüame ennustada binaarse y-muutuja oodatavaid väärtusi tõenäosustena, ehk 1-de arvu suhet katsete koguarvu, siis tavaline lineaarne regressioon ei garanteeri, et ennustused jäävad 0 ja 1 vahele, ehk tõenäosuste skaalasse.

Eespool õppisime transformeerima andmeid, et paremini täita regressiooni eeldusi (lineaarsust ja normaalsust). Nüüd aga ei transformeeri me mitte andmeid, vaid mudeli võrrandit ennast selleks, et suruda mudeli ennustused tõenäosusskaalasse. Selliste transformeeritud mudelite ehk GLM-ide (*Generalized Linear Model*) levinuim näide on logistilise regressiooni mudel. Logistilises regressioonis ei modelleeri me mitte otse y väärtusi (1 ja 0) erinevatel x-i väärtustel, vaid tõenäosust $P(Y = 1|X)$ [loe: tõenäosus, et $Y = 1$, eeldades kindlat x-i väärtust].

Logistiline regressioon kasutab logistilist transformatsiooni, mis näiteks funktsioonile $y = a + bx$ on

$$P(Y = 1|X) = \frac{exp(a + bx)}{1 + exp(a + bx)}$$

$exp(a)$ tähendab e astmes a, kus e on Euleri arv, ehk arv, mille naturaallogaritm on 1 (seega on e naturaallogaritmi alus). e on umbes 2.71828 ja selle saab valemist $(1 + 1/n)^n$, kui n läheneb lõpmatusele.

Logistiline transformatsioon viib lineaarse regressiooni tavapärasest y-muutuja skaalast [$-\infty$,$+\infty$] tõenäosuste skaalasse [0, 1], andes sirge asemele S-kujulise kurvi, mis läheneb asümptootiliselt ühelt poolt 0-le ja teiselt poolt 1-le. Logistilise funktsiooni põõrdväärtus on logit funktsioon, mis annab "*odds*-i" ehk shansi ehk kihlveosuhte tõenäosusele p: $odds = \frac{p}{1-p}$. Tõenäosuse p logit ehk logit(p) on sama, mis log(odds):

$$logit(p) = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p)$$

y = a + bx mudeli korral tavalises meetrilises skaalas on *odds* eksponent fititud lineaarsest mudelist:

$$odds = \frac{P(Y = 1 \mid X)}{1 - P(Y = 1 \mid X)} = exp(a + bx)$$

ja ekvivalentselt

$$log(odds) = logit(p) = a + bx$$

.

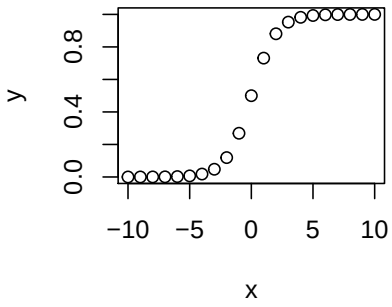
Matemaatiliselt pole vahet, kas me transformeerime prediktorid logistilise funktsiooniga või ennustatava muutuja logit funktsiooniga – need on sama asja erinevad kirjeldused.

Kuidas suhtuvad *odds*-d tõenäosustesse? Näiteks tõenäosus 0.2 (20%) tähendab, et $odds = 0.2/(1 - 0.2) = 1/4$ ehk üks neljale ja tõenäosus 0.9 tähendab, et $odds = 0.9/(1 - 0.9) = 9$ ehk üheksa ühele. *Odds*-e kasutavad näiteks hipodroomid, sest nii on mänguril lihtne näha, et kui kihlveokontori poolt mingile hobusele omistatud odds on näiteks üks nelja vastu ja ta maksab kihlveo sõlmimisel 1 euro, siis ta saab võidu korral 4 eurot kasu (ehk 5 eurose kupüüri). Logaritm *odds*-idest ongi logit transformatsioon, mille pöördväärtus on omakorda logistiline transformatsioon!

Suvalise arvu α logistiline funktsioon on logiti põõrdväärtus:

$$logit^{-1}\alpha = logistic(\alpha) = \frac{exp(\alpha)}{1 + exp(\alpha)}$$

```
x <- -10:10
y <- exp(x)/(1 + exp(x))
plot(y ~ x)
```



Kui me logistilise regressiooniga fititud mudeli $y = a + bx$ korral muudame x-i väärtust ühe ühiku võrra, siis muutub *log-odds* b võrra, mis on sama, mis õelda, et *odds* muutub $exp(b)$ võrra. Samas b ei vasta $P(Y = 1|X)$ muutusele X-i muutumisel ühe ühiku võrra. See, kui kiiresti $P(Y = 1|X)$ muutub, sõltub X-i väärtusest. Siiski, senikaua kuni $b > 0$, kaasneb X-i kasvuga alati tõenäosuse $P(Y = 1)$ kasv (ja vastupidi).

Kahe tõenäosuse logitite vahe on sama, mis logaritm *odds-ratio*-st (log(OR) ehk shanside suhe)

$$log(OR) = logit(p_1) - logit(p_2)$$

Odds-ratio

Kui meil on 2 katsetingimust (ravim/platseebo) ning 2 väljundit (näit elus/surnud), siis

- a - ravim/elus juhutude arv,
- b - ravim/surnud juhutude arv,

- c - platseebo/elus juhutude arv,
- d - platseebo/surnud juhutude arv.

$$OR = \frac{a/b}{c/d}$$

- OR = 1 Katsetingimus ei mõjuta väljundi *odds*-e
- OR > 1 Katsetingimus tõstab väljundi *odds*-e
- OR < 1 Katsetingimus langetab väljundi *odds*-e

Logistiline regressioon üldistab OR-i kaugemale kahest binaarsest muutujast. Kui meil on binaarne y-muutuja ja binaarne x-muutuja (X_1), pluss rida teisi x-muutujaid ($X_2 ... X_n$), siis mitmese logistilise regressiooni X_1-e tõusukoefitsient β_1 on seotud tingimusliku OR-ga. $\exp(\beta_1)$ annab Y ja X vahelise OR-i, tingimusel, et teiste X-muutujate väärtused on fikseeritud (see on tavaline sõltumatute muutujatega lineaarse regressiooni beta-koefitsientide tõlgendamise tingimus).

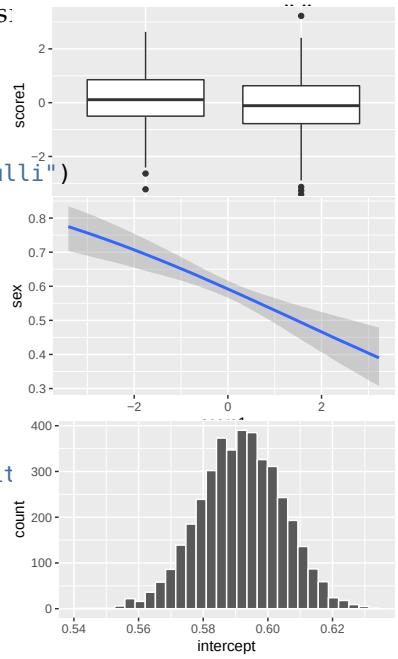
OR-i kui suhtelise efekti suuruse tõlgendamine sõltub sündmuse y = 1 baastõenäosusest. Näiteks kui surm põhjusel x on tavapäraselt väga haruldane ja mingi keskkonnamõju annab OR = 10, siis tegelik tõus suremuses (surma tõenäosus keskkonnamõju tingimustes) võib olla tühine.

Rakendame logistilist regressiooni brms-is. Selleks ennustame inglise kooliõpilaste sugu nende matemaatikatestide tulemuste põhjal. 0 - on poiss ja 1 - on tüdruk, score1 on kirjaliku testi tulemus ja score2 suulise vabas vormis testi oma. Me standardiseerime score1 nii, et selle keskväärtus oleks 0 ja sd oleks1. Poistel on score1 veidi kõrgem:

```
schools <- read_csv("raamat/data/schools.csv")
```

```
## Parsed with column specification:
## cols(
##   school = col_double(),
##   student = col_double(),
##   sex = col_double(),
##   score1 = col_double(),
##   score2 = col_double()
## )
```

```
schools$score1 <- scale(schools$score1)
ggplot(schools, aes(factor(sex), score1)) + geom_boxplot()
```




```
m_logistic1 <- brm(sex ~ score1, data = schools, family = "bernoulli")
write_rds(m_logistic1, "raamat/data/m_logistic1.fit")

m_logistic1 <- read_rds("raamat/data/m_logistic1.fit")
conditional_effects(m_logistic1)
```

posterior interceptile tõenäosusksaalas:

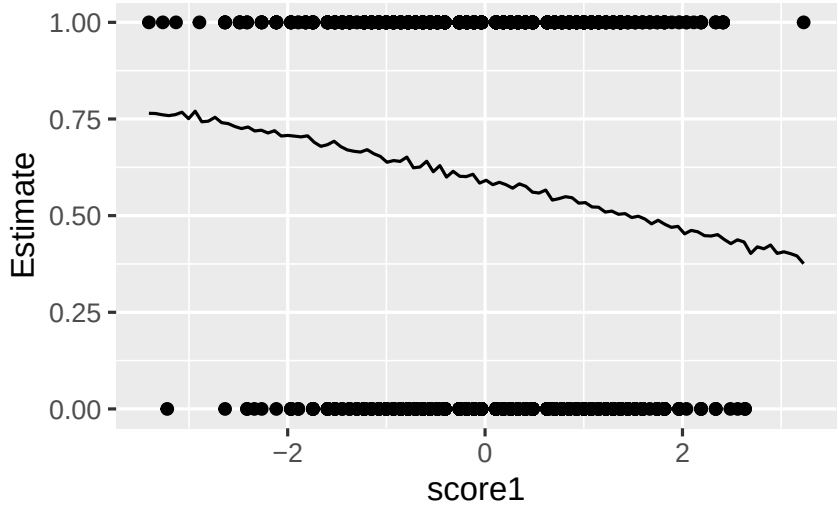
```
mlog1_post <- posterior_samples(m_logistic1)
intercept <- inv_logit_scaled(mlog1_post$b_Intercept)
ggplot(data = NULL, aes(intercept)) + geom_histogram(color = "whit

median(inv_logit_scaled(mlog1_post$b_Intercept))

## [1] 0.5920171
```

Kui õpilase score1 = kõikide õpilaste keskväärtusel, siis tõenäosus, et see õpilane on tüdruk = 59%

```
median(inv_logit_scaled(mlog1_post$b_Intercept + mlog1_post$b_score1))

## [1] 0.5296253
```

Kui õpilase score1 ületab kõikide õpilaste keskväärtuse ühe standardhälbe võrra, siis tõenäosus, et see õpilane on tüdruk = 53%.

```
newx <- expand.grid(score1 = modelr::seq_range(schools$score1, n = 100))

predict_interval_brms2 <- predict(m_logistic1, newdata = newx, re_formula = NULL) %>%

cbind(newx, .)

ggplot(data = predict_interval_brms2, aes(x = score1, y = Estimate)) +
geom_point(data = schools, aes(score1, sex)) +
geom_line()
```



