 Vilniaus Universitetas

Regresinės analizės

projektinis darbas

Darbą atliko:

Vainius Gataveckas, Matas Gaulia, Dovydas Martinkus

Duomenų Mokslas

3 kursas 2 gr.

Vilnius, 2022

**Turinys**

[Naudoti metodai 3](#_Toc104494481)

[Duomenys ir jų šaltiniai 4](#_Toc104494482)

[Tikslas ir uždaviniai 6](#_Toc104494483)

[Atliktos analizės aprašymas 7](#_Toc104494484)

[Pradinis duomenų apdorojimas ir analizė 7](#_Toc104494485)

[Išvados 51](#_Toc104494486)

[1 Priedas. Tiesinio modelio migracijos prieaugiui diagnostiniai grafikai. 52](#_Toc104494487)

[2 Priedas. Tiesinio modelio natūraliam prieaugiui diagnostiniai grafikai. 55](#_Toc104494488)

# Naudoti metodai

Šiame darbe naudota tiesinė ir kvantilių regresijos. Taip pat naudoti apibendrintieji tiesiniai modeliai su glodniaisiais splainais, multinominis logistinės regresijos modelis. Darbas atliktas naudojant R.

Naudoti R paketai:

*tidyverse*

*rsample*

*corrplot*

*car*

*effects*

*lm.beta*

*quantreg*

*mgcv*

*gratia*

*effect*

*yardstick*

*mnet*

*themis*

*recipes*

*broom*

*ggrepel*

# Duomenys ir jų šaltiniai

Gyventojų skaičiaus prieaugio prognozavimas: atskirai tiriamas natūralus procentinis pokytis ir procentinis pokytis dėl migracijos.

Duomenų aibės sudarymui panaudoti duomenys iš skirtingų šaltinių. Duomenų šaltiniai:

Our World in Data. Natūralaus ir bendro gyventojų prieaugio šalims duomenys. Prieiga per internetą: <https://ourworldindata.org/world-population-growth>

UNData. Įvairūs ekonominiai, socialiniai, su aplinkosauga ir infrastruktūra susiję šalių indikatoriai. Prieiga per internetą[: https://www.kaggle.com/datasets/sudalairajkumar/undata-country-profiles/code](:%20https:/www.kaggle.com/datasets/sudalairajkumar/undata-country-profiles/code)

World Happiness Report. Apklausomis paremti gyvenimo kokybės šalyse įvertinimų pagal skirtingus kriterijus duomenys. Prieiga per internetą: <https://www.kaggle.com/datasets/mathurinache/world-happiness-report-20152021?select=2017.csv>

Atlikus pirminį apdorojimą duomenų aibę sudaro šie požymiai:

“employment\_industry\_percent\_of\_employed" – dalis dirbančių industriniame sektoriuje.

"unemployment\_percent\_of\_labour\_force" - bedarbių dalis visoje darbo rinkoje.

"agricultural\_production\_index\_2004\_2006\_100" – šalyje pagamintų agrikultūros produktų inkesas pasvertas pagal kainas.

"urban\_population\_percent\_of\_total\_population" - miestuose gyvenanti gyventojų dalis.

"health\_total\_expenditure\_percent\_of\_gdp" – dalis BVP išleidžiama sveikatos apsaugai.

"education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop" - gyventojų, lankančių pradinį mokymąsį skaičius 100 gyventojų.

"education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop" – - gyventojų, lankančių aukštajį (ar kitą trečio lygio) mokymąsį skaičius 100 gyventojų.

"pop\_using\_improved\_drinking\_water\_urban" - miesto gyventojų, naudojančių geros kokybės geriamą vandenį skaičius 100 gyventojų.

"freedom" – asmeninės laisvės įvertinimas.

"generosity" – dosnumo įvertinimas (labdara, savanoriavimas ir t.t.).

"trust\_government\_corruption" - pasitikėjimo vyriausybe ir korupcijos lygio įvertinimas.

"migration\_growth" – procentinis gyventojų skaičiaus prieaugis dėl migracijos.

"natural\_growth” – procentinis natūralus gyventojų skaičiaus prieaugis (vien tik dėl mirčių ir gimimų šalyje).

"category" – vėliau sudarytas kategorinis kintamasis priskiriantis šalims klases pagal migracijos ir natūralaus gyventojų skaičiaus prieaugio reikšmes.

# Tikslas ir uždaviniai

Tikslas: Atlikti regresinę analizę natūraliam ir migracijos gyventojų prieaugiui pagal ekonominius ir socialinius šalių indikatorius.

Uždaviniai:

Požymių aibės sudarymas.

Tiesinių regresijos modelių gyventojų prieaugiui sudarymas.

Kvantilių regresijos ir apibendrintųjų adityviųjų modelių su glodniaisiais splainais gyventojų prieaugiui sudarymas.

Multinominės logistinės regresijos modelio sudarymas šalių klasifikavimui į pagal gyventojų prieaugį sudarytas klases.

Modelių tinkamumo analizė.

Kovariančių įtakos gyventojų prieaugiui įvertinimas.

Modelių panaudojimas prognozėms gauti.

# Atliktos analizės aprašymas

Pradinis duomenų apdorojimas ir analizė

Tyrimui naudoti 2017 metų duomenys. Visi naudoti duomenų rinkiniai sujungti į vieną jungiant pagal šalies pavadinimą.

Iš duomenų aibės pašalinti su migracija, gimstamumu susiję požymiai, nes su jais tiesiogiai susijusios siekiamos prognozuoti reikšmės. Taip pat pašalinti su aplinkosauga, energijos sritimi susiję požymiai, nes jie laikyti ne tokie svarbūs išsikeltam tyrimo tikslui. Papildomai atsisakyta požymių su daug praleistų reikšmių, mažais požymio reikšmių skirtumais tarp šalių.

Sudaryta duomenų aibė padalinta į apmokymo ir testavimo aibes naudojant 85-15 santykį. Iš duomenų iš anksto pašalintos stipriai tarpusavyje koreliuojančios kovariantės naudojant 0.7 Pirsono koreliacijos ribą. Kovariantės, likusios po šio ir prieš tai aprašyto požymių filtravimo, aprašytos praėjusiame skyriuje.

Sudarytas kategorinis kintamasis šalims priskiriantis klases pagal jų natūralaus ir migracijos prieaugio reikšmes (klasė 0 – teigiamas migracijos prieaugis ir teigiamas natūralus, 1 – teigiamas migracijos ir neigiamas natūralus, 2 – neigiamas migracijos ir teigiamas natūralus, 3 – neigiamas migracijos ir neigiamas natūralus).

library(tidyverse)  
library(janitor)  
library(countrycode)  
  
# Natūralaus gyventojų prieaugio duomenys  
pop\_natural <- read\_csv("natural-population-growth.csv") %>%  
 filter(Year == 2017) %>%  
 dplyr::select(1, 4) %>%  
 set\_names(c("country", "natural\_growth")) %>%  
 mutate(country = countryname(country))  
  
# Bendras gyventojų prieaugis iš kurio bus gaunamas migracijos prieaugis  
pop\_total <- read\_csv("population-growth-rates.csv") %>%  
 filter(Year == 2017) %>%  
 dplyr::select(1, 4) %>%  
 set\_names(c("country", "total\_growth")) %>%  
 mutate(country = countryname(country))  
  
  
# UNData duomenys  
country\_stats <- read\_csv("country\_profile\_variables.csv") %>%  
 clean\_names() %>%  
 dplyr::select(-c(2, 3, 4, 5, 6, 7)) %>%  
 mutate(country = countryname(country))  
  
  
# World Happiness Report duomenys  
happiness <- read\_csv("2017.csv") %>%  
 clean\_names() %>%  
 dplyr::select(-c(2), -starts\_with("whisker"), -c("dystopia\_residual", "happiness\_score", "family")) %>%  
 mutate(country = countryname(country))

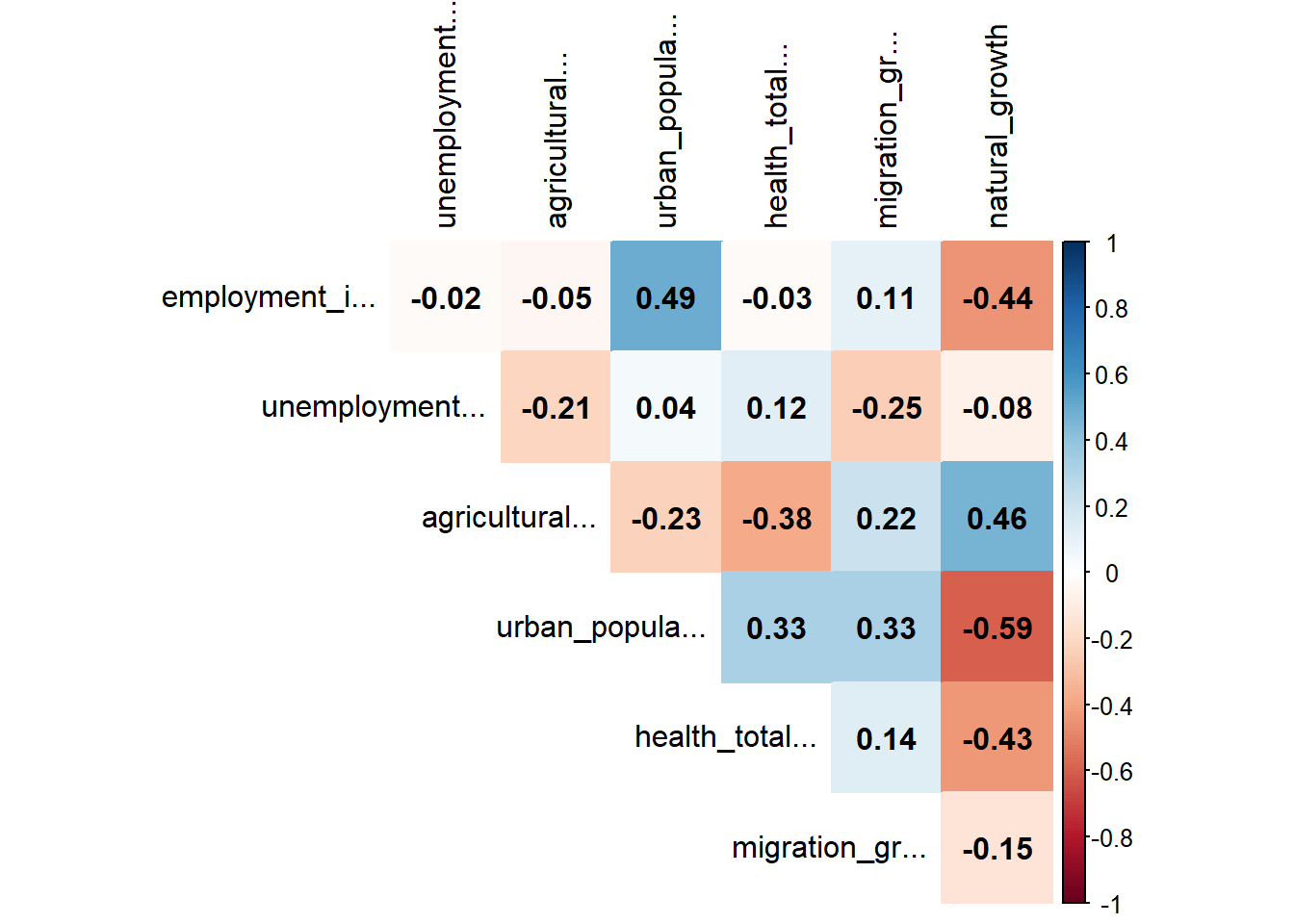
x <- reduce(list(pop\_natural, pop\_total, country\_stats, happiness), inner\_join, by = "country")  
  
  
# Išfiltruojami nenaudinti kintamieji  
x <- x %>%  
 dplyr::select(  
 -starts\_with("gdp"),  
 -starts\_with("labour"),  
 -starts\_with("international"),  
 -starts\_with("balance"),  
 -starts\_with("population"),  
 -starts\_with("fertility"),  
 -starts\_with("net"),  
 -starts\_with("energy\_prod"),  
 -starts\_with("forest"),  
 -starts\_with("threatened"),  
 -starts\_with("seats"),  
 -starts\_with("urban\_population\_growth"),  
 -starts\_with("refugees"),  
 -starts\_with("infant"),  
 -starts\_with("life\_expectancy"),  
 -starts\_with("co2"),  
 -starts\_with("economy"),  
 -starts\_with("education\_government"),  
 -starts\_with("energy"),  
 -health\_physicians\_per\_1000\_pop,  
 -individuals\_using\_the\_internet\_per\_100\_inhabitants,  
 -mobile\_cellular\_subscriptions\_per\_100\_inhabitants\_40,  
 -pop\_using\_improved\_sanitation\_facilities\_urban\_rural\_percent  
 ) %>%  
 mutate(across(everything(), ~ replace(., . %in% c("...", "-99", ".../..."), NA))) %>%  
 mutate(across(starts\_with("education"), ~ str\_split(., "/") %>% map(~ mean(as.numeric(.)))))  
  
  
pop <- x$pop\_using\_improved\_drinking\_water\_urban\_rural\_percent  
f1 <- possibly(~ `[[`(.x, 1), 1)  
x$pop\_using\_improved\_drinking\_water\_urban <- pop %>%  
 str\_split("/") %>%  
 map(f1)  
f2 <- possibly(~ `[[`(.x, 2), 1)  
x$pop\_using\_improved\_drinking\_water\_rural <- pop %>%  
 str\_split("/") %>%  
 map(f2)  
  
  
x <- x %>%  
 dplyr::select(-pop\_using\_improved\_drinking\_water\_urban\_rural\_percent) %>%  
 mutate(across(-country, as.numeric)) %>%  
 mutate(migration\_growth = total\_growth - natural\_growth) %>%  
 drop\_na() %>%  
 dplyr::select(-total\_growth)

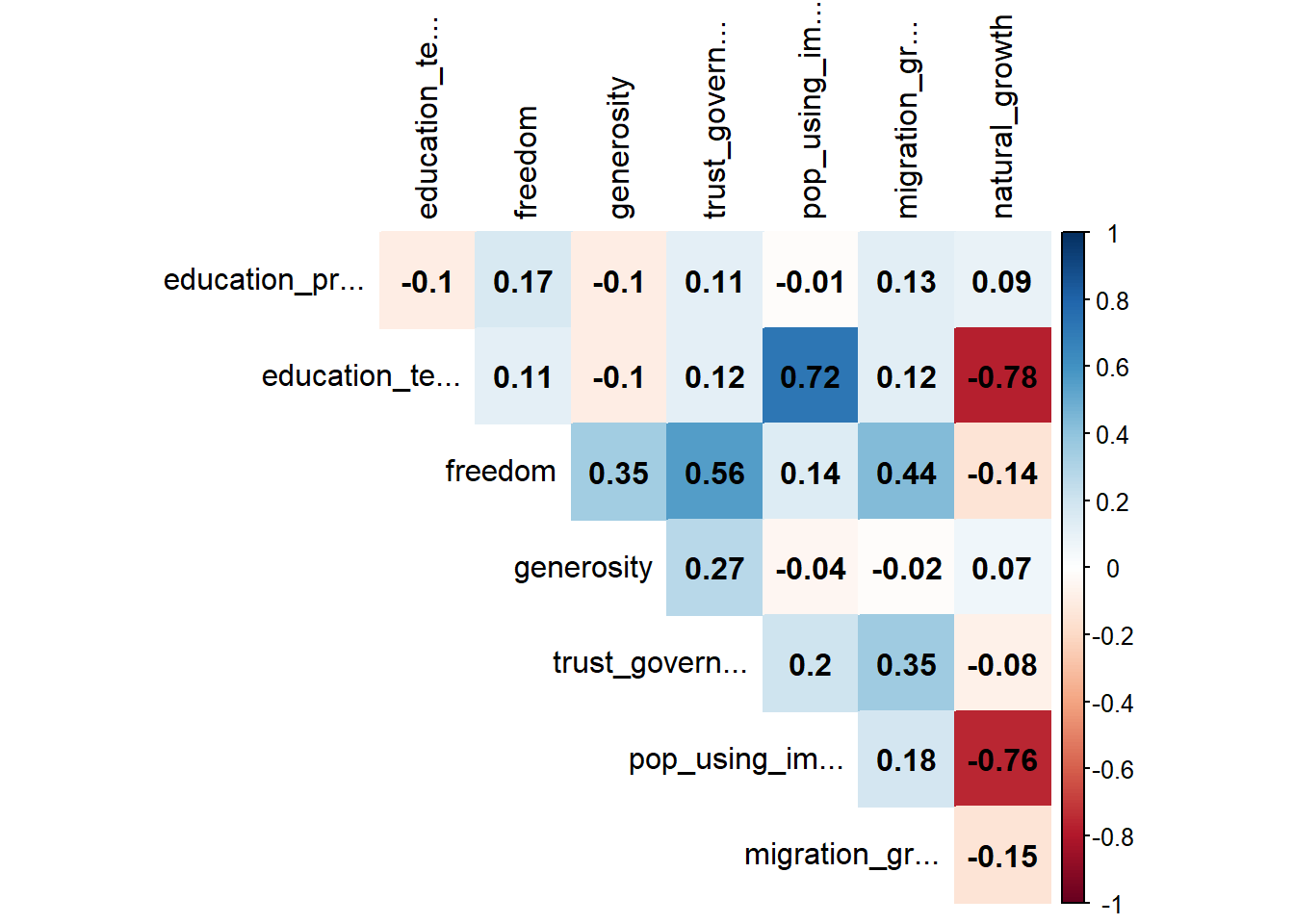
library(rsample)  
  
set.seed(123)  
  
# sudaramos kategorijos pagal tai ar migracijos/natūralus prieaugiai yra teigiami ar neigiami  
x <- x %>% mutate(  
 category = factor(case\_when(  
 migration\_growth >= 0 & natural\_growth >= 0 ~ 0, # "P migation, P natural",  
 migration\_growth >= 0 & natural\_growth < 0 ~ 1, # "P migation, N natural",  
 migration\_growth < 0 & natural\_growth >= 0 ~ 2, # "N migation, P natural",  
 TRUE ~ 3 # "N migration, N natural"  
 ))  
)  
  
  
# padalijimas į mokymo ir testavimo aibes  
train\_test\_split <- initial\_split(x, prop = 0.8)  
train <- training(train\_test\_split)  
test <- testing(train\_test\_split)  
  
country\_train <- train$country  
country\_test <- test$country  
  
train <- train %>% dplyr::select(-country)

library(recipes)  
  
# iš anksto panaikinami kintamieji, kurie labai stipriai koreliuoja su kitais  
correlated\_recipe <- recipe(natural\_growth ~ ., data = train) %>%  
 add\_role(migration\_growth, new\_role = "outcome") %>%  
 add\_role(category, new\_role = "outcome") %>%  
 step\_corr(all\_numeric\_predictors(), threshold = 0.8) %>%  
 step\_nzv(all\_numeric\_predictors())  
  
  
correlated\_recipe <- prep(correlated\_recipe, training = train)  
  
train <- bake(correlated\_recipe, NULL)

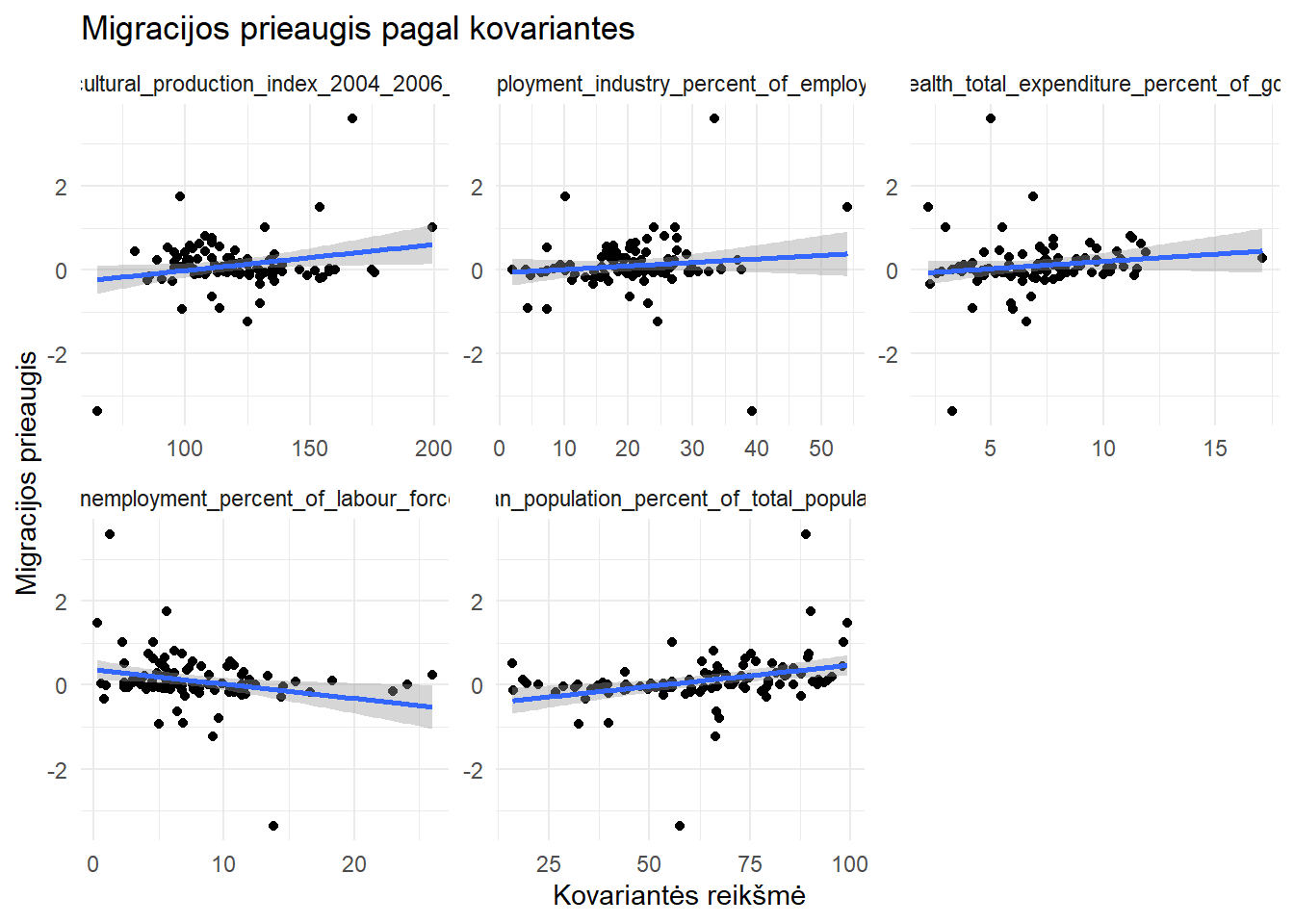
Nubraižytos koreliacijų diagramos. Pastebėta, kad didelė dalis požymių kurie teigiamai koreliuoja su migracijos gyventojų prieaugiu neigiamai koreliuoja su natūraliu prieaugiu (ir atvirkščiai). Taip pat atskirai natūraliam ir migracijos prieaugiams nubraižytos sklaidos diagramos su regresijos kreive pagal naudojamas kovariantes. Ryšiai tarp atsako ir kovariančių dažnai tiesiniai. Tiesa, braižant prieš tai minėtas regresijos kreives nėra atsižvelgiama į kitų kovariančių reikšmes todėl šie grafikai gali tik sufleruoti apie šių kovariančių įtaką pilname modelyje. Pastebėtas dvi labai stiprios išskirtys pagal migracijos prieaugį (Sirija ir Bahreinas). Pasirinkta prieš konstruojant modelius migracijos prieaugiui šias reikšmes iš anksto pašalinti iš duomenų aibės.

