Математички факултет
Универзитет у Београду

Мастер рад

Бајесовски линеарни регресиони модели

Студент:
Биљана Јовановић 1021/2016

Ментор:
др Бојана Милошевић

септембар 2019.

Предговор

Зашто користити бајесовско закључивање уместо класичног фреквенционистичког приступа? У књизи [2] аутор наводи да су бајесовске методе увек имале природну предност у односу на класичне због свог приступа да све непознате квантитете третирају пробабилистички. Међутим, без употребе рачунара постојали су проблеми у употреби тих метода - немогућност оцене параметара модела, априорне расподеле које су укључивале неке лоше претпоставке. Повећањем моћи и капацитета рачунара као и појавом нових алгоритама дошло је до великог напретка у решавању тих проблема. Данас живимо у свету где осим наше могућности имагинације не постоји још много других ограничења. Зато данас истраживачи користе и бајесовске и класичне методе. Почевши од 1990-их година приметан је пораст употребе бајесовских метода у медицини, природним наукама, инжењерству, друштвеним наукама. Један од разлога за тај тренд пораста су повећана потреба за статистичким анализама било које врсте, нпр. клиничка испитивања, резултати анкета, политичке и економске анализе, анализа података са интернета, истраживања тржишта. Затим, многи научници су дошли до закључака да алтернативе оваквом приступу имају логичких недоследности и недостатака. И напредак рачунарских метода који је довео до тога да реалистични бајесовски модели дају резултате који се могу јако добро апроксимирати па нам то што их аналитички не можемо извести не представља проблем.

Бајесовска анализа се заснива на почетном скупу претпоставки и омогућава истраживачима да у процес закључивања као додатак подацима којима располажу укључе и поуздане информације које већ поседују. Закључци о непознатим параметрима нису изражени на уобичајен начин, као тачкаста оцена[1] и њен интервал поверења[2]. Уместо тога, овде непознати параметри имају своју расподелу, тј. могу се сматрати случајним величинама. Пре посматрања података непознати параметри имају своју

[1] Point estimate

[2] Confidence interval

априорну расподелу која осликава наше априорно знање - информације које су нам доступне пре него што узмемо у обзир добијене податке. Након узимања података у обзир добија се апостериорна расподела. Апостериорна расподела затим може послужити за извођење разних закључка о непознатом параметру као што су: квантили расподеле, вероватноћа да параметар узима вредности из неког интервала, затим интервали прекривања који представљају бајесовски аналогон интервалима поверења као и многи други. Такође, апостериорна расподела користи се за одређивање апостериорне предиктивне расподеле нових података.
Две кључне претпоставке у бајесовском закључивању су следеће:

1. Функција веродостојности[3] описује зависност целог узорка података од вредности непознатих параметара.

2. Непознати параметри третирају се као случајне величине и за њих се претпоставља нека априорна расподела која не зависи од података.

Са овим претпоставкама основе бајесовског закључивања могу се свести на три корака:

1. Спецификовање вероватносног модела који укључује функцију веродостојности и априорну расподелу непознатог параметра.

2. Ажурирање знања о непознатом параметру рачунањем условне расподеле непознатог параметра при услову да су нам дати ти конкретни подаци. Ту расподелу зовемо апостериорном расподелом параметра.

3. Процена (евалуација) колико модел одговара подацима и процена осетљивости закључака на претпоставке модела.

Уколико је неопходно, модел се може изменити а затим се могу поновити ова три основна корака. Ови кораци се могу поновити и након добијања нових података при чему за априорну расподелу изаберемо претходно добијену апостериорну расподелу.
Често постоји слагање између бајесовске и класичне фреквенционистичке анализе тј. њиховом применом можемо добити исте закључке. Постоје два веома важна случаја где је ово увек тачно. Први случај је када се за априорну расподелу параметра изабере расподела која даје једнаку вероватноћу, односно густину свим вредностима одговарајућег

[3] Likelihood function

скупа, односно интервала. То можемо записати као: $\pi(\theta) \propto c, \forall\theta \in \Theta$, где је c нека константа. То је униформна расподела у случају да је Θ дискретан скуп или ограничен интервал односно неправа расподела у случају неограниченог интервала. У том случају су апостериорна расподела и функције веродостојности пропорционалне. Уколико се за бајесовску тачкасту оцену изабере мода апостериорне расподеле, она се поклапа са оценом добијеном методом максималне веродостојности. Други случај је случај у коме је узорак веома великог обима па избор априорне расподеле има занемарљив утицај на апостериорну расподелу.

Садржај

Поглавље 1

Основни појмови

У овом поглављу изложен је преглед свих резултата неопходних за бајесовско закључивање у општем случају као и за бајесовски приступ линеарној регресији док ће примери са кодовима бити представљени у последњем поглављу. Представљени су: Бајесово правило, оцењивање параметара на бајесовски начин, типови априорних расподела, извођење апостериорних расподела, методе за апроксимацију апостериорних расподела у случајевима када није могуће добити њихов аналитички облик, апостериорна предиктивна провера, анализа сензитивности модела.

1.1 Бајесово правило

Бајесово правило или Бајесова теорема описује вероватноћу догађаја засновану на априорном знању о условима који могу бити повезани са тим догађајем. Теорема је име добила по Томасу Бајесу[1] који је први формулисао специјалан случај теореме. Тај случај је презентован 1763. године, две године након његове смрти, у склопу есеја[2], захваљујући његовом пријатељу Ричарду Прајсу. Иако је Бајесово име везано за закључивање многи би се сложили да његов следбеник Пјер - Симон Лаплас[3] има много више заслуга за овакву анализу и да је везивање једне битне гране статистике за Бајесово име у многоме претеривање. Бајес јесте први експлицитно извео познато правило, али Лаплас је био

[1] Thomas Bayes (1702-176) - енглески статистичар, филозоф а уједно и презвитеријански свештеник

[2] "An Essay towards solving a Problem in the Doctrine of Chances"

[3] Pierre-Simon, marquis de Laplace (1749-1781) - француски научник чији рад је допринео развоју инжењерства, математике, статистике, физике, астрономије и филозофије

први који је у свом делу[4] обезбедио детаљнију анализу која је значајнија за данашњу бајесовску статистику.
Бајесово правило можемо извести из формуле за условну вероватноћу:

$$P(A|B) = \frac{P(B,A)}{P(B)} \tag{1.1}$$

где су A и B догађаји такви да $P(B) \neq 0$. Бајесово правило гласи:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{1.2}$$

Ако је $\{A_j\}$ разбијање сигурног догађаја на дисјунктне догађаје онда применом формуле потпуне вероватноће једнакост (1.2) може да се запише и на следећи начин:

$$P(A|B) = \frac{P(B|A)P(A)}{\sum_j P(B|A_j)P(A_j)} \tag{1.3}$$

Према бајесовској интерпретацији вероватноћа мери степен веровања. Бајесова теорема тада служи као веза између степена веровања да ће се десити догађај A пре и после узимања у обзир догађаја B. У том контексту $P(A)$ називамо априорном вероватноћом и она представља иницијални степен веровања у A. $P(A|B)$ зовемо апостериорна вероватноћа и за њено добијање су нам поред априорне $P(A)$ потребни још и $P(B|A)$ и $P(B)$ који представљају условну односно безусловну вероватноћу догађаја B. Знамо да је догађај B реализован и самим тим нећемо га третирати пробабилистички. То не значи да је познато све о свим могућим догађајима или о оним који ће се десити у будућности, већ само да је познато све о том конкретном догађају. Једина улога $P(B)$ као имениоца у Бајесовом правилу је да обезбеди нормираност и зато можемо да запишемо:

$$P(A|B) \propto P(B|A)P(A) \tag{1.4}$$

где $\propto$ означава пропорционалност. Условна веротноћа $P(A|B)$ је баланс онога у шта смо већ веровали ($P(A)$) и доприноса добијеног од нове опсервације ($P(B|A)$). Постоје ситуације када су подаци утицајнији од априорног знања, као и обрнуто. То нам одговара јер када немамо довољно података или су они лошег квалитета пожељно је да се ослонимо што више на претходна знања док уколико имамо довољно података априорна знања нису од великог интереса.

[4] "Theorie analytique des probabilites".

Бајесово правило можемо применити и на случајне променљиве. Нека су X и Y случајне променљиве. Постоје многе ситуације у којима желимо да знамо X али можемо да меримо само величину Y која је на неки начин повезана са њом. У зависности од тога ког су типа X и Y следе 4 различите верзије Бајесове теореме за случајне променљиве.
Уколико су обе променљиве апсолутно непрекидног типа са заједничком функцијом густине расподеле $f_{X,Y}(x,y)$ знамо да је

$$f_{X,Y}(x,y) = f_{X|Y}(x|y)f_Y(y) = f_{Y|X}(y|x)f_X(x). \tag{1.5}$$

Тада условну густину за X можемо да изразимо као количник:

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)f_X(x)}{f_Y(y)} \tag{1.6}$$

$$= \frac{f_{Y|X}(y|x)f_X(x)}{\int_{-\infty}^{\infty} f_{Y|X}(y|u)f_X(u)du}. \tag{1.7}$$

За дискретне случајне величине X и Y, где са $P_X(x)$ означавамо вероватноћу догађаја $\{X = x\}$ а са $P_Y(y)$ вероватноћу догађаја $\{Y = y\}$, имамо следећу формулу:

$$P_{X|Y}(x|y) = \frac{P_{Y|X}(y|x)P_X(x)}{P_Y(y)} \tag{1.8}$$

$$= \frac{P_{Y|X}(y|x)P_X(x)}{\sum_k P_{Y|X}(y|k)P_X(k)} \tag{1.9}$$

при чему друга једнакост важи применом формуле потпуне вероватноће. Уколико је X дискретног а Y апсолутно непрекидног типа и знамо да је Y узела вредност y, тада важи:

$$P_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)P_X(x)}{f_Y(y)} \tag{1.10}$$

$$= \frac{f_{Y|X}(y|x)P_X(x)}{\sum_k f_{Y|X}(y|k)P_X(k)}. \tag{1.11}$$

Преостали случај који разматрамо је случај када доносимо закључке о апсолутно непрекидној случајној величини X уколико знамо да је дискретна случајна величина Y узела вредност y. Тада је:

$$f_{X|Y}(x|y) = \frac{P_{Y|X}(y|x)f_X(x)}{P_Y(y)} \tag{1.12}$$

$$= \frac{P_{Y|X}(y|x)f_X(x)}{\int_{-\infty}^{\infty} P_{Y|X}(y|u)f_X(u)du}. \tag{1.13}$$

1.2 Бајесовско оцењивање параметара

Нека је θ непознати параметар који је циљ анализе. Идеја је донети закључке о њему на основу већ доступних информација. Те информације су добијени узорак података који зависи од вредности непознатог параметра. Податке ћемо означавати са D.
Процес закључивања почиње спецификовањем параметарског модела за процес генерисања података. Моделом задајемо како очекујемо да подаци изгледају за све могуће вредности непознатог параметра. У зависности од тога да ли су подаци апсолутно непрекидног или дискретног типа модел је представљен функцијом густине расподеле или расподелом вероватноће. У оба случаја користићемо ознаку $p(D|\theta)$ која указује на то да су подаци генерисани из неког модела који зависи од параметра. С обзиром на то да податке сматрамо познатим једном када их добијемо а параметар је непозната случајна величина уместо претходне више смисла имала би ознака $L(\theta|D)$ која се користи за функцију веродостојности код метода максималне веродостојности јер суштински представљају исту функцију. У сваком случају држаћемо се уобичајене ознаке $p(D|\theta)$ и зваћемо је функцијом веродостојности или краће веродостојношћу. Такође, задајемо и априорну расподелу параметара коју ћемо без обзира на тип параметра означавати са $\pi(\theta)$. Априорна расподела представља субјективно убеђење о параметру пре посматрања добијених података и у зависности од тога колико знамо о параметру расподела може бити мање или више информативна. Она мора бити задата али не мора бити превише утицајна[5]. Апостериорну расподелу непознатог параметра за дате податке добијамо коришћењем Бајесовог правила и за њу важи:

$$p(\theta|D) = \frac{p(D|\theta)\pi(\theta)}{p(D)} \tag{1.14}$$

$$= \frac{p(D|\theta)\pi(\theta)}{\int_\Theta p(D|\theta)\pi(\theta)d\theta}. \tag{1.15}$$

За $p(D)$ постоји више назива у литератури, међу којима су и нормирајућа константа, нормирајући фактор, маргинална веродостојност и априорна предиктивна расподела. Улога $p(D)$ је да омогући нормираност апостериорне расподеле параметра од интереса. Не зависи од непознатог параметра θ и самим тим не носи информације од значаја. Из тог разлога

[5]Постоји приступ који се зове објективни Бајес (objective Bayes) који покушава да што више умањи утицај априорне расподеле.

претходну једнакост можемо записати на компактнији начин - коришћењем пропорционалности. Тада је:

$$p(\theta|D) \propto p(D|\theta)\pi(\theta) \tag{1.16}$$

тј. апостериорна расподела је пропорционална производу априорне расподеле и функције веродостојности.
Након добијања апостериорне расподеле можемо изнети оне закључке за које сматрамо да су од значаја за анализу. Неке од могућности су графички приказ апостериорне расподеле, разне сумарне статистике као што су средња вредност, медијана, мода (мере положаја), стандардна девијација, ранг, интерквартилни ранг (мере расејања), вероватноће да параметар узме вредност из неког интервала (региона у случају вишедимензионог параметра) итд.
Уколико је циљ анализе оценити непознате параметре, постоји велики број могућности које се могу поделити у две основне групе: тачкасте и интервалне оцене.

1.2.1 Тачкасте оцене

Са $L(\theta, \hat{\theta}(D))$ је означена функција губитка[6] која мери одступање стварне вредности непознатог параметра од његове оцењене вредности. Бајесов ризик, као функција од $\hat{\theta}$, дефинише се као очекивана вредност функције губитка, где је очекивање у односу на маргиналну расподелу параметра θ, $p(\theta)$. За оцену кажемо да је Бајесова оцена уколико минимизира Бајесов ризик у класи свих оцена, тј. једнака је

$$\arg\min \int\int L(\theta, \hat{\theta}) p(\theta|D) p(D) dD d\theta = \arg\min \int\int L(\theta, \hat{\theta}) p(\theta|D) d\theta p(D) dD. \tag{1.17}$$

Како је $p(D)$ увек веће или једнако од нуле, за добијање Бајесове оцене довољно је наћи вредност $\hat{\theta}$ која минимизира очекивану вредност функције губитка у односу на апостериорну расподелу параметра θ, $p(\theta|D)$, за конкретне податке D.
За различите изборе функције губитка добијамо различите Бајесове оцене непознатог параметра. Биће наведене три функције губитка и њима одговарајуће Бајесове оцене.

1. Функција грешке је квадрат разлике вредности, $L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2$. $\hat{\theta} = \int \theta p(\theta|D) d\theta$, тј. Бајесова оцена је једнака очекивању апостериорне расподеле.

[6] Loss function

2. Функција грешке је апсолутна вредност разлике, $L(\theta, \hat{\theta}) = |\theta - \hat{\theta}|$. Тада је Бајесова оцена једнака медијани апостериорне расподеле.

3. Функција грешке узима вредност 0, уколико је $|\theta - \hat{\theta}| < \delta$, односно 1, уколико је $|\theta - \hat{\theta}| \geq \delta$. Тада је Бајесова оцена мода апостериорне функције густине расподеле. Ту оцену зовемо MAP[7] оцена.

1.2.2 Интервалне оцене

Интервалне оцене се називају интервали прекривања[8] и представљају бајесовски аналогон фреквенционистичким интервалима поверења. За задату вредност α, $100(1-\alpha)\%$ интервал поверења покрива стварну вредност параметра у просеку у $100(1-\alpha)\%$ поновљених експеримената и не можемо га схватити изван контекста поновљених експеримената док за разлику од њега $100(1-\alpha)\%$ интервал прекривања даје област параметарског простора у коме је вероватноћа покривања θ једнака $1-\alpha$. Интервал прекривања за апостериорну расподелу можемо дефинисати као скуп C, подскуп параметарског простора Θ за кога важи:

$$1 - \alpha = \int_C p(\theta|D)d\theta \tag{1.18}$$

у случају апсолутно непрекидног параметра θ док интеграл мењамо сумом а знак једнакости знаком веће или једнако у случају дискретног параметра. Тако дефинисани интервали нису јединствени јер можемо дефинисати скуп C на разне начине тако да претходна једнакост важи. Специјално, уколико желимо да креирамо интервале са једнаким реповима[9] то постижемо тако што су апостериорне вероватноће да вредност параметра буде лево и десно од интервала једнаке и износе по $\alpha/2$. У случају симетричних расподела овако дефинисан интервал веродостојности биће центриран око средње вредности.
Други специјални случај интервала прекривања је интервал највеће апостериорне густине кога ћемо надаље означавати са ИНАГ. Додатни услов је да је густина унутар интервала увек већа или једнака од густине ван њега. Уколико је апостериорна расподела вишемодална, ИНАГ ће заправо бити унија интервала. У случају унимодалне и симетричне апостериорне расподеле интервал прекривања са једнаким реповима и ИНАГ се поклапају.