Mida kõrgem on score1, seda väiksema tõenäosusega on mudeli arvates selle välja teeninud tüdruk.

confusion matrix ja ROC

Kui meie ennustatud tõenäosus >= 0.5, siis võtame vastu otsuse "tüdruk" ehk 1, vastasel korral aga ütleme "poiss" ehk 0. See cutoff 0.5 on arbitraarne. Seejärel vaatame, kuidas sellise tõenäosuse cutoffi puhul meie ennustused (otsused) vastavad tegelikkusele.

```
sch <- drop_na(schools)
glm.probs <- predict(m_logistic1, type = "response") %>% as.data.frame()
glm.probs <- glm.probs %>%

  mutate(pred=case_when(Estimate >= 0.5 ~ 1,

                        Estimate < 0.5 ~ 0))
sch <- sch %>% bind_cols(glm.probs)
```

```
act_1_pred_0 <- dplyr::filter(sch, sex == 1, pred == 0) %>% nrow()
act_1_pred_1 <- dplyr::filter(sch, sex == 1, pred == 1) %>% nrow()
act_0_pred_1 <- dplyr::filter(sch, sex == 0, pred == 1) %>% nrow()
act_0_pred_0 <- dplyr::filter(sch, sex == 0, pred == 0) %>% nrow()
tribble(~sex, ~predicted_right, ~predicted_wrong, "poiss", act_0_pred_0, act_0_pred_1,
    "tüdruk", act_1_pred_1, act_1_pred_0) %>% kableExtra::kable()
```

sex	predicted_right	predicted_wrong
poiss	52	572
tüdruk	845	54

```
count(sch, sex, pred) %>% kableExtra::kable()
```

sex	pred	n
0	0	52
0	1	572
1	0	54
1	1	845

sensitiivsus = korrektselt ennustatud tüdrukute (1) osakaal spetsifilisus = korrektselt ennustatud poiste (0) osakaal

Tõstes sensitiivsust langetame spetsiifilisust ja vastupidi. Seda näitab ROC-i kurv. Kui me ennustame alati 0-i (poissi), siis on meie ennustuse spetsiifilisus 1 ja sensitiivsus 0.

sensitivity

```
851/(851 + 48)
```

```
## [1] 0.9466073
```

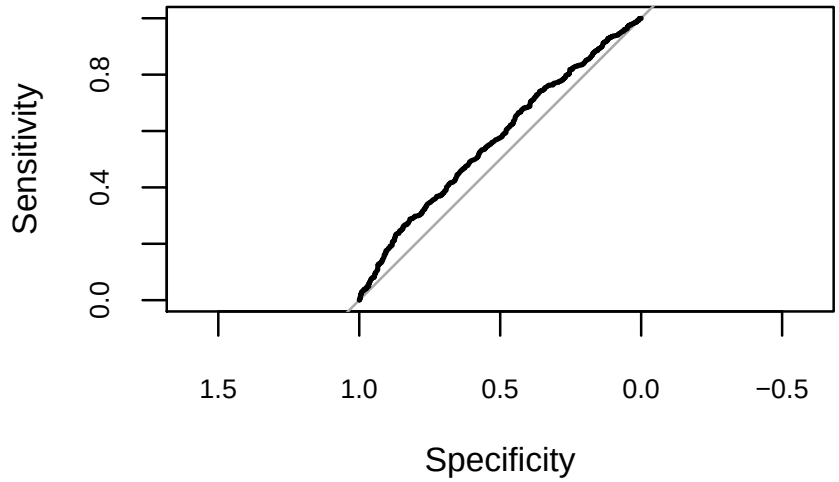
specificity

```
55/(55 + 569)
```

```
## [1] 0.08814103
```

Siin oleme selgelt raskustes, sest meie tõenäosuscutoff 0.5 eelistab sensitiivsust spetsiifilisele. Meil oleks vaja leida enda jaoks sobiv cut-off, mis meie jaoks optimiseeriks sensitiivsuse ja spetsiifilisuse. Selle optimumi leidmiseks joonistame välja sensitiivsuste ja spetsiifilisuste seose kõikidel ennustustõenäosuste cutoffidel. Seda spetsiifilisus vs sensitiivsus kurvi kutsutakse ajaloolistel põhjustel ROC kurviks ehk *Receiver Operating Characteristic curve* (see meetod töötati välja 2. maailmasõja ajal radarisignaalide analüüsimiseks).

```
roccurve <- roc(sch$sex ~ glm.probs$Estimate)
plot(roccurve, cex.axis = 0.7, cex.lab = 0.8)
```



Kui me klassifitseerime poisse ja tüdrukuid juhuslikult münti visates, siis ROC kurv tuleb sirge, nagu näidatud joonisel ja kurvialune pindala = 0.5. Ideaalse ennustuse korral on kurvi alune pindala 1. Mei ROC kurv on näitab mudeli üsna tagasihoidlikku ennustusjõudu, auc on ainult 0.57.

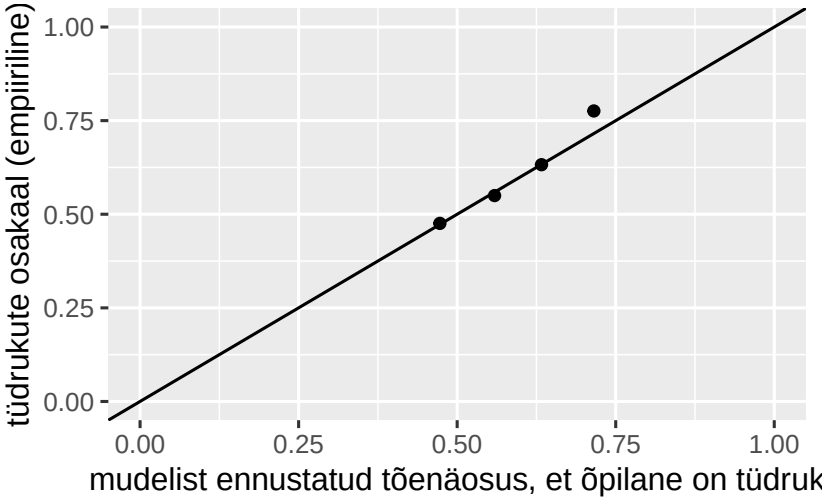
```
auc(roccurve)
```

```
## Area under the curve: 0.5703
```

Kurvialuse pindala = 0.57 tähenab, et kui me võtame juhusliku poisi ja juhusliku tüdruku, siis 57% tõenäosusega annab mudel tüdrukule (1) suurema tõenäosuse olla tüdruk (1) kui poisile (0) olla tüdruk (1).

Kalibreerime mudeli ennustatud tõenäosused empiiriliste suhete (tüdrukud/õpilased) vastu:

```
# jagan score1 andmed 4 võrdsesse intervalli
sch$sc1_fact <- cut_interval(sch$score1, n = 4)
df1 <- count(sch, sc1_fact, sex) %>%
pivot_wider(names_from = sex, values_from = n) %>%
mutate(empirical_fraction = `1`/(`0` + `1`))

df2 <- sch %>% dplyr::filter(sex == 1) %>%
group_by(sc1_fact, sex) %>%
summarise(estimate = median(Estimate))
df2 <- full_join(df1, df2[, -2])
ggplot(df2, aes(estimate, empirical_fraction)) +
geom_point() +
geom_abline() +
xlim(0, 1) + ylim(0, 1) +
ylab("tüdrukute osakaal (empiiriline)") +
xlab("mudelist ennustatud tõenäosus, et õpilane on tüdruk")
```



ROC kurv näitab, kui hästi meie ennustused kahte gruppi üksteisest lahutavad ja kalibreerimine näitab, kas tõenäosused tähendavad päriselt seda, mida me neilt ootame (kas ennustatud tõenäosused vastavad empiirilistele suhetele)

Poissoni jaotus

Kui sündmuse toimumiseks on väga palju võimalusi, mis on aga kõik ükshaaval väga vähetõenäolised, siis võime sündmuste arvu modelleerida Poissoni jaotusega, mis matemaatiliselt on binoomjaotuse erijuht.[34]

Me eeldame veel, et modelleeritavad sündmused on ajas üksteisest sõltumatud ja vahetatavad (nende mõõtmise järjekord ei oma tähtsust), et sündmuste toimumise sagedus ei muutu, et kaks sündmust

[34] Poissoni jaotus modelleerib üksikuid haruldasi ja sõltumatuid diskreetseid sündmusi, mille arvu me saame üles lugeda. Näiteks surmi ajaühiku kohta või pommitabamusi pindalaühiku kohta.

ei saa toimuda täpselt samal ajal/kohas ning et sündmuse toimumise tõenäosus on proportsionaalne intervalli pikkusega/suurusega ajas või ruumis. Poissoni jaotusel on üksainus parameeter: keskväärtus, ehk keskmine sündmuste arv intevallis, ehk λ. Poissoni jaotuse standardhälve on lihtsalt ruutjuur sündmuste arvust $\sqrt{\lambda}$.

Matemaatiliselt näeb Poissoni jaotus välja niiviisi (k on sündmuste arv intervallis):

$$P(k) = e^{\lambda} \times \frac{\lambda^k}{k!}$$

Poissoni (ja negatiivne binoom) regressioonimudel näeb välja niimoodi:

$$log[E(Y|X)] = \beta X$$

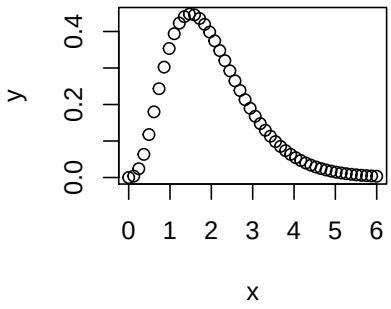
ehk

$$E(Y|X) = e^{\beta X}$$

Beeta-koefitsient näitab sündmuste (Y-i countide) protsentuaalset kasvu X-i ühe-ühikulise kasvu korral.