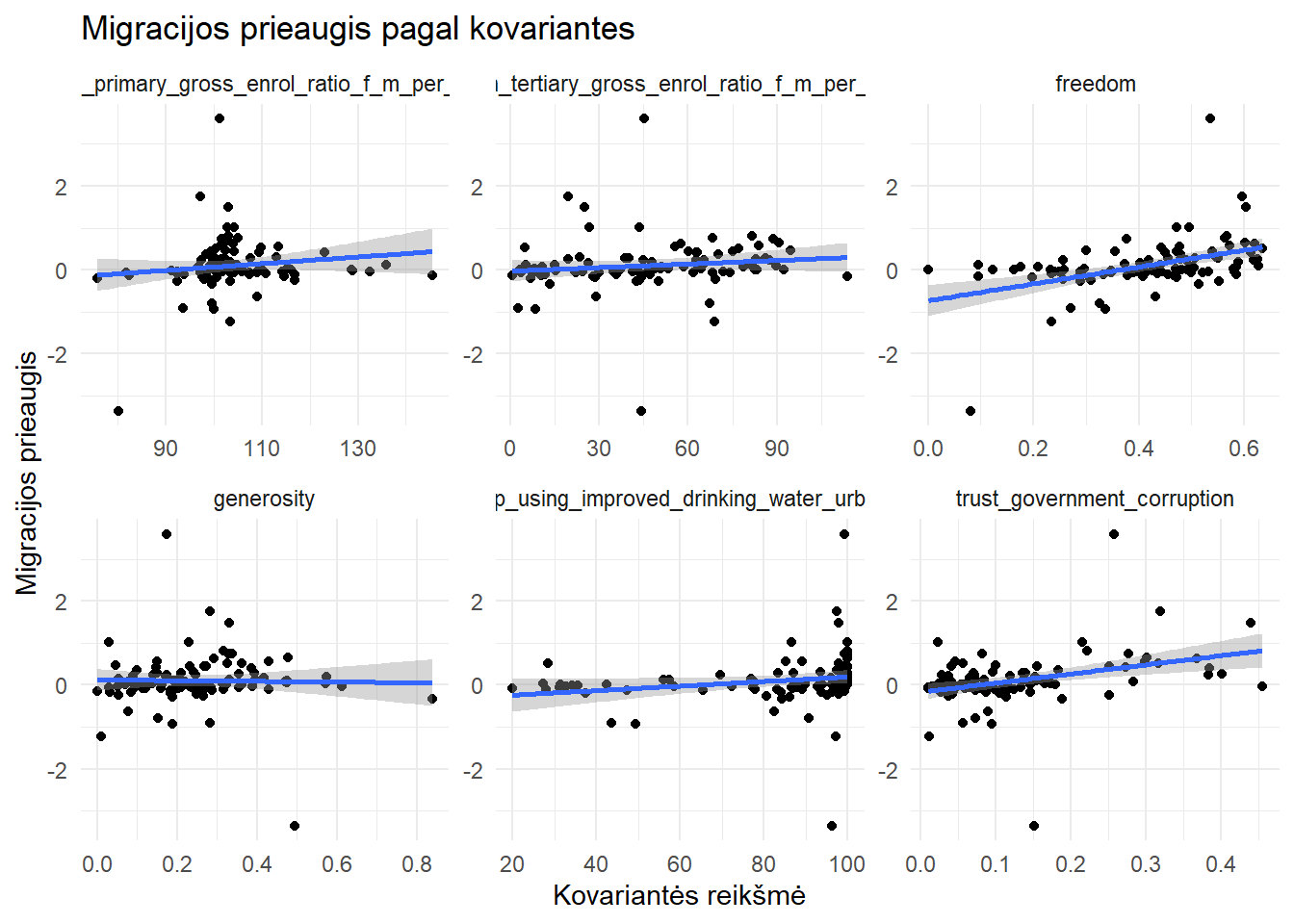
library(corrplot)  
  
# koreliacijų grafikai  
regression\_train <- train %>% dplyr::select(-category)  
  
correlation <- function(name, name2) {  
 correlation\_matrix <- regression\_train %>%  
 dplyr::select(1:5, {{ name }}, {{ name2 }}) %>%  
 set\_names(., str\_trunc(names(.), 15)) %>%  
 cor()  
  
 corrplot(correlation\_matrix, order = "original", method = "color", type = "upper", diag = FALSE, tl.col = "black", addCoef.col = "black")  
  
 correlation\_matrix <- regression\_train %>%  
 dplyr::select(6:length(regression\_train), {{ name }}, {{ name2 }}) %>%  
 set\_names(., str\_trunc(names(.), 15)) %>%  
 cor()  
  
 corrplot(correlation\_matrix, order = "original", method = "color", type = "upper", diag = FALSE, tl.col = "black", addCoef.col = "black")  
}  
  
  
correlation(migration\_growth, natural\_growth)



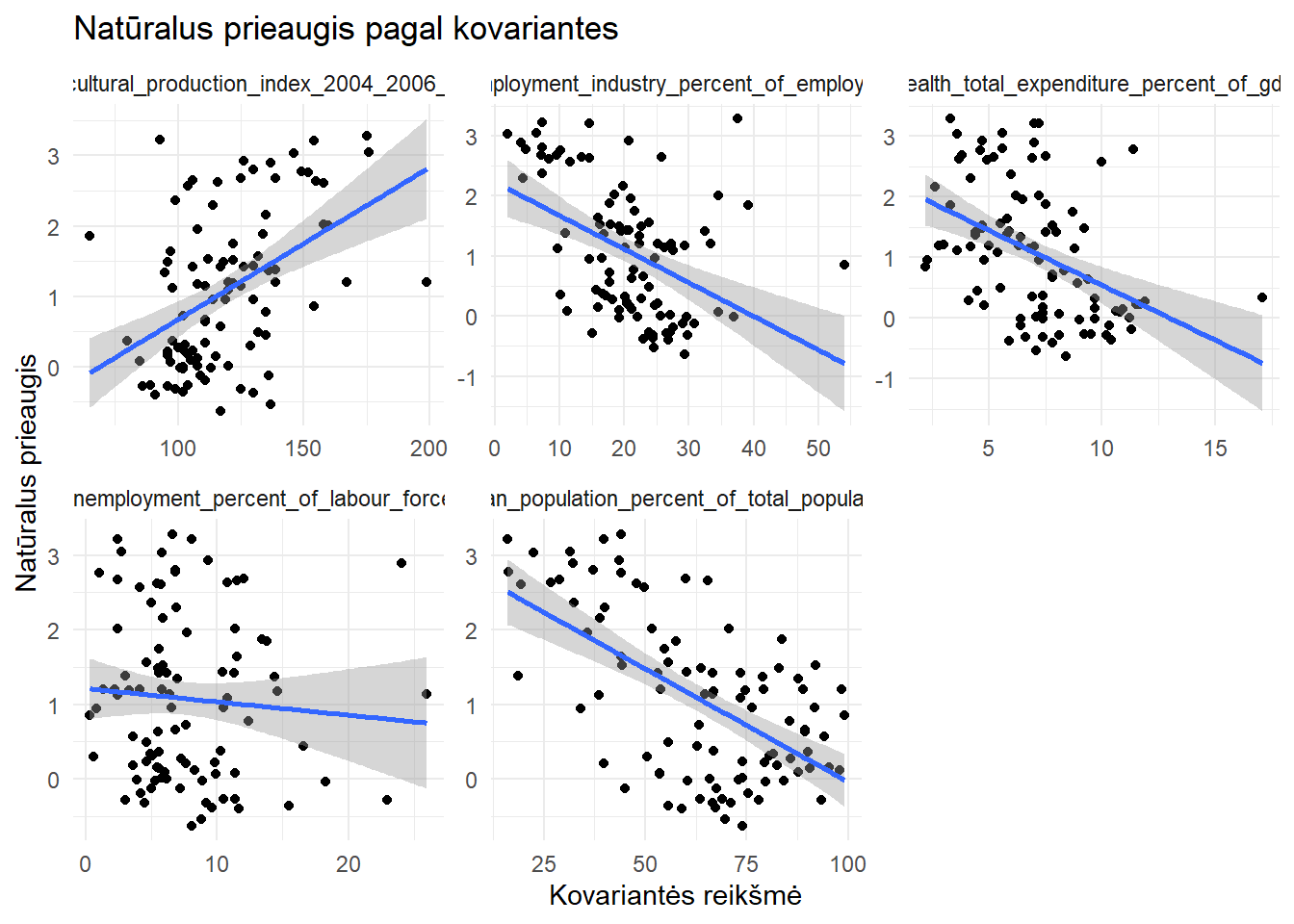


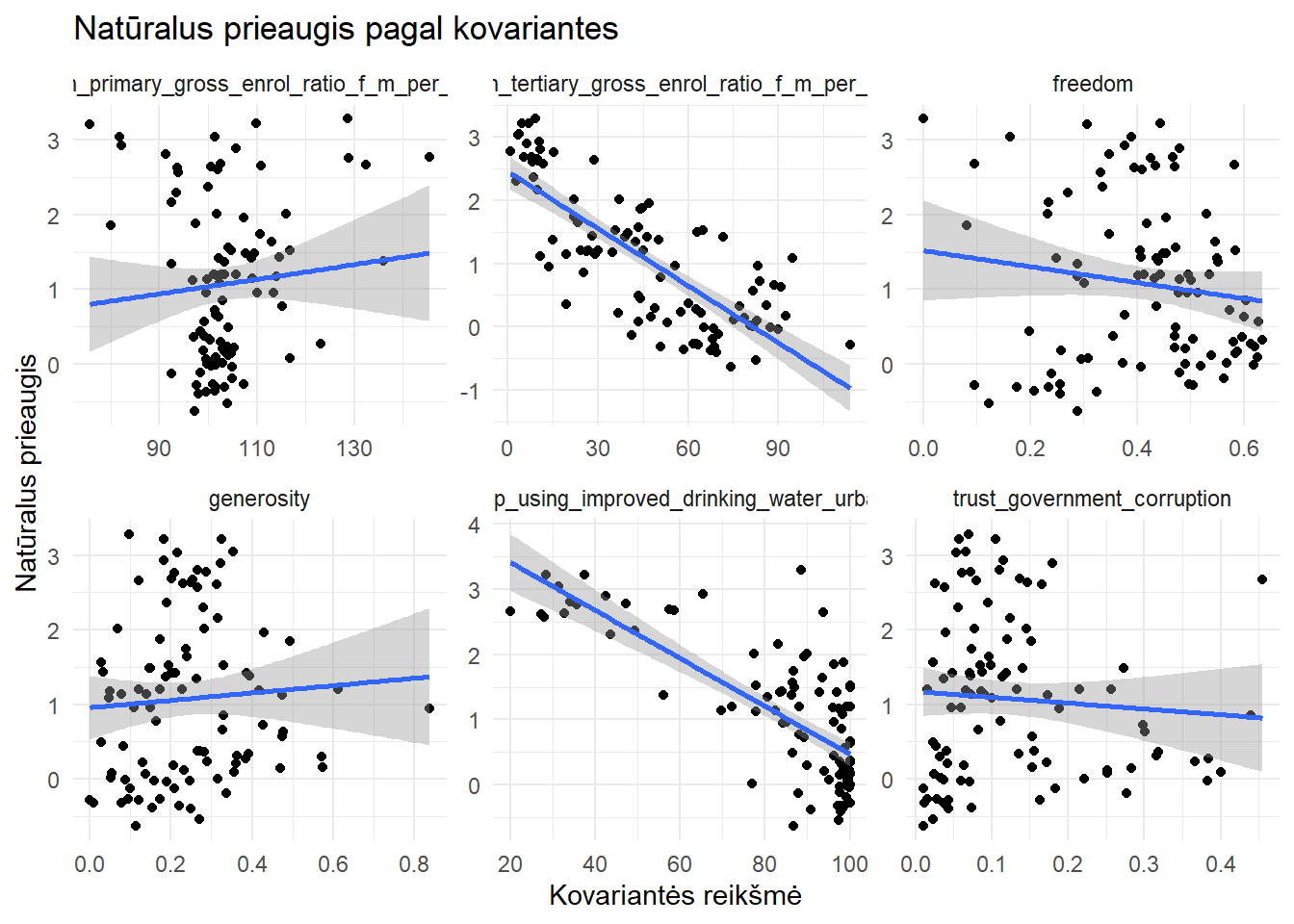
# sklaidos diagramos su kiekviena kovariante  
scatterplot <- function(name, name2, main, ylab) {  
 a <- regression\_train %>%  
 dplyr::select(1:5, {{ name }}, -{{ name2 }}) %>%  
 pivot\_longer(-{{ name }}) %>%  
 ggplot(aes(x = value, y = {{ name }})) +  
 facet\_wrap(vars(name), scales = "free") +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 theme\_minimal() +  
 labs(title = main) + xlab("Kovariantės reikšmė") + ylab(ylab)  
  
  
 b <- regression\_train %>%  
 dplyr::select(6:length(regression\_train), {{ name }}, -{{ name2 }}) %>%  
 pivot\_longer(-{{ name }}) %>%  
 ggplot(aes(x = value, y = {{ name }})) +  
 facet\_wrap(vars(name), scales = "free") +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 theme\_minimal() +  
 labs(title = main) + xlab("Kovariantės reikšmė") + ylab(ylab)  
  
 plot(a)  
  
 plot(b)  
}  
  
  
scatterplot(migration\_growth, natural\_growth, "Migracijos prieaugis pagal kovariantes","Migracijos prieaugis")



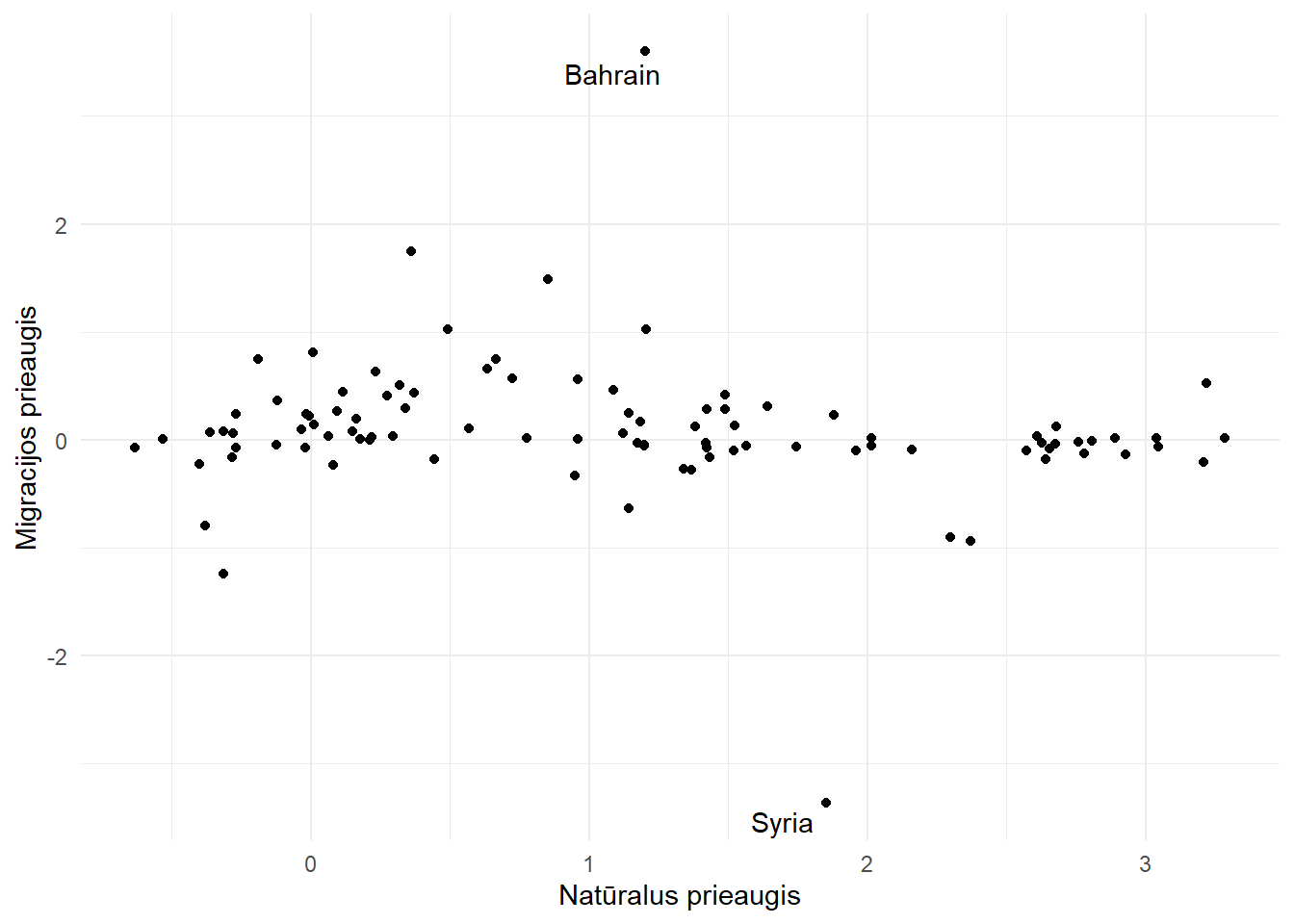


scatterplot(natural\_growth, migration\_growth, "Natūralus prieaugis pagal kovariantes","Natūralus prieaugis")





library(ggrepel)  
# migracijos ir natūralaus prieaugio sklaidos grafikas  
ggplot(regression\_train, aes(natural\_growth, migration\_growth)) +  
 geom\_point() +  
 theme\_minimal() +  
 xlab("Natūralus prieaugis") +  
 ylab("Migracijos prieaugis") + labs(main="Migracijos ir natūralus prieaugis") +  
 geom\_text\_repel(data=(regression\_train %>% cbind(country\_train))[abs(regression\_train$migration\_growth)>2,]  
 ,aes(label=country\_train))



# matomos dvi labai stiprios išskirtys  
outlier\_indices <- regression\_train$migration\_growth %>%  
 abs() %>%  
 order(decreasing = TRUE) %>%  
 `[`(1:2)

library(car)  
library(effects)  
library(lm.beta)  
library(broom)

Sudaryti atskiri tiesiniai modeliai natūraliam ir migracijos gyventojų prieaugiuibės. Atlikta pažingsninė regresija. Sumažintame modelyje gauta, kad migracijos prieaugis labiausiai priklauso nuo.