[7] Maximum a Posteriori
[8] Bayesian credible intervals
[9] equal tail intervals

1.2.3 Пример оцењивања параметара

Уколико за пример узмемо испитивање утицаја разних фактора на тренутну зараду појединца истраживање можемо да извршимо тако што најпре прикупимо одређене податке. За сваког појединца имаћемо његову месечну зараду као и нпр. број година радног искуства, број година рада на тренутном радном месту, степен образовања, зараду на претходном радном месту, просечну оцену на највишем степену студија, пол, итд.
Податке можемо поделити у две целине. Са y можемо означити вектор зарада испитаника. Уколико имамо n испитаника вектор y можемо представити као вектор са n компоненти, $y = (y_1, y_2 \ldots, y_n)^T$ где је y_i зарада i-тог појединца. У овом примеру, вектор y је зависна променљива. Остале променљиве зовемо експланаторне, објашњавајуће или независне променљиве или предиктори. Означавамо их са x_j, док са X означавамо цео скуп предиктора. Уколико имамо n опсервација и p објашњавајућих променљивих онда X можемо представити матрицом са n врста и p колона. За њих можемо али не морамо да задајемо расподелу и кад их једном добијемо сматрамо их константама или реализованим вредностима случајне променљиве. Сада су подаци облика $D = (y, X)$.
Уколико претпоставимо линеарну везу између зараде и осталих променљивих тада су нам непознати параметри коефицијенти линеарне регресије као и стандардна девијација грешке. Све непознате параметре означимо са θ. Обично је полазна тачка статистичке анализе претпоставка да заједничка расподела за n вредности y_i, $p(y_1, y_2, \ldots, y_n)$ не зависи од пермутације индекса. Додатна претпоставка која може бити корисна је да су условне расподеле за y_i при датој вредности θ непознатог параметра независне и једнако расподељене. Бајесовске статистичке закључке о непознатом параметру доносимо на основу апостериорне расподеле, коришћењем тачкастих или интервалних оцена.
Након сакупљања нових података $D^{'}$, који су независни од почетних D али су генерисани на исти начин, дотадашњу апостериорну расподелу $p(\theta|D)$ можемо сматрати новом априорном и полазећи од ње и нових података добијамо нову апостериорну расподелу. На овај начин добијамо исти резултат какав бисмо добили да смо све податке имали дате на почетку:

$$p(\theta|D, D^{'}) \propto p(D^{'}|\theta)p(\theta|D) \tag{1.19}$$
$$\propto p(D^{'}|\theta)p(D|\theta)\pi(\theta) \tag{1.20}$$
$$= p(D^{'}, D|\theta)\pi(\theta). \tag{1.21}$$

Такође, при овом услову независности нових података од старих имамо и резултат да апостериорна расподела не зависи од редоследа убаци-

вања података у модел. На овај начин можемо ажурирати апостериорну расподелу колико год пута желимо и самим тим добијањем нових података доносити нове закључке о непознатом параметру.

1.3 Апостериорна предиктивна расподела

Поред доношења закључака о непознатом параметру, занима нас да за новодобијене вредности независних променљивих израчунамо одговарајуће вредности зависне променљиве. Уколико за илустрацију искористимо претходни пример, циљ је да на основу добијеног модела донесемо закључке о непознатим зарадама $\tilde{y}$, особа чији су остали подаци дати са $\tilde{X}$. То можемо да урадимо коришћењем апостериорне предиктивне расподеле:

$$p(\tilde{y}|y) = \int p(\tilde{y}, \theta|y)d\theta \tag{1.22}$$

$$= \int p(\tilde{y}|\theta, y)p(\theta|y)d\theta \tag{1.23}$$

$$= \int p(\tilde{y}|\theta)p(\theta|y)d\theta, \tag{1.24}$$

где последња једнакост важи јер су y и $\tilde{y}$ условно независне за дату вредност параметра θ.

1.4 Избор априорне расподеле

Једна од највећих критика бајесовског закључчивања је неопходност избора априорне расподеле. Критике се углавном заснивају на томе да је избор расподеле субјективан, или чак да дозвољава статистичарима да тим избором утичу на резултате анализе. Ипак, истина је да у било којој статистичкој анализи постоји извесна доза субјективизма, којој се ипак не придаје толики значај.

Један од најубедљивијих аргумената за укључивање априорне расподеле је то што често постоји претходно знање које би било добро искористити, док у ситуацијама када претходно знање не постоји може се искористити нека од расподела која неће значајно утицати на апостериорну расподелу.

Априорне расподеле можемо да поделимо у две категорије, праве и неправе. Праве расподеле испуњавају аксиоме Колмогорова. Неправе расподеле се разликују од правих по томе што интеграл њихове густине у случају апсолутно непрекидних случајних величина, односно сума

вероватноћа у случају дискретних није коначан број. Коришћење неправих априорних расподела је дозвољено, оне се често употребљавају, међутим важно је извршити проверу да ли је тако добијена апостериорна расподела права.
Постоји још једна категорија априорних расподела које се користе у бајесовским моделима, а то су конјуговане априорне расподеле. Следи дефиниција конјугованости:

Дефиниција 1.4.1. *Уколико са $\mathcal{F}$ означимо фамилију расподела $p(y|\theta)$ а са $\mathcal{P}$ фамилију априорних расподела параметра θ, $\pi(\theta)$, тада кажемо да је фамилија $\mathcal{P}$ конјугована фамилији $\mathcal{F}$ уколико важи да и апостериорна расподела, $p(\theta|y)$, припада фамилији $\mathcal{P}$.*

У прошлости, пре постојања техника за апроксимацију апостериорне расподеле, о којим ће бити речи у наредном одељку, за многе предложене моделе није било могуће израчунати апостериорну расподелу и самим тим је коришћење конјугованих априорних расподела било једино решење. Предности се огледају у једноставности и у добијању аналитичког облика апостериорне расподеле. Уколико се знање које имамо о параметру може изразити у облику конјуговане расподеле или уколико је она добра апроксимација, то треба искористити. Наравно, то често није случај и тада користимо информативне расподеле које одговарају нашем знању. Када нам је циљ да ублажимо фреквенционистичку критику субјективности приликом избора априорне расподеле, тј. када желимо да априорна има што мању улогу у креирању апостериорне расподеле или у ситуацијама када о параметру не постоји никаква позната информација, користићемо неинформативне расподеле. Неки од примера неинформативне априорне расподеле су униформна расподела на ограниченом носачу или неправа расподела која даје једнаку вероватноћу неограниченом интервалу вредности, $\pi(\theta) \propto const$, $\forall\theta \in (-\infty, \infty)$. Једна од критика коришћења униформне расподела као априорне је то што својство униформности није инваријантно при трансформацијама, тј. уколико немамо никакву информацију о непознатом параметру не би требало ни да имамо информацију о било каквој његовој трансформацији. Један познати начин креирања неинформативне априорне расподеле која је инваријантна при трансформацијама је Џефрисов[10] принцип по коме је $\pi(\theta) \propto \sqrt{|J(\theta)|}$, где је са $J(\theta)$ означена Фишерова информациона функција. У случају једнодимензионог параметра, Фишерова инфор-

[10] Sir Harold Jeffreys (1891 - 1989) - британски математичар, статистичар, геофизичар и астроном

мација се дефинише као:

$$J(\theta) = E[(\frac{d \log p(y|\theta)}{d\theta})^2|\theta]. \tag{1.25}$$

Под додатним условима регуларности важи да је:

$$J(\theta) = -E[\frac{d^2 logp(y|\theta)}{d\theta^2}|\theta]. \tag{1.26}$$

У случају вишедимензионог параметра Фишерова информација има матрични облик:

$$[J(\theta)]_{i,j} = E[(\frac{d \log p(y|\theta)}{d\theta_i})(\frac{d \log p(y|\theta)}{d\theta_j})|\theta]. \tag{1.27}$$

Под додатним условима регуларности важи да је:

$$[J(\theta)]_{i,j} = -E[\frac{d^2 logp(y|\theta)}{d\theta_i d\theta_j}|\theta]. \tag{1.28}$$

1.5 Монте Карло методе засноване на ланцима Маркова

Одређивање апостериорне расподеле директном применом Бајесовог правила укључује израчунавање маргиналне веродостојности, $p(D)$. Обично то захтева решавање интеграла који је немогуће аналитички решити. У прошлости, потешкоће до којих је долазило су се превазилазиле ограничавањем на моделе са једноставнијом функцијом веродостојности и конјугованом априорном расподелом што је доводило до интеграла који се може израчунати. Када овај метод није било могуће применити једна од могућих алтернатива је апроксимација подинтегралне функције неком другом која је једноставнија и чији интеграл умемо да решимо.
Поред тога, једна класа метода укључује нумеричку апроксимацију интеграла. Када је непознати параметар мале димензије или, на други начин речено, када постоји мали број непознатих параметара можемо да апроксимирамо интеграл сумом. Уколико модел има више параметара, што је случај у реалистичним моделима, овај приступ неће моћи да изврши апроксимацију. Када сваки параметар представимо са n дискретних вредности, а има p параметара, долазимо до тога да нам за њих треба n^p вредности њихових комбинација.
Трећи начин за апроксимацију апостериорне расподеле је узимање

случајног узорка великог броја репрезентативних комбинација вредности параметара из апостериорне расподеле. Развијени су многи такви алгоритми, и зовемо их Монте Карло методе засноване на ланцима Маркова[11], надаље МКЛМ методе. За коришћење МКЛМ метода довољно је да се за дату вредност θ вредност $\pi(\theta)$ лако израчунава, као и $p(D|\theta)$ за дате вредности θ и D, док израчунавање $p(D)$ није потребно што ове методе чини једноставним за коришћење.
Монте Карло методе за симулацију представљају широку класу алгоритама за рачунање који се заснивају на понављаном узимању случајног узорка како бисмо извукли нумеричке закључке о проблемима који могу да буду и детерминистички. Често се користе у проблемима који се јављају у математици и физици када је тешко или немогуће применити неки други приступ. Користе се углавном у три различите класе проблема: оптимизација, нумеричка интеграција и извлачење узорака из неке расподеле. У принципу, могу се користити за решавање било ког проблема који има пробабилистичку интерпретацију. Према закону великих бројева, интеграле који су заправо очекиване вредности неких случајних величина можемо апроксимирати узимањем емпиријске средње вредности тј. средње вредности случајног узорка из расподеле те случајне величине. Уколико је расподела вероватноће случајне величине параметризована математичари често користе споменуте МКЛМ методе чија је основна идеја да креирају ланац Маркова са унапред одређеном стационарном расподелом тако да ће на крају узорак генерисан методом бити заправо узорак из циљне расподеле.
МКЛМ методе генеришу случајно лутање такво да је свеки следећи корак у потпуности независан од свих претходних позиција осим тренутне. Сваки такав процес се назива марковски процес, по математичару Андреју Маркову[12], а сваки низ узастопних корака зовемо марковским ланцем.
Неки од представника МКЛМ метода су Гибсов, Метрополис-Хастингс алгоритам, као и његов специјалан случај, Метрополис алгоритам. Ови алгоритми као и софтвер за аутоматско креирање узорака и брзи хардвер који су их подржали су најзаслужнији за развој бајесовских метода.

1.5.1 Метрополис-Хастингс алгоритам

Метрополис-Хастингс алгоритам је један од МКЛМ метода за добијања низа случајних узорака из расподеле из које је директно узимање узорка

[11] *MCMC* - Markov chain Monte Carlo

[12] Андреј Марков (1856 - 1922) - руски математичар, највише познат по доприносу теорији случајних процеса

тешко. Често се користи за вишедимензионе параметре. Алгоритам је име добио по Николасу Метрополису[13] и В. К. Хастингсу[14]. Овим алгоритмом можемо узети узорак из било које расподеле $p(x)$ уколико знамо вредност неке њој пропорционалне функције $f(x)$. То овај алгоритам чини нарочито погодним за коришћење у бајесовској линеарној регресији с обзиром на то да није неопходно израчунати константу нормализације. У том случају за функцију f бирамо производ априорне расподеле и функције веродостојности. У свакој итерацији алгоритма бира се кандидат за следећу вредност узорка на основу тренутне вредности и функције предлога. Ту вредност кандидата са неком вероватноћом прихватимо као нову вредност или одбијемо и користимо постојећу у следећој итерацији. Кораке понављамо онолико пута колики узорак желимо да добијемо.

Најпре ће бити описан специјалан случај - Метрополис алгоритам. На произвољан начин бирамо и функцију $g(x|y)$ која представља функцију густине расподеле нове вредности x за дату тренутну вредност y. Ову функцију зовемо расподела предлога или расподела скока и она мора да задовољава услов симетричности, тј. да је $g(x|y) = g(y|x)$, док у Метрополис-Хастингс алгоритму то није случај. Уобичајен избор је нормална расподела центрирана у вредности y јер у том случају ће предложене нове позиције бити близу тренутних.

За иницијалну вредност изаберемо произвољну вредност x_0 у којој је задовољено да је $f(x_0) > 0$. Вредности се генеришу као узорак из случајног лутања. У свакој итерацији t најпре генеришемо кандидата $x^{'}$ користећи расподелу $g(x^{'}|x_t)$. Након тога израчунамо $\alpha = f(x^{'})/f(x_t)$ и са вероватноћом $min(1, \alpha)$ прихватамо предложену вредност у узорак. Уколико је предложена вредност прихваћена, биће $x_{t+1} = x^{'}$ а иначе $x_{t+1} = x_t$. Вредност α зовемо количник прихватања[15] и показатељ је колико је вероватна предложена вредност у односу на тренутну вредност. Што је α мања, мања је вероватноћа да ће предложена вредност ући у узорак. Понављамо овај поступак онолико пута колики узорак желимо и тиме на крају добијамо репрезентативни узорак из циљне расподеле.

[13] Nicholas Metropolis, (1915 - 1999) - грчко-амерички физичар познат по свом доприносу развоју Монте-Карло метода. Један је од аутора Equation of State Calculations by Fast Computing Machines заједно са Arianna W. Rosenbluth, Marshall Rosenbluth, Augusta H. Teller and Edward Teller у коме је предложен алгоритам који користи симетричне расподеле предлога.

[14] Wilfred Keith Hastings, (1930 - 2016) - канадски статистичар познат по свом доприносу Метрополис-Хастингс алгоритму који се састоји у уопштавању Метрополис алгоритма тако да расподеле предлога не морају бити симетричне

[15] Acceptance ratio

Метрополис-Хастингс и друге МКЛМ методе дају корелисане узорке. То значи да ако желимо скуп независних резултата узорака требало би да одбацимо већину узорка и задржимо сваки n-ти, за неко n. Аутокорелисаност можемо редуковати повећавањем "ширине скока" што се постиже повећањем стандарне девијације расподеле предлога. Међутим тиме се увећава и вероватноћа да ће предложена вредност бити одбијена и самим тим задржана тренутна вредност. Тако да и превелика и премала величина скока доводе до велике аутокорелисаности. Такође, како крећемо од произвољне иницијалне вредности пожељно је одбацити почетне вредности из узорка како бисмо спречили да избор почетне вредности утиче на резултате. Тај почетни део узорка који одбацујемо зовемо загревање.
Метрополис-Хастингс алгоритам је јако сличан и, као последица тога што расподела предлога није симетрична, једина разлика је у томе како изгледа количник прихватања. Један од честих избора је такозвани Метрополисов избор:

$$A(x^{'}, x_t) = min(1, \frac{f(x^{'})}{f(x_t)} \frac{g(x_t|x^{'})}{g(x^{'}|x_t)}).$$

У вишедимензионим проблемима тешко је пронаћи адекватну расподелу предлога и у таквим ситуацијама се Гибсов алгоритам показао као ефикасније решење.

1.5.2 Гибсов алгоритам

Други тип МКЛМ алгоритма, Гибсово узимање узорка[16] се користи за моделе са вишедимензионим параметром. Алгоритам је име добио по физичару Гибсу[17], а увели су га браћа Стјуарт[18] и Доналд Геман[19] 1984. године док су радили на истраживању какви закључци се могу извући о слици на основу улаза који се састоји из њених пиксела.
Процедура Гибсовог алгоритма је донекле иста као код Метрополис-Хастингс алгоритма, узорци се добијају случајним лутањем кроз простор параметара. Лутање креће из случајно изабране тачке и у свакој

[16] Gibbs sampling

[17] Josiah Willard Gibbs (1839 - 1903) - амерички научник који је имао значајан допринос физици, хемији и математици

[18] Stuart Geman (1949 -) - амерички математичар, познат по свом доприносу компјутерској визији, теорији вероватноће и статистици, као и машинском учењу и неуронауци

[19] Donald Jay Geman(1943 -) - амерички примењени математичар и водећи истраживач у пољу машинског учења и препознавања образаца

тачки следећи корак зависи само од тренутне позиције. Разлика је то што овде у сваком кораку бирамо једну компоненту параметра. Редослед избора компоненти може бити случајан, али ипак се из практичних разлога користи цикличан избор, тј. компоненте се бирају јасно дефинисаним редоследом који се не мења. Претпоставимо да смо изабрали компоненту θ_i. Тада нову вредност тог параметра узимамо директно из условне расподеле, $p(\theta_i|\{\theta_j\}_{j\neq i}, D)$. Нова вредност θ_i, комбинована са непромењеним вредностима $\{\theta_j\}_{j\neq i}$ представља нову вредност случајног лутања. Процес затим понављамо са следећом компонентом непознатог параметра. Након што за све компоненте утврдимо новодобијене позиције, процес се понавља.
Овај метод је јако користан када не можемо да одредимо заједничку расподелу, $p(\theta|D)$, али можемо условне расподеле $p(\theta_i|\{\theta_j\}_{j\neq i}, D)$ и самим тим можемо да генеришемо узорак. Уколико бисмо упоредили ова два приступа видели бисмо да тачке које се добијају помоћу ових алгоритама изгледају слично, али не и њихове трајекторије. У сваком случају, конвергирају ка истој расподели.
Један од недостатака Гибсовог алгоритма је тај што треба да изведемо условне вероватноће сваког параметра и да генеришемо случајне узорке из тих расподела. Још један недостатак је то што због промене само једног параметра у једној итерацији процес може да буде заустављен због високе корелације међу параметрима. Данас постоје бројне модификације[20] основног алгоритма које су направљене са циљем редуковања аутокорелације у довољној мери да се превазиђу било какви додатни трошкови израчунавања.

1.5.3 Репрезентативност, тачност и ефикасност

Репрезентативност, тачност и ефикасност представљају три основна циља при генерисању узорка из апостериорне расподеле.

1. Вредности ланца морају бити репрезентативни представници апостериорне расподеле. Не би требало да избор иницијалне вредности параметра утиче на њих и требало би да у потпуности истраже ранг апостериорне расподеле.

2. Ланац би требало да буде довољне величине тако да оцене буду тачне и стабилне што значи да се поновљеним генерисањем МКЛМ ланца добију приближни резултати.