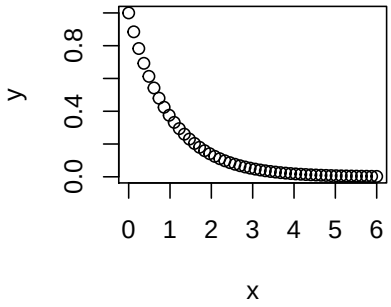
Kui Poissoni andmemudelile lisada gammajaotuse prior, saame gammajaotusega posteeriori $Gamma(\alpha + y, \beta + x)$ kus α ja β on gammajaotuse parameetrid. Gammajaotuse keskväärtus on α/β ja sd on $\sqrt{\alpha}/\beta$.

```
x <- seq(0, 6, length.out = 50)
y <- dgamma(x, 4, 2)
plot(x, y)
```

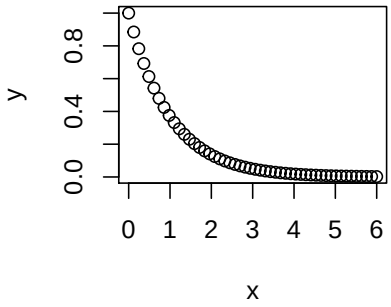


Poissoni näide: parameeter θ, millele vajame hinnangut, on haigussurmade arv inimese kohta aastas. n on populatsiooni suurus, y on empiiriline surmade arv 10 aasta jooksul. Siis andmemudel on $y \sim Poisson(10n\theta)$. Prior võiks olla $Gamma(20, 430000)$, mille keskväärtus tuleb 20/430000 = 4.65e-5 ja sd = sqrt(20)/430000 = 1.04e-5. Siis posteerior on $Gamma(20 + y, 430000 + 10n)$, mille keskväärtus ja sd on lihtsalt arvutatav (vt ülalpool). Kui me tahame ennustada uusi y-i väärtusi (andmepunkte ehk surmade arvusid), siis ennustav jaotus on negatiivne binoomial $negBinom(\alpha, \beta/10n)$. Teine võimalus selleks ennustuseks on: 1) tõmba $Gamma(20, 430000)$ jaotusest näit 1000 juhuslikku arvu (need on θ väärtused); 2) igale neist tuhandest tõmba Poissoni jaotusest $Pois(10000\theta)$ vastav 1 arv (meie populatsiooni suurus on 1000 ja surmad on ennustatud 10 aasta peale). Nüüd lugedes kokku saadud 0, 1, 2, 3 jne saadki ennustused igale 10-aasta surmade arvu tõenäosusele.

```
x <- seq(0, 6, length.out = 50)
y <- dgamma(x, 1, 1)
plot(x, y)
```




```
x <- seq(0, 6, length.out = 50)
y <- dexp(x, 1)
plot(x, y)
```



Millal kasutada Poissoni jaotust, ja millal binoomjaotust? Kui iga andmepunkti saab vaadelda kui edukate katsete arvu suhet kõikide katsete arvule, siis kasuta binoomjaotust/logistilist regressiooni. Kui aga andmepunkti väärtusel pole loomulikku piiri (see on lihtsalt mingit tüüpi sündmuste arv), kasuta Poissoni/logaritmilist regressiooni.

Poissoni jaotuse kasutamisel bioloogiliste andmetega on üks väike probleem - enamasti on bioloogilistel countidel suurem dispersioon kui Poissoni jaotus seda ette näeb. Sellisel juhul tuleks tõepärafunktsioonina kasutada Poissoni jaotuse asemel negatiivset binoomjaotust. Selleks et testida, kas teie mudeli y-mmutuja, mis koosneb countidest, on liiga suure dispersiooniga, kasutame funktsiooni `pscl::odtest()`, mis võrdleb mudelis fititud residuaalide jaotust poissoni- ja negbinoom jaotustega. Kui poissoni mudel ei sobi, siis p väärtus tuleb madal ja on mõistlik kasutada negbinoomset mudelit.

```
library(pscl)
library(MASS)
r <- MASS::glm.nb(y ~ x, data = data)
# kasuta seda mudelit, mida tahad ise brm()-s fittida.
odTest(r)
```

eksponentsiaaljaotus

Seda jatust kasutatakse poissoni jaotusega sündmuseni kuluva aja modelleerimiseks (sündmused on ajas sõltumatud ja vahetatavad). Üldisemalt kasutatakse eksponentsiaaljatustjaotust nullst suuremate, sageli ajas muutuvate, reaalarvuliste muutujate modelleerimiseks. Eksponentsiaaljaotus on mäluta, ehk tõenäosus, et mõõteobjekt elab veel ajaühiku t ei sõltu sellest, kui kaua ta on juba elanud. Eksponentsiaaljaotus on gammajaotuse erijuht, kus $gamma(\alpha, \beta)$ parameetrite asemel on parameetrid 1 ja θ (ja θ on "rate"). Kui me lisame eksponentsiaalsele andmemudelile gammajaotusega priori tuleb posteerior gammajaotusega $Gamma(\alpha + 1, \beta + x)$. Gamma priorit võib vaadelda $\alpha - 1$ eksponentsiaalse andmepunktina, mille totaalne ooteaeg on β.

Üle-dispersioonilised binaarsete sündmuste mudelid

Kuidas käituda, kui teie andmeid (Y-muutujat) ei tooda mitte üks protsess, vaid protsesside segu, st. iga andmepunkt võib olla toodetud omaenese protsessi poolt. Ja need protsessid tulevad omakorda mingist jaotusest. Näiteks, kui me mudeldame erinevate vanusegruppide pikkusi koos, siis peaks tõepärafunktsioon olema normaaljaotuste segu. Selline funktsioon on õnneks olemas – see on studenti t-jaotus, mille kujuparameeter – nu – ütleb, kui erinevate SD-ga on selle segu komponentjaotused.

Kui meil on tegu mündiviskesituatsiooniga, kus me viskame mitut münti, millel igaühel on erinev kulli saamise tõenäosus, siis saame kasutada binoomjaotuste segujaotust, mida kutsutakse beeta-binoomjaotuseks. Ja poissoni protsessi korral, kus iga sündmus võib olla pärit erinevast possoni jaotusest, saame kasutada negatiivset binoomjaoust, ehk gamma-piossoni jaotust (need nimed on sünonüümid).

Kui meil on tegu mõne sellise protsessiga, siis oleks parim lahendus konditsioneerida mudel X-muutuja(te)ga, mis viiksid Y-jaotuse lihtsale kujule (normaaljaotus, binoomjaotus, jne). Kui seda teha ei saa, siis järgmine trikk on kasutada neid eelpoolmainitud nn üledisperseeritud jaotusmudeleid.

Need on nn pidevad mixture mudelid, kus iga binoom- või poissoni jaotusega count eeldatakse omavat isiklikku beta- või gammajaotusega edutõenäosust. Pidevad selle pärast, et beta ja gammajaotus on pidevad jaotused. Praktikas on sageli parem alternatiiv pidevatele segumudelitele mitme-tasemelised (hierarhilised) mudelid, millest lähemalt allpool.

Poissoni mudeli dispersioon on λ ja $sd = \sqrt{\lambda}$. Binoomjaotuse dispersioon on $np(1-p)$. Kui tegelik dispersioon on suurem, siis on meil tegu üledispersiooniliste andmetega, mida mudeldame kasutades teisi jaotusi.

Beeta-binoomjaotus eeldab, et igal katsel on oma realiseerumise tõenäosus (sündmuse toimumise tõenäosus) ja mudel annab meile nende tõenäosuste jaotuse, mitte ühe tõenäosuse kõigile sündmustele. X-muutujad muudavad selle jaotuse kuju, aga ei määra iga individuaalse sündmuse toimumise tõenäosust. Beeta-jaotus on 2 parameetriline: $\hat{p}$ on keskmine tõenäosus ja θ kirjeldab jaotuse ulatust (laiust). $\theta = 2$ annab ühtlase jaotuse 0 ja 1 vahel, kui $\theta < 2$, siis koondub jaotuse mass 0 ja 1 juurde kahte tippu.

Mudel näeb välja selline

$$y \sim dBetaBinomial(n, \hat{p}, \theta)$$

$$logit(\hat{p}) = a + bx$$

pluss priorid a-l, b-le ja θ-le. Beetajaotuse alune parameetriruum on 0st 1ni, seega tõenäosusskaala. Selleks, et saada tasast beetajaotust: $\theta = 2$ ja kui $\theta > 2$, siis koondub priori tihedus kuhugi. Seega on hea, kui θ priori tihedus algab 2-st ja langeb sealt. Me saame väikese trikiga θ priori ümber defineerida:

$$\theta = \phi + 2$$

$$\phi \sim exponential(1)$$

Selline mudel lubab igale andmepunktile oma intercepti ja tõusu, mis tõmmatakse beeta-jaotusest, mille keskväärtus on $\hat{p}$ ja dispersioon on θ.

Negatiivne binoomjaotus (ehk gamma-poisson) eeldab, et iga Poissoni sündmus toimub oma sagedusega. Gammajaotus annab nende sageduste jaotuse. Jaotusel on kaks parameetrit, λ ja ϕ.

$$y_i \sim NegBinomial(\lambda_i, \phi)$$

Kui ϕ läheneb nullile, läheneb jaotus sama μ-ga normaaljaotusele. ϕ on alati positiivne ja määrab gamma-poissoni jaotuse dispersiooni nii: $\lambda + \lambda^2/\phi$. Suurem ϕ lähendab seda jaotust puhtale poissonile.

λ-le saab lisada lineaarse mudeli log-linki abil. NB! Bioloogilised protsessid, mida te hataksite mudeldada Poissoni tõepäramudeliga, tasub alati proovida mudeldada ka negbinoom tõepäraga, ja siis mudeleid võrrelda (loo ja pp-checkiga, näiteks).

Zero-inflated mudelid

Meil on kaks erinevat mehhanismi, mis võivad anda tulemuseks null sündmust, millistest ainult üks võib anda ka >0 sündmuse. Seega lahendame koos 2 mudelit: binoomjaotuse mudel mudeldab protsessi, mis võib anda 0, 1, 2, ... sündmust ja poissoni mudel jagab binoomjaotuse mudeliga null-vaatlused kahte ossa, ehk kahe mehhanismi vahel, millest üks toodab ainult nulle ja teine toodab nulle ja kõike muud pealekauba. Seega saame binoomjaotusest selle puhtalt nulle andva protsessi osakaalu kõikides nullides ja poissoni jaotus mudeldab kõik muu.

$$y_i \sim ZIPoisson(p_i, \lambda_i)$$

$$logit(p_i) = \alpha_p + \beta_p x_i$$

$$log(\lambda_i) = \alpha_\lambda + \beta_\lambda x_i$$

Seega on meil 2 lineaarset mudelit, 4 parameetrit (ning seega 4 priorit) ja 2 erinevat link-funktsiooni. Iseenesest võivad need kaks lineaarset mudelit olla ka erinevate prediktoritega.

Näide: üledispersioonilised puusaluumurrud

Siin on meil haigekassa andmed puusaluumurdude ravi kohta (9913 raviepisoodi). Me teame patsiendi sugu (sex), vanust (age), maakonda, kus teda raviti (county), komorbiidsuste arvu (comorbidity), dementia staatust (dementia), luumurru tüüpi (fracture type), ravimeetodit (management method), akuutse haiglaravi kestust pävades (acute LOS), akuutse ravifaasi teraapia kestust tundides (acute_therapy), postakuutse haiglaravi kestust (posacute LOS) ja postakuutse teraapia kestust (postacute_therapy) ning elulemust 3 kuu mõõdudes (status_3m).

```
head(hf) %>% kableExtra::kable()
```

id	sex	age	county	comorbidity	dementia	fracture_type	management_method	acute_LOS	a
1	f	-0.7281573	harju	-0.0420625	1	pertrochanteric	osteosynthesis	7	
2	f	1.8836261	viljandi	-1.1181027	0	femoral_neck	conservative	35	
3	f	0.2979005	idaviru	-0.5800826	0	femoral_neck	osteosynthesis	12	
4	f	1.7903481	harju	-0.0420625	0	femoral_neck	osteosynthesis	4	
5	f	0.3911785	harju	-1.1181027	0	pertrochanteric	osteosynthesis	4	
6	m	-2.5937168	idaviru	-1.1181027	0	femoral_neck	osteosynthesis	44	

Mudel on zero_inflated negative binomial, mis mudeldab üledispersiooniga Poissoni olukorras, kus

```
prior = c(prior(normal(0, 10), class = b))
hf_m1_zi <- brm(bf(postacute_LOS ~ age + sex + comorbidity + dementia + management_method +
    acute_LOS + acute_therapy + county, zi ~ dementia + management_method + county),
    family = zero_inflated_negbinomial(), data = hf, prior = prior, cores = 3, chains = 1)
```

Mitmetasemelised mudelid

Mitmetasemeline mudel on regressioonimudel, kus andmed on struktureeritud gruppidesse ja mudeli koefitsiendid võivad erineda grupist gruppi.