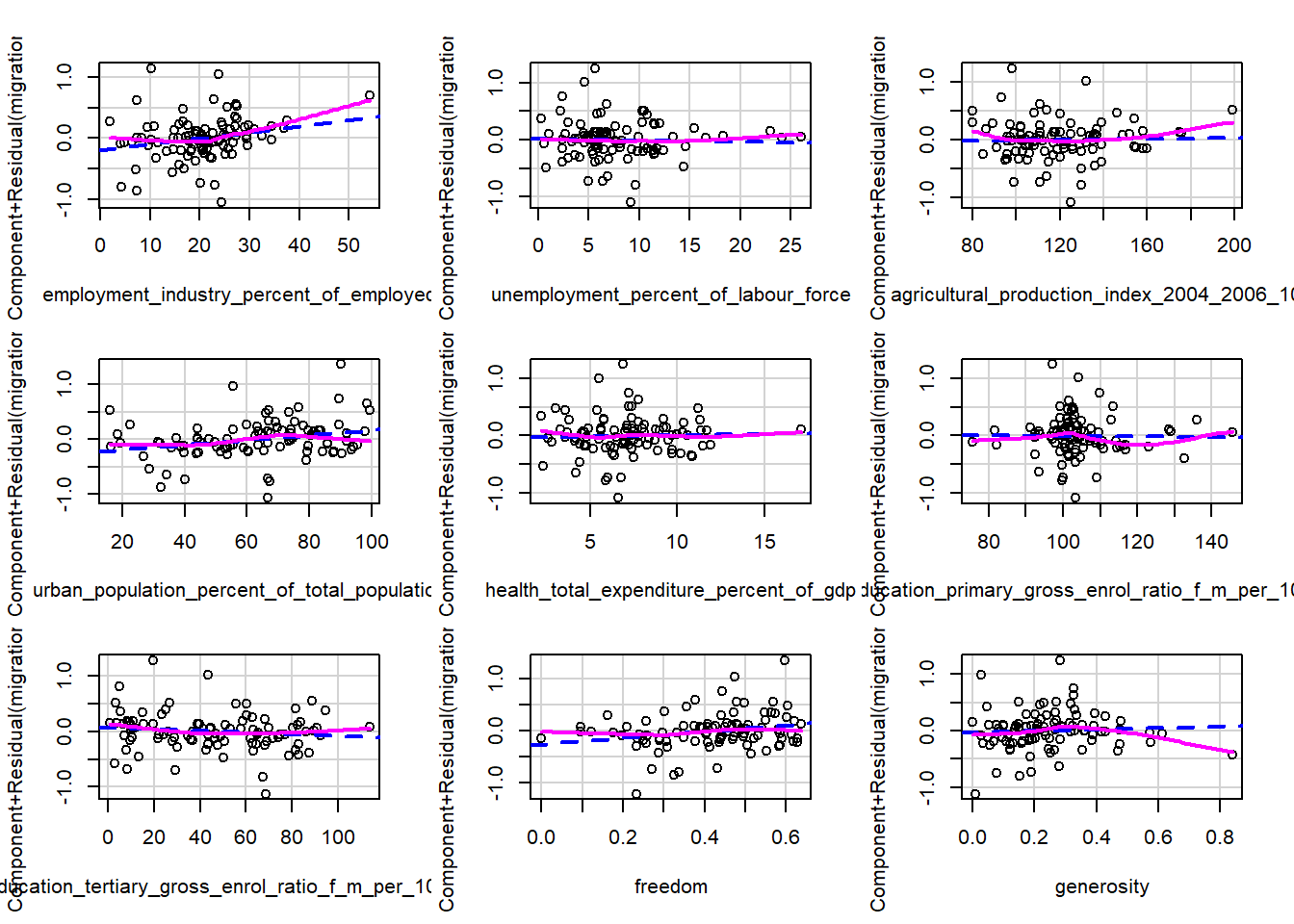
Tokia pati procedūra naudota ir sudarant natūralaus gyventojų prieaugio modelį. Šį kartą ryškių išskirčių nerasta

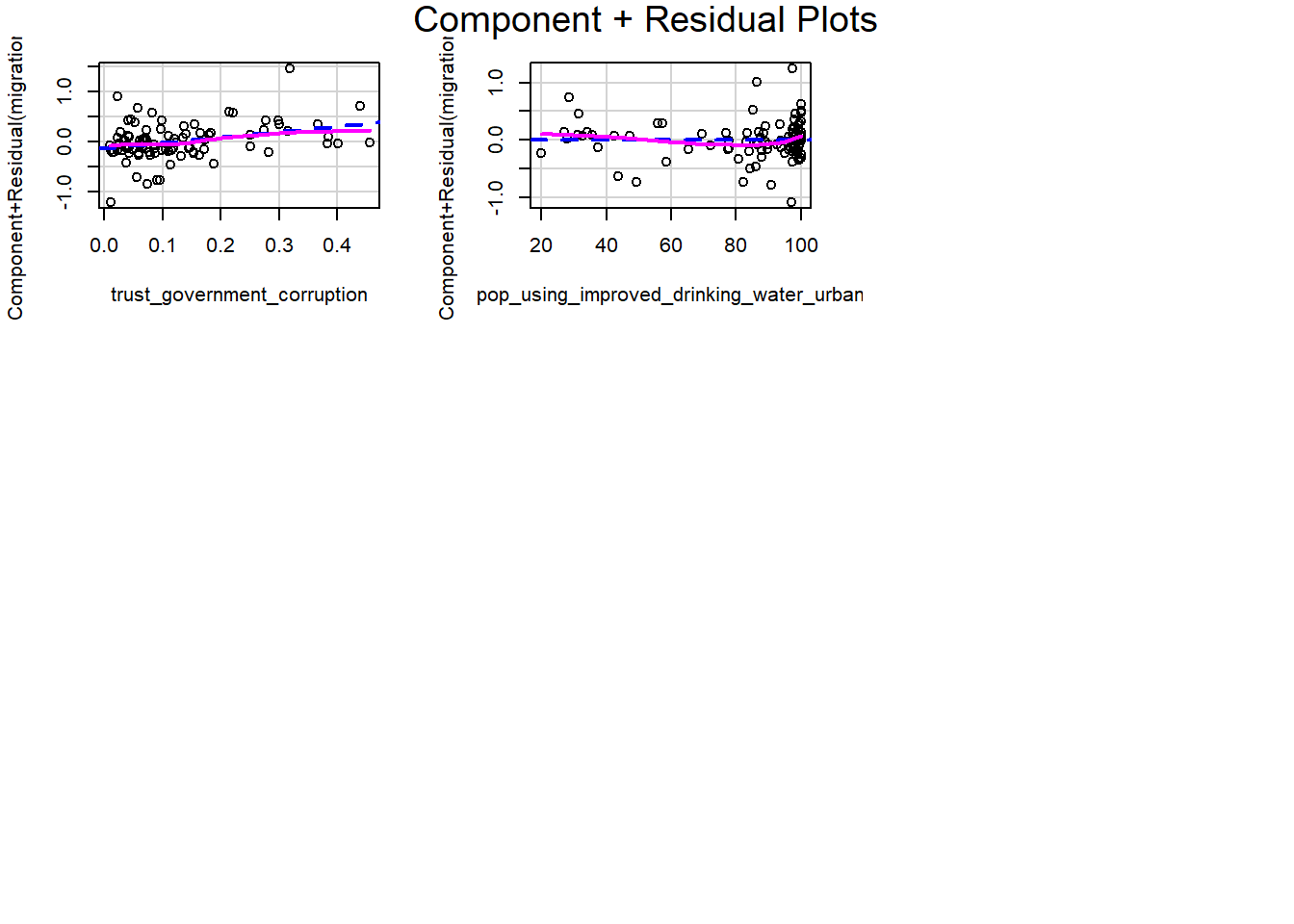
# sudaromas paprastos regresijos modelis, atliekama pažingsninė regresija  
linear\_fit <- function(formula) {  
 model\_linear <- lm(formula, data = data)  
  
 # diagnostiniai grafikai  
 crPlots(model\_linear)  
 plot(model\_linear)  
 plot(cooks.distance(model\_linear))  
  
 # pažingsninė regresija  
 model\_linear\_small <- MASS::stepAIC(model\_linear, direction = "both", trace = 0)  
  
 # ar yra statistiškai reikšmingas skirtumas  
 print(anova(model\_linear, model\_linear\_small))  
  
 # kovariančių efektų grafikas  
 plot(predictorEffects(model\_linear\_small))  
 print(summary(model\_linear\_small))  
  
  
 stand <- lm.beta(model\_linear\_small)  
 # standartizuotų koeficientų grafikas  
 coeff\_plot <- tidy(stand) %>%  
 filter(term != "(Intercept)") %>%  
 ggplot(aes(term, estimate)) +  
 geom\_pointrange(aes(ymin = estimate - std.error, ymax = estimate + std.error), color = "blue") +  
 scale\_x\_discrete() +  
 coord\_flip() +  
 theme\_minimal() +  
 labs(x = "Kovariantė", y = "Standartizuotos koeficientų reikšmės")  
  
 plot(coeff\_plot)  
  
 model\_linear\_small  
}

# Atskirai apmokomi modeliai migracijos ir natūraliam prieaugiui  
print("Tiesinės regresijos modelis migracijos prieaugiui")

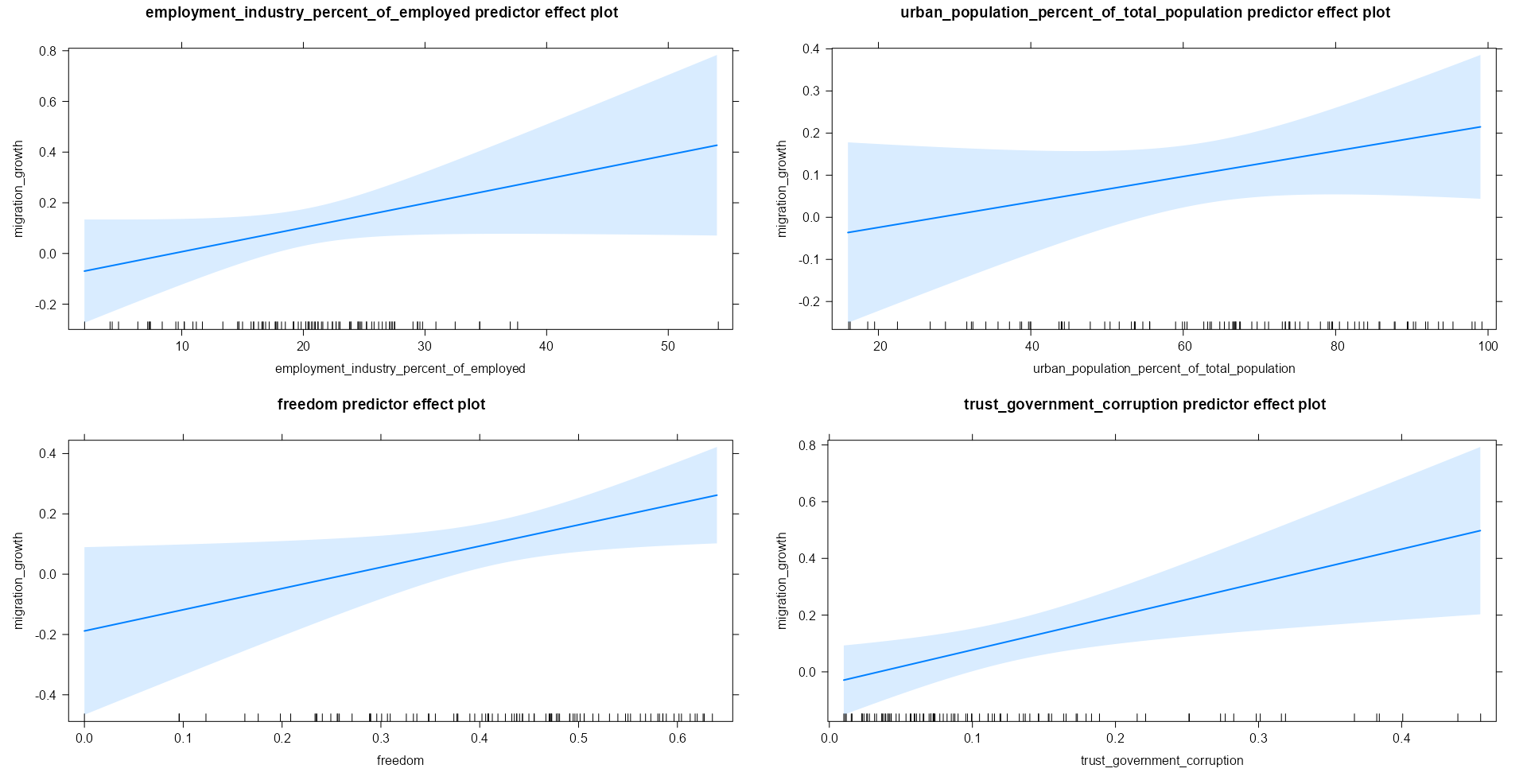
## [1] "Tiesinės regresijos modelis migracijos prieaugiui"

data <- regression\_train %>%  
 dplyr::select(-natural\_growth) %>%  
 slice(-outlier\_indices)  
model\_linear\_migration <- linear\_fit(migration\_growth ~ .)



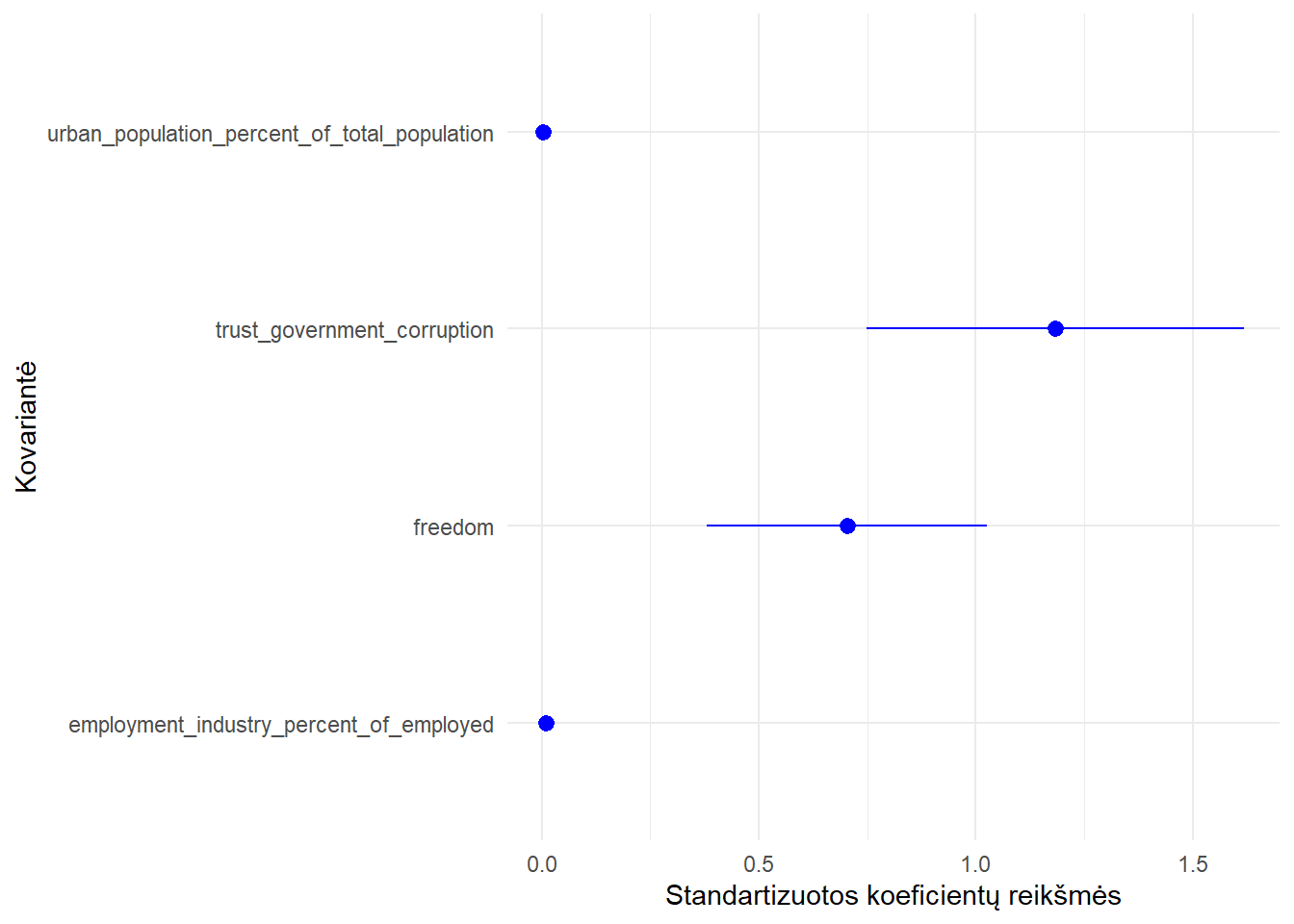


## Analysis of Variance Table  
##   
## Model 1: migration\_growth ~ employment\_industry\_percent\_of\_employed +   
## unemployment\_percent\_of\_labour\_force + agricultural\_production\_index\_2004\_2006\_100 +   
## urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp +   
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop +   
## freedom + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## Model 2: migration\_growth ~ employment\_industry\_percent\_of\_employed +   
## urban\_population\_percent\_of\_total\_population + freedom +   
## trust\_government\_corruption  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 83 11.211   
## 2 90 11.356 -7 -0.1452 0.1536 0.9931



hhdhfh

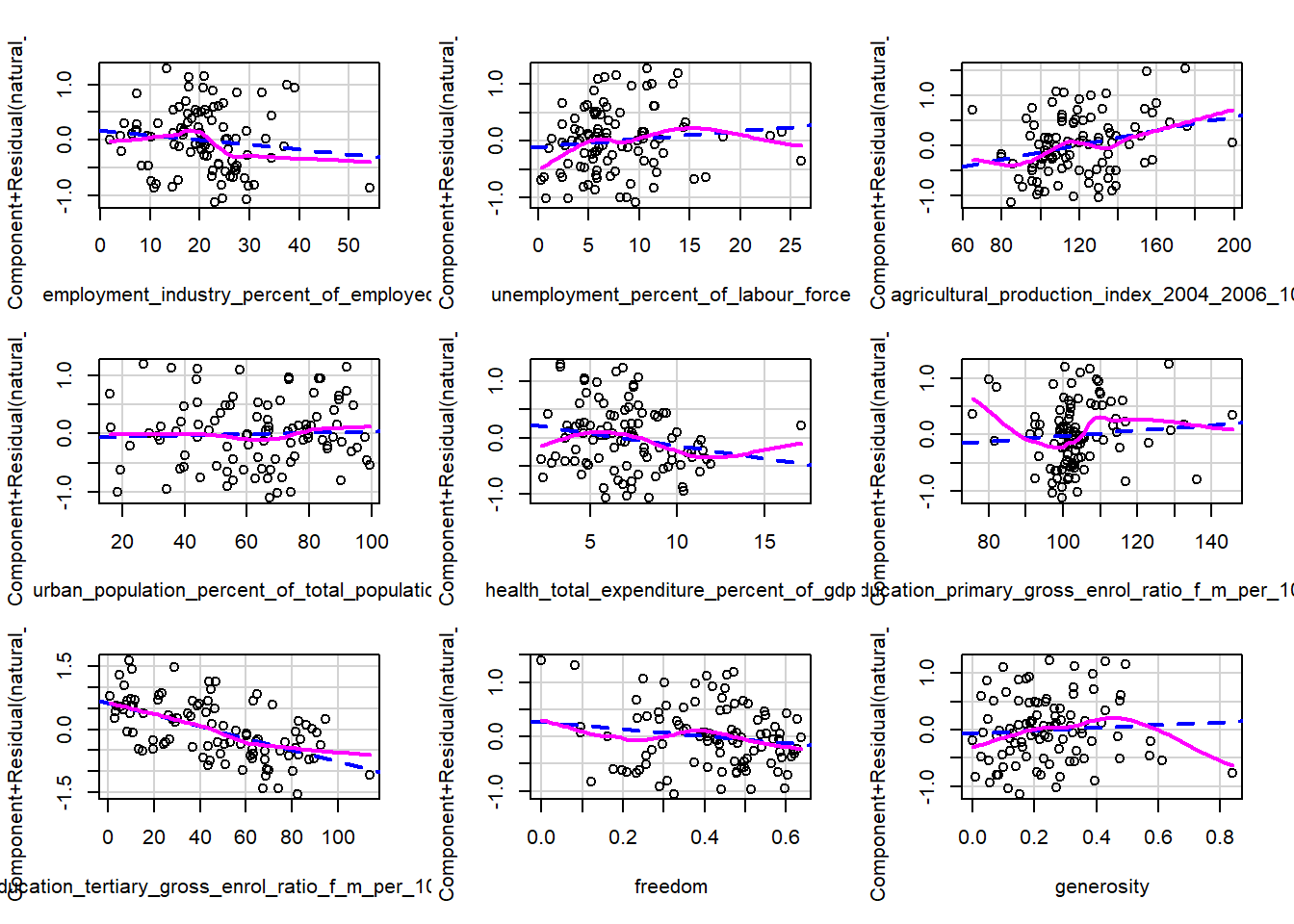
##   
## Call:  
## lm(formula = migration\_growth ~ employment\_industry\_percent\_of\_employed +   
## urban\_population\_percent\_of\_total\_population + freedom +   
## trust\_government\_corruption, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.13140 -0.17003 0.01515 0.13945 1.30339   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -0.720036 0.160058 -4.499  
## employment\_industry\_percent\_of\_employed 0.009551 0.005220 1.830  
## urban\_population\_percent\_of\_total\_population 0.003024 0.002161 1.399  
## freedom 0.703689 0.323155 2.178  
## trust\_government\_corruption 1.184538 0.434899 2.724  
## Pr(>|t|)   
## (Intercept) 2.04e-05 \*\*\*  
## employment\_industry\_percent\_of\_employed 0.07062 .   
## urban\_population\_percent\_of\_total\_population 0.16517   
## freedom 0.03205 \*   
## trust\_government\_corruption 0.00775 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3552 on 90 degrees of freedom  
## Multiple R-squared: 0.3368, Adjusted R-squared: 0.3074   
## F-statistic: 11.43 on 4 and 90 DF, p-value: 1.518e-07