[20] Blocked Gibbs sampler, Collapsed Gibbs sampler, Gibbs sampler with ordered overrelaxation...

3. Ланац би требало да буде генерисан ефикасно са што мање корака.

Оно што је загарантовано у сваком МКЛМ ланцу је да би бесконачно дуги ланци представљали идеалну репрезентацију апостериорне расподеле, што се никад не може десити због ограничених ресурса којима располажемо (првенствено време и меморија рачунара). Зато је потребно да проверимо квалитет коначних ланаца како бисмо проверили постоје ли знаци нерепрезентативности или нестабилности. Са порастом сложености модела ланци постају проблематичнији и њихово проверавање постаје важније и већи изазов. У последњем поглављу ће на конкретном примеру бити испитан квалитет узорка генерисаног из апостериорне расподеле коришћењем МКЛМ метода.

1.6 Апостериорна предиктивна провера

Апостериорна предиктивна провера је провера уклапања резултата добијених моделом са стварним подацима. Не постоји јединствен начин за проверу да ли предвиђања модела систематски и значајно одступају од стварних података јер постоји много начина за дефинисање одступања. Неколико начина за проверу одступања биће наведено у наставку.
Модел можемо проверити валидацијом, тако што за нови скуп података $\tilde{X}$ симулирамо $\tilde{y}$ а затим тако добијене вредности упоредимо са стварним. Уколико немамо приступ новим подацима сличан ефекат постижемо поделом почетног узорка на два дела, део који се користи за развој модела - тренинг скуп и део који се користи за тестирање - тест скуп. Још један приступ валидацији је унакрсна валидација[21]. Најпре дефинишемо k група а затим у свакој од k итерација развијемо модел на $k-1$ групи а тестирамо га на преосталој. У свакој итерацији бирамо различиту групу за тестирање и самим тим различити су и скупови за развој модела.
Такође, модел је могуће проверити и графички. На једном графику представимо вредности које би предвидео модел као и податке које већ имамо. Најпре за цео скуп X генеришемо узорак из апостериорне предиктивне расподеле за y. Уколико модел одговара подацима, тада подаци које добијамо генерисањем из модела треба да изгледају слично постојећим. Било какво одступање је показатељ потенцијалне неадекватности модела. Уколико би се десило да постоји систематско одступање, требало би размотрити друге моделе, нпр. неке који имају нелинеаран тренд. Такође, могли бисмо да испитамо својства података. Уколико би се испоставило да подаци имају аутлајере у односу на оно што је предвиђено

[21] Cross validation

нормалном расподелом, могли бисмо да изменимо модел тако да користи расподеле са тешким репом као што је Студентова расподела.
Провера адекватности модела је веома битан корак у свакој анализи података, па тако и у бајесовској линеарној регресији. Типичан случај јесте да више од једног модела може да обезбеди адекватно уклапање података и модела.

1.7 Анализа сензитивности модела

У књизи [1] аутор наводи да након што утврдимо да је неки модел адекватан за дате податке можемо да урадимо анализу осетљивости, тј. да проверимо колико промена неких претпоставки модела утиче на добијене закључке. Модели се могу разликовати по много чему. Спецификација априорне расподеле, функција веродостојности или избор предиктора које укључујемо у модел су само неки од примера потенцијалних разлика. Могуће је да више модела приближно одговарају подацима а да закључци добијени из тих модела буду различити. Промене модела различито утичу на различита питања. Нпр. већи је утицај на средње вредности и екстремне квантиле, него на медијану апостериорне расподеле. Понекад је могуће извршити анализу осетљивости употребом робуснијих модела који обезбеђују да екстремне вредности података не доведу до погрешних закључака. Типичан пример робусног модела је употреба Студентове t расподеле за функцију веродостојности, уместо нормалне која се обично претпоставља.
И провера модела и анализа сензитивности се могу сматрати делом уобичајене анализе која се врши након формирања узорка из апостериорне расподеле. У било ком проблему примене модела на доношење закључака, постојаће знање које није формално укључено ни у априорну расподелу ни у функцију веродостојности. Уколико додатна информација доводи до закључка да су апостериорни закључци погрешни то указује на потребу креирања прецизнијих и тачнијих модела.

Поглавље 2

Класична - фреквенционистичка линеарна регресија

У овом поглављу изложен је кратак осврт на класичну линеарну регресију. Регресиона анализа се користи за објашњавање или моделовање везе између једне променљиве, Y, коју зовемо одговор, исход или зависна променљива и једне или више независних или објашњавајућих променљивих, $X_1, \ldots X_p$, које још зовемо и предикторима. Када је $p = 1$ реч је о простој а када важи $p > 1$ ради се о вишеструкој линеарној регресији. Зависна променљива је непрекидног типа док независне променљиве могу бити било непрекидног било категоријског типа. Неки од основних циљева линеарне регресије су:

- Предвиђање будућих опсервација зависне променљиве за дате вредности независних
- Процена ефекта независних променљивих на зависну променљиву
- Одређивање аналитичко-математичког облика одговарајуће везе.

Након што одредимо облик модела треба оценити параметре а затим испитати квалитет тако добијеног модела. Случај вишеструке линеарне регресије можемо записати у матричном облику на следећи начин:

$$y = X\beta + \epsilon \tag{2.1}$$

где је

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & & & \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \epsilon = \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_p \end{bmatrix}. \tag{2.2}$$

Са y смо означили вектор вредности које одговарају зависној променљивој. Матрицу X зовемо дизајн матрицом. Прва колона је ту како би модел укључио слободан члан док су остале колоне добијени подаци који представљају предикторе. Случајност модела потиче од случајних грешака ϵ за које претпостављамо да испуњавају одређене услове: центрираност, хомоскедастичност и некорелисаност, тј. претпостављамо да важи следеће:

1. $E(\epsilon) = 0$
2. $Cov(\epsilon) = \sigma^2 I$
3. X и ϵ су независни случајни вектори.

Вектор β је вектор параметара модела, где његова димензија p одговара броју независних променљивих. Коефицијенте модела оцењујемо методом најмањих квадрата, тј. важи да је $\hat{\beta}$ она вредност за коју се достиже минимум суме квадрата грешака, $\sum_i \epsilon_i^2$. Дакле циљ нам је да минимизујемо

$$\sum_i \epsilon_i^2 = \epsilon^T \epsilon = (y - X\beta)^T (y - X\beta) = y^T y - 2\beta X^T y + \beta^T X^T X \beta. \tag{2.3}$$

Диференцирањем и изједначавањем са нулом долазимо до тога да оцена $\hat{\beta}$ задовољава следећи систем који зовемо систем нормалних једначина:

$$X^T X \hat{\beta} = X^T y. \tag{2.4}$$

Уколико је ранг дизајн матрице једнак броју p тада је матрица $X^T X$ инвертибилна, систем нормалних једначина има јединствено решење и добијамо оцену параметара β,

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{2.5}$$

Оцењена вредност вектора y је тада

$$\hat{y} = X(X^T X)^{-1} X^T y = Hy. \tag{2.6}$$

Матрица H[1] је пројектор, тако да $\hat{y}$ представља ортогоналну пројекцију вектора y на раван генерисану са X. Разлику између стварних и оцењених вредности вектора y зовемо вектором резидуала модела и он се може приказати у следећем облику:

$$e = y - \hat{y} = (I - H)y. \tag{2.7}$$

Одавде закључимо да је

$$E(e) = (I - H)E(y) = 0 \tag{2.8}$$

$$Cov(e) = \sigma^2(I - H)(I - H)^T = \sigma^2(I - H). \tag{2.9}$$

Као меру одступања стварних вредности података од вредности предвиђених моделом можемо користити суму квадрата резидуала, SSE[2]. Мале вредности указују на добро уклапање модела са подацима.

$$SSE = \sum_i {e_i}^2 = e^T e = y^T(I - H)y = y^T y - y^T H y.$$

Непристрасну оцену за σ^2 добијамо управо дељењем SSE са бројем степени слободе, тј. $\hat{\sigma^2} = \frac{SSE}{n-p-1}$.

$$\sum_i (y_i - \bar{y})^2 = \sum_i (y_i - \hat{y_i})^2 + \sum_i (\hat{y_i} - \bar{y})^2 \tag{2.10}$$

$$SST = SSE + SSR. \tag{2.11}$$

SSE је необјашњено одступање које потиче од модела, SSR објашњено одступање а SST је укупно одступање.
За процену квалитета модела можемо да користимо коефицијент детерминације, $R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$. За коефицијент детерминације важи да је $0 \leq R^2 \leq 1$, где су вредности ближе 1 показатељи бољег уклапања. Са $R = \sqrt{R^2}$ је дефинисан вишеструки коефицијент корелације. Пошто коефицијент детерминације расте са порастом броја предиктора који улазе у модел погодно је користити алтернативу која узима у обзир непристрасне оцене грешака и дефинисана је помоћу:

$$R_A^2 = 1 - \frac{\frac{SSE}{n-p-1}}{\frac{SST}{n-1}}. \tag{2.12}$$

Уколико додатно претпоставимо да $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ тада се оцене за параметар β добијене помоћу метода најмањих квадрата и метода максималне веродостојности поклапају. Пошто је $\hat{\beta}$ линеарна трансформација случајног вектора са нормалном расподелом под том додатном претпоставком важиће да $\hat{\beta} \sim \mathcal{N}(\beta, (X^T X)^{-1}\sigma^2)$.

[1] Матрицу H зовемо hat matrix

[2] SSE -The sum of squared errors of prediction

Поглавље 3

Бајесовска линеарна регресија

У овом поглављу ће кроз два примера бити илустрована бајесовска линеарна регресија.
У првом примеру представљен је једноставан линеарни модел који је често полазна тачка бајесовске анализе регресионих модела. Наведени су теоријски резултати, упоређена су два приступа регресији - бајесовски и класични. Такође, приказани су начини провере уклапања модела са подацима, као и апостериорна предиктивна расподела. Резултати су илустровани на примеру просте линеарне регресије на генерисаном скупу података а могу се применити на произвољан скуп података и произвољан број предиктора.
Други пример је пример вишеструке линеарне регресије на скупу реалних података. Закључци се доносе на основу узорка из апостериорне расподеле који је добијен применом Гибсовог алгоритма. Како је наведено у одељку1.5, након добијања узорка потребно је проверити да ли је дошло до конвергенције, тако да је у овом примеру приказано неколико начина за дијагностику. Такође, извршена је и анализа осетљивости упоређивањем резултата који се добијају при различитим моделима.
Сви кодови писани су у програмском језику R [5], у RStudio [6] окружењу, а сви значајни делови су представљени у одељцима.

3.1 Пример 1

Кренућемо од најједноставнијег и највише примењиваног случаја линеарне зависности променљиве y од предиктора $(x_1, \ldots, x_p)$. То је случај нормалног линеарног модела[1] где је расподела за y при датом X нор-

[1] Normal linear model

мална и чија је средња вредност линеарна функција од X:

$$E(y_i|\beta, X) = \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \forall i \in 1 \ldots n.$$

$(x_{i1}, \ldots, x_{ip})$ је вектор врста вредности независне променљиве који одговара i-тој опсервацији, $(\beta_1, \ldots \beta_p)$ вектор колона непознатих вредности регресионих коефицијената, а X је дизајн матрица описана у претходном поглављу. Допустимо да је $x_{i1} = 1, \forall i$, како би модел имао и слободни члан.
Под додатним претпоставкама да су y_i условно независни за дате вредности параметара и предиктора, као и да је $D(y_i|\beta, X) = \sigma^2, \forall i$, модел постаје још једноставнији. Тај случај зовемо обичном линеарном регресијом[2]. Вектор непознатих параметара је $\theta = (\beta_1, \ldots, \beta_p, \sigma^2) = (\beta, \sigma^2)$. Коначно запишемо модел:

$$y|\beta, \sigma^2, X \sim \mathcal{N}(X\beta, \sigma^2 I),$$

где је са $\mathcal{N}$ означена вишедимензионална нормална расподела а са I јединична матрица димензија $n \times n$.
Полазећи од тако дефинисане функције веродостојности и изабране априорне расподеле израчунавамо аналитички облик апостериорне расподеле или, уколико то није могуће, закључке о непознатим параметрима доносимо на основу добијеног узорка из апостериорне расподеле.
Како се наводи у [1], поглавље 14, један од честих и погодних избора априорне расподеле непознатих параметара је неправа априорна расподела која је униформна за $(\beta, \log \sigma)$, односно:

$$\pi(\beta, \sigma^2) \propto \frac{1}{\sigma^2}.$$

То је стандардна неинформативна априорна расподела. Када постоји мало параметара и довољно тачака, овакав избор неинформативне априорне расподеле је погодан и даје прихватљиве резултате. Заједничку апостериорну расподелу непознатих параметара можемо представити као производ условне и маргиналне расподеле:

$$p(\beta, \sigma^2|y) = p(\beta|\sigma^2, y)p(\sigma^2|y).$$

Условна апостериорна расподела за β при датом σ^2 је вишедимензиона нормална:

$$\beta|\sigma^2 \sim \mathcal{N}(\hat{\beta}, \sigma^2(X^T X)^{-1}),$$

[2] Ordinary linear regression

где је $\hat{\beta}$ оцена која се добија методом најмањих квадрата и, као што је наведено у претходном поглављу, за њу важи да је једнака $(X^TX)^{-1}X^Ty$. Апостериорна расподела параметра σ^2 се може представити као скалирана инверзна χ^2 расподела, за коју уводимо ознаку $\chi^{2^{-1}}$, и као инверзна гама расподела, за коју уводимо ознаку γ^{-1}. Прецизније, важи следећа једнакост у расподели:

$$\sigma^2 \sim \gamma^{-1}(\frac{n-p}{2}, (n-p)\frac{s^2}{2}) = \chi^{2^{-1}}(n-p, s^2),$$

где је са n означен обим узорка, са p број предиктора, укључујући и слободан члан а са s^2 означен је SSE из претходног поглавља, $s^2 = \frac{(y-X\hat{\beta})^T(y-X\hat{\beta})}{n-p}$.

За случајну величину V кажемо да има $\gamma^{-1}(a,b)$ расподелу уколико је можемо представити као $V = 1/U$ где је U случајна величина са $\gamma(a,b)$ расподелом, где је a параметар облика и b инверз параметра скалирања. Уколико случајна величина U има $\chi^{2^{-1}}(\nu, \tau^2)$ тада $\frac{U}{\tau^2\nu}$ има $\chi^{2^{-1}}_\nu$ расподелу, коју можемо представити као инверз случајне величине са χ^2_ν расподелом.

С обзиром на то да смо користили неправу априорну расподелу непознатих параметара, важно је проверити да ли је добијена апостериорна расподела права. По [1], $p(\beta, \sigma^2|y)$ јесте права уколико су задовољена 2 услова: $n > p$ и $rang(X) = p$.

Такође, могуће је израчунати маргиналну апостериорну расподелу параметра β и то је вишедимензиона Студентова расподела са $n-p$ степени слободе. Међутим, у пракси се то ретко користи, већ за доношење закључака о апостериорној расподели непознатих параметара користимо симулације. Симулација узорка из апостериорне расподеле параметара (β, σ^2) може се вршити тако што се најпре симулира вредност σ^2 из маргиналне апостериорне расподеле, а затим се за тако добијену вредност σ^2 симулира β из условне апостериорне расподеле. Та два корака понављамо онолико пута колики узорак желимо.

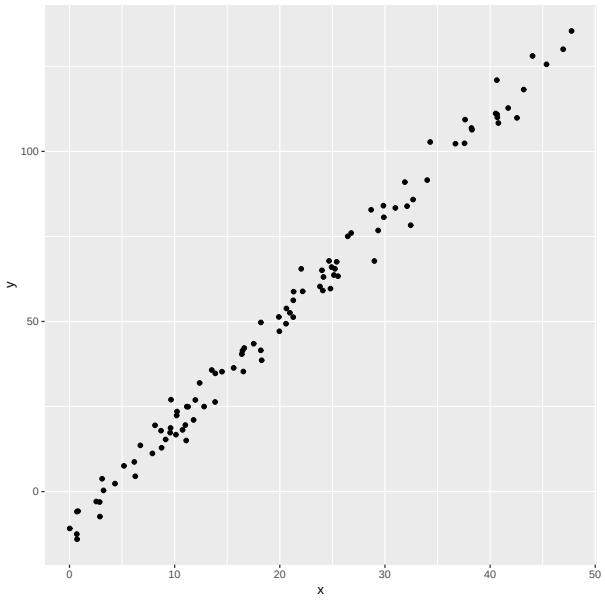
Претпоставимо да имамо нови скуп података $\tilde{X}$ и да желимо да предвидимо вредности $\tilde{y}$. У класичном приступу бисмо за предвиђену вредност изабрали $\tilde{X}\hat{\beta}$. У бајесовском имамо апостериорну предиктивну расподелу $p(\tilde{y}|y)$. За симулирање вредности $\tilde{y}$ изабраћемо узорак (β, σ^2) из заједничке апостериорне расподеле а затим $\tilde{y}$ из $\mathcal{N}(X\beta, \sigma^2 I)$. За потребе ове симулације можемо искористити претходно симулирани узорак из заједничке апостериорне расподеле непознатих параметара.