Statistika teooria ütleb, et me peaksime oma mudelitesse hõlmama need faktorid, mida kasutati eksperimendi disainis. Mitmetasandilised mudelid on parim viis, kuidas mudelisse panna katsedisainis esinevaid erinevatel tasemetel klastreid ja samas vältida mudeli ülefittimist. Mitmetasandiline mudel kajastab sellise katse või vaatluse struktuuri, kus andmed ei grupeeru mitte ainult katse- ja kontrolltingimuste vahel, vaid ka lisaklastritesse ehk gruppidesse. Näiteks, kui me mõõdame platseebo-kontrollitud uuringus kümmet patsienti ja teeme igale patsiendile viis kordusmõõtmist (kahetasemeline mudel). Või kui meil on geeniuuring, kus uuritakse korraga 1000 valgu taset ja uuring toimub 10 laboris (3-tasemeline mudel). Või kui mõõdame kalamaksaõli mõju matemaatikaeksami tulemustele kümnes koolis, ja igas neist viies klassis (3-tasemeline mudel).

Tavapärane lähenemine oleks kõigepealt keskmistada andmed iga klassi sees ning seejärel keskmistada iga kooli sees (võtta igale koolile 5 klassi keskmine). Ning seejärel, võttes iga kooli keskmise üheks andmepunktiks, teha soovitud statistiline test (N = 10, sest meil on 10 kooli). Paraku, sellisel viisil talitades alahindame varieeruvust, mistõttu meie statistiline test alahindab ebakindluse määra arvutatud statistiku ümber. Hierarhilised mudelid, mis kajastavad adekvaatselt katse struktuuri, aitavad sellest murest üle saada. Üldine soovitus on, et kui teie katse struktuur seda võimaldab, siis peaksite alustama modelleerimist hierarhilistest mudelitest.

Mitmetasemelised mudelid on eriti kasulikud, kui teil on osades klastrites vähem andmepunkte kui teistes, sest nad vaatavad andmeid korraga nii klastrite vahel kui klastrite sees ning kannavad informatsiooni üle klastritest, kus on rohkem andmepunkte, nendesse klastritesse, kus on vähe andmeid. See tõstab hinnangute täpsust.

Tavapärane regressioonimudel on sageli vaadeldav mitmetasandilise mudeli erijuhuna. Näiteks kujutage ette mudelit, kus laste õppedukust on mõõdetud mitmes koolis. Kui gruppide (koolide) vaheline

varieeruvus on väga madal, siis annab mitmetasemeline mudel sarnase tulemuse lihtsa mudeliga, kus kõik koolid on ühte patta kokku pandud. Ja vastupidi, kui koolid on üksteisest väga erinevad, siis võime sama hästi modelleerida iga kooli eraldi ja teistest sõltumatult. Samuti, kui meil on andmeid väga väheste koolide kohta, siis võib mitmetasandilsest mudelist saadav kasu olla tagasihoidlik, sest meil pole piisavalt andmeid, et modelleerida koolide vahelist varieeruvust. Samuti, kui meil on iga kooli kohta piisavalt palju andmeid, siis saame iga kooli eraldi modelleerides praktiliselt sama tulemuse kui mitmetasemelisest mudelist. Muudel juhtudel on tõenäoliselt mõistlikum modelleerida õppedukust kahetasemelises mudelis, korraga õpilase tasemel ja kooli tasemel.

kahetasemeline mudel algebra keeles

1. tase on õpilse tase, 2. tase on klassi tase. meil on j klassi, milles on erinev arv õpilasi. Iga õpilase kohta teame populaarsusindeksit (y - muutuja) ja sugu (x - muutuja). Klassi tasemel teame õpetaja staazi aastates (z - muutuja). Alustuseks on meil j regressioonivõrrandit, eraldi võrrand igale klassile

$$y_{ij} = b_{0j} + b_{1j}X_{ij} + e_{ij}$$

(1. võrrand)

kus subskript j tähistab klassi ja i tähistab õpilast. Näit b_{1j} tähendab, et me fitime igale klassile oma b_1 koefitsiendi (ja b_{0j}, et fitime igale klassile oma b_0-i). Me eeldame, et igal klassil on erinevad b_0 ja b_1, mis tulevad vastavatest klassiülestest normaaljaotustest. Enamasti eeldame, et kõikide klasside varieeruvus on sama. Me modelleerime b_0 ja b_1 jaotusi, tuues sisse klassi tasemel muutuja z:

$$b_{0j} = \gamma_{00} + \gamma_{01}Z_j + u_{0j}$$

(2. võrrand)

$$b_{1j} = \gamma_{10} + \gamma_{11}Z_j + u_{1j}$$

(3. võrrand)

2. võrrand ennustab klassi keskmist populaarsusindeksit vastavalt õpetaja staazile.
3. võrrand ütleb, et populaarsuse ja soo seose tugevus sõltub õpetaja staazist – kui $\gamma_{11} > 0$, siis on seos seda tugevam, mida staazikam on õpetaja. u_{0j} ja u_{1j} on residuaalide vead klassi tasemel, mille kohta me eeldame, et mean = 0 ja sõltumatust residuaalide vigadest

õpilase tasemel (e_{ij}). u_{0j} residuaalide vigade dispersioon on σ^2_{u0} jne. u_{0j} ja u_{1j} covariance on σ^2_{u01} ja me ei eelda, et see = 0. Gamma koefitsiendid ei varieeru klasside vahel.

Asendades 1. võrrandis b_{0j} ja b_{1j}, saame oma võrrandisüsteemi muuta üheks pikaks võrrandiks.

$$Y_{ij} = \gamma_{00} + \gamma_{10}X_{ij} + \gamma_{01}Z_j + \gamma_{11}X_{ij}Z_j + u_{1j}X_{ij} + u_{0j} + e_{ij}$$

Siis näeme uues võrrandis liiget $\gamma_1 Z_j X_{ij}$, mis on interaktsiooniliige. Ilma interaktsioonita mudel näeb välja niimodi

$$Y_{ij} = \gamma_{00} + \gamma_{10}X_{ij} + \gamma_{01}Z_j + u_{1j}X_{ij} + u_{0j} + e_{ij}$$

juhusliku vealiige u_{ij} korrutub X_{ij}-ga, mistõttu sellest tulenev totaalne viga on erinev erinevatel X_{ij} väärtustel. Seega on meie mudel heteroskedastiline ja erineb selle poolest tavalistest lineaarsetest mudelitest, mis eeldavad homoskedastilisust e residuaalvea sõltumatust X-i väärtusest.

Teine oluline erinevus tavalisest lin mudelist on, et gupeeritud andmete puhul ei ole täidetud andmete iseseisvuse eeldus (gruppide sees modelleeritakse andmed korreleerituna - klassisisene korrelatsioon). Klassisisese korrelatsiooni saame intercept-only mudelist

$$Y_{ij} = \gamma_{00} + u_{0j} + e_{ij}$$

(selle mudeli saad, kui viskad pikast mudelist välja kõik X ja Z sisaldavad liikmed.) See mudel ajab varieeruvuse lahku kahe iseseisva komponendi vahel: σ^2_e (1. taseme vigade dispersioon e_{ij}) ja σ^2_{u0} (2. taseme vigade dispersioon u_{0j}).

Klassisisene korrelatsioon:

$$\rho = \frac{\sigma^2_{u0}}{\sigma^2_{u0} + \sigma^2_e}$$

rho annab grupistruktuuri poolt seletatud variace proportsiooni, mida võib tõlgendada kui kahe sama grupi juhusliku liikme vahelist oodatavat korrelatsiooni.

Üldiselt on kasulik töötada standardiseeritud regressioonikoefitsientidega, mida saab tõlgendada sd ühikutes. Erandiks on olukord, kus analüüsi eesmärk on võrrelda erinevaid valimeid omavahel. Enamasti on kasulik standardiseerida andmed, mis mudelisse sisse lähevad, aga on võimalik ka standardiseerida koefitsiente kui selliseid.

st_coef = (unstand_coef x sd(X))/sd(Y)

Standardiseeritud andmete kasutamine muudab varieeruvuskomponentide fitte, aga jätab b koefitsientide fitid olemuselt samaks (see

kehtib X-muutujate suvaliste lineaarsete transformatsioonide korral). Kui me jagame X-muutuja 2-ga, siis *uus* $b_1 = 2$ *x vana* b_1. Mudeli poolt seletamata varieeruvuse proportsioon ei muutu. Eelnev kehtib senikaua, kuni mudeli tõusud (b_1) ei ole vabaks lastud, st need ei varieeru klassist klassi.

Tsentreerimine ($x_i - mean(x)$) mõjutab b_0 aga mitte b_1 koefitsiente, mis võib rääkida selle meetodi kasuks üle standardiseerimise ($(\frac{x_i - mean(x)}{sd(x)})$, mis tekitab X muutujate jaotused, mille mean = 0 ja sd = 1.

NB! grupi tasemel tsentreerimine, ehkki vahest kasulik, töötab hoopis teistmoodi kui üle kõikide gruppide tsentreerimine ja viib täiesti erineva mudelini - sellest tuleks hoiduda, senikaua kui te ei tea täpselt, mida te teete.

> Mitmetasemelised mudelid on erilised, sest nad hõlmavad mitut varieeruvuse allikat, ei eelda konsantset vigade jaotust, ning modelleerivad seoseid erinevate mudeli tasemete vahel.

Aegread

Aegridasid saab analüüsida mitmetasemilisena, kus korduvad mõõtmised (1. tase) on grupeeritud indiviidide sisse (2. tase). Nii saab analüüsida ka ebaühtlase ajavahemiku järel tehtud mõõtmisi.

> Aegridade analüüsi lisaprobleem võrreldes tavalise mitmetasemelise mudeliga on, et me ei saa enam eeldada, et 1. taseme (indiviidi taseme) vead on üksteisest sõltumatud. Aegridade vaatluste vead on sageli ajas autokorreleeritud.