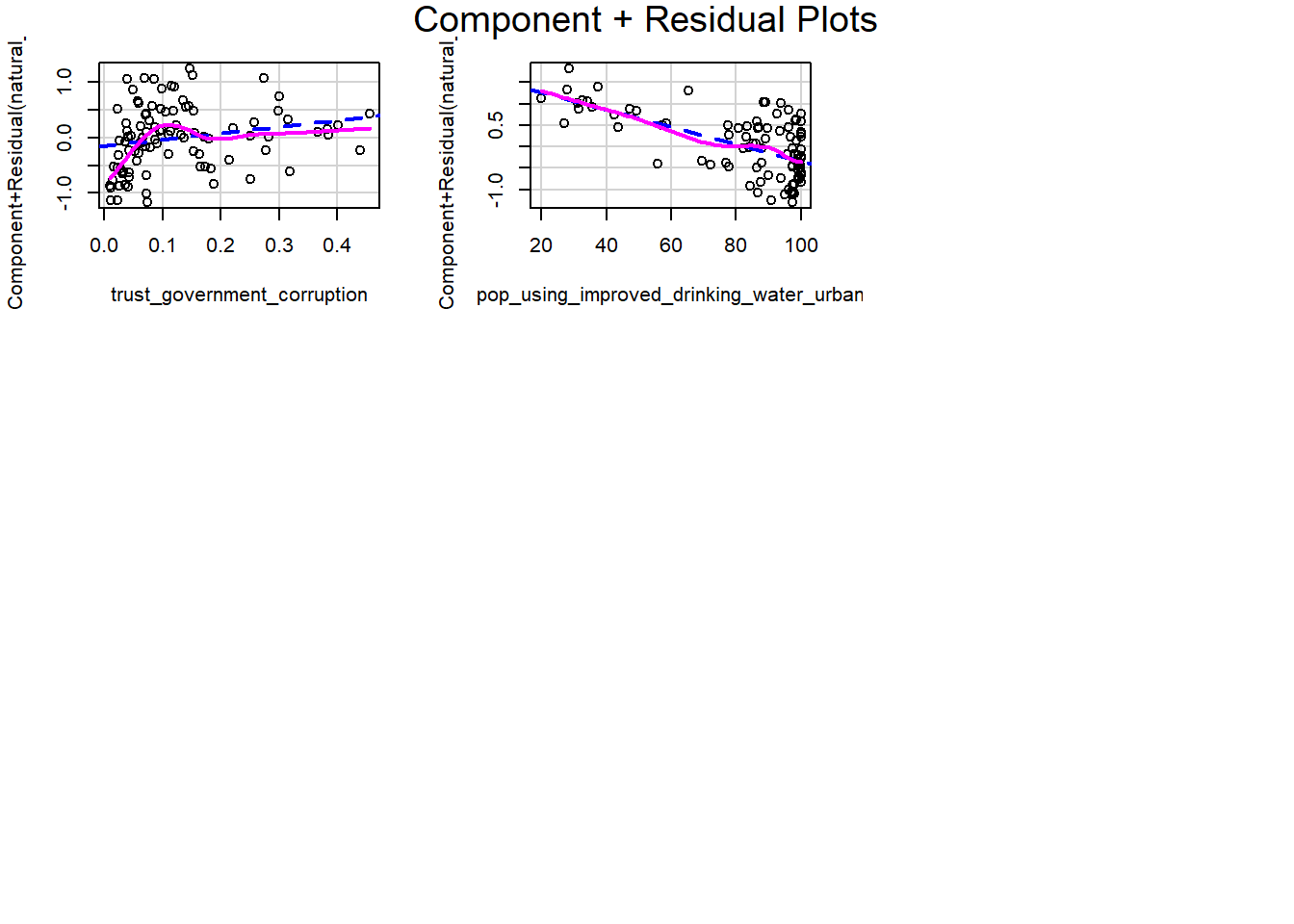


print("Tiesinės regresijos modelis natūraliam prieaugiui")

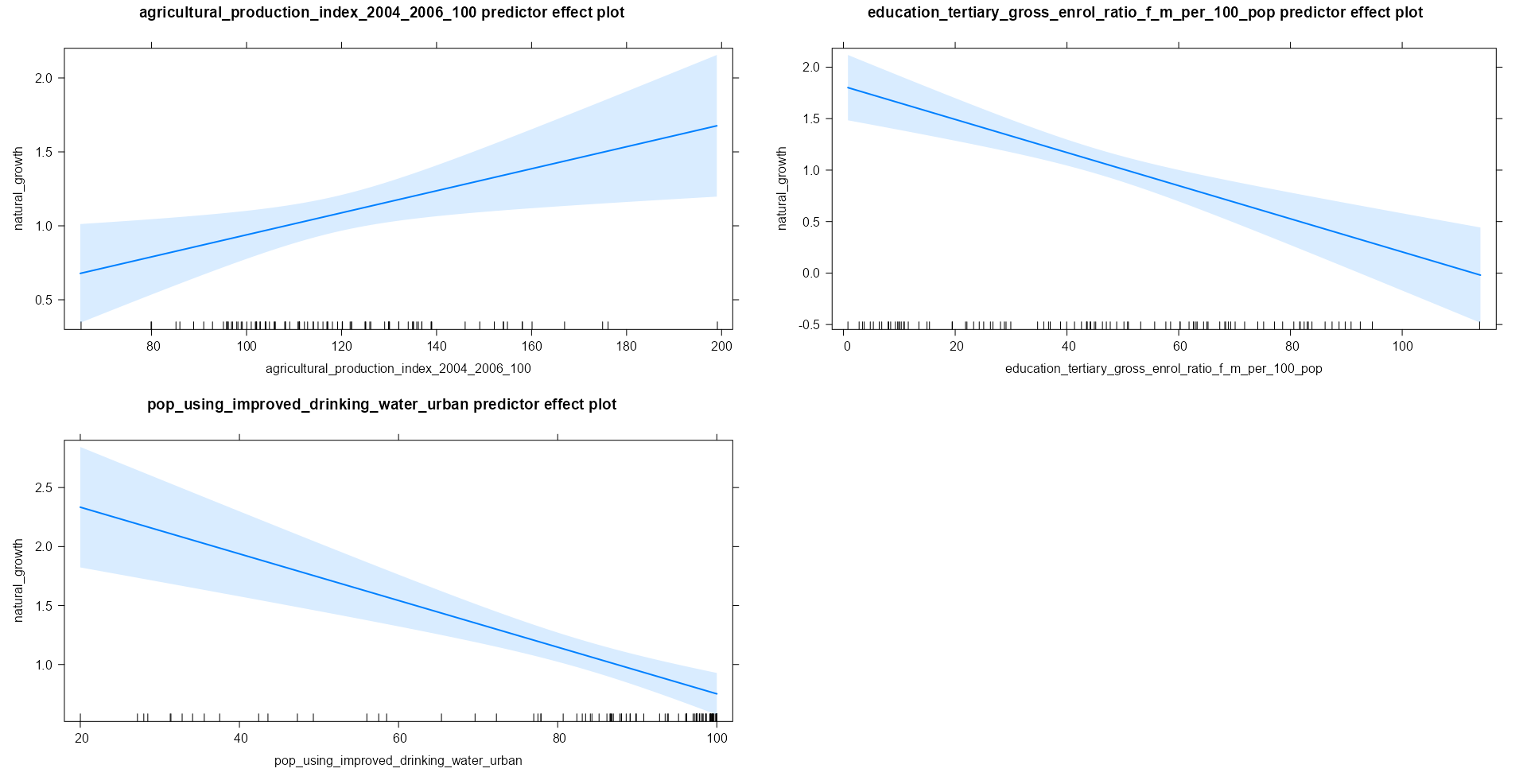
## [1] "Tiesinės regresijos modelis natūraliam prieaugiui"

data <- regression\_train %>% dplyr::select(-migration\_growth)  
model\_linear\_natural <- linear\_fit(natural\_growth ~ .)

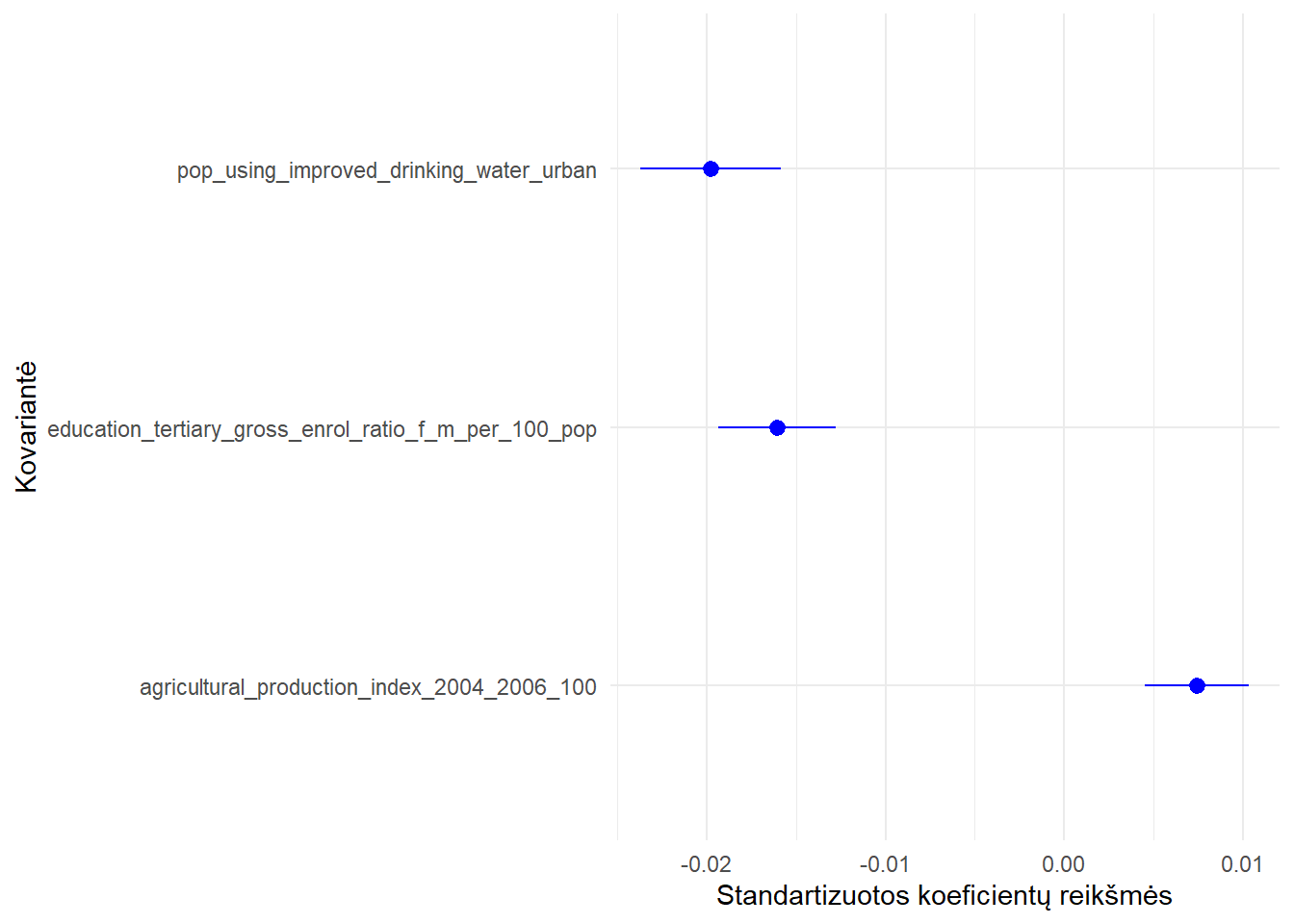




## Analysis of Variance Table  
##   
## Model 1: natural\_growth ~ employment\_industry\_percent\_of\_employed + unemployment\_percent\_of\_labour\_force +   
## agricultural\_production\_index\_2004\_2006\_100 + urban\_population\_percent\_of\_total\_population +   
## health\_total\_expenditure\_percent\_of\_gdp + education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop +   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + freedom +   
## generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## Model 2: natural\_growth ~ agricultural\_production\_index\_2004\_2006\_100 +   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + pop\_using\_improved\_drinking\_water\_urban  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 85 30.826   
## 2 93 33.392 -8 -2.5665 0.8846 0.533



##   
## Call:  
## lm(formula = natural\_growth ~ agricultural\_production\_index\_2004\_2006\_100 +   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + pop\_using\_improved\_drinking\_water\_urban,   
## data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.17333 -0.52663 0.02113 0.33529 1.40262   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 2.579233 0.468106  
## agricultural\_production\_index\_2004\_2006\_100 0.007448 0.002911  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -0.016088 0.003293  
## pop\_using\_improved\_drinking\_water\_urban -0.019777 0.003939  
## t value Pr(>|t|)   
## (Intercept) 5.510 3.18e-07 \*\*\*  
## agricultural\_production\_index\_2004\_2006\_100 2.559 0.0121 \*   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -4.885 4.29e-06 \*\*\*  
## pop\_using\_improved\_drinking\_water\_urban -5.021 2.47e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5992 on 93 degrees of freedom  
## Multiple R-squared: 0.7062, Adjusted R-squared: 0.6968   
## F-statistic: 74.52 on 3 and 93 DF, p-value: < 2.2e-16



dfdfbgfbgfb

# Matoma, kad migracijos prieaugiui tiesiniu modeliu gaunami daug prastesni rezultatai negu  
 # natūraliam prieaugiui

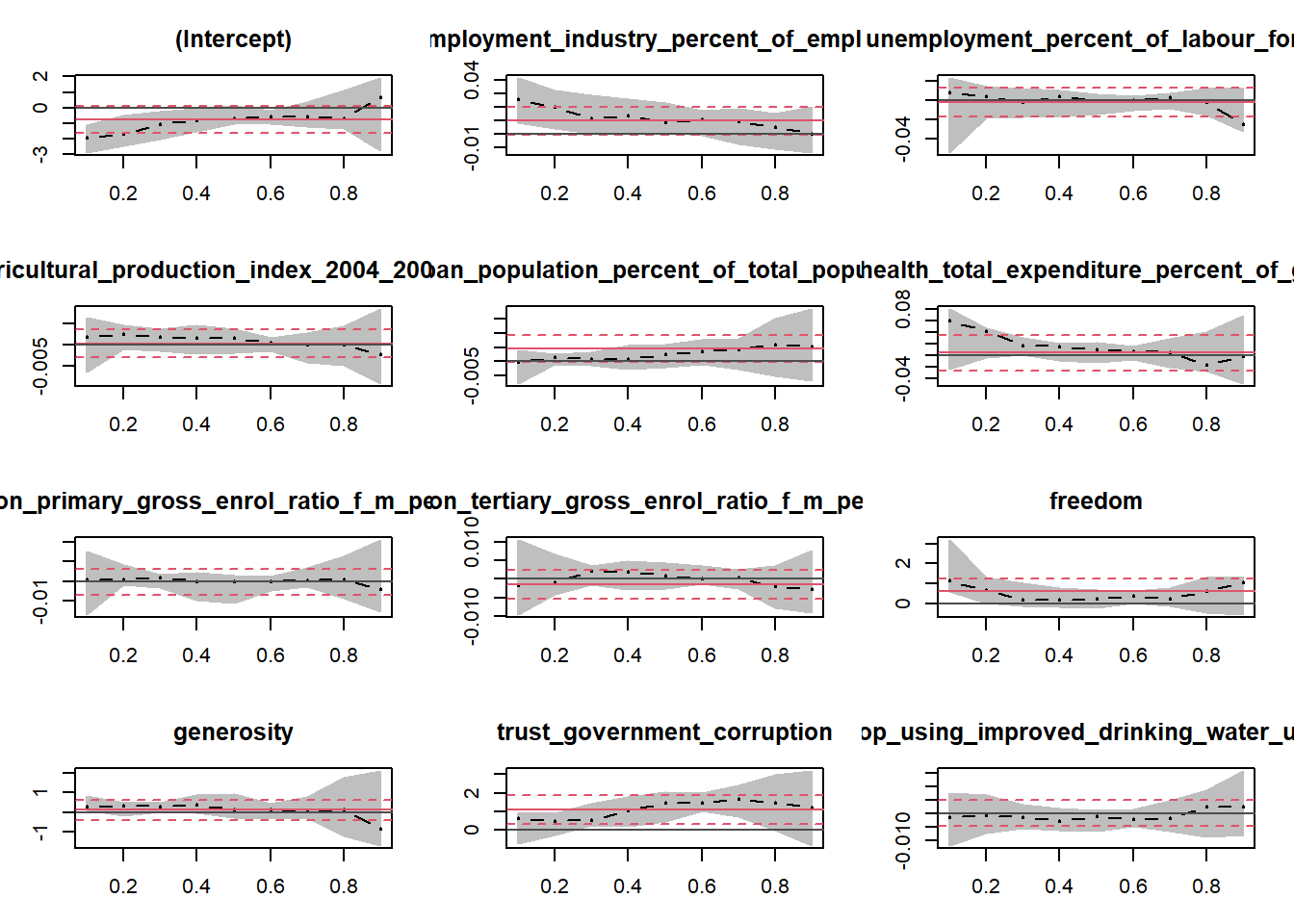
# Kvantilių regresija  
library(quantreg)  
  
quantile\_fit <- function() {  
 model\_quantile <- rq(formula, data = data, tau = tau)  
  
 print(summary(model\_quantile, se = "boot"))  
 plot(summary(model\_quantile))  
 print(anova(model\_quantile, test = "Wald", joint = FALSE))  
  
 model\_quantile  
}

print("Kvantilių regresija migracijos prieaugiui")

## [1] "Kvantilių regresija migracijos prieaugiui"

data <- regression\_train %>%  
 dplyr::select(-natural\_growth) %>%  
 slice(-outlier\_indices)  
tau <- seq(0.1, 0.9, 0.1)  
formula <- migration\_growth ~ .  
  
model\_quantile\_migration <- quantile\_fit()

##   
## Call: rq(formula = formula, tau = tau, data = data)  
##   
## tau: [1] 0.5  
##   
## Coefficients:  
## Value Std. Error  
## (Intercept) -0.70883 0.62031   
## employment\_industry\_percent\_of\_employed 0.00868 0.00830   
## unemployment\_percent\_of\_labour\_force -0.00092 0.00800   
## agricultural\_production\_index\_2004\_2006\_100 0.00160 0.00265   
## urban\_population\_percent\_of\_total\_population 0.00243 0.00289   
## health\_total\_expenditure\_percent\_of\_gdp 0.01011 0.01460   
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -0.00033 0.00419   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 0.00091 0.00259   
## freedom 0.22736 0.35053   
## generosity 0.15808 0.35722   
## trust\_government\_corruption 1.44830 0.58068   
## pop\_using\_improved\_drinking\_water\_urban -0.00114 0.00282   
## t value Pr(>|t|)  
## (Intercept) -1.14270 0.25645  
## employment\_industry\_percent\_of\_employed 1.04623 0.29849  
## unemployment\_percent\_of\_labour\_force -0.11501 0.90872  
## agricultural\_production\_index\_2004\_2006\_100 0.60567 0.54639  
## urban\_population\_percent\_of\_total\_population 0.83999 0.40333  
## health\_total\_expenditure\_percent\_of\_gdp 0.69240 0.49062  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -0.07900 0.93722  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 0.34953 0.72757  
## freedom 0.64862 0.51838  
## generosity 0.44253 0.65925  
## trust\_government\_corruption 2.49414 0.01461  
## pop\_using\_improved\_drinking\_water\_urban -0.40334 0.68774  
##   
## Call: rq(formula = formula, tau = tau, data = data)  
##



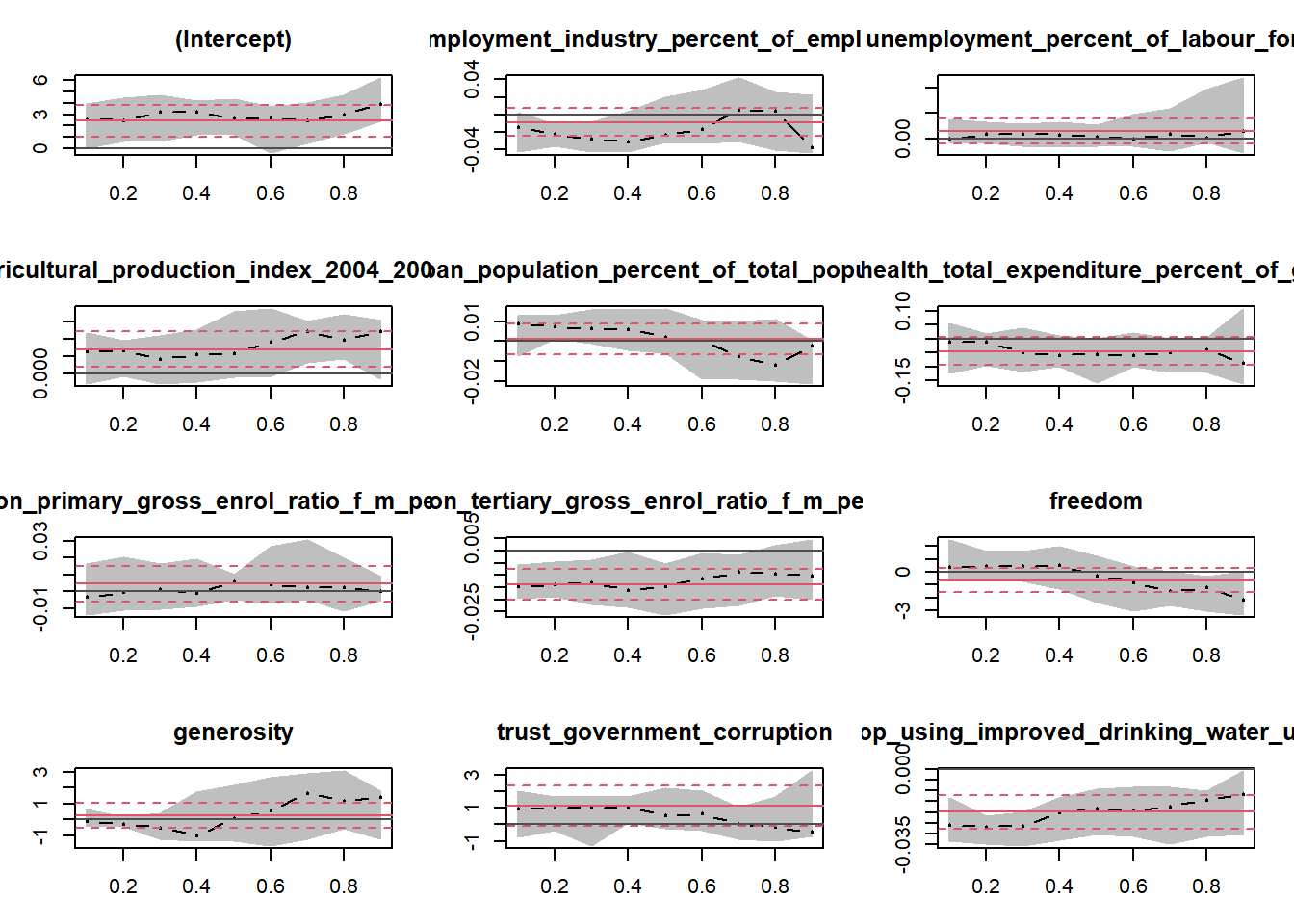
## Quantile Regression Analysis of Deviance Table  
##   
## Model: migration\_growth ~ employment\_industry\_percent\_of\_employed + unemployment\_percent\_of\_labour\_force + agricultural\_production\_index\_2004\_2006\_100 + urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp + education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + freedom + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## Tests of Equality of Distinct Slopes: tau in { 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 }  
##   
## Df Resid Df F value Pr(>F)  
## employment\_industry\_percent\_of\_employed 8 847 0.4387 0.8980  
## unemployment\_percent\_of\_labour\_force 8 847 0.8835 0.5298  
## agricultural\_production\_index\_2004\_2006\_100 8 847 0.3063 0.9638  
## urban\_population\_percent\_of\_total\_population 8 847 0.2722 0.9749  
## health\_total\_expenditure\_percent\_of\_gdp 8 847 0.4797 0.8711  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 8 847 0.2504 0.9808  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 8 847 0.4294 0.9037  
## freedom 8 847 0.5251 0.8382  
## generosity 8 847 1.4714 0.1636  
## trust\_government\_corruption 8 847 0.7046 0.6877  
## pop\_using\_improved\_drinking\_water\_urban 8 847 0.4466 0.8931