За потребе илустровања бајесовске линеарне регресије под наведеним претпоставкама користићу функције из пакета LearnBayes [6]. Креатор пакета је уједно и аутор књиге [4] која је послужила као инспирација за

овај пример. Овде је наведен пример просте линеарне регресије а поступак би био аналоган у случају вишеструке линеарне регресије.
Најпре је генерисан узорак x обима 100 из униформне $\mathcal{U}[0, 50]$ расподеле. Затим је на $-10 + 3x$ додат шум, узорак истог обима из нормалне расподеле са очекивањем једнаким 0 и стандардном девијацијом једнаком 4, и тиме је добијен y.

```
set.seed(11)
n = 100
x <- runif(n, 0, 50)
random_effect1 <- rnorm(n, sd = 4)
y <- -10 + 3 * x + random_effect1
data1 <- data.frame(y = y, x = x)
```

Ове вредности параметрара шума и кофицијената линеарне везе произвољно су изабране. Идеја је да скуп података задовољава све неопходне услове како бисмо могли да применимо класичну линеарну регресију. За потребе репродуковања резултата вредност seed-a постављена је на 11.
На следећем графику приказан је дијаграм зависности y од x:



Коришћењем функције lm креирамо линеарни модел методом најмањих квадрата. Позивом функције summary прикажемо добијени модел, тј. оцењене вредности коефицијената, њихову стандардну грешку, вредност t-статистике, p-вредности, затим дескриптивне статистике резидуала,

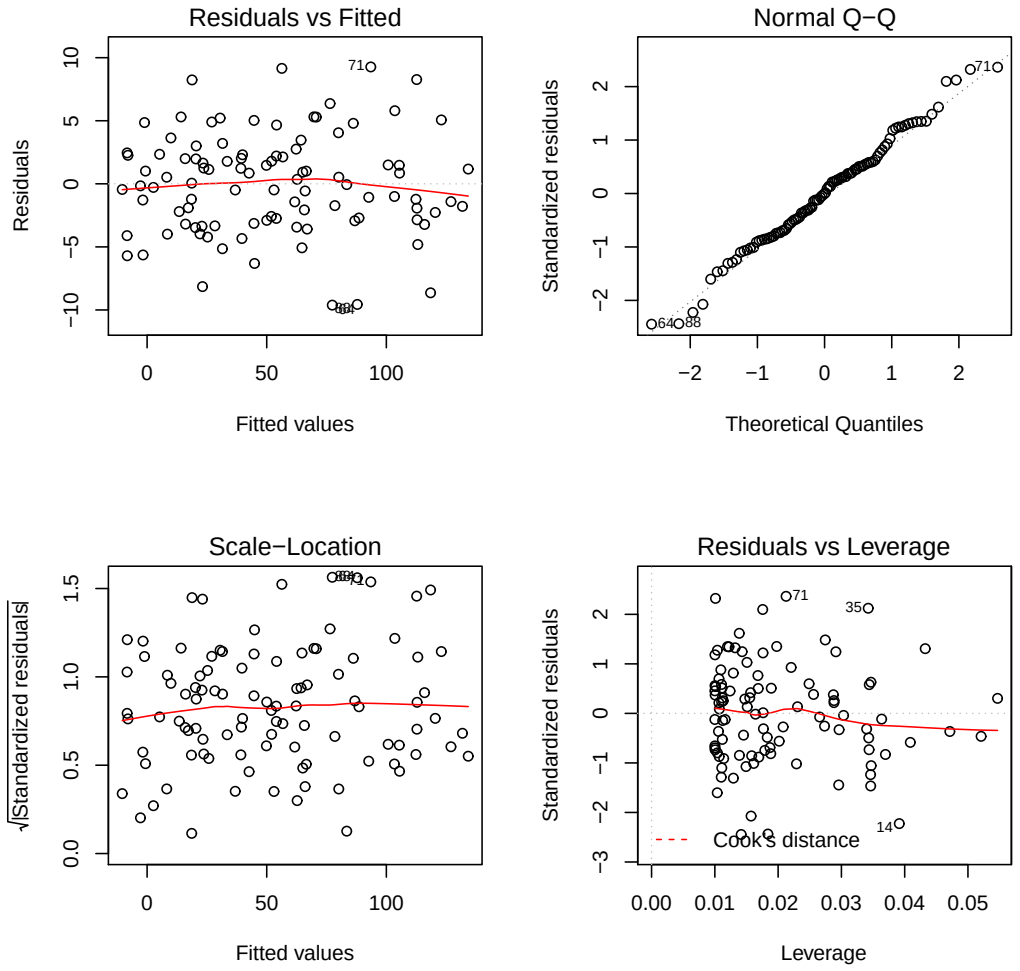
као и оцењену вредност стандардне девијације резидуала, вредности R^2 и $\hat{R}^2$ као и вредности F-статистике и резултујуће p-вредности. Прегледом резултата који је добијен позивом summary функције можемо да закључимо да модел одговара подацима. Овакви резултати су били очекивани имајући у виду начин на који су подаци генерисани.

```
model1 <- lm(y ~ x, data = data1, x = TRUE, y = TRUE)
```

```
> summary(model1)
Call:
lm(formula = y ~ x, data = data1, x = TRUE, y = TRUE)

Residuals:
Min        1Q    Median        3Q       Max
-9.61764 -2.86392 -0.00595  2.27075  9.26403

Coefficients:
Estimate Std. Error t value  Pr(>|t|)
(Intercept) -10.473542   0.756162 -13.851 < 2.2e-16 ***
x             3.030890   0.031031  97.672 < 2.2e-16 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *
   0.05    .    0.1          1

Residual standard error: 3.962 on 98 degrees of freedom
Multiple R-squared:  0.98983, Adjusted R-squared:  0.98973
F-statistic: 9539.8 on 1 and 98 DF,  p-value: < 2.22e-16
```

Резидуале ћемо испитати позивом plot функције:

```
> plot(model1)
```



На основу првог и трећег графика можемо да закључимо да су резидуали хомоскедастични. На основу другог графика да су приближно нормални.

Као мера одступања стварних вредности од вредности које су предвиђене моделом често се изабере средњеквадратна грешка[3] - средња вредност квадрата резидуала. Уколико рачунамо средњеквадратну грешку на целом скупу података, једноставно је израчунамо као

```
> mean((model1$residuals) ^ 2)
[1] 15.38
```

[3] MSE - mean squared error

Значајније је видети како модел функционише на новим подацима које нисмо употребили за развој модела. То можемо постићи на више начина од којих ће бити приказана подела на тренинг и тест скуп, унакрсна валидација и поновљена унакрсна валидација.
Уколико за тренинг скуп изаберемо случајан узорак кога чине 75% иницијалних података а за тест скуп преосталих 25% добијемо вредност средњеквадратне грешке на тест узорку која износи 19.23.

```
test <- sample(1:100, size = 25)
testData <- data1[test, ]
trainData <- data1[-test, ]
actual = testData$y
predicted = predict(lm(y ~ x, data = trainData), data.frame
    (x = testData$x))
mse = mean((actual - predicted) ^ 2)
```

```
> mse
[1] 19.23
```

Наравно, овај поступак можемо поновити више пута, сваки пут изабравши другачији тренинг и тест скуп, како бисмо израчунали средњу вредност тако добијених средњеквадратних грешака што представља репрезентативнији резултат.
На сличан начин вршимо и унакрсну валидацију. Најпре случајним узимањем узорака поделимо цео скуп на четири групе и у свакој итерацији по једну групу користимо за тест а преостале три за развој модела. Тако добијемо четири различите вредности MSE и за резултат унакрсне валидације изаберемо средњу вредност тако добијених вредности. Тим поступком добије се резултат 15.99.

```
data2 <- data1[sample(nrow(data1)), ]
folds <- cut(seq(1, nrow(data2)), breaks = 4, labels =
    FALSE)
mse <- c()
for (i in 1:4) {
  testIndexes <-  which(folds == i, arr.ind = TRUE)
  testData <- data2[testIndexes,]
  trainData <- data2[-testIndexes,]
  actual = testData$y
  predicted = predict(lm(y ~ x, data = trainData), data.
    frame(x = testData$x))
  mse[i] = mean((actual - predicted) ^ 2)
  }
```

```
> mean(mse)
[1] 15.99
```

И овај поступак можемо поновити више пута, сваки пут бирајући групе на другачији начин и тако добијемо резултат поновљене унакрсне валидације 16.08.

```
rep_mse <- c()
for (j in 1:100) {
data2 <- data1[sample(nrow(data1)), ]
folds <- cut(seq(1, nrow(data2)), breaks = 4, labels =
   FALSE)
mse <- c()
for (i in 1:4) {
testIndexes <- which(folds == i, arr.ind = TRUE)
testData <- data2[testIndexes,]
trainData <- data2[-testIndexes,]
actual = testData$y
predicted = predict(lm(y ~ x, data = trainData), data.frame
   (x = testData$x))
mse[i] = mean((actual - predicted) ^ 2)
}
rep_mse[j] <- mean(mse)
}
```

```
> mean(rep_mse)
[1] 16.08
> range(rep_mse)
[1] 15.55 17.54
```

Закључак је да је средњеквадратна грешка приближно 16.
Прелазимо на бајесовску линеарну регресију. Коришћењем функције blinreg из LearnBayes paketa креирамо одговарајући узорак обима 5000 из апостериорне расподеле непознатих параметара.

```
y = data1$y
X = matrix(nrow = n, ncol = 2)
X[, 1] = rep(1, n)
X[, 2] = data1$x
theta.sample1 = blinreg(y, X, 5000)
beta_1 <- theta.sample1$beta[, 2]
beta_0 <- theta.sample1$beta[, 1]
sigma <- theta.sample1$sigma
```

Као што је објашњено у одељку 1.2.1, у зависности од избора функције губитка имамо различите оцене непознатог параметра, и то: очекивање и медијана уколико су нам функције губитка средњеквадратна грешка и средња апсолутна грешка, респективно.
Када је апостериорна расподела параметара симетрична и унимодална, те оцене ће се поклапати. То је случај са коефицијентима линеарне регресије. Како закључке изводимо на основу узорка из апостериорне расподеле за оцене ћемо изабрати узорачку средњу вредност, односно узорачку медијану па самим тим резултати неће бити једнаке већ приближне вредности. Такође, биће приближне и вредностима добијеним класичним приступом јер је за априорну расподелу параметра β коришћена неинформативна расподела.
У случају параметра σ, апостериорна расподела није симетрична и уколико нас занимају тачкасте оцене, оне ће имати различите вредности за различите функције губитка. Упоредићемо тако добијене оцене са 3.030890 и -10.473542 што су претходно добијене оцене коефицијената линеарне регресије, као и са 3.962 што је оцена стандардне девијације.

```
> mean(beta_1)
[1] 3.03069
> median(beta_1)
[1] 3.0304

> mean(beta_0)
[1] -10.4667
> median(beta_0)
[1] -10.4656

> mean(sigma)
[1] 3.99429
> median(sigma)
[1] 3.97934
```

Оцењене вредности коефицијената регресије на оба начина су приближно једнаке, као и приближно једнаке оцени добијеној стандардним приступом. Исто важи и за оцењене вредности стандардне девијације грешке.
Поред тачкастих оцена за сваки узорак из апостериорне расподеле можемо приказати и мере одступања: стандардну девијацију, распон вредности, интерквартилно растојање. У следећем делу видимо колико те вредности износе.

```
> sd(beta_1)
```

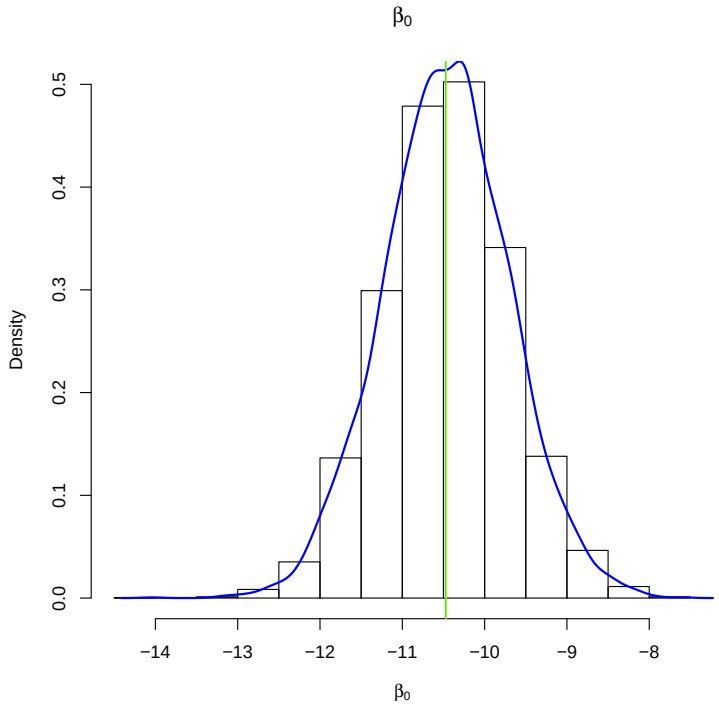
```
[1] 0.0314768
> range(beta_1)
[1] 2.89661 3.16568
> summary(beta_1)
Min. 1st Qu.  Median    Mean 3rd Qu.
2.90    3.01    3.03    3.03    3.05
Max.
3.17

> sd(beta_0)
[1] 0.766842
> range(beta_0)
[1] -14.04267  -7.60063
> summary(beta_0)
Min. 1st Qu.  Median    Mean 3rd Qu.
-14.04  -10.98  -10.47  -10.47   -9.95
Max.
-7.60

> sd(sigma)
[1] 0.28787
> range(sigma)
[1] 3.14359 5.22072
> summary(sigma)
Min. 1st Qu.  Median    Mean 3rd Qu.
3.14    3.79    3.98    3.99    4.18
Max.
5.22
```

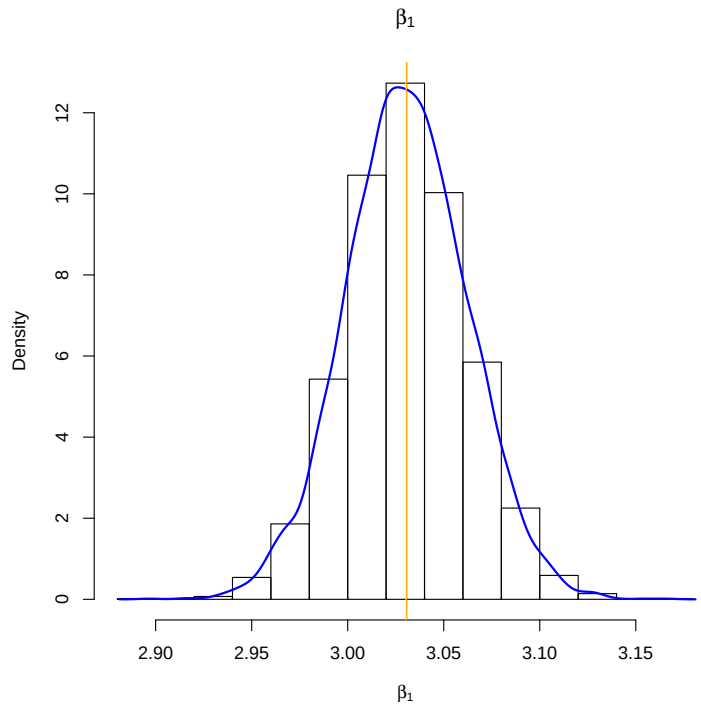
Уколико за интервалну оцену параметра изаберемо интервале прекривања са једнаким реповима, једноставно их можемо израчунати коришћењем узорачких квантила. За произвољну вредност α, интервалну оцену добијемо као интервал чије су границе: $q_{\frac{\alpha}{2}}$ и $q_{1-\frac{\alpha}{2}}$ где је са q_a дефинисан узорачки квантил реда a. Узмимо за пример $\alpha = 0.05$.

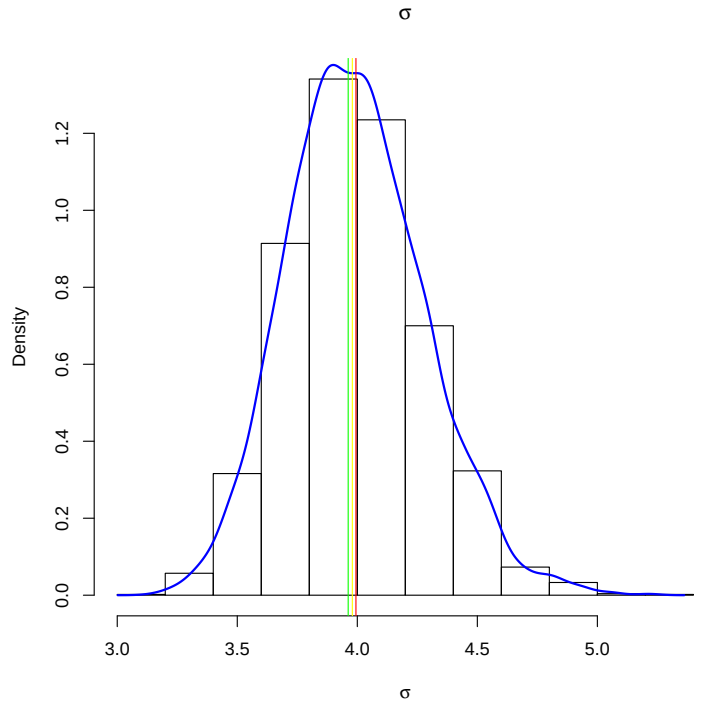
```
> quantile(beta_1, probs = c(0.025, 0.975))
2.5%   97.5%
2.96757 3.09251
> quantile(beta_0, probs = c(0.025, 0.975))
2.5%     97.5%
-11.97650  -8.94543
> quantile(sigma, probs = c(0.025, 0.975))
2.5%   97.5%
```

```
9 3.47294  4.58339
```

Као што је раније било споменуто, у случају унимодалне и симетричне расподеле ови интервали ће се поклапати са интервалима највеће густине. Поред тога што смо приказали оцењене вредности, можемо и графички представити апостериорну расподелу непознатих параметара. На следећа три графика приказани су хистограми узорака из апостериорне расподеле непознатих параметара, њихова оцењена густина (плава боја), оцена параметра методом најмањих квадрата (зелена боја) као и бајесовске оцене које су узорачка средња вредност (црвена) и медијана (жута).