Y_{ti} - indiviidi i õpitulemus ajapunktis t.
T_{ti} - ajapunkt
X_{ti} - ajas muutuv covariaat - kas õpilane töötab
Z_t - ajast mittesõltuv kovariaat - sugu

1. tase (indiviidi tase):

$$Y_{ti} = \pi_{0i} + \pi_{1i} x T_{ti} + \pi_{2i} X_{ti} + e_{ti}$$

2. tase (üle indiviidie):

$$\pi_{0i} = \beta_{00} + \beta_{01} Z_t + u_{01}$$

$$\pi_{1i} = \beta_{10} + \beta_{11} Z_t + u_{11}$$

$$\pi_{2i} = \beta_{20} + \beta_{21} Z_t + u_{21}$$

Oluline punkt: aegridade modelleerimisel pakub meile sageli huvi ka ka korrelatsioon mudeli tõusu ja intercepti vahel. Kahjuks sõltub see näitaja sellest, millises skaalas me ajamuutuja mudelisse sisse anname. Oluline on tagada, et ajaskaala nullpunkt on mõtekas.

To model growth - polynomial, logistic curve (first slow change, then quick, then slow again). Logistic parameters have meaning! Cubic polynomial approximates logistic and exponential curves - but here interpretation is on the level of some predicted growth curves.

`https://facebook.github.io/prophet` sessoonsete andmete fittimine ennustavasse mudelisse

`https://github.com/nwfsc-timeseries` kogu aegridade analüüsi pakette

Temporaalne autokorrelatsioon

eeldus: Mida lähemal on 2 ajapunkti üksteisele, seda suurem on nende vaheline korrelatsioon (ja residuaalide vaheline korrelatsioon). Tavaline lm eeldab, et see korrelatsioon = 0. Seda korrelatsiooni saab kas hinnata, või selle vastu võidelda. Meie teeme siin viimast.

Brms mudel näeb välja niimoodi

```
data.temporalCor.brms <- brm(y ~ x, data = d, autocor = cor_ar(formula = ~year))
```

cor_ar() on brms funktsioon autokorrelatsiooni modelleerimiseks. selle formula on ühepoolne valem ~t või ~ t | g, mis annab ajakovariaadi t ja grupeeriva faktori g. Kui g on antud, siis modelleeritakse iga grupp iseseisvalt ja teistest gruppidest sõltumata (gruppide vahel on korrelatsioon 0).

Teine võimalus inkorporeerib mudelisse AR1 residuaalide autokorrelatsioonistruktuuri, kus korrelatsiooni eksponent väheneb ajas lineaarselt. Me eeldame, et see vähenemine toimub samamoodi sõltumata sellest, millises ajavahemikus me parasjagu oleme (statsionaarsuse eeldus).

```
data.temporalCor.brms = brm(y ~ x, data = d, autocor = cor_ar(~year, cov = TRUE))
```

Lihtsuse mõttes eeldame, et iga ajapunkti oodatud väärtus = tavaline lin prediktor + autokorrelatsiooni parameeter (ρ) korrutatuna eelmise vaatluse residuaaliga + tavapärane sõltumatu müra (σ^2).

mitmetasemeline mudel R-i mudelikeeles

Kui meie muutujad andmetabelis “data” on y = õpilase testiskoor, x = katsetingimus (binaarne faktor katse-kontroll, kalamaksaöli - platseebo), ja kool, siis “ühepajamudel” väljendub R-i mudelikeeles:

```
mudel <- lm(y ~ x, data=data)
```

ja mudel, kus iga kool on eraldi modelleeritud:

```
mudelid <- data %>% group_by(kool) %>% do(model = lm(y ~ x, data = .))
```

või purrr-i abil

```
data %>% split(.$kool) %>% map(~ lm(y ~ x, data = .)) %>% map(summary) %>% map_dfr(~ broom::glance(.), .id = "kool")
```

Seevastu hierarhiline mudel kirjutatakse kui

```
mudel <- lme4::lmer(y ~ x + (1 + x | kool), data=data)
```

või

```
mudel <- lme4::lmer(y ~ x + (1 | kool), data=data)
```

Esimesel juhul modelleeritakse igale koolile nii tõus kui intercept ja teisel juhul modelleeritakse igale koolile ainult intercept, seeläbi eeldades, et kõikidel koolidel on mudelis sama tõus, ehk kalamaksaõli efekt (ES = testitulemus kalamaksaõli grupis - testitulemus platseebogrupis). Intercept tähendab sellises mudelis enamasti baastaset (kontrolltingimus) ja tõus tähendab katseefekti (katsetingimus - kontrolltingimus). Seega eeldab teine mudel, et igas grupis võib küll olla oma baastase, aga katsefekt sellest ei muutu.

Lisaks, mudel

```
mudel <- lme4::lmer(y ~ x + (1 + x \ kool), data=data)
```

modelleerib igale koolile tõusu ja intercepti lisaeeldusega, et tõusude ja interceptide vaheline korrelatsioon puudub. Ilma selle eelduseta püüab mudel selle korrelatsiooni andmete põhjal leida. Kui andmeid on liiga vähe või mudel on liiga keeruline või korrelatsiooni võimalik esinemine tundub teadulsikult väga väheusutav, võib korrelatsiooni hindamisest loobuda, aga muidu tasub seda siiski hinnata.

```
mudel <- lme4::lmer(y ~ x + (1 + x | kool) + (1 + x | linn), data=data)
```

Kui meil on mudelis rohkem kui 2 taset, kirjutame need sõltumata sellest, kas tasemed on hierarhiliselt üksteise sees (õpilane - kool - linn) või mitte (patsient - haigla - ravimi batch)

```
mudel <- lme4::lmer(y ~ x + (1 + x | grupp1) + (1 + x | grupp2), data=data)
```

Kui esimesed 2 mudelit saab fittida lm() funktsiooniga, siis lihtne mitte-bayesiaanlik alternatiiv hierarhilise mudeli tarbeks on lme4 pakett (`https://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf`), mis on lihtsam, kiirem ja ebatäpsem *ad hoc* viis arvutada mitmetasemeliseid mudeleid, kui Stan. Selle eelis Stani ees on eelkõige kiirus ja puuduseks on väiksem paindlikus mudelite formuleerimisel ja see, et väikeste valimite ja väheste gruppide puhul töötab lme4 algoritm palju halvemini, kui bayesi lahendused. Seega kasutame me pigem Stani, kui lme4. Samas, suurepärane pakett nimega brms (`https://cran.r-project.org/web/packages/brms/`

`vignettes/brms_overview.pdf`) suudab tõlkida lme4 mudeli kirjelduse otse Stani keelde ja seda mudelit seal jooksutada. Brms teeb elu magusaks (vt lisa 2).

> Suurem sõltuvus valimi suurusest ja erinevatest lisaeeldustest võrreldes Bayesi mudelitega on see hind, mida ad hoc lahendused maksavad oma lihtsuse eest. Enamust klassikalisi teste (t test, chi ruut test, jms) võib vaadelda selliste ad hoc lahendustena, mis sageli lagunevad laiali väikesetel valimitel, samas kui bayes töötab väikeste valimitega hästi – tõsi küll, sõltudes väikeste valimite korral rohkem priorist ja andes seal realistlikult laiad usaldusintervallid.

Mitmetasemeliste mudelite lisaeeldused

Mitmetasandilised mudelid toovad sisse lisaeelduse, et lineaarsuse/normaalsuse jm eeldused kehtivad igal mudeli tasemel. Samuti, et kõik grupid tulevad samast statistilisest populatsioonist, ja vastavalt sellele on nad mudelis koondatud ühise priori alla. Mitmetasemelises mudelis töötab grupi tasemel mudel priorina indiviidi tasemel mudelile.

Mitmetasemeline mudel töötab korraga mitmel tasmel

Mudeli muudab mitmetasemeliseks see, et me määrame veamudelit kasutades mitte ainult indiviidi tasemel koefitsiente (1. tase), vaid anname neile koefitsientidele omakorda veamudeli (2. tase), mis modelleerib koefitsientide varieeruvust gruppide vahel. Selliseid tasemeid võib lisada põhimõtteliselt ükskõik kui palju. Kõrgema taseme mudel, lisaks sellele, et modelleerida gruppide vahelist varieeruvust, töötab ka priorina madalama taseme suhtes. Seega saab mudeli fittimisel 2. tase informatsiooni 1. tasemelt (andmete näol) ja samal ajal annab informatsiooni esimesele tasemele (priori kujul).

1. Mitmetasemelised mudelid modelleerivad eksplitsiitselt varieeruvust klasrtite sees ja klastrite vahel.
2. Nad modelleerivad indiviidi-tasemel regressioonikoefitsientide varieeruvust.
3. Nad võimaldavad paremini määrata indiviidi tasemel regressioonikoefitsiente endid, eriti kui erinevates gruppides on erinev arv indiviide.

Shrinkage

Oletame, et te plaanite reisi Kopenhaagenisse ja soovite sellega seoses teada, kui kallis on keskeltläbi õlu selle linna kõrtsides. Teile on teada

õlle hind kolmes Kopenhaageni kõrtsis, mida ei ole just palju. Aga sellele lisaks on teile teada ka õlle hind 6-s Viini, 4-s Praha ja 5-s Pariisi kõrtsis. Nüüd on teil põhimõtteliselt kolm võimalust, kuidas sellele probleemile läheneda.

1. Te arvestate ainult Kopenhaageni andmeid ja ignoreerite teisi, kui ebarelevantseid. See meetod töötab hästi siis, kui teil on Kopenhaageni kohta palju andmeid (aga teil ei ole).

2. Te arvestate võrdselt kõiki andmeid, mis teil on — ehk te võtate keskmise kõikidest õllehindadest, hoolimata riigist. See töötab parimini siis, kui päriselt pole vahet, millisest riigist te oma õlle ostate, ehk kui õlu maksab igal pool sama palju. Antud juhul pole see ilmselt parim eeldus.

3. Te eeldate, et õlle hinna kujunemisel erinevates riikides on midagi ühist, aga et seal on ka erinevusi. Sellisel juhul tahate te fittida hierarhilise mudeli, kus teie hinnang õlle hinnale Kopenhaagenis sõltuks mingil määral (aga mitte nii suurel määral, kui eelmises punktis) ka teie kogemustest teistes linnades. Sama moodi, teie hinnang õlle hinnale Pariisis, Prahas jne hakkab mingil määral sõltuma kõikide linnade andmetest.

   Kui teil on olukord, kus te mõõdate erinevaid gruppe, mis küll omavahel erinevad, aga on ka teatud määral sarnased (näiteks testitulemused grupeerituna kooli kaupa), siis on mõistlik kasutada kõikide gruppide andmeid, et adjusteerida iga grupi spetsiifilisi parameetreid. Seda adjusteerimise määra kutsutakse “shrinkage”.

Shrinkage toimub parameetri keskväärtuse suunas ja mingi grupi shrinkage on seda suurem, mida vähem on selles grupis liikmeid ja mida kaugemal asub see grupp kõikide gruppide keskväärtusest. See viib shrinkage koefitsientide kallutatusele (bias), aga samas ka suuremale täpsusele (precision). See tähendab, et shrinkage koefitsiendi hinnang on keskeltläbi lähemal tõelisele koefitsiendi väärtusele kui hannang tavalisele ühetasandilise mudeli koefitsiendile.