Ddiovdfpvodfvjd

print("Kvantilių regresija natūraliam prieaugiui")

## [1] "Kvantilių regresija natūraliam prieaugiui"

data <- regression\_train %>% dplyr::select(-migration\_growth)  
tau <- seq(0.1, 0.9, 0.1)  
formula <- natural\_growth ~ .  
  
model\_quantile\_migration <- quantile\_fit()  
##   
## Call: rq(formula = formula, tau = tau, data = data)  
##   
## tau: [1] 0.5  
##   
## Coefficients:  
## Value Std. Error  
## (Intercept) 2.64220 1.27213   
## employment\_industry\_percent\_of\_employed -0.02361 0.01944   
## unemployment\_percent\_of\_labour\_force 0.00352 0.02022   
## agricultural\_production\_index\_2004\_2006\_100 0.00596 0.00547   
## urban\_population\_percent\_of\_total\_population 0.00228 0.00764   
## health\_total\_expenditure\_percent\_of\_gdp -0.05480 0.05369   
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 0.00591 0.01174   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -0.01463 0.00662   
## freedom -0.27038 1.09466   
## generosity 0.08732 1.10537   
## trust\_government\_corruption 0.54026 0.95658   
## pop\_using\_improved\_drinking\_water\_urban -0.01822 0.00865   
## t value Pr(>|t|)  
## (Intercept) 2.07700 0.04082  
## employment\_industry\_percent\_of\_employed -1.21476 0.22782  
## unemployment\_percent\_of\_labour\_force 0.17418 0.86214  
## agricultural\_production\_index\_2004\_2006\_100 1.09062 0.27852  
## urban\_population\_percent\_of\_total\_population 0.29875 0.76586  
## health\_total\_expenditure\_percent\_of\_gdp -1.02079 0.31025  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 0.50337 0.61600  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -2.21061 0.02975  
## freedom -0.24700 0.80551  
## generosity 0.07900 0.93722  
## trust\_government\_corruption 0.56478 0.57371  
## pop\_using\_improved\_drinking\_water\_urban -2.10564 0.03819  
##



## Quantile Regression Analysis of Deviance Table  
##   
## Model: natural\_growth ~ employment\_industry\_percent\_of\_employed + unemployment\_percent\_of\_labour\_force + agricultural\_production\_index\_2004\_2006\_100 + urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp + education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + freedom + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## Tests of Equality of Distinct Slopes: tau in { 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 }  
##   
## Df Resid Df F value  
## employment\_industry\_percent\_of\_employed 8 865 1.7082  
## unemployment\_percent\_of\_labour\_force 8 865 0.3304  
## agricultural\_production\_index\_2004\_2006\_100 8 865 0.9534  
## urban\_population\_percent\_of\_total\_population 8 865 1.7414  
## health\_total\_expenditure\_percent\_of\_gdp 8 865 0.7544  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 8 865 0.2721  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 8 865 0.5841  
## freedom 8 865 1.7266  
## generosity 8 865 2.7101  
## trust\_government\_corruption 8 865 0.6416  
## pop\_using\_improved\_drinking\_water\_urban 8 865 0.7526  
## Pr(>F)   
## employment\_industry\_percent\_of\_employed 0.092640 .   
## unemployment\_percent\_of\_labour\_force 0.954457   
## agricultural\_production\_index\_2004\_2006\_100 0.471410   
## urban\_population\_percent\_of\_total\_population 0.085268 .   
## health\_total\_expenditure\_percent\_of\_gdp 0.643271   
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 0.974933   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop 0.791577   
## freedom 0.088479 .   
## generosity 0.005959 \*\*  
## trust\_government\_corruption 0.743013   
## pop\_using\_improved\_drinking\_water\_urban 0.644911   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Prieš tai sudaryti paprasti tiesinės regresijos modeliai leidžia lengvai įvertinti kovariančių įtaką gyventojų prieaugiui, tačiau modelyje neatsižvelgiama į galimus netiesinius ryšius tarp kovariančių ir atsako ir prognozuojant šiais reikšmes galimi gauti prasti rezultatai, todėl papildomai pasirinkta sudaryti netiesinės regresijos modelius, šiuo atveju naudojant apibendrintus adityviuosius modelius su glodniaisiais splainais.

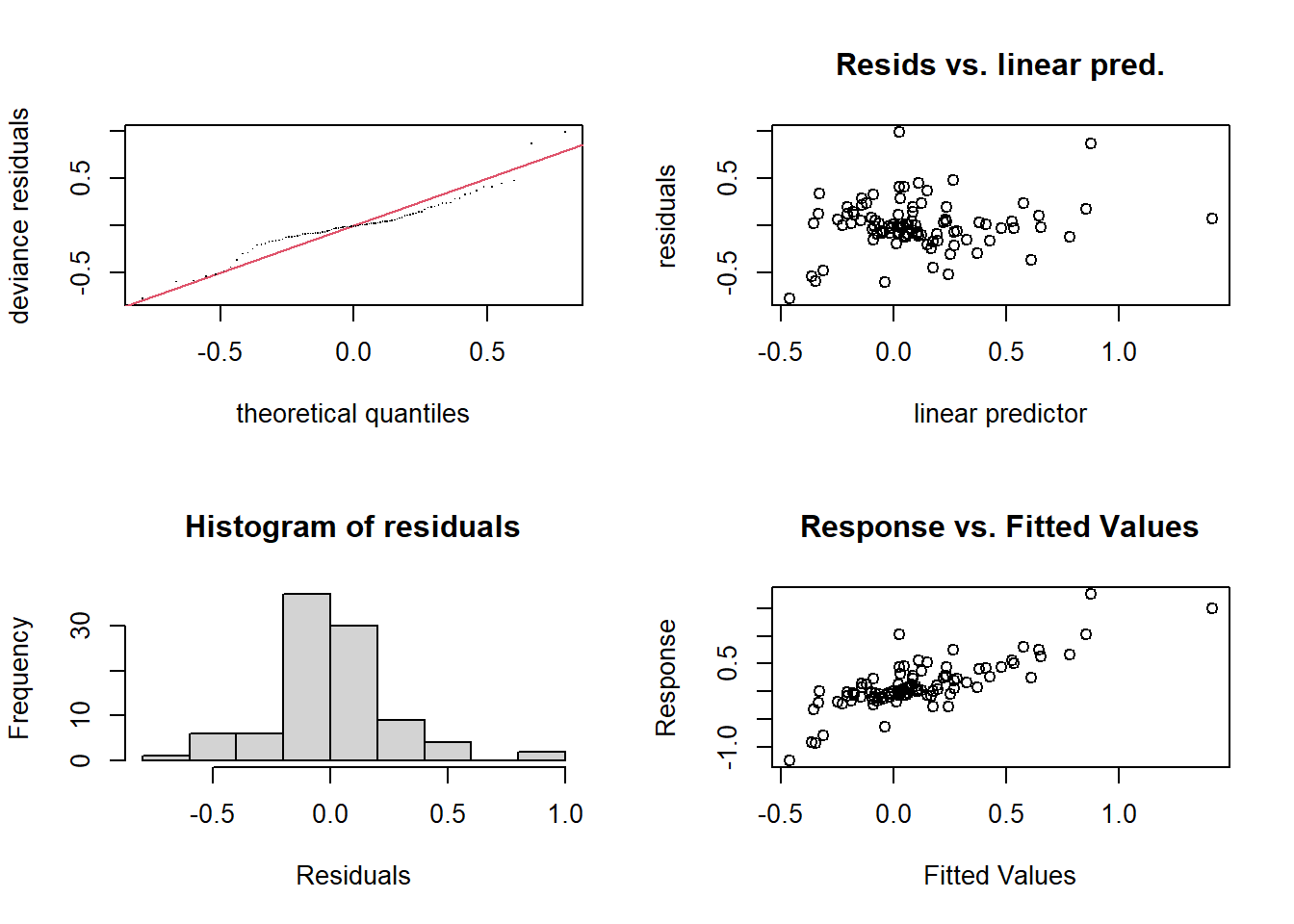
Iš pradžių splainais modeliuoti visos kovariantės, tačiau dalis jų modelyje gautos supaprastintos iki tiesės, todėl splainais toliau modeliuotos tik kovariantes „employment\_industry\_percent\_of\_employed“, „unemployment\_percent\_of\_labour\_force“ , „urban\_population\_percent\_of\_total\_population“, „education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop", „freedom“, „generosity“, „trust\_government\_corruption“. Kaip ir prieš tai sudarytame tiesiniame grafike matomas (R-adj 0.472)

library(mgcv)  
library(gratia)  
  
# siekiant tiksliau prognozuoti reikšmes naudinga sudaryti apibendrintus adityvius modelius,  
# kuriais galima įtraukti netiesinius sąryšius tarp kovariančių ir atsako  
fit\_gam <- function(formula, data) {  
 model\_gam <- gam(formula, data = data, select = FALSE)  
 gam.check(model\_gam)  
 summary(model\_gam)  
 draw(model\_gam)  
 k.check(model\_gam)  
 model\_gam  
}

print("GAM migracijos prieaugiui")

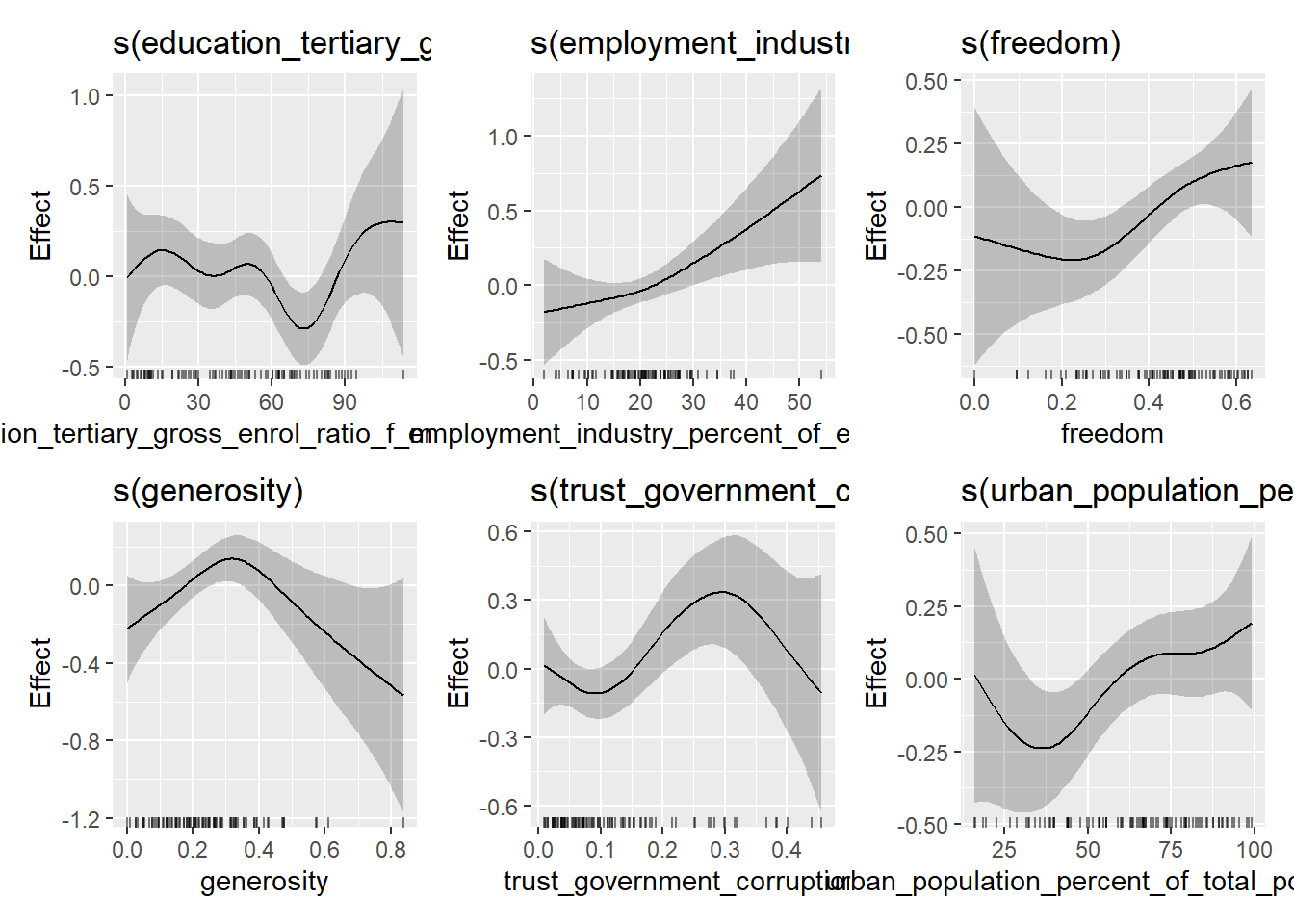
## [1] "GAM migracijos prieaugiui"

data <- regression\_train %>%  
 dplyr::select(-natural\_growth) %>%  
 slice(-outlier\_indices)  
  
  
model\_gam\_migration <- fit\_gam(migration\_growth ~ s(employment\_industry\_percent\_of\_employed) +  
 unemployment\_percent\_of\_labour\_force +  
 agricultural\_production\_index\_2004\_2006\_100 +  
 s(urban\_population\_percent\_of\_total\_population) +  
 health\_total\_expenditure\_percent\_of\_gdp +  
 education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop +  
 s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) +  
 s(freedom) +  
 s(generosity) +  
 s(trust\_government\_corruption), data)



##   
## Method: GCV Optimizer: magic  
## Smoothing parameter selection converged after 7 iterations.  
## The RMS GCV score gradient at convergence was 6.213085e-07 .  
## The Hessian was positive definite.  
## Model rank = 59 / 59   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index  
## s(employment\_industry\_percent\_of\_employed) 9.00 1.82 0.99  
## s(urban\_population\_percent\_of\_total\_population) 9.00 3.62 1.00  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 9.00 6.24 0.97  
## s(freedom) 9.00 2.55 0.84  
## s(generosity) 9.00 3.20 1.03  
## s(trust\_government\_corruption) 9.00 3.57 1.02  
## p-value   
## s(employment\_industry\_percent\_of\_employed) 0.370   
## s(urban\_population\_percent\_of\_total\_population) 0.525   
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 0.385   
## s(freedom) 0.055 .  
## s(generosity) 0.640   
## s(trust\_government\_corruption) 0.560   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

draw(model\_gam\_migration)



k.check(model\_gam\_migration)

## k' edf k-index  
## s(employment\_industry\_percent\_of\_employed) 9 1.816777 0.9868366  
## s(urban\_population\_percent\_of\_total\_population) 9 3.617499 0.9980236  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 9 6.243436 0.9668621  
## s(freedom) 9 2.549298 0.8395376  
## s(generosity) 9 3.197427 1.0346404  
## s(trust\_government\_corruption) 9 3.565093 1.0191396  
## p-value  
## s(employment\_industry\_percent\_of\_employed) 0.4075  
## s(urban\_population\_percent\_of\_total\_population) 0.4275  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 0.3300  
## s(freedom) 0.0600  
## s(generosity) 0.6125  
## s(trust\_government\_corruption) 0.5575

summary(model\_gam\_migration)

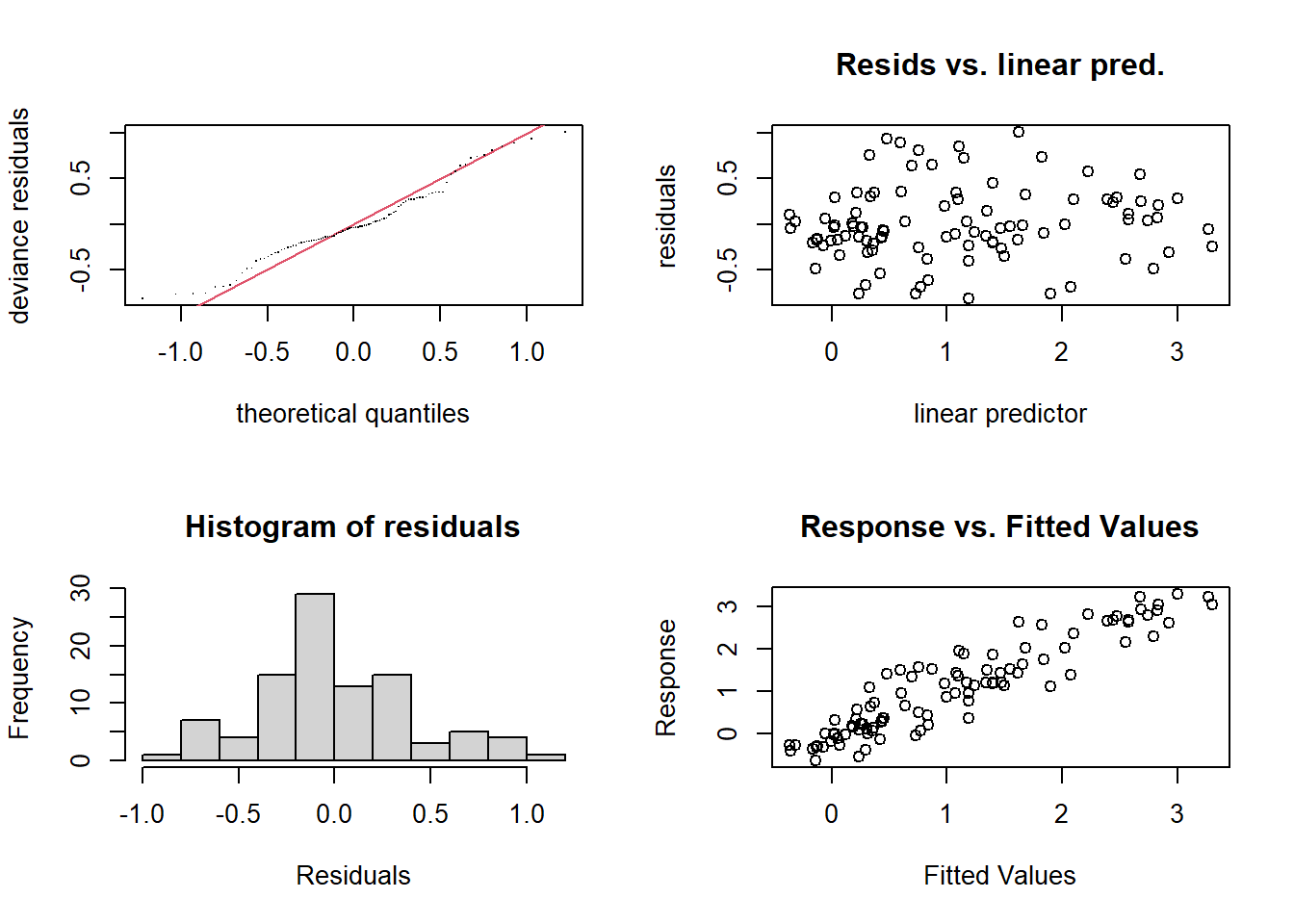
##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## migration\_growth ~ s(employment\_industry\_percent\_of\_employed) +   
## unemployment\_percent\_of\_labour\_force + agricultural\_production\_index\_2004\_2006\_100 +   
## s(urban\_population\_percent\_of\_total\_population) + health\_total\_expenditure\_percent\_of\_gdp +   
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) +   
## s(freedom) + s(generosity) + s(trust\_government\_corruption)  
##   
## Parametric coefficients:  
## Estimate Std. Error  
## (Intercept) 0.3740994 0.4853808  
## unemployment\_percent\_of\_labour\_force -0.0038517 0.0084051  
## agricultural\_production\_index\_2004\_2006\_100 -0.0001339 0.0019064  
## health\_total\_expenditure\_percent\_of\_gdp -0.0040872 0.0170518  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -0.0018611 0.0040183  
## t value Pr(>|t|)  
## (Intercept) 0.771 0.443  
## unemployment\_percent\_of\_labour\_force -0.458 0.648  
## agricultural\_production\_index\_2004\_2006\_100 -0.070 0.944  
## health\_total\_expenditure\_percent\_of\_gdp -0.240 0.811  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop -0.463 0.645  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F  
## s(employment\_industry\_percent\_of\_employed) 1.817 2.302 3.959  
## s(urban\_population\_percent\_of\_total\_population) 3.617 4.454 1.942  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 6.243 7.357 1.702  
## s(freedom) 2.549 3.196 2.817  
## s(generosity) 3.197 3.944 2.867  
## s(trust\_government\_corruption) 3.565 4.370 2.615  
## p-value   
## s(employment\_industry\_percent\_of\_employed) 0.0262 \*  
## s(urban\_population\_percent\_of\_total\_population) 0.1076   
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 0.1230   
## s(freedom) 0.0366 \*  
## s(generosity) 0.0310 \*  
## s(trust\_government\_corruption) 0.0392 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.472 Deviance explained = 61.2%  
## GCV = 0.13236 Scale est. = 0.096153 n = 95