Поред избора априорне расподеле, битан аспект који утиче на апосте-

риорну расподелу је величина узорка података. За процену утицаја обима узорка случајним избором изабрани су подскупови полазног скупа података величина 10, 30 и 70 а затим за сваки од њих креиран узорак из апостериорне расподеле непознатих параметара позивом blinreg функције. Једноставности ради, биће приказан утицај само на параметар β_1. У следећем коду приказано је креирање подскупа обима $n = 10$ а аналогно поступамо за било коју вредност n.

```
n = 10
test <- sample(1:100, size = n)
testData <- data1[test, ]
y_10 = testData$y
X_10 = matrix(nrow = n, ncol = 2)
X_10[, 1] = rep(1, n)
X_10[, 2] = testData$x
theta.sample_10 = blinreg(y_10, X_10, 5000)
beta1_10 = theta.sample_10$beta[, 2]
```

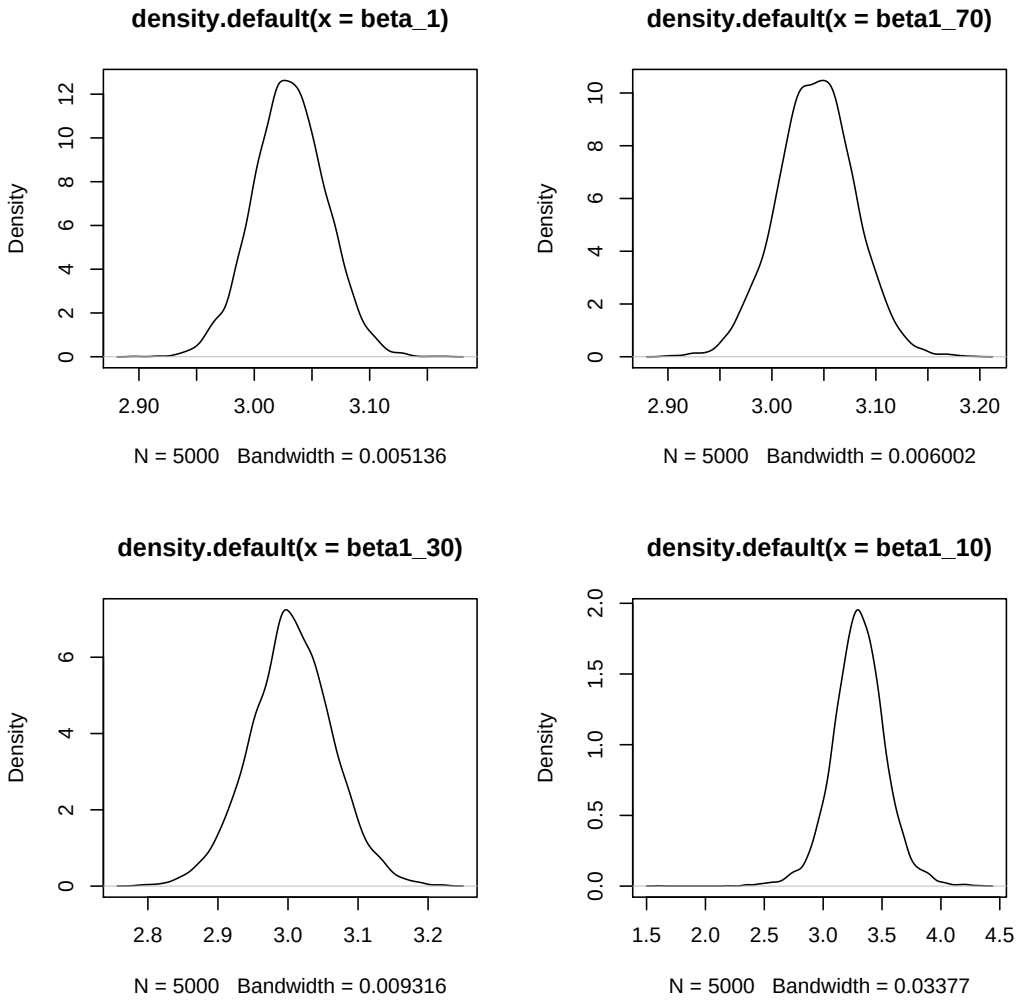
Након што су сви узорци креирани упоредимо 95%-не интервале прекривања са једнаким реповима.

```
> quantile(beta_1, probs = c(0.025, 0.975))
2.5%  97.5%
2.9676 3.0925
> quantile(beta1_70, probs = c(0.025, 0.975))
2.5%  97.5%
2.9684 3.1153
> quantile(beta1_30, probs = c(0.025, 0.975))
2.5%  97.5%
2.8861 3.1245
> quantile(beta1_10, probs = c(0.025, 0.975))
2.5%  97.5%
2.8482 3.7609
```

Можемо упоредити и опсег вредности из узорака.

```
> range(beta_1)
[1] 2.8966 3.1657
>   range(beta1_70)
[1] 2.8974 3.1942
>   range(beta1_30)
[1] 2.7842 3.2222
>   range(beta1_10)
[1] 1.6013 4.3375
```

Као и оцењене густине апостериорних расподела:



Порастом обима узорка смањује се опсег вредности које су генерисане из апостериорне расподеле непознатог параметара као и интервали прекривања. Тиме расте сигурност у закључке добијене из апостериорне расподеле.

Поред одређивања облика везе један од циљева регресионе анализе је одређивање предвиђених вредности зависне променљиве за дате нове вредности независне. Изабремо нпр. вредности: -10, 1, 7, 20, 40, 70, где су све вредности осим -10 и 70 унутар опсега вредности које су коришћене за креирање модела.

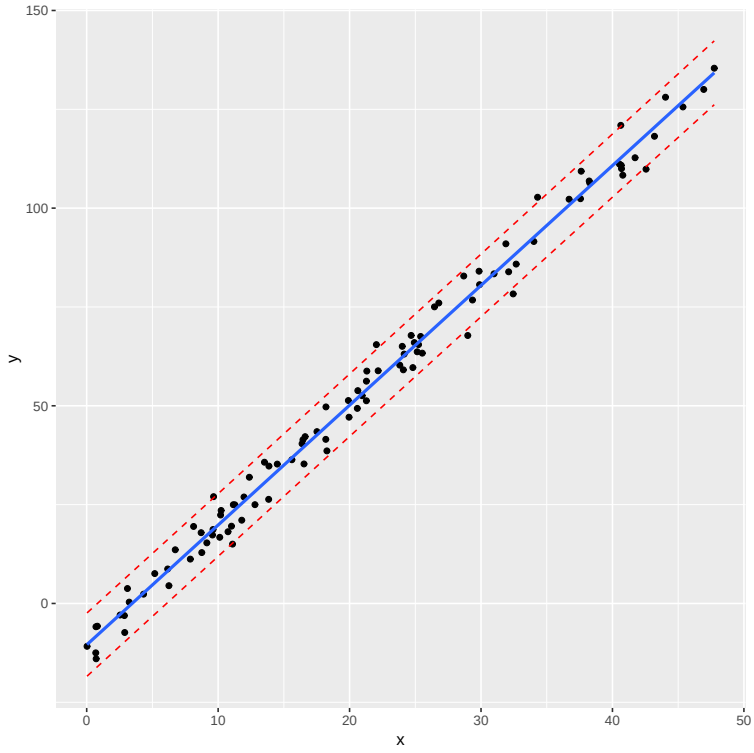
У методу најмањих квадрата, поред оцењених вредности можемо добити и 95% интервале поверења и предвиђања.

```
new_data <- data.frame(x = c(1, 7, 20, 40, -10, 70))
mean_values = predict(model1, newdata = new_data, interval
   = "confidence")
predicted_values = predict(model1, newdata = new_data,
   interval = "prediction")
values = predict(model1, newdata = new_data)
```

```
> mean_values
fit       lwr       upr
1  -7.4427  -8.8911  -5.9942
2  10.7427   9.5870  11.8984
3  50.1443  49.3567  50.9319
4 110.7621 109.3399 112.1843
5 -40.7824 -42.8331 -38.7318
6 201.6888 198.5560 204.8216
> predicted_values
fit      lwr       upr
1  -7.4427 -15.437   0.55191
2  10.7427   2.796  18.68942
3  50.1443  42.243  58.04586
4 110.7621 102.772 118.75190
5 -40.7824 -48.908 -32.65717
6 201.6888 193.225 210.15217
```

Уколико уместо тако изабраних вредности независне промењиве за предвиђање искористимо све вредности x које су коришћене за креирање модела добијамо следећи график. Испрекидане црвене линије представљају границе интервала предвиђања.

```
pred.int <-    predict(model1, interval = "prediction")
mydata <- cbind(data1, pred.int)
library("ggplot2")
p <- ggplot(mydata, aes(x, y)) + geom_point() + stat_smooth
   (method = lm) + geom_line(aes(y = lwr), color = "red",
   linetype = "dashed") + geom_line(aes(y = upr), color = "
   red", linetype = "dashed")
```



Приметимо да постоје тачке за које предвиђене вредности не упадају у 95%-интервал предвиђања, и то су :

```
> options(digits = 5)
> mydata[(mydata$y > mydata$upr | mydata$y < mydata$lwr), ]
          y        x      fit      lwr      upr
12   65.471  22.0375   56.320   48.418   64.222
14  109.855  42.5521  118.497  110.483  126.512
35  120.947  40.6334  112.682  104.686  120.678
55   15.010  11.0964   23.158   15.234   31.082
64   67.779  28.9917   77.397   69.479   85.315
71  102.757  34.3024   93.493   85.548  101.439
88   78.302  32.4450   87.864   79.929   95.798
98   27.016   9.6514   18.779   10.848   26.710
```

У бајесовском приступу линеарној регресији ћемо за сваку изабрану вредност независне промењљиве симулирати узорак из апостериорне предиктивне расподеле а затим за предвиђену вредност изабрати средњу вредност узорка. Користимо функцију blinregpred.

```
cov1 = c(1, 1)
cov2 = c(1, 7)
cov3 = c(1, 20)
```

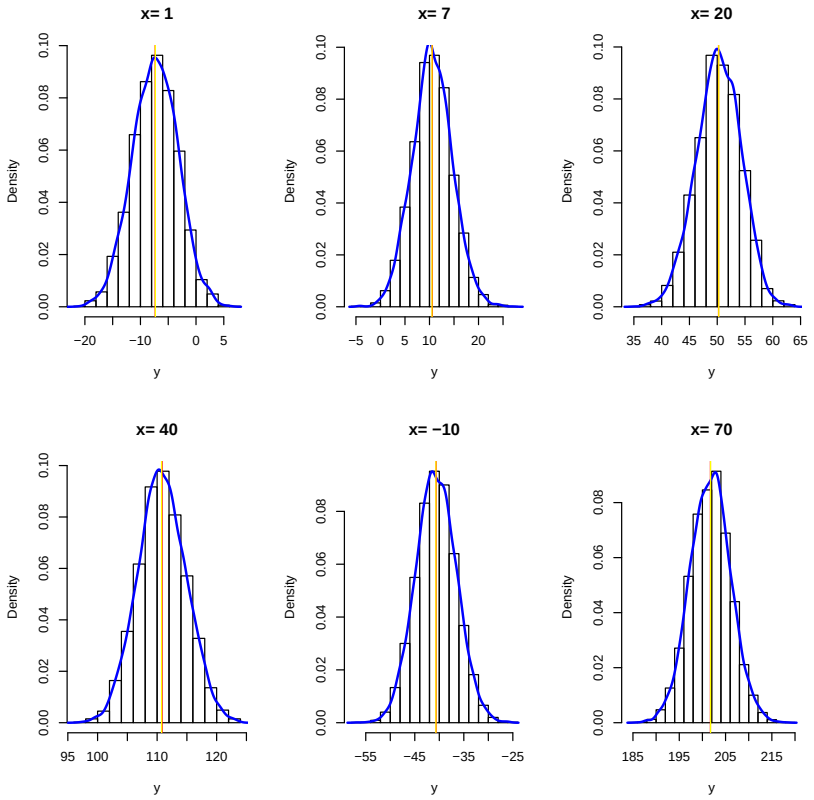
```
cov4 = c(1, 40)
cov5 = c(1, -10)
cov6 = c(1, 70)
X1 = rbind(cov1, cov2, cov3, cov4, cov5, cov6)
pred.draws = blinregpred(X1, theta.sample1)
> colMeans(pred.draws)
[1]  -7.3482  10.5864  50.2350 110.8330 -40.6527 201.7299
```

Тако добијемо оцењене вредности које можемо упоредити са оним вредностима предвиђеним коришћењем predict функције.

```
> colMeans(pred.draws) - values
1          2          3          4          5          6
0.094469 -0.156333  0.090765  0.070973  0.129784  0.041099
```

Видимо да не постоје значајне разлике у предвиђеним вредностима. Можемо да креирамо хистограме узорака из апостериорне предиктивне расподеле. Плавом бојом је означена оцењена густина, црвеном средње вредности, а жутом медијане узорка. Следи код за креирање хистограма.

```
par(mfrow = c(2, 3))
for (j in 1:6) {
hist(
pred.draws[, j],
main = paste("x=", X1[j, 2]),
xlab = "y",
probability = TRUE
)
lines(density(pred.draws[, j]), lwd = 2, col = "blue")
abline(v = mean(pred.draws[, j]), col = "red")
abline(v = median(pred.draws[, j]), col = "yellow")
}
```



Како бисмо могли да закључимо колико се подаци који су предвиђени коришћењем модела слажу са стварним подацима, можемо израчунати MSE на бајесовки начин, аналогно класичном приступу. Једина разлика је то што у бајесовском приступу уместо једне предвиђене вредности $\hat{y}_i$ најпре одредимо узорак из апостериорне предиктивне расподеле за y_i а затим за сваки узорак израчунамо средњу вредност, коју затим искористимо као оцену предвиђене вредности и онда применимо формулу за MSE. Поступак је приказан у следећем коду.

```
pred.draws = blinregpred(X, theta.sample1)
pred.mean = apply(pred.draws, 2, mean)
actual <- data1$y
mse <- mean((actual - pred.mean) ^ 2)
> mse
[1] 15.442
```

На сличан начин можемо добити одступања предвиђених података од стварних коришћењем унакрсне валидације. У следећем коду је дат цео поступак.

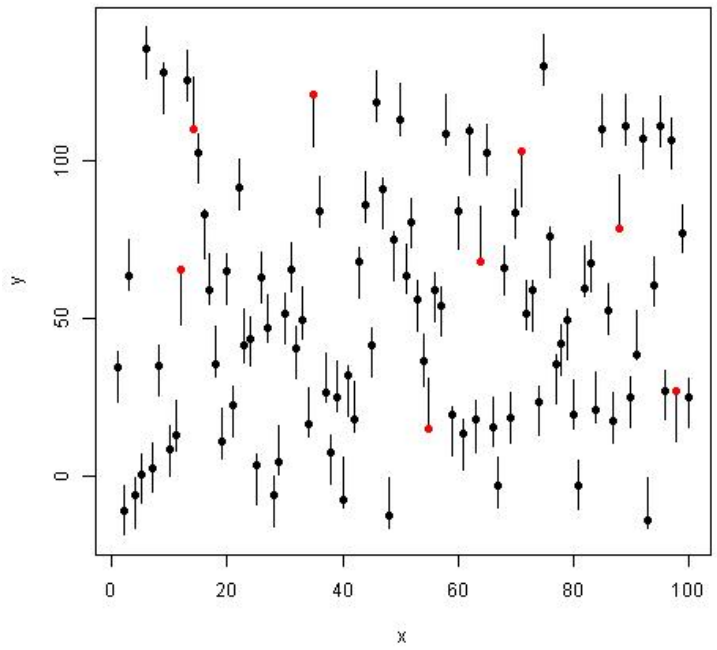
```
data <- data1[sample(nrow(data1)), ]
folds <- cut(seq(1, nrow(data)), breaks = 4, labels = FALSE
    )
mse <- c()
m_train <- matrix(ncol = 2, nrow = 75)
m_train[, 1] <- rep(1, 75)
m_test <- matrix(ncol = 2, nrow = 25)
m_test[, 1] <- rep(1, 25)
for (i in 1:4) {
testIndexes <- which(folds == i, arr.ind = TRUE)
testData <- data[testIndexes,]
trainData <- data[-testIndexes,]
m_train[, 2] <- trainData$x
m_test[, 2] <- testData$x
theta.sample = blinreg(trainData$y, m_train, 5000)
pred.draws = blinregpred(m_test, theta.sample)
pred.mean = apply(pred.draws, 2, mean) #ocene y
actual = testData$y
mse[i] = mean((actual - pred.mean) ^ 2)
}
> mean(mse)
[1] 15.869
```

Тако добијени резултати су нешто нижи од резултата добијених методом најмањих квадрата.