Shrinkage on põhimõtteliselt sama nähtus, mis juba Francis Galtoni poolt avastatud regressioon keskmisele. Regressioon keskmisele on stohhastiline protsess kus, olles sooritanud n mõõtmist ja arvutanud nende tulemuste põhjal efekti suuruse, see valimi ES peegeldab nii tegelikku ES-i kui juhuslikku valimiviga. Kui valimivea osakaal ES-s on suur, siis lisamõõtmised vähendavad keskeltläbi efekti suurust. Shrinkage erineb sellest ainult selle poolest, et lisamõõtmised meenutavad ainult **osaliselt** algseid mõõtmisi.

Kasutades hierarhilisi mudeleid saab võidelda ka valehäirete ehk mitmese testimise probleemiga. See probleem on lihtsalt sõnastatav: kui te sooritate palju võrdluskatseid ja statistilisi teste olukorras, kus

tegelik katseefekt on tühine, siis tänu valimiveale annavad osad teie paljudest testidest ülehinnatud efekti. Seega, kui meil on kahtlus, et enamus võrdlusi on "mõttetud" ja me ei oska ette ennustada, millised võrdlused neist (kui üldse mõni) võiks anda tõelise teaduslikult mõtteka efekti, siis on lahendus kõiki saadud efekte kunstlikult pisendada kõikide efektide keskmise suunas. Mudeli kontekstis kutsutakse sellist lähenemist *shrinkage*-ks. Aga kui suurel määral seda teha? See sõltub nii sellest, kui palju teste me teeme, valimi suurusest, kui ka sellest, kuidas jaotuvad mõõdetud efektisuurused (milline on efektisuuruste varieeruvus testide vahel).

Bayesi lahendus on, et me lisame mudelisse veel ühe hierarhilise priori, mis kõrgub üle gruppide-spetsiifilise priori. Seega anname me olemasolevale priorile uue kõrgema taseme meta-priori, mis tagab, et informatsiooni jagatakse gruppide vahel ja samal ajal ka gruppide sees. Sellise lahenduse õigustus on, et me usume, et erinevad alamgrupid pärinevad samast üli-jaotusest ja neil on omavahel midagi ühist (ehkki alam-gruppide vahel võib olla ka reaalseid erinevusi). Näiteks, et kõik klassid saavad oma lapsed samast lastepopulatsioonist, aga siiski, et leidub ka eriklasse eriti andekatele.

Selline mudel tagab, et samamoodi nagu mudeli ennustused individaalsete andmepunktide kohta iga alam-grupi sees "liiguvad lähemale" oma alam-grupi keskmisele, samamoodi liiguvad ka alamgruppide keskmised lähemale üldisele grupi keskmisele. Selle positiivne mõju on valealarmide vähendamine ja oht on, et me kaotame ka tõelisi efekte. Bayesi eelis on, et see oht realiseerub ainult niipalju, kuipalju meie mudel ei kajasta reaalset katse struktuuri. Klassikalises statistikas rakendatavad multiple testingu korrektsioonid (Bonfferroni, ANOVA jt) on kõik teoreetiliselt kehvemad.

Lihtsaim shrinkage mudeli tüüp on mudel, kus me laseme vabaks interceptid, aga mitte tõusunurgad. Igale klastrile vastab mudelis oma intercepti parameeter ja oma intercepti prior. Lisaks annab mudel meile fittimise käigus valimi andmete põhjal ise parameetrid kõrgema taseme priorisse, mis on ühine kõikidele interceptidele. Seega me määrame korraga interceptide parameetrid ja kõrgema taseme priori parameetrid, mis tähendab, et informatsioon liigub mudelit fittides mõlemat pidi — mööda hierarhiat alt ülesse ja ülevalt alla. Selline mudel usub, et erinevate koolide keskmine tase erineb (seda näitab iga kooli intercept), aga juhul kui me mõõdame näiteks kalamaksaõli mõju õppeedukusele, siis selle mõju suurus ei erine koolide vahel (kõikide koolide tõusuparameetrid on identsed).

Shrinkage kui nähtuse avastas Francis Galton 1870-ndatel aastatel. Galton ja tema sõbrad veetsid nimelt kümme aastat üle Inglismaa taimi kasvatades ja mõõtes erinevate põlvkondade seemnete suurusi. Eesmärk oli luua tühjale kohale uus teadus, pidevate tunnuste ge-

neetika, ja katsete tulemus oli rabav. Nimelt leiti tugev seaduspära, mille kohaselt suurte seemnetega emataimede tütred andsid keskeltläbi väiksemaid seemneid kui nende vanemad ja vastupidi, väikeste seemnetega emade tütred andsid keskeltläbi suuremaid seemneid. Galtoni usk, et ta on avastanud tähtsa bioloogiaseaduse, purunes ca 1885, kui ta pääses analüüsima tuhatkonna inimese pikkusi andmestikus, mis sisaldas vanemate ja täiskasvanud laste pikkusi, ning leidis seal sama nähtuse. Sertifitseeritud geeniusena mõistis Galton, et ta ei olnud avastanud mitte niivõrd geneetikaseaduse, vaid peaaegu, et loogikaseaduse. Tema enda sõnadega: The average regression of the offspring to a constant fraction of their mid-parental deviations, is now shown to be a perfectly reasonable law which might have been deductively foreseen. It is of so simple a character that I have made an arrangement with pulleys and weights by which the probable average height of the children of known parents can be mechanically reckoned. (vt joonis). Sellega avastas Galton regressiooni keskmisele, mis on sisuliselt sama asi, mis shrinkage. Galton nägi, et shrinkagel on järgmised omadused: 1. "The mean filial regression towards mediocrity was directly proportional to the parental deviation from it." Ehk, mida kaugemal on vanemad keskmisest, seda suurema amplituudiga on nende laste shrinkage 2. "The child inherits partly from his parents, partly from his ancestry. . . . the further his genealogy goes back, the more numerous and varied will his ancestry become . . . Their mean stature will then be the same as that of the race; in other words, it will be mediocre." Ehk, shrinkage toimub alati, kui tunnuse väärtus ei ole deterministlikult määratud (shrinkage taandub korrelatsioonile vanemate ja laste varieeruvuse vahel) 3. "This law tells heavily against the full hereditary transmission of any gift. The more exceptional the amount of the gift, the more exceptional will be the good fortune of a parent who has a son who equals him in that respect." Ehk, pidevate tunnuste korral on korrelatsioon alati <1 & >-1 4. "The law is even-handed; it levies the same heavy succession-tax on the transmission of badness as well as of goodness. If it discourages the extravagant expectations of gifted parents that their children will inherit all their powers, it no less discountenances extravagant fears that they will inherit all their weaknesses and diseases." Ehk shrinkage töötab võrdselt mõlemas suunas (ülevalt alla ja alt üles, aga ka vanematelt lastele ja lastelt vanematele). Seega ei ole shrinkage ajas toimuv, põhjuslik, ega isegi mitte füüsikaline protsess, vaid tõenäosusteooriast tulenev loogiline paratamatus. Samamoodi nagu shrinkage esineb vanemate-laste vahel esineb see ka valimi-kordusvalimi vahel kõigi valimite keskmise suunas (valimiefektid taanduvad välja sedamõõda, kuidas valimeid juurde tuleb). Ja samamoodi, kui me võtame valimi testitulemusi mitmest koolist, siis eeldusel, et õpilased on kõikides koolides sarnased (aga mitte identsed), toimub shrinkage kõikide koolide keskmise suunas. Seega, nihutades mingi kooli keskmist testitulemust koolide keskmise suunas, saame parema hinnangu selle kooli õpilaste teadmisetele kui pelgalt selles koolis õpilaste teadmisi mõõtes!

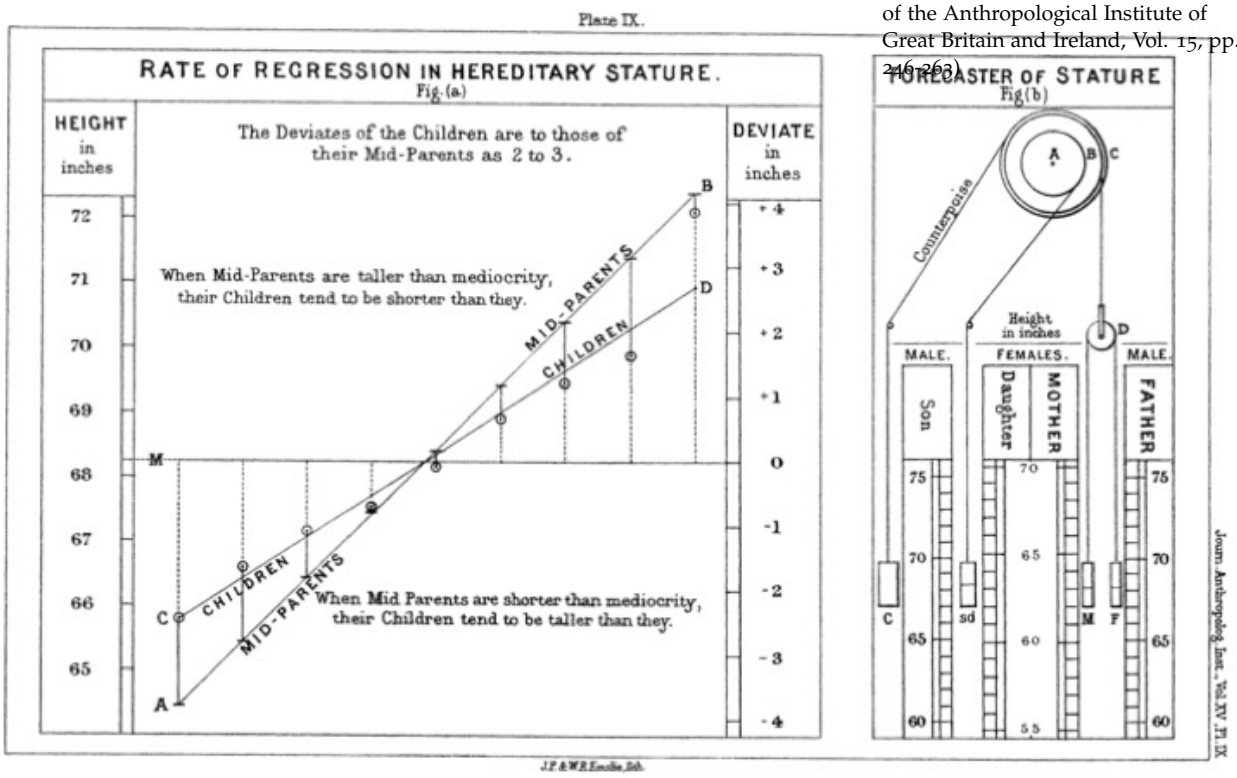


Figure 1: (ref:Galton (1886), The Journal of the Anthropological Institute of Great Britain and Ireland, Vol. 15, pp. 246-263)

ANOVA-laadne mudel

Lihtne ANOVA on sageduslik test, mis võrdleb gruppide keskmisi mitmese testimise kontekstis. Siin ehitame selle Bayesi analoogi, mis samuti hindab gruppide keskmisi mitmese testimise kontekstis. Põhiline erinevus seisneb selles, et kui ANOVA punktennustus iga grupi keskväärtusele võrdub valimi keskväärtusega ja ANOVA pelgalt kohandab usaldusintervalle selle keskväärtuse ümber, siis bayesiaanlik mudel püüab ennustada igale grupile selle tegelikku kõige tõenäolisemat keskväärtust arvestades kõigi gruppide andmeid. Shrinkage-i roll on ekstreemseid gruppe "tagasi tõmmates" vähendada ebakindlust iga grupi keskmise ennustuse ümber. Shrinkage käigus tõmmatakse gruppe kõikide gruppide keskmise poole seda tugevamalt, mida kaugemal nad sellest keskmisest on. Sellega kaasneb paratamatult mõningane süstemaatiline viga, kus tõelised efektid tulevad välja väiksematena, kui nad tegelikult on. Kui ilma tegelike efektideta gruppide arv on väga suur võrreldes päris efektidega gruppidega, siis võib shrinkage meie pärisefektid sootuks ära kaotada. Kahjuks on see loogiline paratamatus; alternatiiviks on olukord, kus meie üksikud pärisefektid upuvad sama suurte pseudoefektide merre.