Jidfvdf

Dcf

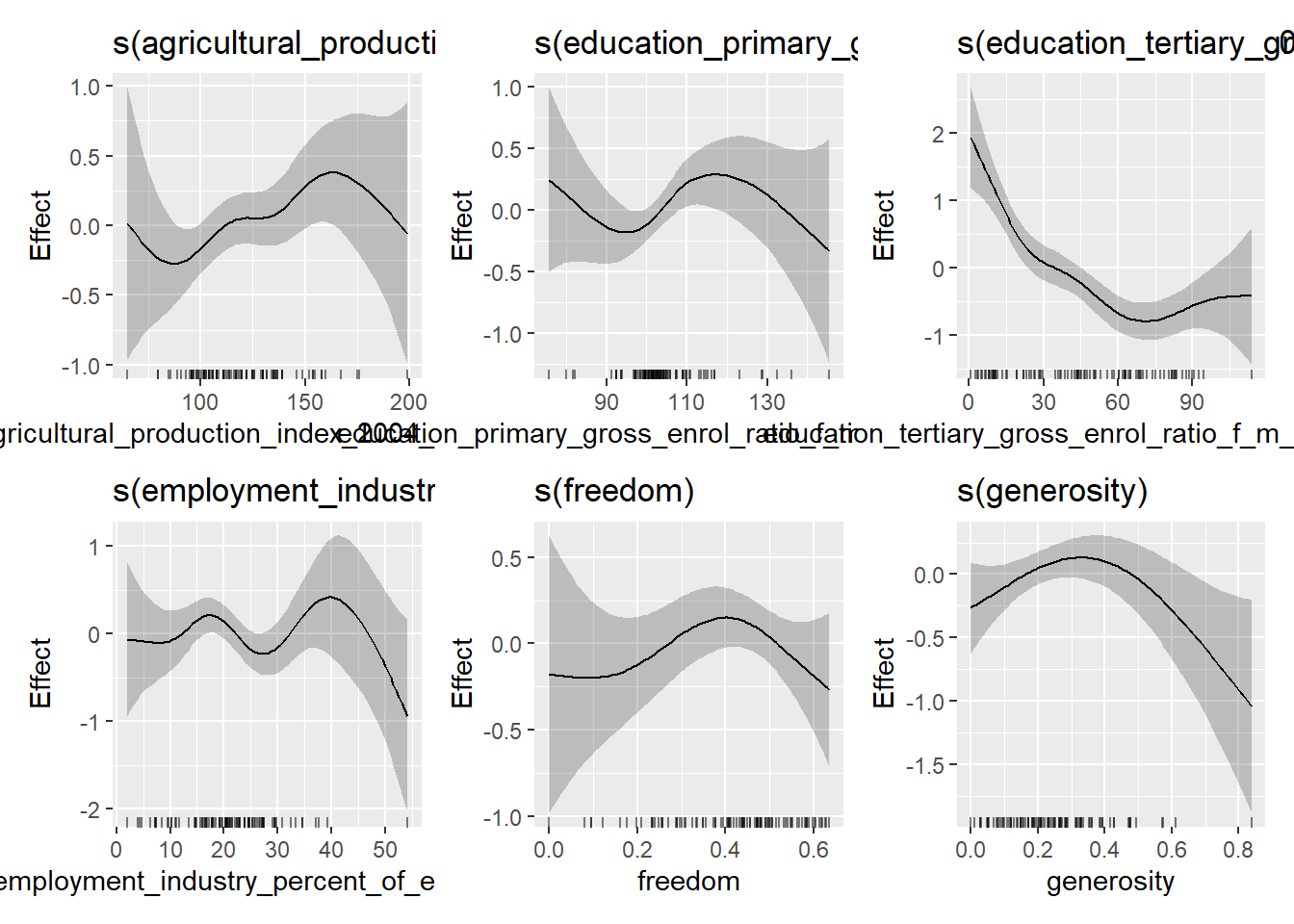
print("GAM natūraliam prieaugiui")

## [1] "GAM natūraliam prieaugiui"  
  
model\_gam\_natural <- fit\_gam(natural\_growth ~ s(employment\_industry\_percent\_of\_employed) +  
 unemployment\_percent\_of\_labour\_force +  
 s(agricultural\_production\_index\_2004\_2006\_100) +  
 urban\_population\_percent\_of\_total\_population +  
 health\_total\_expenditure\_percent\_of\_gdp +  
 s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) +  
 s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) +  
 s(freedom) +  
 s(generosity) +  
 trust\_government\_corruption, regression\_train %>% dplyr::select(-migration\_growth))



##   
## Method: GCV Optimizer: magic  
## Smoothing parameter selection converged after 7 iterations.  
## The RMS GCV score gradient at convergence was 1.371315e-07 .  
## The Hessian was positive definite.  
## Model rank = 59 / 59   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index  
## s(employment\_industry\_percent\_of\_employed) 9.00 5.16 0.96  
## s(agricultural\_production\_index\_2004\_2006\_100) 9.00 4.12 1.00  
## s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 9.00 3.34 1.07  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 9.00 4.83 1.33  
## s(freedom) 9.00 2.68 1.05  
## s(generosity) 9.00 2.41 0.99  
## p-value  
## s(employment\_industry\_percent\_of\_employed) 0.31  
## s(agricultural\_production\_index\_2004\_2006\_100) 0.38  
## s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 0.73  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 1.00  
## s(freedom) 0.66  
## s(generosity) 0.42

draw(model\_gam\_natural)



k.check(model\_gam\_natural)

## k' edf k-index  
## s(employment\_industry\_percent\_of\_employed) 9 5.159902 0.9649547  
## s(agricultural\_production\_index\_2004\_2006\_100) 9 4.123495 0.9987737  
## s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 9 3.340290 1.0692858  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 9 4.832763 1.3262088  
## s(freedom) 9 2.682792 1.0504460  
## s(generosity) 9 2.413316 0.9858652  
## p-value  
## s(employment\_industry\_percent\_of\_employed) 0.3300  
## s(agricultural\_production\_index\_2004\_2006\_100) 0.4350  
## s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 0.7600  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 0.9975  
## s(freedom) 0.6775  
## s(generosity) 0.4150

summary(model\_gam\_natural)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## natural\_growth ~ s(employment\_industry\_percent\_of\_employed) +   
## unemployment\_percent\_of\_labour\_force + s(agricultural\_production\_index\_2004\_2006\_100) +   
## urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp +   
## s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) +   
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) +   
## s(freedom) + s(generosity) + trust\_government\_corruption  
##   
## Parametric coefficients:  
## Estimate Std. Error t value  
## (Intercept) 1.253958 0.311162 4.030  
## unemployment\_percent\_of\_labour\_force 0.001227 0.013146 0.093  
## urban\_population\_percent\_of\_total\_population 0.001981 0.004159 0.476  
## health\_total\_expenditure\_percent\_of\_gdp -0.049611 0.028255 -1.756  
## trust\_government\_corruption 0.316856 0.779622 0.406  
## Pr(>|t|)   
## (Intercept) 0.000141 \*\*\*  
## unemployment\_percent\_of\_labour\_force 0.925931   
## urban\_population\_percent\_of\_total\_population 0.635351   
## health\_total\_expenditure\_percent\_of\_gdp 0.083527 .   
## trust\_government\_corruption 0.685681   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F  
## s(employment\_industry\_percent\_of\_employed) 5.160 6.148 2.391  
## s(agricultural\_production\_index\_2004\_2006\_100) 4.123 5.082 1.848  
## s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 3.340 4.130 2.309  
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 4.833 5.874 8.740  
## s(freedom) 2.683 3.333 1.310  
## s(generosity) 2.413 3.019 3.228  
## p-value   
## s(employment\_industry\_percent\_of\_employed) 0.0363 \*   
## s(agricultural\_production\_index\_2004\_2006\_100) 0.1225   
## s(education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 0.0636 .   
## s(education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop) 9.15e-07 \*\*\*  
## s(freedom) 0.2198   
## s(generosity) 0.0276 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.808 Deviance explained = 86.1%  
## GCV = 0.31706 Scale est. = 0.227 n = 97

Jdcjfdvfv

Dvfvgb

library(yardstick)  
  
# regresijos modelių įvertinimas  
regression\_test <- function(column, model\_linear, model\_gam, data, title) {  
 print(AIC(model\_linear))  
 print(AIC(model\_gam))  
  
  
 regression\_test <- data %>%  
 mutate(  
 predicted\_linear = predict(model\_linear, data),  
 predicted\_gam = predict(model\_gam, data)  
 )  
  
 set <- metric\_set(rmse, mae)  
  
 print("Tiesinis modelis")  
 print(set(regression\_test, {{ column }}, predicted\_linear))  
 print("GAM modelis")  
 print(set(regression\_test, {{ column }}, predicted\_gam))  
  
 regression\_test %>%  
 pivot\_longer(c(predicted\_gam, predicted\_linear)) %>%  
 mutate(name = factor(name, levels = c("predicted\_linear", "predicted\_gam"))) %>%  
 ggplot(aes({{ column }}, value)) +  
 geom\_point(size = 2) +  
 facet\_wrap(vars(name)) +  
 geom\_abline(color = "red", size = 2.25) +  
 labs(  
 x = "Tikros reikšmės", y = "Prognozuotos reikmšės",  
 title = title  
 ) +  
 theme\_minimal()  
}

# GAM modeliu gaunami nežymiai geresni rezultatai su mokymo duomeninis  
# , tačiau naudojant testavimo aibe pagerėjimo negaunama  
# Apskritai abu modeliai netinkami prognozuoti migracijos prieaugį  
print("Regresija migracijos prieaugiui")

## [1] "Regresija migracijos prieaugiui"

AIC(model\_linear\_migration)

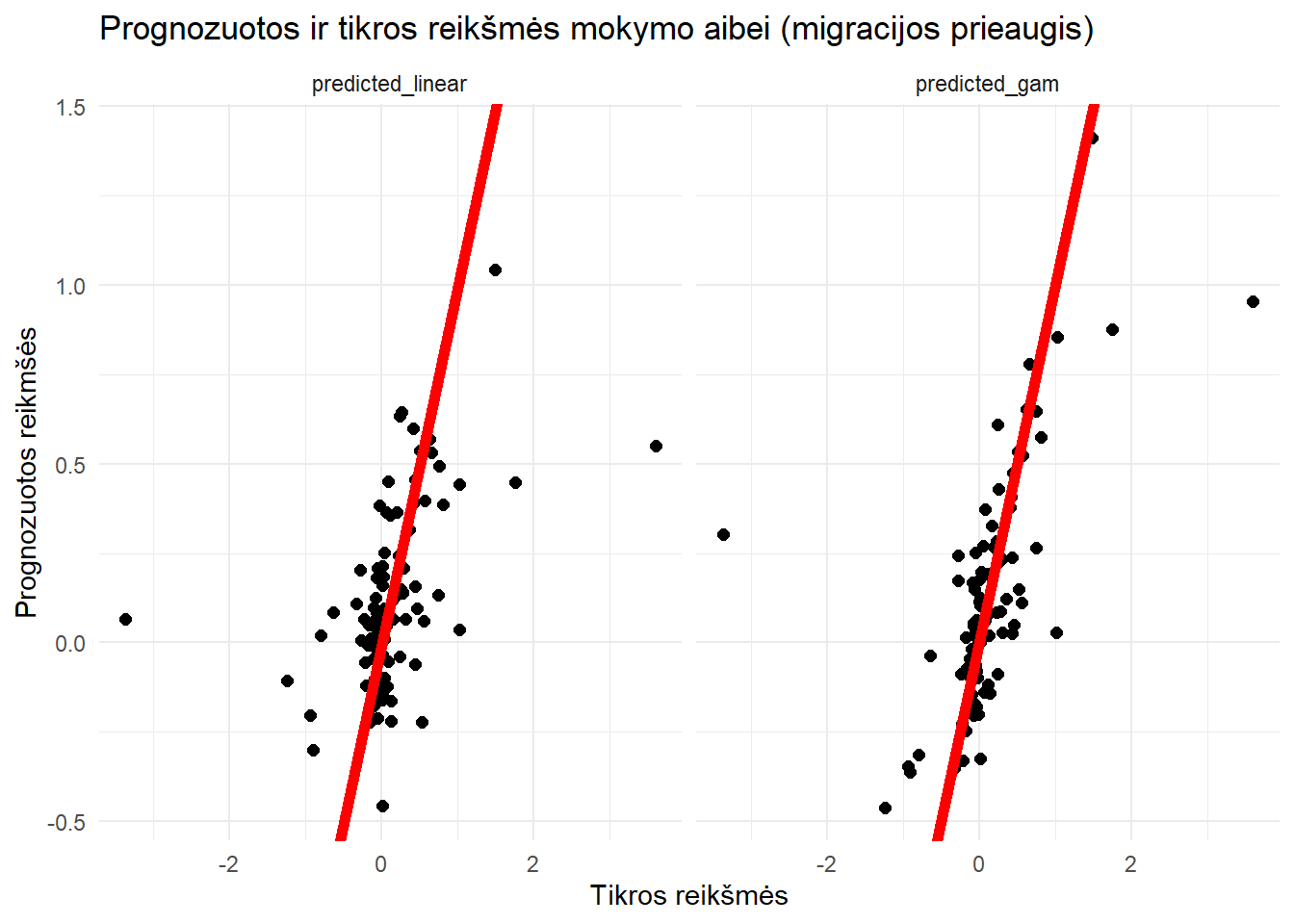
## [1] 79.80556

AIC(model\_gam\_migration)

## [1] 70.74101

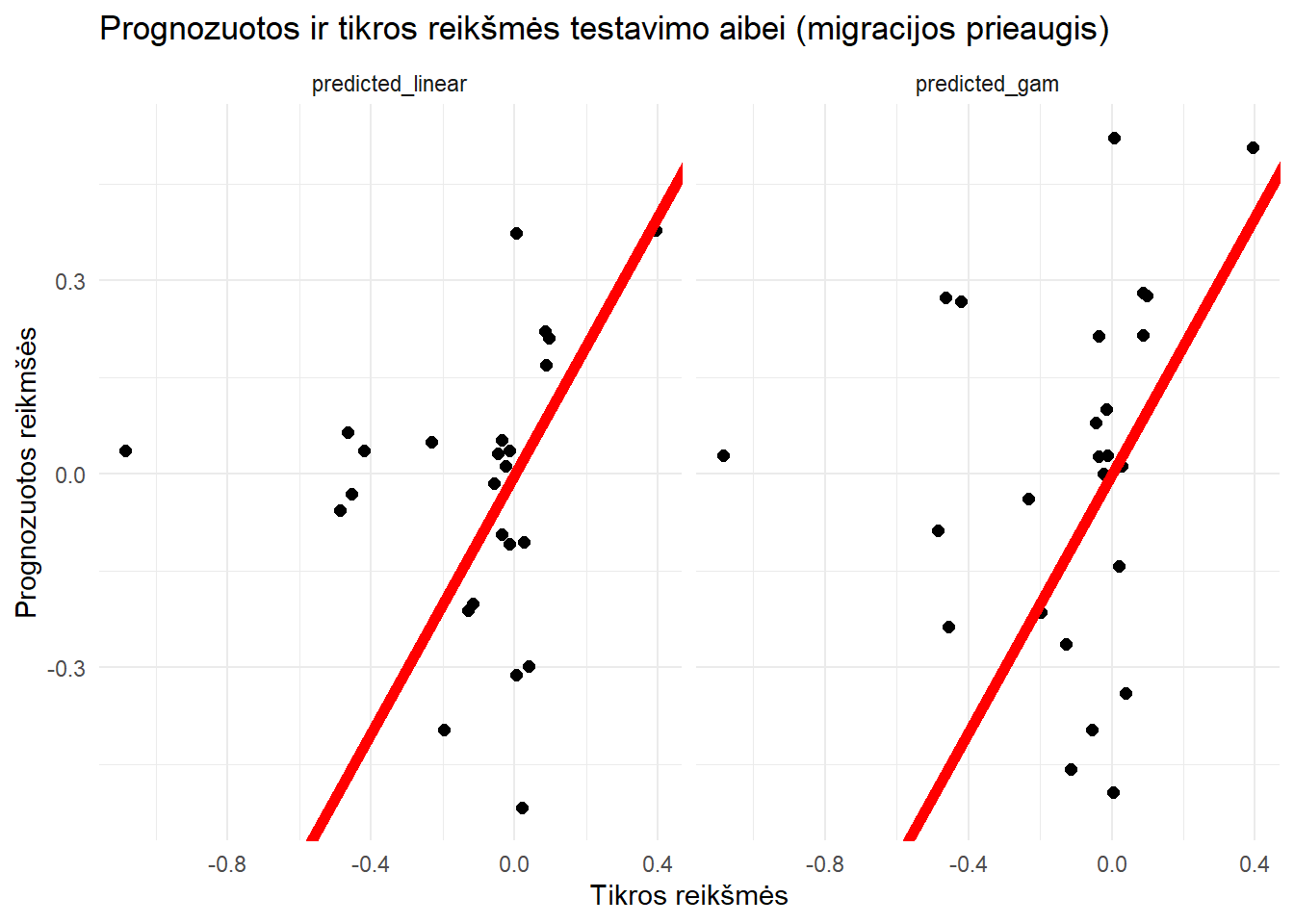
regression\_test(  
 migration\_growth, model\_linear\_migration, model\_gam\_migration, regression\_train,  
 "Prognozuotos ir tikros reikšmės mokymo aibei (migracijos prieaugis)"  
)

## [1] 79.80556  
## [1] 70.74101  
## [1] "Tiesinis modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.578  
## 2 mae standard 0.303  
## [1] "GAM modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.529  
## 2 mae standard 0.240



regression\_test(  
 migration\_growth, model\_linear\_migration, model\_gam\_migration, test,  
 "Prognozuotos ir tikros reikšmės testavimo aibei (migracijos prieaugis)"  
)

## [1] 79.80556  
## [1] 70.74101  
## [1] "Tiesinis modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.345  
## 2 mae standard 0.244  
## [1] "GAM modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.381  
## 2 mae standard 0.279



suahdijfo

# Prognozuojant natūralų prieaugi gaunami geresni rezultatai negu prognozuojant migracijos prieaugį  
# mokymo aibėje matomas stiprus GAM modeliu gautas rezultatų pagerėjimas, tačiau testavimo aibėje skirtumai  
 # tik minimalūs  
print("Regresija natūraliam prieaugiui")

## [1] "Regresija natūraliam prieaugiui"

AIC(model\_linear\_natural)

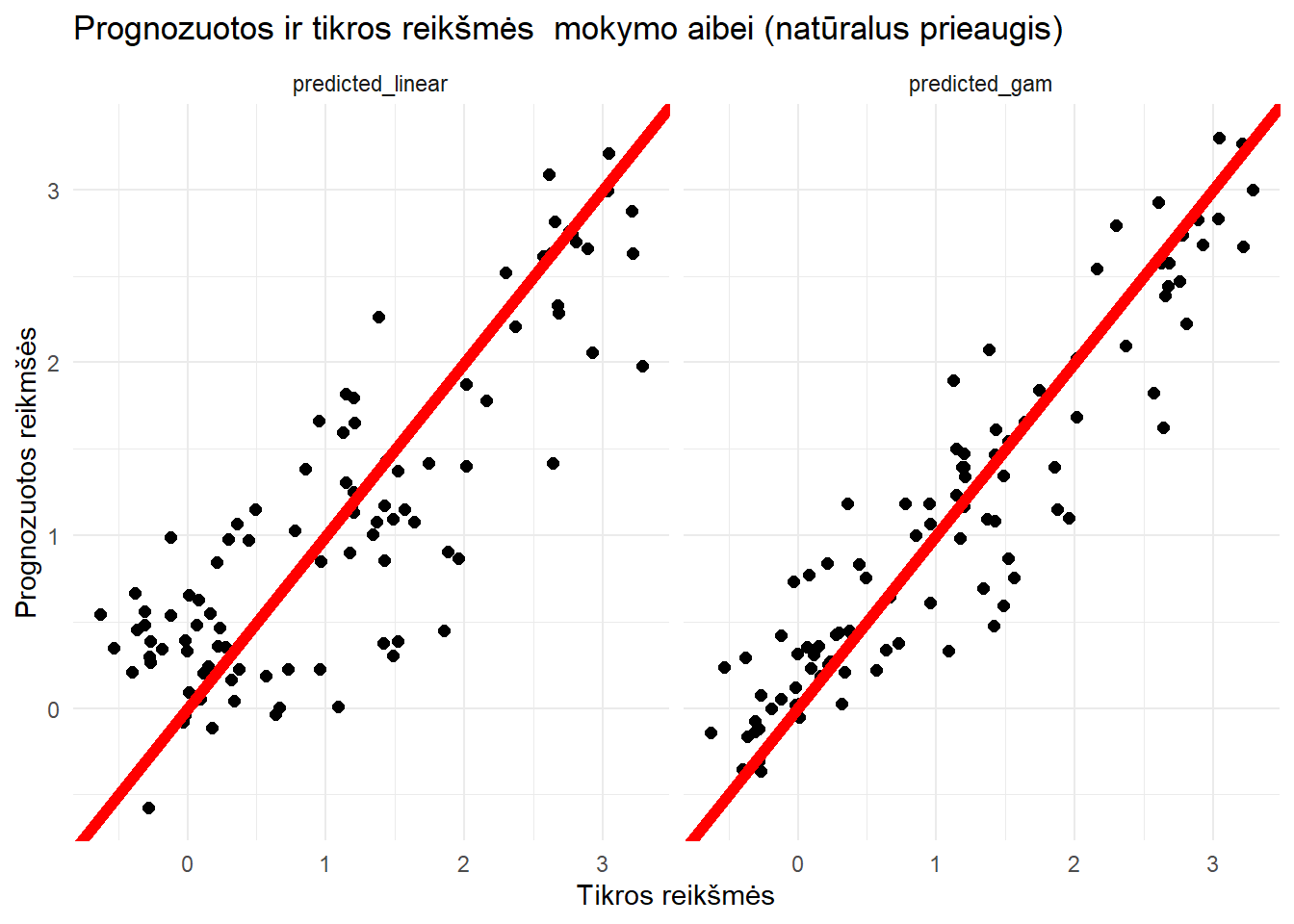
## [1] 181.8344

AIC(model\_gam\_natural)