Други приступ апостериорној предиктивној провери је графички приказ уклапања стварних података са резултатима који су добијени симулирањем из апостериорне предиктивне расподеле. На следећем графику су представљени 95% интервали прекривања добијени на основу узорка из апостериорне предиктивне расподеле y_i као и стварне вредности y_i. Проверавамо да ли стварне вредност упадају у одговарајуће интервале. За интервале прекривања изабрани су они са једнаким реповима који се могу израчунати помоћу квантила из узорка. Како би график био прегледнији, на њему нису приказане вредности независне променљиве па зато након графика приказујемо које вредности су потенцијални аутлајери. График се добија помоћу следећег кода.

```
pred.draws = blinregpred(X, theta.sample1)
pred.sum = apply(pred.draws, 2, quantile, c(.025, .975))
par(mfrow = c(1, 1))
ind = 1:length(data1$y)
matplot(
rbind(ind, ind),
```

```
pred.sum,
type = "l",
lty = 1,
col = 1,
xlab = "1:100",
ylab = "y"
)
matpoints(ind, y, pch = 19, col = 1)
out = (y > pred.sum[2, ] | y < pred.sum[1, ])
matpoints(ind[out], y[out], pch = 19, col = 2)
```



Постоји осам опсервација које не упадају у интервале прекривања и то су исте опсервације које нису упале у одговарајуће интервале предвиђања у класичном приступу. Међутим, прегледом графика јасно је да су све те вредности приближно једнаке једној од граница интервала прекривања па их нећемо сматрати аутлајерима и закључујемо да модел одговара подацима. Уколико би се десило да су одступања већа, један од првих корака би било разматрање неког робуснијег модела. У следећем делу су приказане поменуте опсервације.

```
> cbind(data1$x[out], data1$y[out])
```

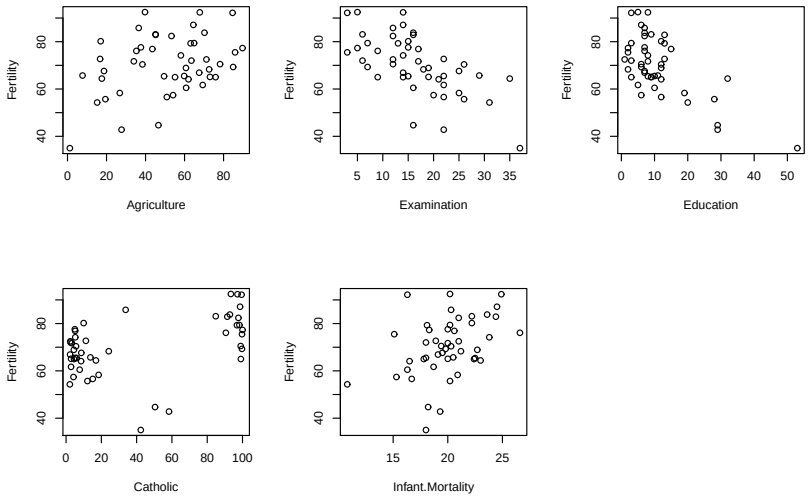
```
     [,1]   [,2]
[1,] 22.038  65.47
[2,] 42.552 109.86
[3,] 40.633 120.95
[4,] 11.096  15.01
[5,] 28.992  67.78
[6,] 34.302 102.76
[7,] 32.445  78.30
[8,]  9.651  27.02
```

3.2 Пример 2

Скуп података који се користи у овом примеру је swiss из datasets пакета и може се пронаћи у R-у. Подаци су сакупљени 1888. године, када се Швајцарска суочила са периодом који је назван демографском транзицијом. - стопа рађања деце је кренула да опада. Скуп се састоји од 47 опсервација које одговарају административним областима Швајцарске у којима се говори француски језик. Свакој опсервацији одговара следећих 6 променљивих:

1. Fertility - стандардизована мера плодности
2. Agriculture - проценат мушкараца чије је главно занимање пољопривреда
3. Examination - проценат војних лица који су остварили највишу оцену на испитивању
4. Education - проценат особа са вишим образовањем
5. Catholic - проценат католика
6. Infant.Mortality - стопа смртности у првој години живота.

Све променљиве су изражене у процентима, док је Fertility скалирана на вредности из интервала $[0, 100]$.
На следећим графицима приказана је зависност Fertility од осталих променљивих.



Уколико креирамо класични линеарни регресиони модел коришћењем свих променљивих можемо да закључимо да променљива Examination није значајан предиктор.

```
Call:
lm(formula = Fertility ~ ., data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-15.2743  -5.2617   0.5032   4.1198  15.3213

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
Agriculture      -0.17211    0.07030  -2.448  0.01873 *
Examination      -0.25801    0.25388  -1.016  0.31546
Education        -0.87094    0.18303  -4.758 2.43e-05 ***
Catholic          0.10412    0.03526   2.953  0.00519 **
Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
---
Signif. codes:  0  ***  0.001  **  0.01  *
 0.05  .  0.1  1

Residual standard error: 7.165 on 41 degrees of freedom
Multiple R-squared:  0.7067,  Adjusted R-squared:  0.671
F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

Због тога креирамо модел у који не укључујемо ту променљиву. Тада добијамо следеће резултате:

```
Call:
lm(formula = Fertility ~ . - Examination, data = data)

Residuals:
Min        1Q    Median        3Q       Max
-14.6765  -6.0522   0.7514   3.1664  16.1422

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)      62.10131    9.60489   6.466 8.49e-08 ***
Agriculture      -0.15462    0.06819  -2.267  0.02857 *
Education        -0.98026    0.14814  -6.617 5.14e-08 ***
Catholic          0.12467    0.02889   4.315 9.50e-05 ***
Infant.Mortality  1.07844    0.38187   2.824  0.00722 **
---
Signif. codes:  0    ***    0.001    **    0.01    *
 0.05    .    0.1         1

Residual standard error: 7.168 on 42 degrees of freedom
Multiple R-squared:  0.6993,  Adjusted R-squared:  0.6707
F-statistic: 24.42 on 4 and 42 DF,  p-value: 1.717e-10
```

Аналогно претходном примеру, креирани су бајесовски линеарни модели полазећи од исте функције веродостојности и истих априорних расподела непознатих параметара. Најпре је креиран модел са свим променљивим. Као што је било и очекивано, оцењене вредности параметара су приближне вредностима добијеним класичним приступом. Од интереса је коефицијент који одговара променљивој Examination.

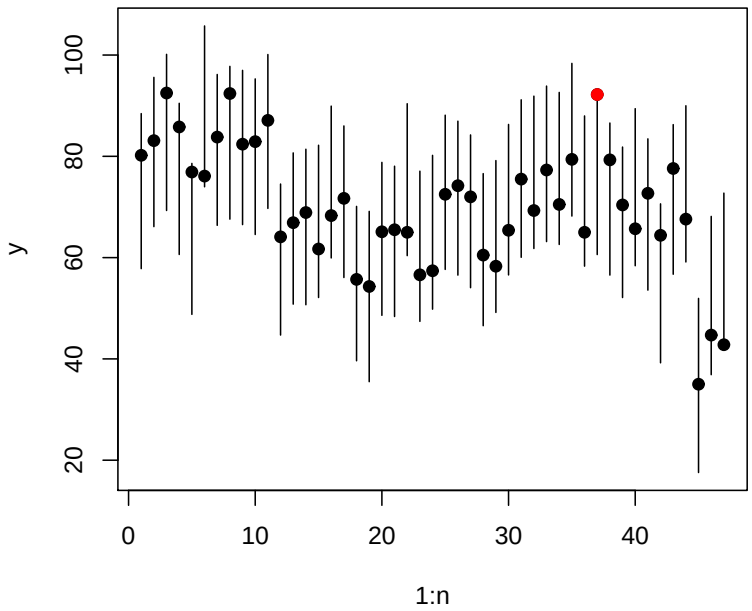
```
> int = quantile(theta.sample1$beta[,3], probs = c(.025,
  .975));
> int
2.5%    97.5%
-0.7714  0.2454
```

Приметимо да ИНАГ апостериорне расподеле тог коефицијента садржи 0 тако да и у овом случају доносимо исти закључак, променљива није значајна. Креирамо зато модел без те променљиве. Као што је и претходно био случај, и овде су оцењене вредности приближне оценама добијеним класичним приступом. Сви остали кораци су у потпуности аналогни претходном примеру, зато ће пре анализе сензитивности мод-

ела бити приказане још само интервалне оцене и апостериорна предиктивна провера.
Интервалне оцене коефицијената линеарне регресије:

```
> intervals = apply(theta.sample2$beta,2,quantile, c
  (.025, .975));
> intervals
Xrep(1, n) XAgriculture XEducation XCatholic
2.5%         42.60       -0.2946    -1.2787    0.06697
97.5%        81.69       -0.0147    -0.6755    0.18248
XInfant.Mortality
2.5%                0.2803
97.5%               1.8521
```

График стварних вредности зависне променљиве и 95%-ни интервал прекривања апостериорне предиктивне расподеле:



Једина тачка чија стварна вредност не упада у интервал је:

```
> data[out,]
Fertility Agriculture Examination Education Catholic
Sierre      92.2        84.6           3         3
 99.46
```

```
4 Infant.Mortality
5 Sierre              16.3
```

Посматрањем графика можемо закључити да не постоје значајна одступања модела од података. Међутим, постоје опсервације чије се стварне вредности налазе у крајевима интервала и због тога можемо покушати да креирамо модел у коме за расподелу грешке уместо нормалне изаберемо неку расподелу са тежим реповима. Биће наведен један такав модел у коме је за расподелу грешке изабрана Студентова t-расподела. Такође, с обзиром на то да је обим узорка релативно мали, а коришћена је неинформативна априорна расподела, резултујући интервали прекривања су релативно широки, а самим тим је већа несигурност у добијене оцене.

Из тих разлога, у наставку ће бити креирано неколико модела са различитим изборима априорне расподеле и функције веродостојности. Добијене резултате можемо упоредити на много начина, као што су: тачкасте и интервалне оцене непознатих параметара, поређења оцењених функција густине расподеле, затим уклапање модела са подацима, што се такође може извршити на више начина од којих су неки наведени у претходном примеру. Овде ће бити илустрован утицај на функцију густине расподеле, а на основу узорака из апостериорне расподеле могу се урадити и преостала поређења. На крају се поставља питање који модел изабрати. Овде ће за финални модел бити изабран модел који се највише уклапа са подацима, а за меру одступања изабрано је средњеквадратно одступање.

За потребе генерисања узорака из апостериорне расподеле користи се Гибсов алгоритам, описан у одељку 1.5.2. Алгоритам је имплементиран у JAGS[4] [7] програму. То је програм који се користи за статистичку анализу бајесовских модела коришћењем МКЛМ метода. Последња верзија програма, JAGS 4.3.0, може се преузети са адресе https://sourceforge.net/projects/mcmc-jags/files/. На истој адреси могу се пронаћи и упутства за инсталацију и коришћење програма, као и низ корисних примера. Иако програм није коришћен директно већ посредством пакета из R-a, неопходно је да претходно буде инсталиран.

Постоји више R пакета који представљају интерфејс за JAGS а овде је коришћен runjags [8]. Више детаља о пакету може се пронаћи на адресама http://runjags.sourceforge.net/ и https://CRAN.R-project.org/package=runjags

Делови кодова су преузети из програма који прате књигу [3] и могу се пронаћи на сајту [10]. У наставку је приказан помоћни код у коме

[4] JAGS - Just Another Gibbs Sampler

је дефинисана функција sampleCreation којом се генерише узорак из апостериорне расподеле за конкретан избор функције веродостојности и априорне расподеле непознатих параметара.

```
sampleCreation = function( data , xName="x" , yName="y" ,
numSavedSteps=15000 , thinSteps=1 , saveName=NULL ,
runjagsMethod=runjagsMethodDefault ,
nChains=3,
adapt = 1000 , # Za objasnjenja videti JAGS user manual
burnin = 4000 # Za objasnjenja videti JAGS user manual
) {

# Podaci
y = data[,yName]
X = as.matrix(data[,xName],ncol=length(xName))

# Ovde mozemo ubaciti provere podataka

# Kreiranje liste od pocetnih podataka
dataList = list(
X = X ,
y = y ,
Nx = dim(X)[2] ,
Ntotal = dim(X)[1]
)

# Definicija modela
modelString = "

# Standardizovanje podataka, i X i y
data {
ym <- mean(y)
ysd <- sd(y)
for ( i in 1:Ntotal ) {
zy[i] <- ( y[i] - ym ) / ysd
}
for ( j in 1:Nx ) {
xm[j]  <- mean(X[,j])
xsd[j] <-  sd(X[,j])
for ( i in 1:Ntotal ) {
zx[i,j] <- ( X[i,j] - xm[j] ) / xsd[j]
}
}
}
```

```
41
42  # Specifikacija modela za standardizovane podatke:
43  model {
44  for ( i in 1:Ntotal ) {
45  zy[i] ~ dnorm( zbeta0 + sum( zbeta[1:Nx] * zx[i,1:Nx] ) ,
      1/zsigma^2 ) #ovde je 1/zsigma^2 preciznost, odnosno
     zsigma^2  disperzija
46  }
47
48  # Izbor apriornih raspodela:
49  zbeta0 ~ dnorm( 0 , 1/2^2 )   # stdev je 2
50  for ( j in 1:Nx ) {
51  zbeta[j] ~ dnorm( 0 , 1/2^2 )
52  }
53  lnzsigma  ~ dunif( -3, 3 )
54  zsigma <-exp(lnzsigma)
55
56  # Transformacija na originalnu skalu
57  beta[1:Nx] <- ( zbeta[1:Nx] / xsd[1:Nx] )*ysd
58  beta0 <- zbeta0*ysd  + ym - sum( zbeta[1:Nx] * xm[1:Nx] /
      xsd[1:Nx] )*ysd
59  sigma <- zsigma*ysd
60
61  }
62  "
63  # inicijalne vrednosti
64  initsList <- replicate(3,
65  list(zbeta0= rnorm(1,0,2),
66  zbeta=rnorm(dataList$Nx,0,2),
67  lnzsigma= runif(1,-3, 3)),
68  simplify=FALSE)
69
70  parameters=c("beta0" , "beta" , "sigma" )
71
72  # Kreiranje lanaca pomocu run.jags funkcije
73  runJagsOut <- run.jags( method=runjagsMethodDefault,
74  model=modelString,
75  monitor=parameters,
76  data=dataList ,
77  inits=initsList ,
78  n.chains=nChains ,
79  adapt=adapt ,
80  burnin=burnin ,
```

```
  sample=ceiling(numSavedSteps/nChains) ,
  thin=thinSteps ,
  summarise=TRUE ,
  plots=TRUE )

  # Mozemo da sacuvacemo rjags objekat
  if ( !is.null(saveName) ) {
  save( runJagsOut , file=paste(saveName,".Rdata",sep="") )
  }

  # Vratimo rjags, kasnije transformisemo u mcmc.list
  return( runJagsOut )
```

Помоћни кодови су скоро идентични, са једином разликом у спецификацији модела. Сачувани су у посебним фајловима, из којих се позивају у главни код помоћу source функције. Сваки код се састоји из дефинисања листе података, модела, листе иницијалних вредности параметара, избора параметара који су нам од интереса и дела у коме се позива run.jags функција и тиме се креира rjags објекат у коме су, између осталог, сачувани узорци из апостериорне расподеле. Како би се обезбедило да подаци буду на истој скали стандардизоване су и зависна и независне промењљиве. За тако трансформисане промењљиве изабране су информативне апостериорне расподеле које имају велику стандардну девијацију па утицај на апостериорну расподелу не би требало да буде велики. Пошто нас занимају расподеле параметара које одговарају почетним, нестандардизованим промењљивим, добијени узорци се трансформишу у циљне и генеришемо узорак њихове апостериорне расподеле. У следећем делу приказан је део главног кода у коме позивамо претходно дефинисану функцију помоћу које креирамо rjags објекат који одговара задатом моделу.

```
set.seed(11)

library(coda)
library(runjags)

nChainsDefault = 3 # Kreiraju se 3 lanca
runjagsMethodDefault = "rjags"
# Ukoliko procesor ima vise jezgara mozemo koristiti
# metod za paralelno izracunavanje - "parallel"
numSavedSteps=15000
# Ukupan broj elemenata u svim lancima  (=3x5000)
thinSteps=1
```

```
# Ukoliko primetimo visoku acf onda povecamo thinSteps,
# inace cuvamo svaki element u generisanom uzorku
adaptSteps = 1000
# Inicijalni koraci koji se ne cuvaju u uzorku
# neophodni za adaptiranje generatora
burnInSteps = 4000
# Neophodan za eliminisanje uticaja inicijalnih vrednosti

data=swiss
yName = "Fertility"

source("Normal-Normal-ExpUniform.R")
fileNameRoot = "Normal-Normal-ExpUniform"
xName = c("Agriculture","Education","Catholic","Infant.
  Mortality")

runjags1 = sampleCreation( data=data , xName=xName ,
  yName=yName ,
numSavedSteps=numSavedSteps , thinSteps=thinSteps ,
saveName=fileNameRoot,
adapt=adaptSteps, burnin=burnInSteps )
```

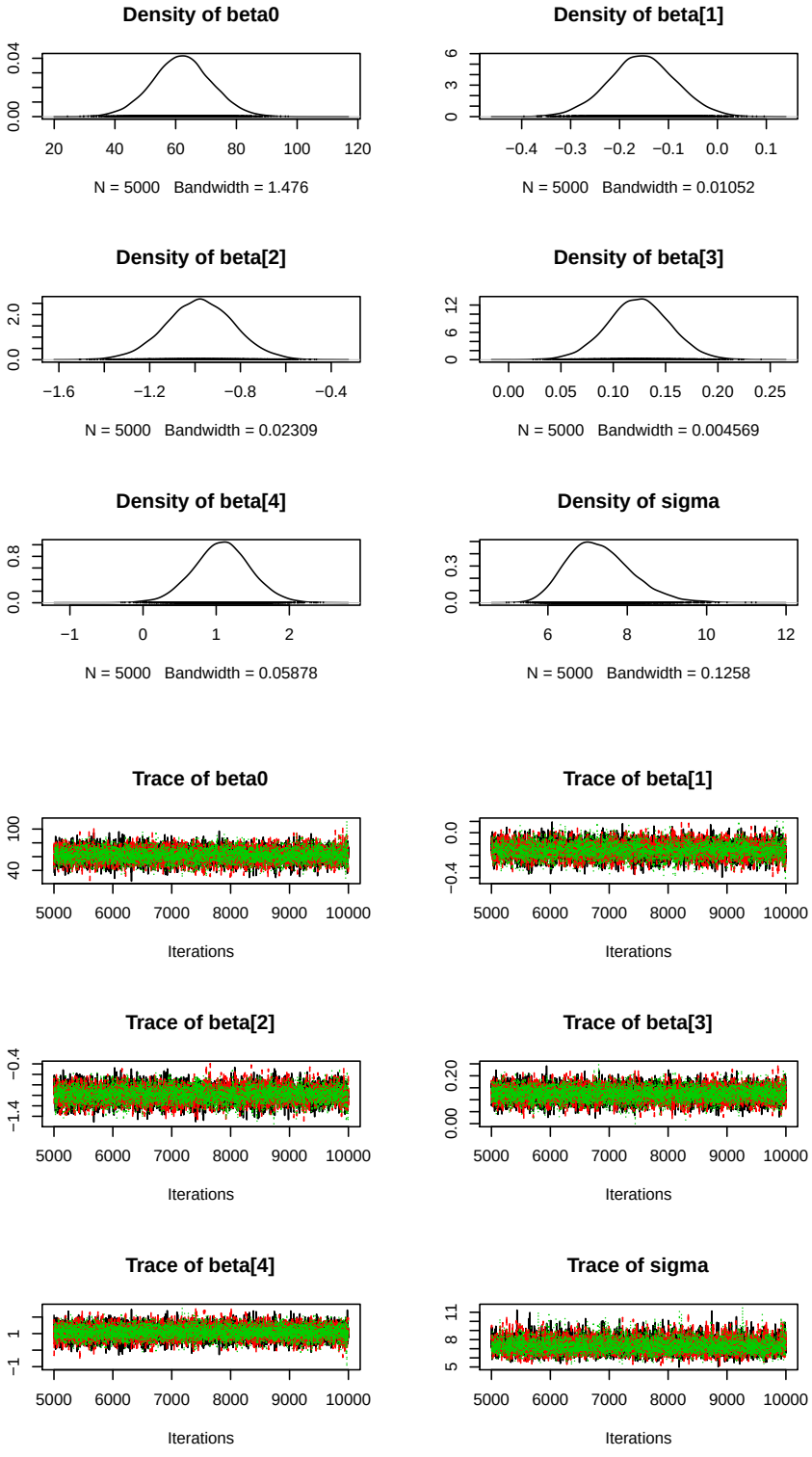
У овом конкретном случају претпостављен је модел који има нормалну функцију веродостојности, нормалну априорну расподелу коефицијената који одговарају стандардизованим зависним променљивим и претпостављена је априорна расподела девијације грешке таква да њен природни логаритам има униформну расподелу на симетричном интервалу.

Након добијања резултата функције прелазимо на испитативање конвергенције ланаца ка циљној расподели. Као што је наведено, користимо функције из coda пакета.

Први корак је преглед графика[5] који за све ланце приказују вредности узорка као и графика оцењених густина узорака из апостериорне расподеле непознатих параметара. Графици су приказани у наставку и добијамо их следећим кодом:

```
mcmcCoda1<-as.mcmc.list(runjags1)
par(mfrow=c(3,2))
traceplot(mcmcCoda)
densplot(mcmcCoda)
```

[5] Traceplot


Density of beta0
N = 5000 Bandwidth = 1.476
Density of beta[1]
N = 5000 Bandwidth = 0.01052
Density of beta[2]
N = 5000 Bandwidth = 0.02309
Density of beta[3]
N = 5000 Bandwidth = 0.004569
Density of beta[4]
N = 5000 Bandwidth = 0.05878
Density of sigma
N = 5000 Bandwidth = 0.1258
Trace of beta0
Trace of beta[1]
Trace of beta[2]
Trace of beta[3]
Trace of beta[4]
Trace of sigma
Iterations

Прегледом графика закључимо да не постоје очигледни показатељи да ланци нису исконвергирали.
Следећи корак су тестови конвергенције. У овом пакету су имплементирани следећи тестови: Гевеке[6] (geweke.diag, geweke.plot), Гелман и Рубин[7] (gelman.diag, gelman.plot), Рафтери и Луис[8] (raftery.diag), Хајделбергер и Велч[9] (heidel.diag). Овде ће детаљније бити описан Гевеке тест, а описи свих тестова се могу пронаћи у документацији која прати одговарајуће функције.
Гевеке дијагностика заснована је на тестирању једнакости средњих вредности првог и последњег дела ланца. Тест статистика је Z-скор који се добија дељењем разлике узорачких средина њиховом оцењеном стандардном грешком. Такође, битна претоставка је да су та два дела асимптотски независна. Функција geweke.diag рачуна вредност Z-скора и можемо задати величине првог и последњег дела ланца. Уколико их не наведемо претпостављају се вредности 10% и 50%, респективно. Уколико добијемо вредности које су у екстремним реповима стандардне нормалне расподеле закључујемо да није дошло до конвергенције. У тим случајевима можемо да позовемо функцију geweke.plot помоћу које се израчунају вредности Z-скора након одбацивања почетних делова ланца на основу чега можемо да закључимо колики део узорка би требало одбацити. У нашем случају добијемо да су све вредности Z-скора на мање од 2 стандардне девијације од 0 па овим тестом закључимо да је дошло до конвергенције.

```
> geweke.diag(mcmcCoda)
[[1]]

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

beta0 beta[1] beta[2] beta[3] beta[4]    sigma
-0.4208  0.4727  0.7269  0.6445 -0.1561  0.1025


[[2]]

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5
```

[6] Geweke
[7] Gelman и Rubin
[8] Raftery и Lewis
[9] Heidelberger и Welch

```
15
16  beta0 beta[1] beta[2] beta[3] beta[4]    sigma
17  0.5600 -0.3228 -0.4965 -0.2726 -0.3562  0.1525
18
19
20  [[3]]
21
22  Fraction in 1st window = 0.1
23  Fraction in 2nd window = 0.5
24
25  beta0   beta[1]   beta[2]   beta[3]   beta[4]     sigma
26  -1.17102  0.02586  0.49191 -0.25188  1.63111  1.37723
```

Гелман и Рубин дијагностика за сваки параметар рачуна потенцијални фактор скалирања[10] заједно са горњом и доњом границама интервала поверења. Приближна конвергенција се достиже када је горња граница близу вредности 1. Интервали поверења су израчунати под претпоставком нормалности променљиве коју испитујемо. Додатно, уколико та претпоставка није испуњена, функција може да трансформише променљиву користећи логаритамску или logit трансформацију што се постиже додавањем аргумента transform=TRUE у позив функције. Тест је заснован на поређењу дисперзије унутар ланаца са дисперзијом међу ланцима. И овај тест потврђује конвергенцију.

```
1  > gelman.diag(mcmcCoda, transform=TRUE)
2  Potential scale reduction factors:
3
4  Point est. Upper C.I.
5  beta0           1          1
6  beta[1]         1          1
7  beta[2]         1          1
8  beta[3]         1          1
9  beta[4]         1          1
10 sigma           1          1
11
12 Multivariate psrf
13
14 1
```

Рафтери и Луис дијагностика се користи за пробна покретања ланаца. За сваки параметар из сваког ланца рачуна се најмањи број итерација потребан за оцењивање квантила q са тачношћу $+/-r$, са

[10] Potential scale reduction factor

вероватноћом s. Минимални број итерацијаја се односи на ланац у коме нема корелација међу узастопним вредностима. Позитивна корелација увећава захтевану величину узорка. Функција враћа вредност фактора зависности[11], I, где је $I = (M + N)/N_{min}$ а са M, N и N_{min} су означени захтевани период загревања, захтевани обим узорка потребан за оцењивање квантила q, и захтевани обим узорка потребан за оцењивање истог квантила под претпоставком да је узорак iid. Велике вредности су показатељи јаке аутокорелисаности. Према ауторима овог пакета за велике вредности сматрамо вредности изнад 5.

```
raftery.diag(mcmcCoda)
[[1]]

Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

Burn-in  Total Lower bound  Dependence
(M)      (N)   (Nmin)       factor (I)
beta0    3         4447  3746         1.190
beta[1]  5         5771  3746         1.540
beta[2]  4         5211  3746         1.390
beta[3]  3         4129  3746         1.100
beta[4]  2         3680  3746         0.982
sigma    5         5577  3746         1.490


[[2]]

Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

Burn-in  Total Lower bound  Dependence
(M)      (N)   (Nmin)       factor (I)
beta0    3         4338  3746         1.16
beta[1]  8         10112 3746         2.70
beta[2]  4         4792  3746         1.28
beta[3]  3         4267  3746         1.14
beta[4]  2         3995  3746         1.07
sigma    5         6078  3746         1.62

```

[11] Dependence factor

```
33
34  [[3]]
35
36  Quantile (q) = 0.025
37  Accuracy (r) = +/- 0.005
38  Probability (s) = 0.95
39
40  Burn-in  Total Lower bound  Dependence
41  (M)      (N)   (Nmin)       factor (I)
42  beta0    3        4198  3746          1.12
43  beta[1]  5        5673  3746          1.51
44  beta[2]  4        4955  3746          1.32
45  beta[3]  3        4198  3746          1.12
46  beta[4]  2        3995  3746          1.07
47  sigma    5        5483  3746          1.46
```

Хајделбергер и Велч (heidel.diag) дијагностике се заснивају на два теста. Тест конвергенције тестира стационарност расподеле. Најпре је примењен на цео узорак а затим уколико стационарност није прихваћена узорак се смањује одбацивањем 10%, 20%,... оригиналног ланца, све док хипотеза о стационарности не буде прихваћена или одбацимо 50% ланца. У том случају је неопходна већа дужина ланца. Уколико се тест прође након одбацивања одређеног процента узорка, тај део узорка се неће користити за даља закључивања. Након тога, на делу узорка који је прошао тест се рачуна 95% интервал поверења за средњу вредност, и уколико је количник половине дужине интервала и оцењене средње вредности мањи од унапред задате вредности, узорак пролази тест. Иначе се сматра да дата дужина ланца није довољна за оцењивање средње вредности са задатом прецизношћу. Узорци су прошли и овај тест.

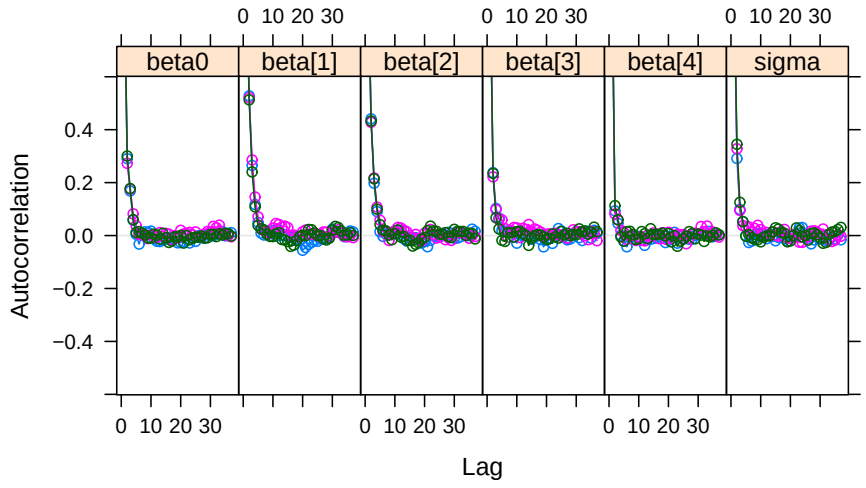
```
1   > heidel.diag(mcmcCoda)
2   [[1]]
3
4   Stationarity start     p-value
5   test         iteration
6   beta0    passed       1         0.987
7   beta[1]  passed       1         0.840
8   beta[2]  passed       1         0.555
9   beta[3]  passed       1         0.919
10  beta[4]  passed       1         0.970
11  sigma    passed       1         0.957
12
13  Halfwidth Mean   Halfwidth
```

```
14  test
15  beta0    passed     61.921  0.37905
16  beta[1]  passed     -0.152  0.00307
17  beta[2]  passed     -0.975  0.00680
18  beta[3]  passed      0.124  0.00115
19  beta[4]  passed      1.080  0.01147
20  sigma    passed      7.312  0.02955
21
22  [[2]]
23
24  Stationarity start     p-value
25  test         iteration
26  beta0    passed       1         0.408
27  beta[1]  passed       1         0.258
28  beta[2]  passed       1         0.279
29  beta[3]  passed       1         0.324
30  beta[4]  passed       1         0.547
31  sigma    passed       1         0.139
32
33  Halfwidth Mean   Halfwidth
34  test
35  beta0    passed     62.008  0.40347
36  beta[1]  passed     -0.154  0.00338
37  beta[2]  passed     -0.977  0.00674
38  beta[3]  passed      0.124  0.00129
39  beta[4]  passed      1.081  0.01261
40  sigma    passed      7.320  0.03288
41
42  [[3]]
43
44  Stationarity start     p-value
45  test         iteration
46  beta0    passed       1         0.350
47  beta[1]  passed       1         0.752
48  beta[2]  passed       1         0.527
49  beta[3]  passed       1         0.580
50  beta[4]  passed       1         0.130
51  sigma    passed       1         0.278
52
53  Halfwidth Mean   Halfwidth
54  test
55  beta0    passed     62.228  0.40267
56  beta[1]  passed     -0.155  0.00328
```

```
beta[2]  passed     -0.985  0.00690
beta[3]  passed      0.124  0.00104
beta[4]  passed      1.077  0.01232
sigma    passed      7.301  0.03290
```

Можемо приказати вредности аутокорелационе функције. Високе вредности су показатељи спорије конвергенције. У таквим ситуацијама пожељно је истањити ланац пре рачунања сумарних статистика јер ће тиме бити сачувана већина информација а уједно и меморија рачунара. У овом примеру вредности аутокорелационе функције брзо опадају.

```
> acfplot(mcmcCoda)
```



На основу резултата спроведених тестова закључујемо да су у овом случају ланци исконвергирали ка апостериорној расподели.
Аналогно наведеном поступку биће креирана још два модела мењајући почетне претпоставке. Након што за њих потврдимо конвергенцију можемо упоредити резултате добијене применом свих метода.
Један од креираних модела је модел који за функцију веродостојности уместо нормалне има Студентову t-расподелу. Број степени слободе, ν,

није унапред задат, већ је за његову априорну расподелу изабрана експоненцијална расподела са очекивањем 30, док су априорне расподеле за вектор β и σ непромењене. Највећа промена је у помоћном коду, где је модел сада дефинисан на другачији начин:

```
model {
for ( i in 1:Ntotal ) {
zy[i] ~ dt( zbeta0 + sum( zbeta[1:Nx] * zx[i,1:Nx] ) , 1/
 zsigma^2, nu)
}
zbeta0 ~ dnorm( 0 , 1/2^2 )
for ( j in 1:Nx ) {
zbeta[j] ~ dnorm( 0 , 1/2^2 )
}
lnzsigma  ~ dunif( -3, 3 )
zsigma <-exp(lnzsigma)
nu ~ dexp(1/30)
```

За исте периоде адаптације и загревања као у претходном случају овде не долази до конвергенције. Судећи по Гевеке тесту, није дошло до конвергенције за параметре β_3, β_4 и σ у 3. ланцу и за параметар β_3 у другом ланцу. Можемо одбацити почетне делове ланца или изнова креирати ланац са дужим периодом загревања како бисмо елиминисали утицај почетних вредности.

```
[[1]]

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

beta0  beta[1]  beta[2]  beta[3]  beta[4]     sigma
 nu
-1.42608  0.99387  1.28360 -0.18023  0.96150  0.01448
 -1.33691


[[2]]

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

beta0 beta[1] beta[2] beta[3] beta[4]    sigma       nu
-0.4934  0.9672 -0.1275 -2.1484  0.4961 -1.4057  0.5186

```

```
18
19  [[3]]
20
21  Fraction in 1st window = 0.1
22  Fraction in 2nd window = 0.5
23
24  beta0 beta[1] beta[2] beta[3] beta[4]    sigma       nu
25  1.5771 -0.4094 -0.3852  2.4775 -2.6062  2.4085  1.7576
```

Пре измена анализирамо резултате преосталих дијагностика како бисмо решили више потенцијалних проблема истовремено. У наставку је приказана Рафтери и Луис дијагностика за нпр. трећи ланац. Иако вредности фактора зависности нису веће од 5, приметан је утицај високе аутокорелисаности, што ћемо покушати да решимо поновним креирањем узорка али са параметром истањивања који је већи од 1.