Ok, aitab mulast, laadime vajalikud raamatukogud ja andmed ning vaatame mis saab.

Andmed: *The data contain GCSE exam scores on a science subject. Two components of the exam were chosen as outcome variables: written paper and course work. There are 1,905 students from 73 schools in England. Five fields are as follows.*

1. School ID
2. Student ID
3. Gender of student

   0 = boy
   1 = girl

4. Total score of written paper
5. Total score of coursework paper

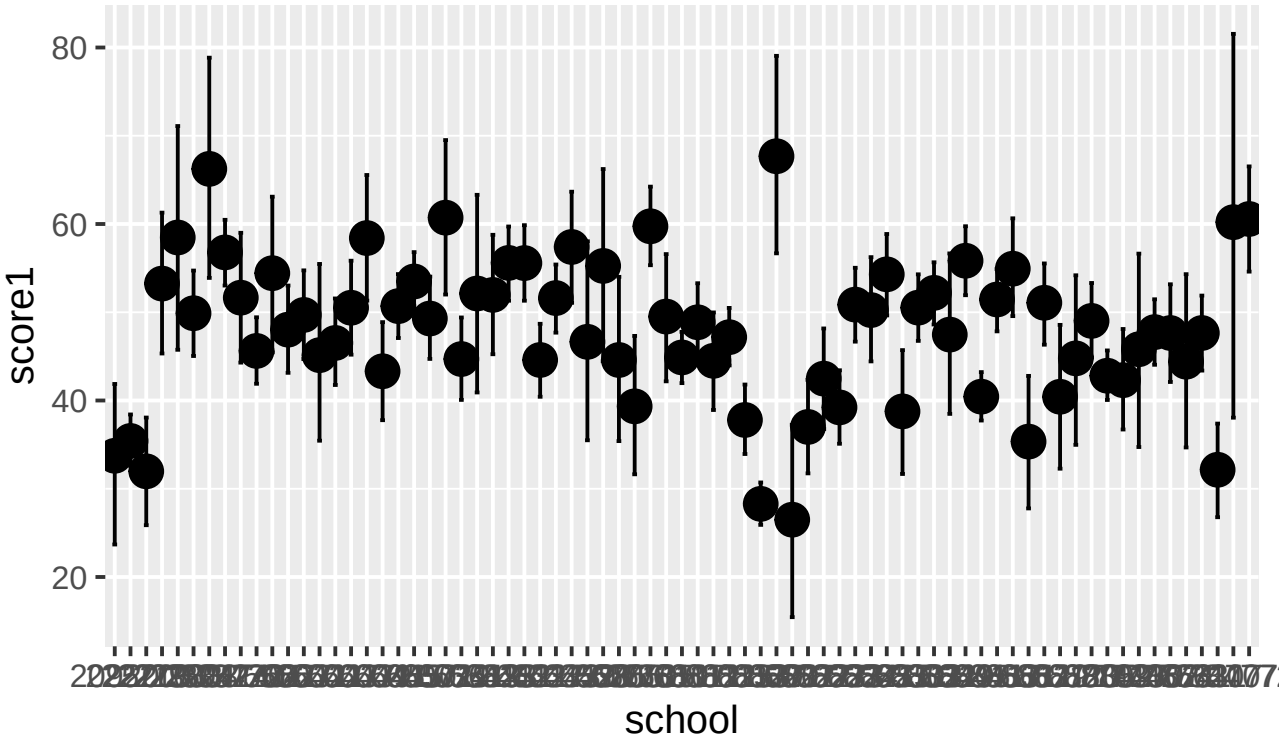
   Missing values are coded as -1.

```
schools <- read_csv("raamat/data/schools.csv")
schools <- schools %>%
dplyr::filter(complete.cases(.)) %>%
mutate_at(vars(sex, school), as.factor)
```

```
## map2stan requires data.frame
class(schools) <- "data.frame"
```

Alustuseks mitte-hierarhiline mudel, mis arvutab keskmise score1 igale koolile eraldi. See on intercept-only mudel, mis tähendab, et me hindame testitulemuse keskväärtust kooli kaupa ja igale koolile sõltumatult kõigist teistest koolidest. Me ei püüa siin ennustada testitulemuste väärtusi x-i väärtuste põhjal. Kuna siin puuduvad kõrgema taseme priorid, siis vaatab mudel igat kooli eraldi ja ühegi kooli hinnang ei arvesta ühegi teise kooli andmetega.

```
school_m2 <- brm(score1 ~ school, data = schools)
write_rds(school_m2, "raamat/data/school_m2.rds")

marginal_effects(school_m2)

## Warning: Method 'marginal_effects' is deprecated. Please use
## 'conditional_effects' instead.
```



Igale koolile antud hinnang on sõltumatu kõigist teistest koolidest.

Ja nüüd hierarhiline mudel, mis teab koolide vahelisest varieeruvusest. Siin leiab a_school-i priorist teise taseme meta-parameetri nimega sigma_school, millele on defineeritud oma meta-prior.

```
school_m3 <- brm(score1 ~ 1 + (1 | school), data = schools)
write_rds(school_m3, "raamat/data/school_m3.rds")

tidy(school_m3) %>%
mutate_if(is.numeric, round, 2) %>%
head() %>% kable()
```

Nagu näha on sd_school < sigma, mis tähendab, et koolide vaheline varieeruvus on väiksem kui õpilaste vaheline varieeruvus neis koolides. Seega sõltub testi tulemus rohkem sellest, kes testi teeb kui sellest, mis koolis ta käib. Loogika on siin järgmine: samamoodi nagu testitulemustel on jaotus õpilasekaupa, on neil ka jaotus koolikaupa. Koolikaupa jaotus töötab priorina õpilasekaupa jaotusele. Aga samas vajab kooli kaupa jaotus oma priorit — ehk meta-priorit. Seega saame me samast mudelist hinnangu nii testitulemustele kõikvõimalike õpilaste lõikes, kui ka kõikvõimalike koolide lõikes. Mudel ennustab ka nende koolide ja õpilaste tulemusi, keda tegelikult olemas ei ole, aga kes võiksid kunagi sündida.

Ning veel üks hierarhiline mudel, mis teab nii koolide skooride keskmiste varieeruvust kui koolide vahelist varieeruvust.

Võrdleme mudeleid.

```
loo(school_m2, school_m3)
```

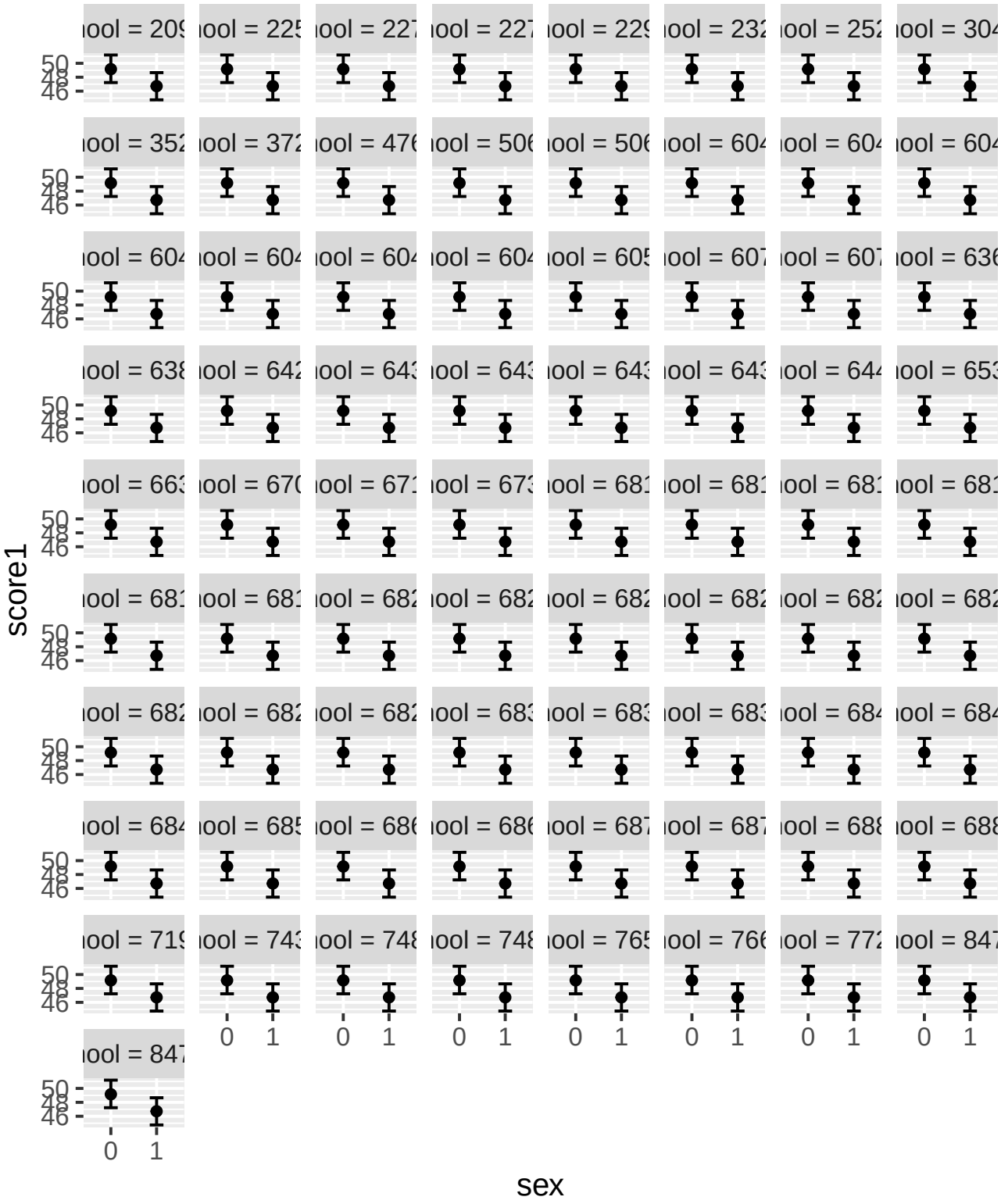
Model comparisons: elpd_diff se_diff school_m3 0.0 0.0
school_m2 -5.0 3.5
Siit nähtub, et m3 on parim mudel, aga napilt.