## [1] 156.1355

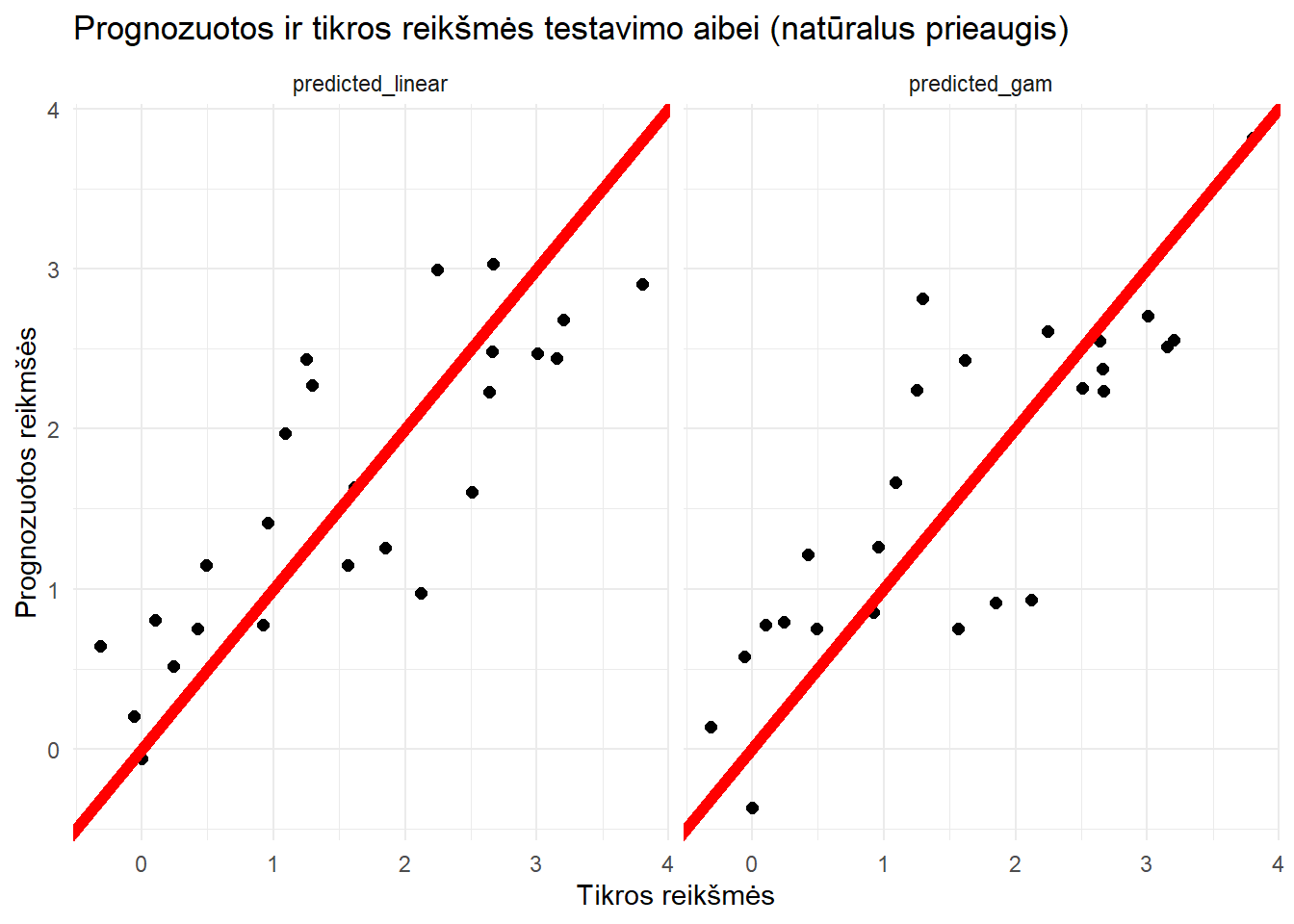
regression\_test(  
 natural\_growth, model\_linear\_natural, model\_gam\_natural, regression\_train,  
 "Prognozuotos ir tikros reikšmės mokymo aibei (natūralus prieaugis)"  
)

## [1] 181.8344  
## [1] 156.1355  
## [1] "Tiesinis modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.587  
## 2 mae standard 0.470  
## [1] "GAM modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.403  
## 2 mae standard 0.309



regression\_test(  
 natural\_growth, model\_linear\_natural, model\_gam\_natural, test,  
 "Prognozuotos ir tikros reikšmės testavimo aibei (natūralus prieaugis)"  
)

## [1] 181.8344  
## [1] 156.1355  
## [1] "Tiesinis modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.660  
## 2 mae standard 0.574  
## [1] "GAM modelis"  
## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.662  
## 2 mae standard 0.561

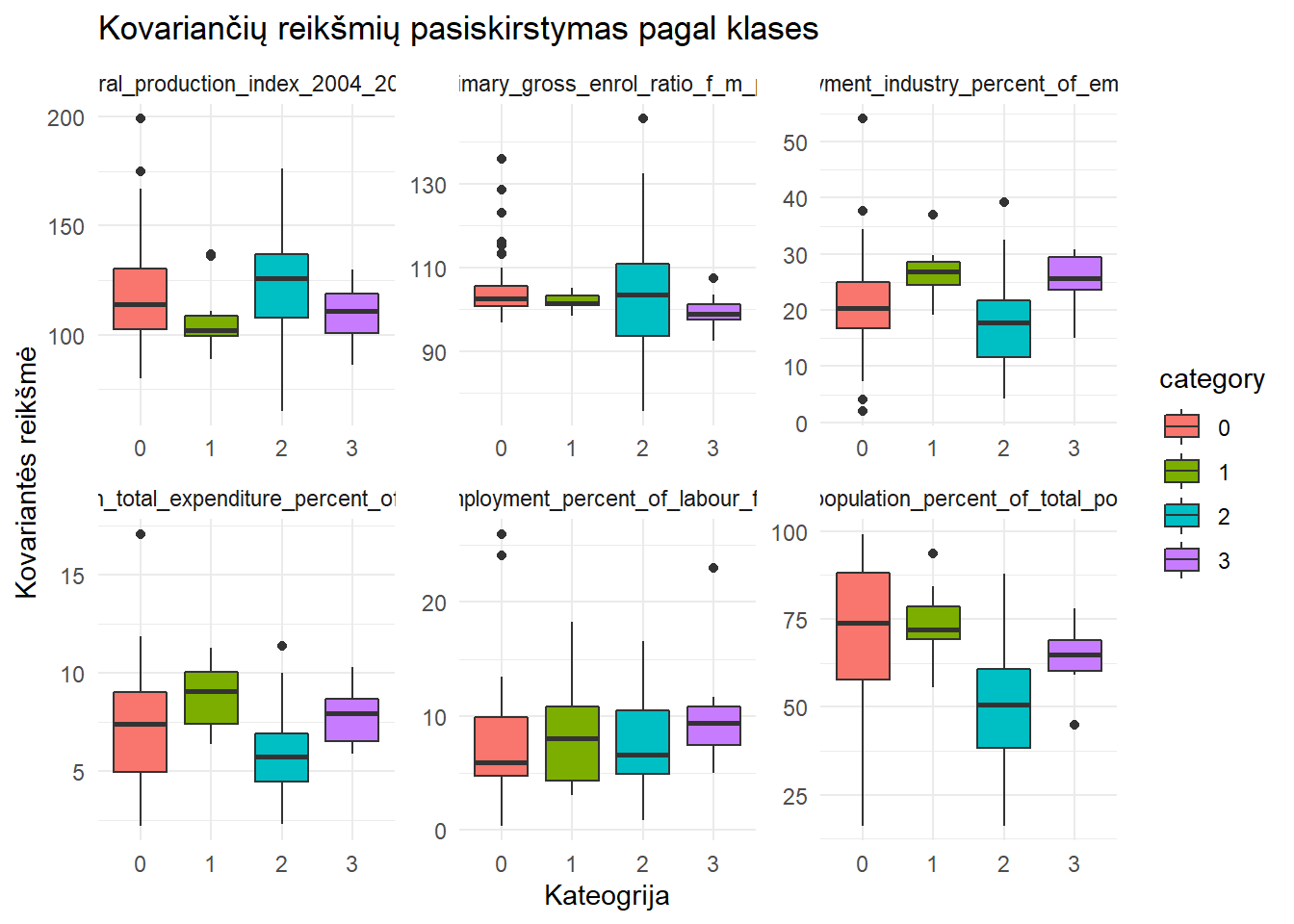


Pasirinkta sudaryti multinominės logistinės regresijos modelį, kuriuo siekiama gauti geresnius rezultatus negu prieš tai naudotais dviem atskirais regresijos modeliais, kai siekiama sužinoti tik kokiai klasei priklauso šalis (ar šalies natūralus/migrantų prieaugiai teigiami ar neigiamai). Akivaizdu, kad kitas šio modelio privalumas yra, kad gauti prognozei reikalingas tik vienas modelis vietoje dviejų.

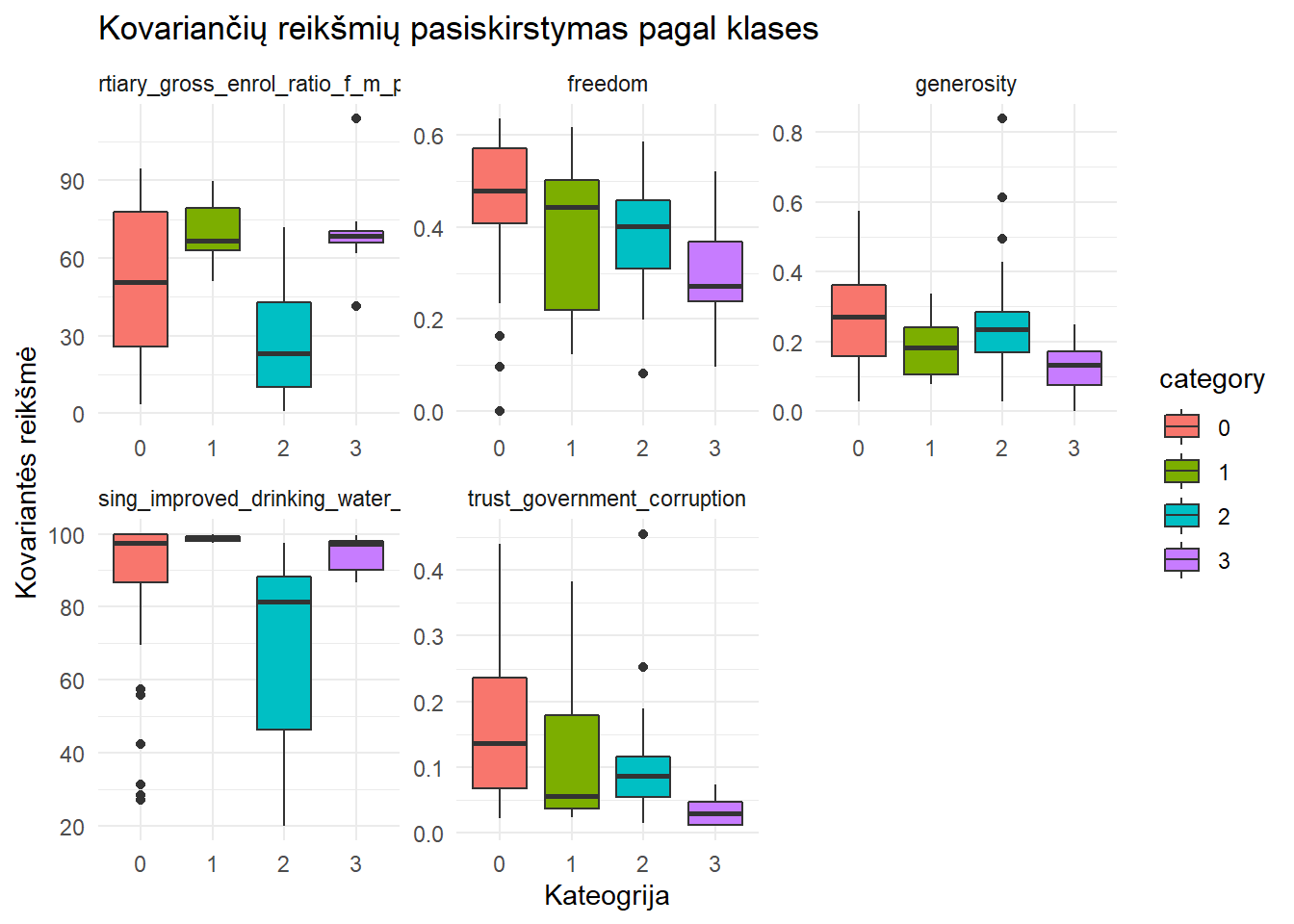
Stačiakampėmis diagramo

classification\_train <- train %>% dplyr::select(-migration\_growth, -natural\_growth)

# Kadangi gautos prastos migracijos prieaugio prognozės, vietoje tikslios   
 # prieaugio reikšmės prognozuojama tik ar tam tikro tipo gyventojų prieaugis teigiamas  
 # ar neigiamas (naudojamos prieš tai sudarytos klasės)  
  
# Stačiakampės diagramos pagal kiekvieną kovariantę  
classification\_train %>%  
 dplyr::select(1:6, category) %>%  
 pivot\_longer(-category) %>%  
 ggplot(aes(x = category, y = value, fill = category)) +  
 facet\_wrap(vars(name), scales = "free") +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(title = "Kovariančių reikšmių pasiskirstymas pagal klases") +   
 xlab("Kateogrija") + ylab("Kovariantės reikšmė")



classification\_train %>%  
 dplyr::select(7:length(classification\_train), category) %>%  
 pivot\_longer(-category) %>%  
 ggplot(aes(x = category, y = value, fill = category)) +  
 facet\_wrap(vars(name), scales = "free") +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(title = "Kovariančių reikšmių pasiskirstymas pagal klases") +  
 xlab("Kateogrija") + ylab("Kovariantės reikšmė")



Dsjfjpofdv

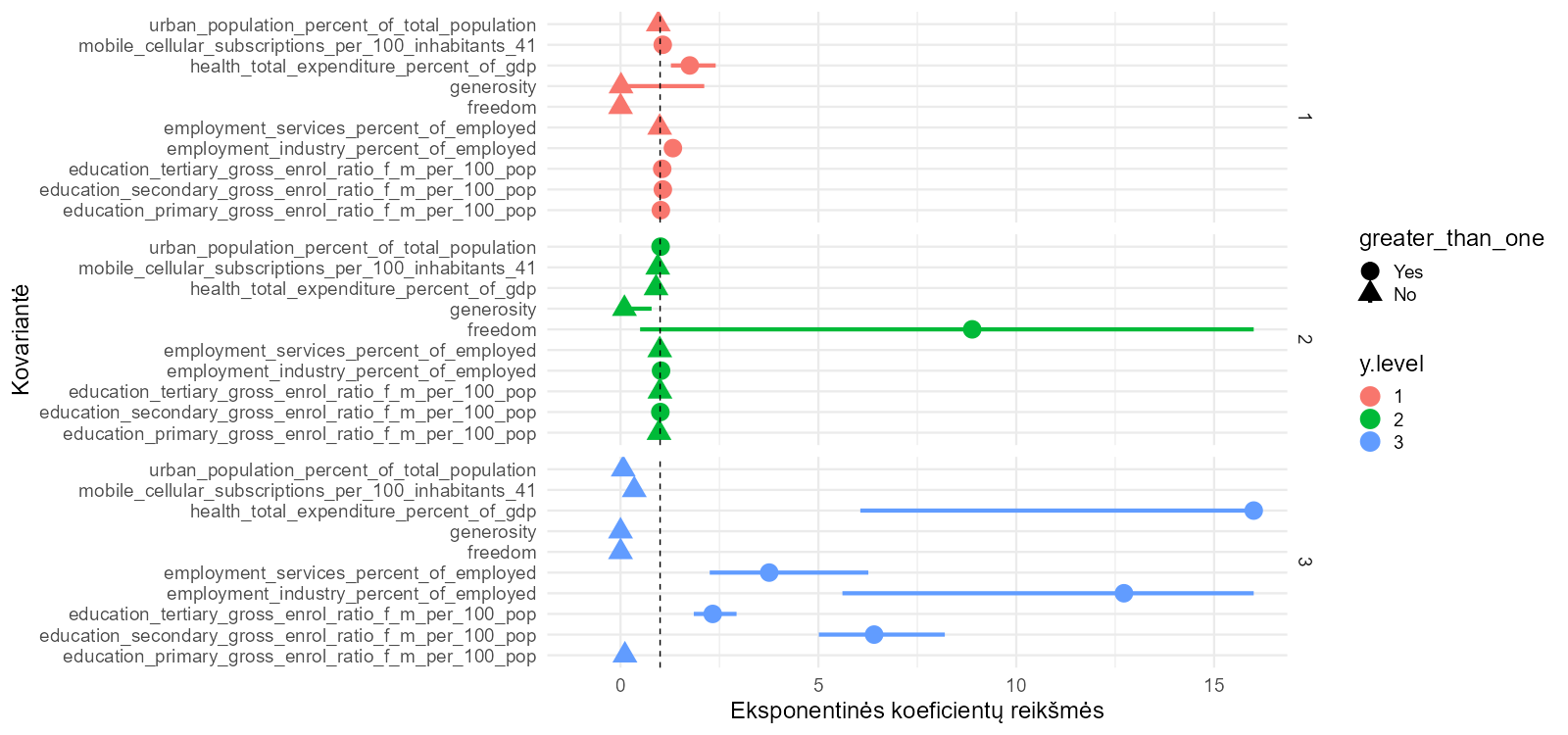
# Pažingsnine regresija sumažintas modelis statistiškai reikšmingai nesiskiria   
anova(model\_logistic, model\_logistic\_small)

## Likelihood ratio tests of Multinomial Models  
##   
## Response: category  
## Model  
## 1 urban\_population\_percent\_of\_total\_population + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## 2 employment\_industry\_percent\_of\_employed + unemployment\_percent\_of\_labour\_force + agricultural\_production\_index\_2004\_2006\_100 + urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp + education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + freedom + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## Resid. df Resid. Dev Test Df LR stat. Pr(Chi)  
## 1 273 145.8441   
## 2 255 125.9399 1 vs 2 18 19.90425 0.3382363

summary(model\_logistic\_small)

## Call:  
## nnet::multinom(formula = category ~ urban\_population\_percent\_of\_total\_population +   
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + generosity +   
## trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban,   
## data = classification\_train, trace = FALSE)  
##   
## Coefficients:  
## (Intercept) urban\_population\_percent\_of\_total\_population  
## 1 -39.8819145 -0.04607011  
## 2 2.2575147 -0.01676782  
## 3 -0.7546367 -0.20816219  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop generosity  
## 1 0.03716542 -8.362663  
## 2 -0.03755632 -1.258363  
## 3 0.14190074 -11.776205  
## trust\_government\_corruption pop\_using\_improved\_drinking\_water\_urban  
## 1 -2.163034 0.427085643  
## 2 -4.428862 0.007482486  
## 3 -41.363126 0.086422684  
##   
## Std. Errors:  
## (Intercept) urban\_population\_percent\_of\_total\_population  
## 1 0.2959585 0.03887555  
## 2 1.1680728 0.01799239  
## 3 7.7821840 0.09635350  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop generosity  
## 1 0.02967933 4.347576  
## 2 0.01890919 1.815050  
## 3 0.06040160 6.360179  
## trust\_government\_corruption pop\_using\_improved\_drinking\_water\_urban  
## 1 4.2367679 0.02636595  
## 2 3.1372343 0.01702100  
## 3 0.2589815 0.08187579  
##   
## Residual Deviance: 145.8441   
## AIC: 181.8441

# multinominės logistinės regresijos modelio koeficientų grafikas  
plot\_coefficients <- function(model) {  
 tidy(model) %>%  
 filter(term != "(Intercept)") %>%  
 mutate(greater\_than\_one = if\_else(estimate > 0, "Yes", "No")) %>%  
 ggplot(aes(term, exp(estimate), color = y.level, shape = greater\_than\_one)) +  
 geom\_pointrange(aes(ymin = exp(estimate - std.error), ymax = exp(estimate + std.error)), ) +  
 scale\_x\_discrete() +  
 coord\_flip() +  
 theme\_minimal() +  
 geom\_hline(yintercept = 1, linetype = "dashed") +  
 scale\_y\_continuous(oob = scales::squish, limits = c(-1, 16)) +  
 facet\_grid(cols = vars(y.level), scales = "free") +  
 labs(x = "Kovariantė", y = "Eksponentinės koeficientų reikšmės")  
}  
  
plot\_coefficients(model\_logistic\_small)



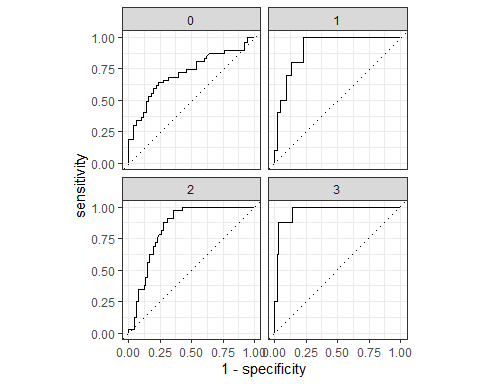
Klasifikavimo modelio kokybė naudojant maišos matricas (angl. confusion matrices), bendrą tikslumą (angl. accuracy), F-score. Kadangi turimas daugelio klasių (multiclass) uždavinys paskutinės minėtos modelio kokybės vertinimo metrikos bendra reikšmė gauta „macro“ vidurkinimą imant metrikos reikšmių visoms 4 klasėms vidurkį (taip kiekvienai klasei priskiriant tokį pat svorį). Taip pat nubraižytos ROC kreivės ir apskaičiuota AUC.