```
1   [[3]]
2
3   Quantile (q) = 0.025
4   Accuracy (r) = +/- 0.005
5   Probability (s) = 0.95
6
7   Burn-in  Total Lower bound  Dependence
8   (M)      (N)   (Nmin)       factor (I)
9   beta0    6         6406  3746          1.71
10  beta[1]  9         10098 3746          2.70
11  beta[2]  10        12122 3746          3.24
12  beta[3]  6         6406  3746          1.71
13  beta[4]  6         7003  3746          1.87
14  sigma    7         7263  3746          1.94
15  nu       5         5483  3746          1.46
```

Ови ланци су прошли Гелман и Рубин, као и Хајделбергер и Велч тестове. Након креирања ланца са два пута дужим периодом загревања и са параметром истањивања који је једнак 2, аутокорелисаност узорака је смањена, и сви ланци су прошли Гевеке тест. Надаље користимо овако добијене ланце.

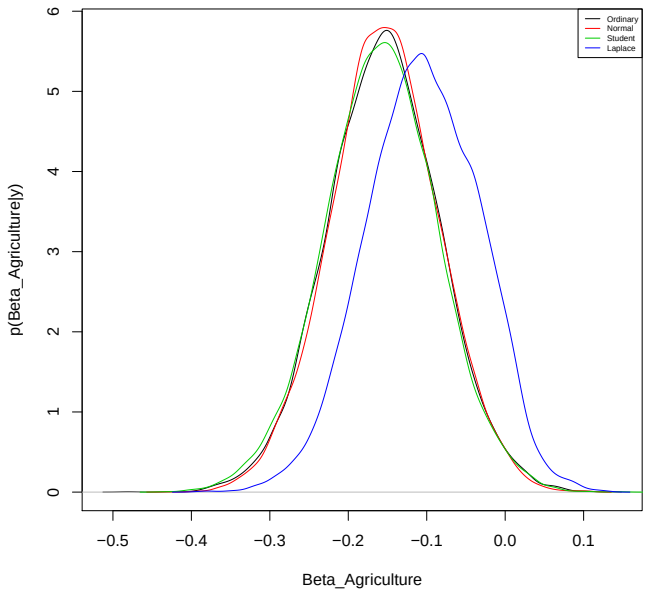
Биће представљен још један модел који се у односу на први креирани разликује у избору априорне расподеле кофицијената који одговарају стандардизованим зависним променљивим. Уместо нормалне расподеле са фиксираним параметрима за условну априорну расподелу је изабрана Лапласова односно дупла експоненцијална расподела са очекивањем 0 и rate параметром који је једнак $\lambda/z\sigma$. Овај модел је такав да мода

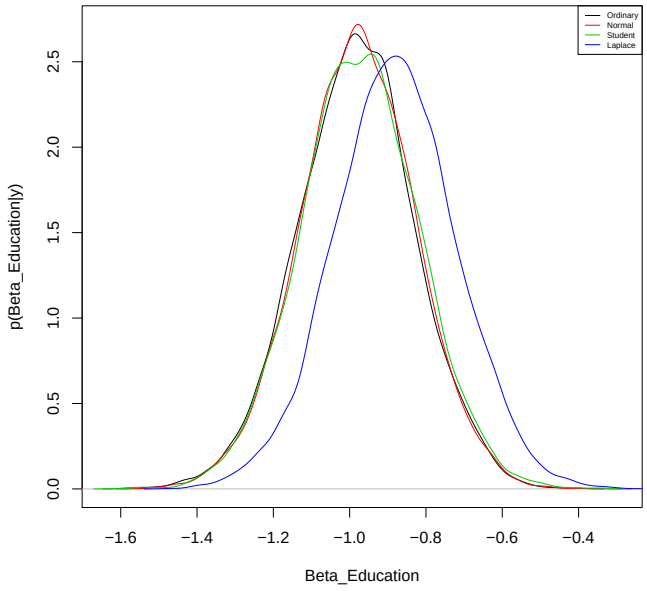
апостериорне расподела β коефицијената одговара резултатима који се добију LASSO регресијом и самим тим нам може бити користан за избор променљивих јер кажњава велике вредности β коефицијената.
За априорну расподелу параметра λ изабрана је униформна расподела на ограниченом интервалу, $U[0.01, 10]$. У наставку је приказано како изгледа део помоћне функције у коме се дефинише модел.

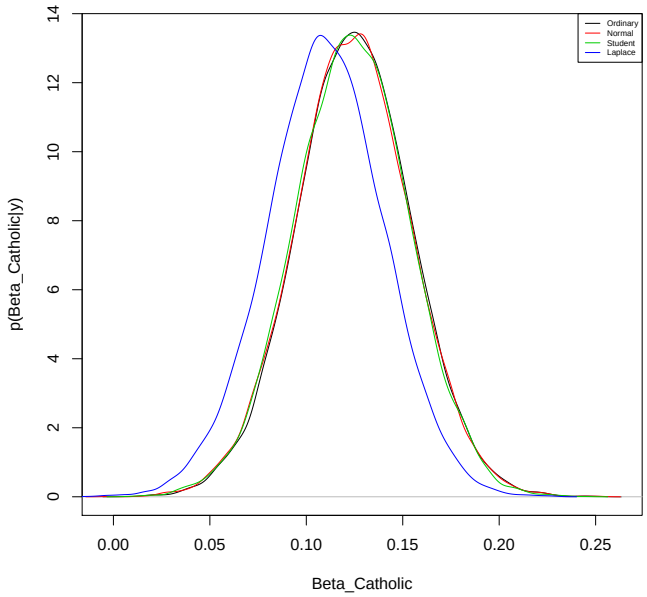
```
model {
for ( i in 1:Ntotal ) {
zy[i] ~ dnorm( zbeta0 + sum( zbeta[1:Nx] * zx[i,1:Nx] ) ,
  1/zsigma^2 )
}
zbeta0 ~ dnorm( 0 , 1/2^2 )
for ( j in 1:Nx ) {
zbeta[j] ~ ddexp( 0 , lambda/zsigma )
}
lnzsigma  ~ dunif( -3, 3 )
zsigma <-exp(lnzsigma)
lambda ~ dunif( 0.01, 10)
```

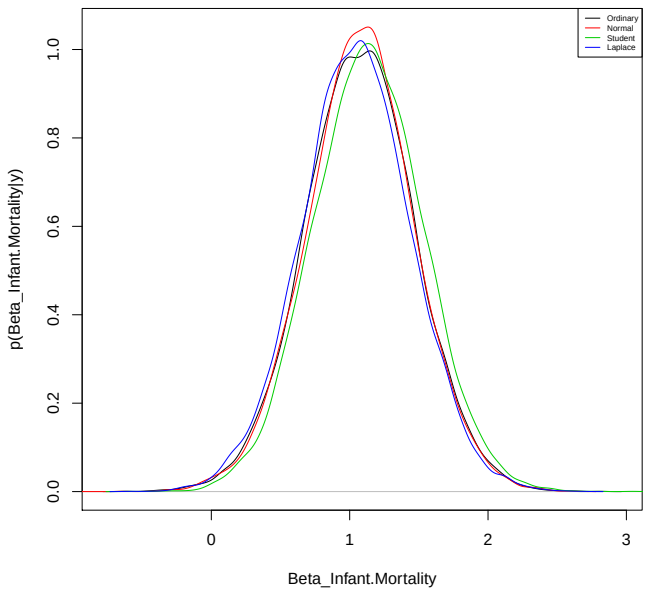
Резултати добијени овим моделом су прошли све тестове конвергенције.
У наставку су приказани графици оцењених густина апостериорних расподела непознатих параметара. На сваком графику представљен је један параметар и оцењене густине добијене помоћу претходних модела. Црном бојом је означен модел обичне линеарне регресије са неправом униформном априорном расподелом, црвеном модел где су за априорну расподелу изабране нормална и логарирам од униформне, зеленом модел са Студентовом функцијом веродостојности и плавом бојом модел са Лапласовом априорном расподелом параметра β.

Коефицијенти који одговарају предикторима:

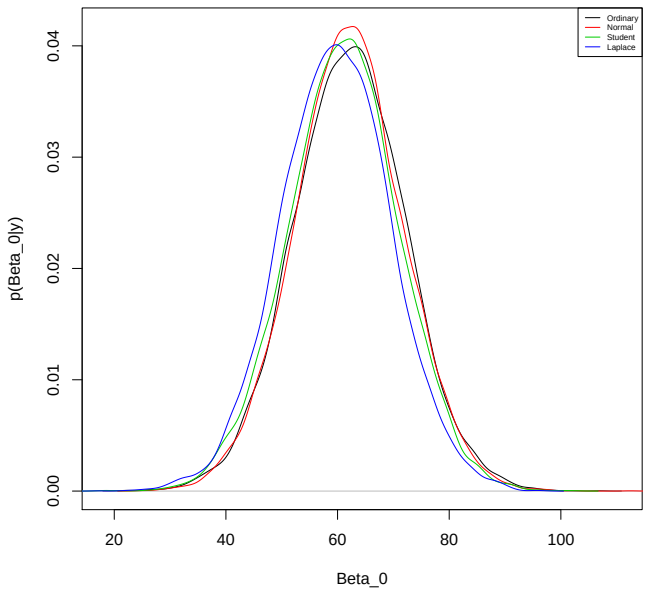




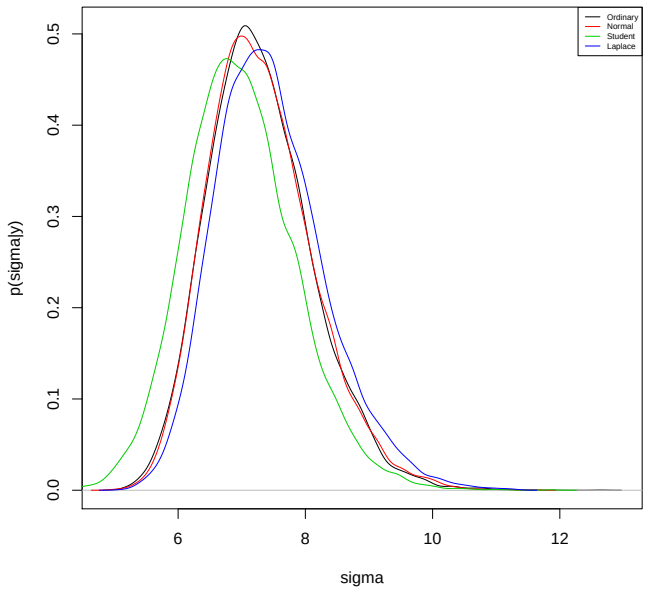

Ordinary
Normal
Student
Laplace
p(Beta_Catholic|y)
0
2
4
6
8
10
12
14
0.00
0.05
0.10
0.15
0.20
0.25
Beta_Catholic


Ordinary
Normal
Student
Laplace
p(Beta_Infant.Mortality|y)
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
Beta_Infant.Mortality

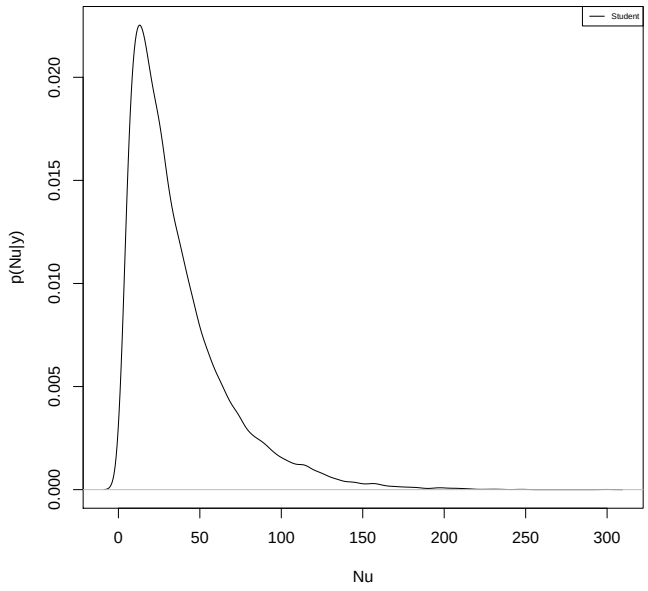
Слободни члан:



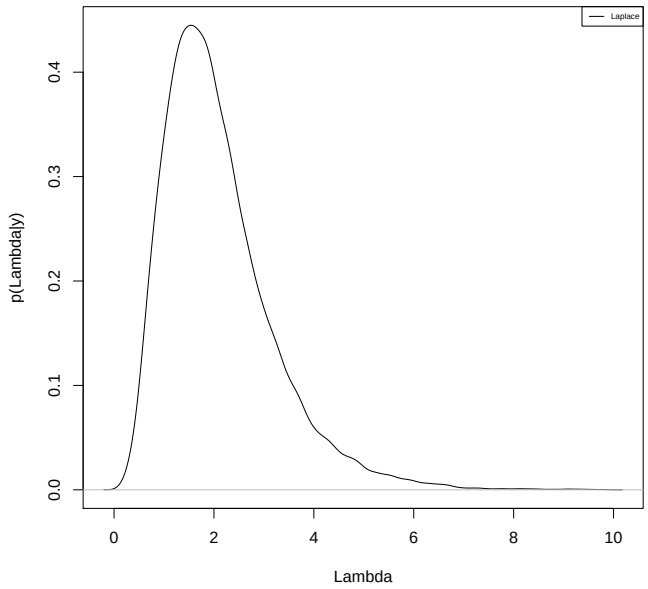
Стандардна девијација:



Помоћни параметри:
ν - број степени слободе из другог модела



λ - параметар из априорне расподеле параметра β из трећег модела.



Посматрањем графика долазимо до закључка да значајније разлике постоје у случају када је за априорну расподелу изабрана Лапласова и да су у том случају расподеле померене ка 0. Такав резултат се могао и очекивати с обзиром на то да је овај модел бајесовски аналогон LASSO регресији.
У случају стандардне девијације приметна је разлика код модела са Студентовом функцијом веродостојности.
За креиране моделе упоређена су средњеквадратна одступнања предвиђених од стварних вредности на скупу који се користио за развој модела. У случају обичне линеарне регресије, искоришћен је претходно генерисани узорак из апостериорне предиктивне расподеле. За оцену предвиђене вредности изабрана је средња вредност узорка и помоћу ње се рачунамо средњеквадратно одступање.

```
pred.mean = apply(pred.draws, 2, mean)
mse <- mean((actual - pred.mean) ^ 2)
> mse
[1] 46.08068389
```

Узорке који су добијени коришћењем Гибсовог алгоритма и сачувани у rjags објектима најпре трансформишемо у матрични облик, а затим генеришемо узорак из апостериорне предиктивне расподеле. Поступак је сличан за сва три модела, овде је илустрован случај са Студентовом расподелом.

```
# Ucitavanje sacuvanih podataka i
# transformacija u matricni oblik
# install.packages("metRology")
# Paket potreban za rt.scaled funkciju
library(metRology)

load("Student-Normal-ExpUniform-afterDiagnostics.Rdata")
runjags2 <- runJagsOut
mcmcCoda2 <- as.mcmc.list(runjags2)
mcmcMatrix2 <- as.matrix(mcmcCoda2, iters = FALSE, chains =
    FALSE)

n <- dim(swiss)[1] # br opservacija
y = swiss$Fertility
X = as.matrix(swiss[, -c(1, 3)])

p = 4 # Broj prediktora
N = 15000 # Obim uzorka

```

```
M <- mcmcMatrix2
yrep <- matrix(data = rep(0, n * N),
nrow = n,
ncol = N)
# Za cuvanje uzorka iz aposteriorne prediktivne raspodele

for (i in 1:N) {
sd = M[i, p + 2]
beta0 = M[i, 1]
beta = M[i, 2:(p + 1)]
nu = M[i, p + 3]
for (k in 1:n) {
m = beta0 + beta %*% X[k, ]
yrep[k, i] <- rt.scaled(1, df = nu, mean = m, sd = sd)
}
}
yrep2 = yrep
mse2 = mean((y - rowMeans(yrep)) ^ 2)
```

У случају друга два модела са нормалном функцијом веродостојности, не постоји параметар ν и користимо rnorm функцију за генерисање узорка из нормалне расподеле.
Добијени су следећи резултати:

1. $MSE_1 = 46.08068389$
2. $MSE_2 = 45.93145059$
3. $MSE_3 = 45.61975745$
4. $MSE_4 = 46.71787993$

Разлике између добијених вредности нису велике, али с обзиром на то да модел креиран коришћењем Студентове расподеле најбоље одговара подацима у смислу средњеквадратног одступања, њега бирамо као финални модел.

Поглавље 4

Закључак

У првом поглављу представљени су теоријски резултати који се налазе иза сваке бајесовске анализе података и како се могу применити на линеарне регресионе моделе.
У следећем поглављу укратко су изложени основни резултати добијени класичним приступом линеарној регресији, тј. прецизније, резултати добијени методом најмањих квадрата.
Коначно, последње поглавље је посвећено конкретним примерима бајесовске линеарне регресије.
Први пример је илустрација обичне линеарне регресије, где полазећи од неправе униформне расподеле добијамо апостериорну расподелу непознатих параметара. Осим што је наведен аналитички облик апостериорне расподеле, генерисани су узорци на основу којих доносимо закључке о параметрима. Такође, илустрован је утицај обима узорка на доношење закључака, генерисање узорка из апостериорне предиктивне расподеле, као и начини на које вршимо проверу добијених резултата. Оцене параметара и предвиђених вредности су упоређене са резултатима добијеним методом најмањих квадрата. С обзиром на то да је у овом примеру коришћена униформна априорна расподела, дошло је до поклапања оцена које су добијене помоћу ова два приступа. У овом случају где су подаци заиста генерисани помоћу фиксираних вредности параметара нам таква ситуација одговара и бајесовски линеарни регресиони модели се показују као једнако успешни као и модели добијени класичним приступом. Јасно је да у реалним примерима то неће бити случај тако да се у таквим ситуацијама бајесовска линеарна регресија показује као боље решење. Креирање модела на реалним подацима илустровано је у другом примеру. Креирани су линеарни регресиони модели за различите изборе априорних расподела и функција веродостојности и за тако добијене моделе је упоређено уклапање са подацима. Од суштинског значаја

су се показали МКЛМ методе које омогућавају да за произвољне изборе почетних претпоставки генеришемо узорак из апостериорне расподеле.

Литература

[1] Gelman, Andrew, et al. Bayesian data analysis. Chapman and Hall/CRC, 2013.

[2] Gill, Jeff. Bayesian methods: A social and behavioral sciences approach. Chapman and Hall/CRC, 2014.

[3] Kruschke, John. Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan. Academic Press, 2014.

[4] Albert, Jim. Bayesian Computation with R. Springer Science & Business Media, 2007.

[5] R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/. URL http://www.R-project.org/.

[6] RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/.

[6] Jim Albert (2018). LearnBayes: Functions for Learning Bayesian Inference. URL https://CRAN.R-project.org/package=LearnBayes

[7] http://mcmc-jags.sourceforge.net/

[8] Matthew J. Denwood (2016). runjags: An R Package Providing Interface Utilities, Model Templates, Parallel Computing Methods and Additional Distributions for MCMC Models in JAGS. Journal of Statistical Software, 71(9), 1-25. doi:10.18637/jss.v071.i09

[9] Martyn Plummer, Nicky Best, Kate Cowles and Karen Vines (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC, R News, vol 6, 7-11

[10] https://sites.google.com/site/doingbayesiandataanalysis/software-installation

Биографија

Биљана Јовановић рођена је 11.02.1994. године у Ужицу а детињство је провела у селу Костојевићи. 2008. године завршила је Основну школу "Душан Јерковић" као носилац Вукове дипломе и ђак генерације. Средњошколско образовање је завршила 2012. године у гимназији "Јосиф Панчић" у Бајиној Башти, такође са Вуковом дипломом. Исте године је уписала основне студије на Математичком факултету Универзитета у Београду, смер Статистика, актуарска и финансијска математика. Дипломирала је 2016. године са просечном оценом 9.02 а затим уписала мастер академске студије на истом смеру.
Радно искуство је започела на летњој пракси у Народној Банци Србије у одељењу за информационо-комуникационе технологије, сектор за развој аналитичких апликација, у периоду од јула до септембра 2016. У периоду од октобра до децембра 2016. године радила је као практикант у ревизорској компанији PricewaterhouseCoopers, у одељењу IT-ревизије. У периоду од маја до јула 2017. радила је на позицији статистичар и аналитичар података у Joker Games. Од септембра 2017. до априла 2018. године радила је у компанији Дунав Осигурање, на позицији консултанта за актуарство и извештавање. Од априла 2018. до данас ради у Addiko банци на позицији аналитичара за моделирање кредитног ризика. У слободно време воли да путује, упознаје нове људе, борави у природи, вози бицикл, дружи се са пријатељима.