Vabad interceptid klassikalises regressioonimudelis

Ennustame score1 sõltuvust sex-ist. Küsimus: kui palju poiste ja tüdrukute matemaatikaoskused erinevad? Fitime mudeli, mis laseb vabaks intercepti. **Selle mudeli eeldus on, et igal koolil on oma baastase (oma intercept), aga kõikide koolide efektid (mudeli tõusu-koefitsient) on identsed.**

Me kasutame prediktorina binaarset kategoorilist muutujat. See on analoogiline olukord ANOVA mudelile, mis võtab arvesse multiple testingu olukorra, mis meil siin on.

```
schools_anova1 <- brm(score1 ~ sex + (1 | school), data = schools)
write_rds(schools_anova1, "raamat/data/schools_anova1.fit")

conditions <- make_conditions(schools_anova1, "school")
marginal_effects(schools_anova1, conditions = conditions)
```


score1
sex
0
1
46
48
50

```
tidy(schools_anova1) %>%
mutate_if(is.numeric, round, 2) %>%
head() %>% kable()
```

term	estimate	std.error	lower	upper
b_Intercept	49.16	1.01	47.53	50.83
b_sex1	-2.46	0.62	-3.44	-1.46
sd_school__Intercept	7.11	0.71	6.01	8.34
sigma	11.18	0.21	10.85	11.53
r_school[20920,Intercept]	-9.85	4.18	-16.85	-3.02
r_school[22520,Intercept]	-11.61	1.75	-14.44	-8.75

Siin on r_school[nr,Intercept] kooli-spetsiifiline korrektsioonifaktor, mis tuleb liita üldisele b_Interceptile. mean(v_Intercept) = 0 ja me eeldame, et korrektsioonid on normaaljaotusega.

sex = 1 ehk sex1 on tüdruk.

Intercept annab siin sex = 0 (poisid) keskmise skoori kooli kaupa (kui liita üldisele interceptile kooli-spetsiifiline intercept). Kui tahame näiteks hinnangut kooli 20920 tüdrukute skoorile (ehk tõelisele matemaatikavõimekusele) siis:

`b_Intercept + b_sex1 + r_school[20920, Intercept]`

annab meile selle posteeriori. Poistele sama kooli kohta:

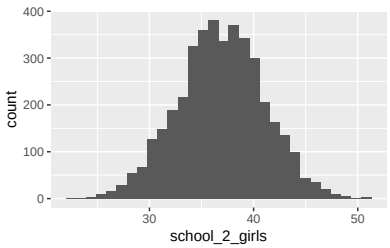
`Intercept + + r_school[20920, Intercept]`

Ja poiste-tüdrukute erinevus skooripunktides on

`b_sex1`

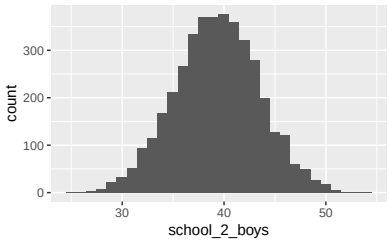
Arvutame siis kooli nr 20920 tüdrukute keskmise skoori posteeriori.

```
schools_m1_samples <- posterior_samples(schools_anova1)
school_2_girls <- schools_m1_samples$b_Intercept +
schools_m1_samples$b_sex1 +
schools_m1_samples$`r_school[20920,Intercept]`
ggplot(data = NULL, aes(school_2_girls)) + geom_histogram()
```



Ja Poiste oma

```
school_2_boys <- schools_m1_samples$b_Intercept +
schools_m1_samples$`r_school[20920,Intercept]`
ggplot(data = NULL, aes(school_2_boys)) + geom_histogram()
```



Siin on eeldus, et kõikides koolides on sama poiste ja tüdrukute vaheline erinevus (b_sex1), kuid erinevad matemaatikateadmiste baastasemed (mudeli intercept on koolide vahel vabaks lastud, kuid tõus mitte).

Vabad tõusud ja interceptid

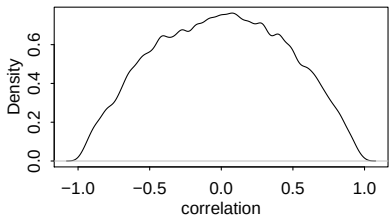
Milline näeb välja mudel, kus me laseme vabaks nii intercepti kui tõusu?

```
schools.f2 <- brm(score1 ~ sex + (1 + sex | school), data = schools, cores = 3, chains = 3)
write_rds(schools.f2, "raamat/data/schools.f2.rds")
```

nüüd on meil lisaparameetrid v_sex1, mis annab tõusu igale koolile eraldi ning Rho-school, mis annab korrelatsiooni intercepti ja tõusu vahel. Nüüd me jagame informatsiooni erinevat tüüpi parameetrite, nimelt interceptide ja tõusude, vahel. Selleks ongi vaja Rho lisa-parameetrit. Nüüd ei modelleeri me intercepti ja tõusu enam 2 eraldi normaaljaotuste abil vaid ühe 2-dimensionaalse normaaljaotusega (mvnorm2).

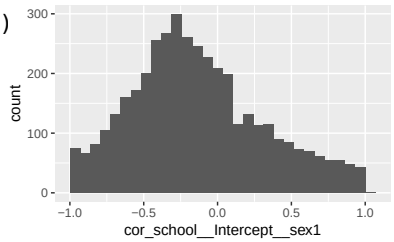
Prior korrelatsioonile Interceptide ja tõusude vahel on `lkj()`. Selle ainus parameeter on K. Mida suurem K, seda rohkem on prior konsentreeritud 0 korrelatsiooni ümber. K = 1 annab tasase priori. Meie kasutame K = 2, mis töötab laia vahemiku mudelitega.

```
R <- rethinking::rlkjcorr(10000, K = 2, eta = 2)
dens(R[, 1, 2], xlab = "correlation")
```



Posteerior korrelatsioonile intercepti ja tõusu vahel:

```
schools_m2_samples <- posterior_samples(schools.f2)
ggplot(data = schools_m2_samples, aes(cor_school__Intercept__sex1)
geom_histogram()
```



Meil on negatiivne korrelatsioon intercepti ja tõusu vahel. Seega, mida väiksem on poiste keskmine skoor koolis (=intercept), seda suurem om erinevus poiste ja tüdrukute skooride vahel (= tõus).

```
tidy(schools.f2) %>% head(8) %>%
mutate_if(is.numeric, round, 2) %>% kable()
```

term	estimate	std.error	lower	upper
b_Intercept	49.15	0.99	47.51	50.76
b_sex1	-2.46	0.63	-3.51	-1.41
sd_school__Intercept	7.37	0.83	6.11	8.87
sd_school__sex1	1.45	1.06	0.12	3.48
cor_school__Intercept__sex1	-0.14	0.45	-0.81	0.72
sigma	11.16	0.21	10.83	11.53
r_school[20920,Intercept]	-10.46	4.56	-18.20	-3.16
r_school[22520,Intercept]	-12.14	2.05	-15.62	-8.84

Nüüd saab kooli 20920 skoori tüdrukutele valemiga:

```
b_Intercept + b_sex1 + r_school[20920,Intercept] + r_school[20920,sex1]
```

Sama skoor poistele:
`b_Intercept + b_sex1 + r_school[20920,Intercept]`
ja tüdrukute ja poiste erinevus :
`b_sex1 + r_school[20920,sex1]`
tüdrukute-poiste erinevus üle kõikide koolide:
`b_sex1`
tüdrukute keskmine skoor üle kõikide koolide:
`b_Intercept + b_sex1`
ja poiste keskmine skoor üle kõikide koolide:
`b_Intercept`

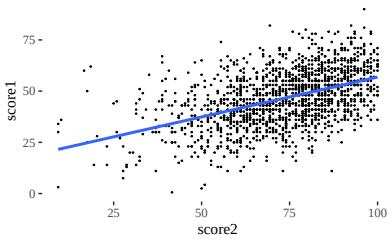
```
loo(schools_anova1, schools.f2)
```

Model comparisons: elpd_diff se_diff schools_anova1 0.0 0.0
schools.f2 -0.2 1.0

Hierarhiline mudel pidevate prediktoritega

Siin püüame ennustada score1 mõju score2 väärtusele.

```
ggplot(schools, aes(score2, score1)) +
geom_point(size = 0.2, aplha = 0.2) +
geom_smooth(method = lm, se = F) +
ggthemes::theme_tufte()
```



Kõigepealt lihtne regressioon `lm()` funktsiooniga (see ei ole hierarhiline mudel).

```
lm(score1 ~ score2, data = schools)
```

```
##
## Call:
## lm(formula = score1 ~ score2, data = schools)
##
## Coefficients:
## (Intercept)       score2
##     17.9713       0.3888
```

score2 tõus 1 punkti võrra tõstab score1-e 0.39 punkti võrra.

Modelleerime seost üle Bayesi hierarhilise mudeli, kus ainult Intercept on vabaks lastud.

```
school_m3 <- brm(score1 ~ score2 + (1 | school), data = schools, prior = prior(normal(80,
    30), class = b), cores = 3, chains = 3)
write_rds(school_m3, "raamat/data/school_m3.fit")
```

Siin ei ole individuaalsed interceptid tõlgenduslikult informatiivsed, aga nende sissepanek parandab mudeli ennustust beta koefitsiendile (beta läheb väiksemaks ja ebakindlus selle hinnangu ümber kasvab).

```
tidy(school_m3) %>% head(6) %>%
mutate_if(is.numeric, round, 2) %>% kable()
```

term	estimate	std.error	lower	upper
b_Intercept	21.30	1.59	18.71	23.86
b_score2	0.36	0.02	0.33	0.39
sd_school__Intercept	6.61	0.69	5.56	7.80
sigma	10.06	0.19	9.76	10.38
r_school[20920,Intercept]	-7.48	3.75	-13.77	-1.13
r_school[22520,Intercept]	-6.18	1.60	-8.79	-3.51

Siin tuleb beta veidi väiksem - 0.36. Kuna $sd_{school} < sigma$, siis tundub, et koolide vaheline varieeruvus on väiksem kui laste vaheline varieeruvus (sigma on üle kõigi koolide). iga kooli baastase tuleb `Intercept + r_school[...,Intercept]` aga selle mudeli järgi on kõikide koolide score2 ja score1 sõltuvus sama tugevusega.

Laseme siis ka tõusud vabaks

```
school_m4 <- brm(score1 ~ score2 + (1 + score2 | school), data = schools, prior = prior(normal(80,
    30), class = b), cores = 3, chains = 3)
write_rds(school_m4, "raamat/data/school_m4.fit")
```

nüüd saame igale koolile arvutada oma intercepti ja oma tõusu (ikka samamoodi: `b_Intercept + r_school[,Intercept] ja b_score2 + r_school[,school]`)