# pačių sudaryta modelio kokybės metrika, kuri "pataiso" bendrą tikslumą  
# priskirdama 0.5 - jeigu teisingai prognozuotas vieno tipo prieaugis  
# 1 - jeigu teisingi abu prieaugiai  
# 0 - jeigu abiejų tipų prieaugiai neteisingi  
  
custom\_metric <- function(y\_true, y\_pred) {  
 c(  
 "custom\_metric", "multiclass",  
 case\_when(  
 y\_true %in% c(0, 3) & y\_pred %in% c(1, 2) ~ 0.5,  
 y\_true %in% c(1, 2) & y\_pred %in% c(0, 3) ~ 0.5,  
 y\_true == y\_pred ~ 1,  
 TRUE ~ 0  
 ) %>%  
 mean()  
 )  
}  
  
# Multinominės logistinės regresijos modelio įvertinimas  
classification\_eval <- function(model, data) {  
 df\_pred\_truth <- tibble(  
 predicted = factor(predict(model, data)),  
 truth = data$category  
 ) %>%  
 cbind(as.data.frame(model$fitted.values))  
  
  
  
 classification\_metrics <- metric\_set(accuracy, j\_index, f\_meas)  
  
 print("Maišos matrica")  
 print(conf\_mat(df\_pred\_truth,  
 truth = truth,  
 estimate = predicted  
 ))  
  
 print("Modelio kokybės metrikos")  
 print(classification\_metrics(df\_pred\_truth,  
 truth = truth,  
 estimate = predicted  
 ) %>%  
 rbind(custom\_metric(df\_pred\_truth$truth, df\_pred\_truth$predicted)))  
  
  
  
 print(roc\_auc(df\_pred\_truth, truth = truth, c("0", "1", "2", "3"), estimator = "macro"))  
  
 roc\_curve(df\_pred\_truth, truth = truth, c("0", "1", "2", "3")) %>%  
 autoplot()  
}  
  
print("Pradinis multinominės logistinės regresijos modelis")

## [1] "Pradinis multinominės logistinės regresijos modelis"

classification\_eval(model\_logistic\_small, classification\_train)

## [1] "Maišos matrica"  
## Truth  
## Prediction 0 1 2 3  
## 0 32 4 10 1  
## 1 2 4 0 0  
## 2 12 0 22 0  
## 3 1 2 0 7  
## [1] "Modelio kokybės metrikos"  
## # A tibble: 4 x 3  
## .metric .estimator .estimate   
## <chr> <chr> <chr>   
## 1 accuracy multiclass 0.670103092783505  
## 2 j\_index macro 0.525509827074684  
## 3 f\_meas macro 0.656323877068558  
## 4 custom\_metric multiclass 0.824742268041237  
## # A tibble: 1 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 roc\_auc macro 0.856



Kadangi duomenų rinkinyje turimos stipriai išbalansuotos klasės (klasės “1” - teigiamas migracijos prieaugis, neigiamas natūralus ir „3“ - neigiami abiejų tipų prieaugiai, sudaro mažiau nei po 5% stebėjimų), pasirinkta šią problemą spręsti sugeneruojant dirbtinų stebėjimų mažumos klasėms naudojant SMOTE algoritmą.

Naudojant testavimo duomenų aibę palyginti rezultatai gauti naudojant paprastą multinominį logistinės regresijos modelį, modelį gautą naudojant SMOTE algoritmu sugeneruotus dirbtinius stebėjimus ir klasifikavimui pritaikant du regresijos modelius.

Tiek pagal maišos matricą, tiek pagal modelio kokybės metrikas testavimo aibėje matomi geresni antrojo modelio rezultatai (modelio, kuriam apmokyti buvo naudojami SMOTE algoritmu sugeneruoti stebėjimai). Kaip ir iš anksto tikėtasi, klasifikavimui panaudojus du prieš tai sudarytus regresijos modelius gauti prasčiausi rezultatai.

# Turimas ne itin ryškus klasių išbalasavimas  
# (daugumos klasės stebėjimų beveik 6 kartus daugiau nei mažiausios)   
# Todėl rezultatai gali pagerėti sugeneravus dirbtinius papildomus stebėjimus  
classification\_train %>% count(category)

## # A tibble: 4 x 2  
## category n  
## <fct> <int>  
## 1 0 47  
## 2 1 10  
## 3 2 32  
## 4 3 8

library(themis)  
  
smote\_recipe <- recipe(category ~ .,  
 data = classification\_train  
) %>%  
 step\_smote(category, over\_ratio = 1)  
  
  
smote\_recipe <- prep(smote\_recipe, training = classification\_train)  
  
classification\_train2 <- bake(smote\_recipe, NULL)

model\_logistic2 <- nnet::multinom(category ~ ., data = classification\_train2, trace = FALSE)  
  
model\_logistic2\_small <- stats::step(model\_logistic2, direction = "both")

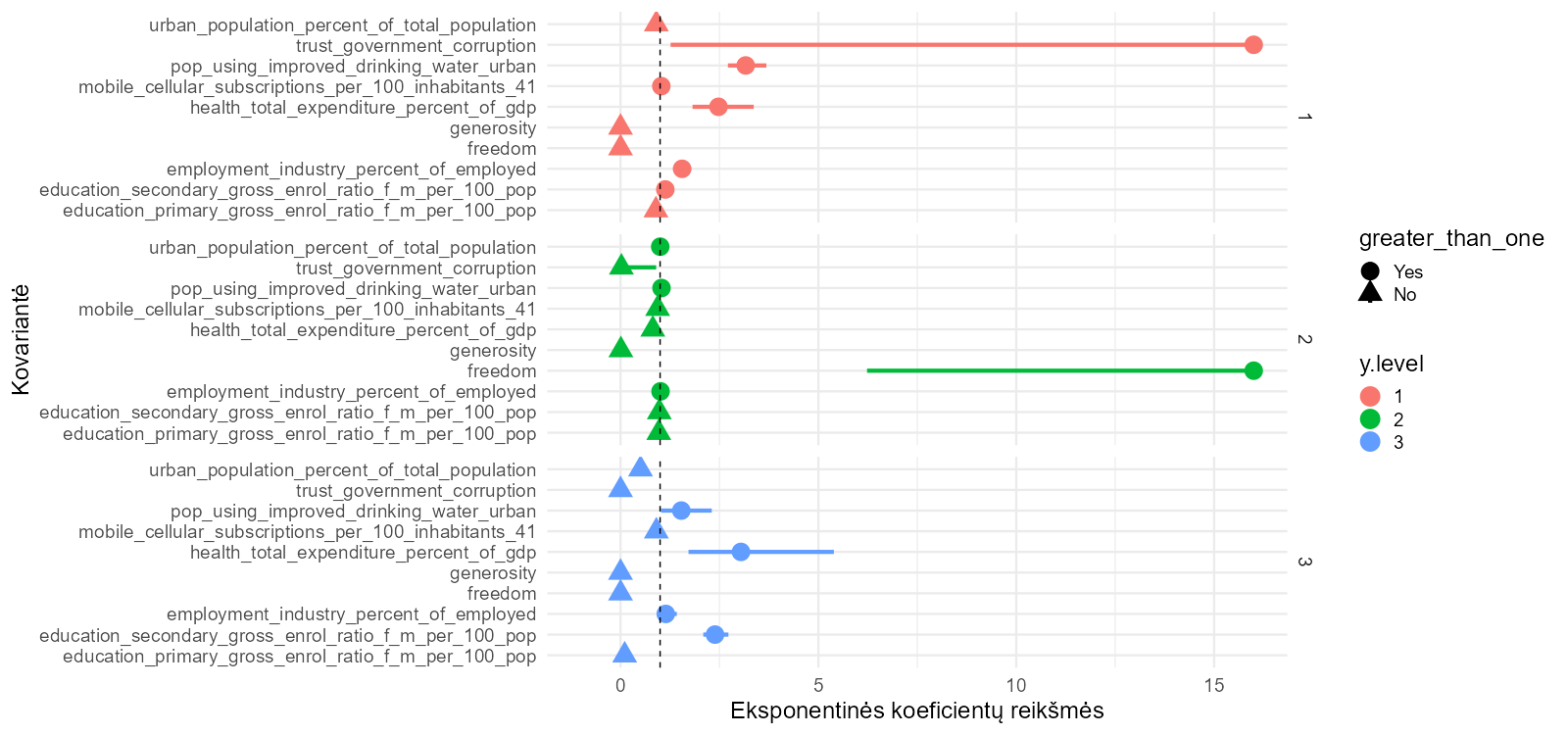
anova(model\_logistic2, model\_logistic2\_small)

## Likelihood ratio tests of Multinomial Models  
##   
## Response: category  
## Model  
## 1 employment\_industry\_percent\_of\_employed + urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp + education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + freedom + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## 2 employment\_industry\_percent\_of\_employed + unemployment\_percent\_of\_labour\_force + agricultural\_production\_index\_2004\_2006\_100 + urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp + education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + freedom + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban  
## Resid. df Resid. Dev Test Df LR stat. Pr(Chi)  
## 1 534 180.2015   
## 2 528 175.6232 1 vs 2 6 4.578352 0.5989112

summary(model\_logistic2\_small)

## Call:  
## nnet::multinom(formula = category ~ employment\_industry\_percent\_of\_employed +   
## urban\_population\_percent\_of\_total\_population + health\_total\_expenditure\_percent\_of\_gdp +   
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop + education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop +   
## freedom + generosity + trust\_government\_corruption + pop\_using\_improved\_drinking\_water\_urban,   
## data = classification\_train2, trace = FALSE)  
##   
## Coefficients:  
## (Intercept) employment\_industry\_percent\_of\_employed  
## 1 -82.56939 0.4258807  
## 2 5.70080 -0.0116691  
## 3 92.81018 -0.2041168  
## urban\_population\_percent\_of\_total\_population  
## 1 -0.05943872  
## 2 -0.02051866  
## 3 -0.51990696  
## health\_total\_expenditure\_percent\_of\_gdp  
## 1 0.74220698  
## 2 -0.19739614  
## 3 0.03267364  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop  
## 1 -0.04120533  
## 2 -0.01072513  
## 3 -1.04980535  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop freedom generosity  
## 1 0.12485108 -13.934443 -20.910237  
## 2 -0.03129278 -1.030607 -2.348438  
## 3 0.39886697 14.382368 -31.583264  
## trust\_government\_corruption pop\_using\_improved\_drinking\_water\_urban  
## 1 12.028641 0.754354905  
## 2 -3.201743 0.008499684  
## 3 -83.873611 0.326507763  
##   
## Std. Errors:  
## (Intercept) employment\_industry\_percent\_of\_employed  
## 1 0.9382478 0.12418061  
## 2 3.0079846 0.04331525  
## 3 2.5050944 0.11138284  
## urban\_population\_percent\_of\_total\_population  
## 1 0.04737429  
## 2 0.01952951  
## 3 0.09579285  
## health\_total\_expenditure\_percent\_of\_gdp  
## 1 0.2687616  
## 2 0.1428908  
## 3 0.4182192  
## education\_primary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop  
## 1 0.09746081  
## 2 0.02512582  
## 3 0.10697038  
## education\_tertiary\_gross\_enrol\_ratio\_f\_m\_per\_100\_pop freedom generosity  
## 1 0.04534927 6.080630 6.239774  
## 2 0.01963119 2.465082 1.871969  
## 3 0.06976496 6.006716 7.726796  
## trust\_government\_corruption pop\_using\_improved\_drinking\_water\_urban  
## 1 7.627677 0.13619649  
## 2 3.019814 0.01851975  
## 3 2.069105 0.11631511  
##   
## Residual Deviance: 180.2015   
## AIC: 240.2015

plot\_coefficients(model\_logistic2\_small)



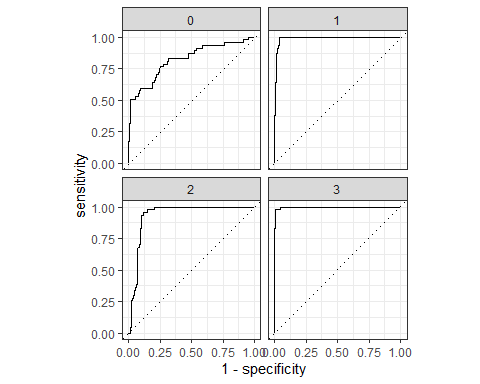
Saudhfikod

print("Multinominės logistinės regresijos modelis su SMOTE")

## [1] "Multinominės logistinės regresijos modelis su SMOTE"

classification\_eval(model\_logistic2\_small, classification\_train2)

## [1] "Maišos matrica"  
## Truth  
## Prediction 0 1 2 3  
## 0 26 0 10 1  
## 1 6 47 0 0  
## 2 13 0 37 0  
## 3 2 0 0 46  
## [1] "Modelio kokybės metrikos"  
## # A tibble: 4 x 3  
## .metric .estimator .estimate   
## <chr> <chr> <chr>   
## 1 accuracy multiclass 0.829787234042553  
## 2 j\_index macro 0.773049645390071  
## 3 f\_meas macro 0.822588817404336  
## 4 custom\_metric multiclass 0.906914893617021  
## # A tibble: 1 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 roc\_auc macro 0.936



sxaidjfodfg

# apskaičiuoja kiek kartų teisingai prognozuotas kiekvieno tipo prieaugis  
# (lengviau interpretuoti negu maišos matricą)  
custom\_confusion <- function(y\_true, y\_pred) {  
 case\_when(  
 y\_true == y\_pred ~ "Correct both",  
 (y\_true %in% c(0, 2) & y\_pred %in% c(0, 2)) | (y\_true %in% c(1, 3) & y\_pred %in% c(1, 3)) ~ "Correct natural",  
 (y\_true %in% c(0, 1) & y\_pred %in% c(0, 1)) | (y\_true %in% c(2, 3) & y\_pred %in% c(2, 3)) ~ "Correct migration",  
 TRUE ~ "Correct none"  
 ) %>%  
 tibble(results = .) %>%  
 count(results)  
}  
  
  
# palyginimui jeigu butų naudojami prieš tai sudaryti 2 regresijos modeliai prognozuoti klases  
class\_predictions <- function() {  
 tibble(  
 migration\_growth = predict(model\_gam\_migration, test),  
 natural\_growth = predict(model\_gam\_natural, test)  
 ) %>%  
 mutate(category = factor(case\_when(  
 migration\_growth >= 0 & natural\_growth >= 0 ~ 0, # "P migration, P natural",  
 migration\_growth >= 0 & natural\_growth < 0 ~ 1, # "P migration, N natural",  
 migration\_growth < 0 & natural\_growth >= 0 ~ 2, # "N migration, P natural",  
 TRUE ~ 3  
 ))) %>%  
 pull(category)  
}  
  
  
classification\_test <- function(model, data, name) {  
 df\_pred\_truth <- tibble(truth = data$category)  
 classification\_metrics <- metric\_set(accuracy, j\_index, f\_meas)  
  
 if (name == "Naudojant du regresijos modelius") {  
 df\_pred\_truth$predicted <- factor(class\_predictions(), levels = c(0, 1, 2, 3))  
 } else {  
 df\_pred\_truth$predicted <- factor(predict(model, test), levels = c(0, 1, 2, 3))  
 }  
  
  
 print("Maišos matrica")  
 conf\_mat(df\_pred\_truth,  
 truth = truth,  
 estimate = predicted  
 ) %>% print()  
  
  
 print(custom\_confusion(df\_pred\_truth$truth, df\_pred\_truth$predicted))  
  
 print("Modelio kokybės metrikos")  
 classification\_metrics(df\_pred\_truth, truth, estimate = predicted) %>%  
 rbind(custom\_metric(df\_pred\_truth$truth, df\_pred\_truth$predicted)) %>%  
 print()  
  
 cat("\n\n")  
}

# Naudojant testavimo aibę.  
# Geriausi rezultatai gauti su modeliu, kuriam naudotas SMOTE algoritmas  
# blogiausi - panaudojus regresijos modelius  
  
# Vėl matoma, kad geresni rezultatai gaunami prognozuojant natūralų gyventojų prieaugį  
print("Naudojant pradinį multinominės logistinės regresijos modelį")

## [1] "Naudojant pradinį multinominės logistinės regresijos modelį"

classification\_test(model\_logistic, test, "Pradinis")

## [1] "Maišos matrica"  
## Truth  
## Prediction 0 1 2 3  
## 0 2 1 2 0  
## 1 0 0 0 0  
## 2 5 0 12 0  
## 3 0 1 1 1  
## # A tibble: 3 x 2  
## results n  
## <chr> <int>  
## 1 Correct both 15  
## 2 Correct migration 2  
## 3 Correct natural 8  
## [1] "Modelio kokybės metrikos"  
## # A tibble: 4 x 3  
## .metric .estimator .estimate   
## <chr> <chr> <chr>   
## 1 accuracy multiclass 0.6   
## 2 j\_index macro 0.333928571428571  
## 3 f\_meas macro 0.527777777777778  
## 4 custom\_metric multiclass 0.8

print("Naudojant multinomės logistinės regresijos modelį su SMOTE")

## [1] "Naudojant multinomės logistinės regresijos modelį su SMOTE"

classification\_test(model\_logistic2\_small, test, "SMOTE")

## [1] "Maišos matrica"  
## Truth  
## Prediction 0 1 2 3  
## 0 2 0 2 0  
## 1 0 1 0 0  
## 2 4 0 13 0  
## 3 1 1 0 1  
## # A tibble: 3 x 2  
## results n  
## <chr> <int>  
## 1 Correct both 17  
## 2 Correct natural 7  
## 3 Correct none 1  
## [1] "Modelio kokybės metrikos"  
## # A tibble: 4 x 3  
## .metric .estimator .estimate   
## <chr> <chr> <chr>   
## 1 accuracy multiclass 0.68   
## 2 j\_index macro 0.514484126984127  
## 3 f\_meas macro 0.585700757575758  
## 4 custom\_metric multiclass 0.82

classification\_test(model\_logistic, test, "Naudojant du regresijos modelius")

## [1] "Maišos matrica"  
## Truth  
## Prediction 0 1 2 3  
## 0 4 1 8 1  
## 1 0 1 0 0  
## 2 3 0 7 0  
## 3 0 0 0 0  
## # A tibble: 4 x 2  
## results n  
## <chr> <int>  
## 1 Correct both 12  
## 2 Correct migration 1  
## 3 Correct natural 11  
## 4 Correct none 1  
## [1] "Modelio kokybės metrikos"  
## # A tibble: 4 x 3  
## .metric .estimator .estimate   
## <chr> <chr> <chr>   
## 1 accuracy multiclass 0.48   
## 2 j\_index macro 0.170634920634921  
## 3 f\_meas macro 0.535873015873016  
## 4 custom\_metric multiclass 0.72

## Išvados

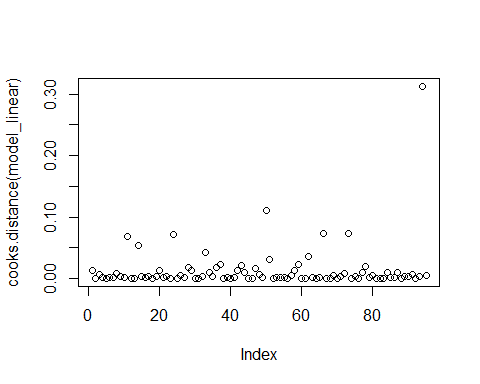
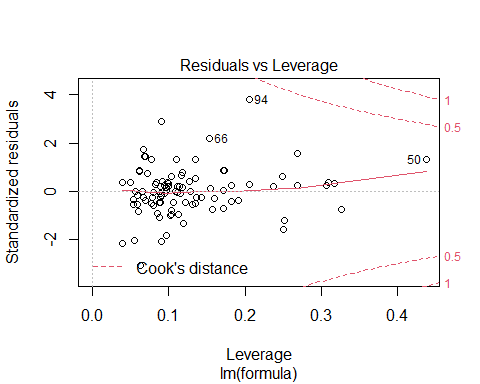
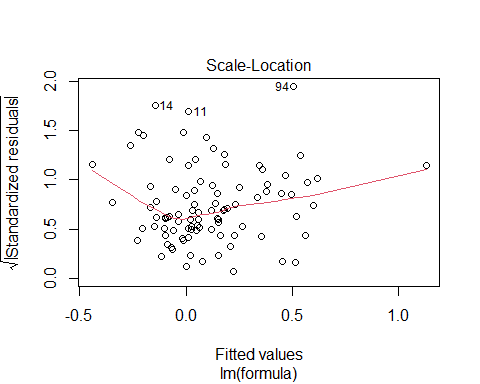
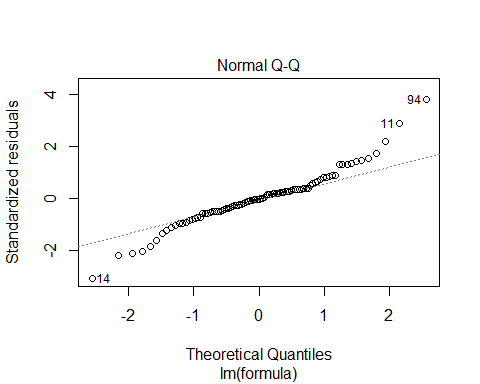
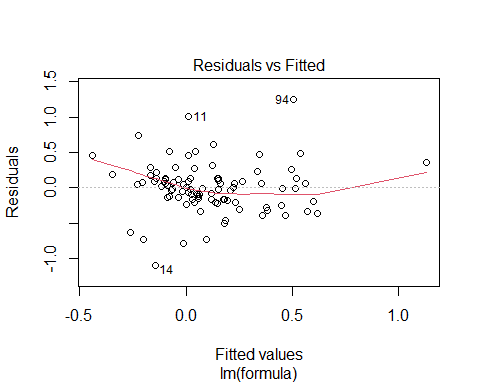
Atlikus pirminę duomenų aibės analizę rasta, kad betono stipris gali netiesiškai priklausyti nuo jo mišinį sudarančių medžiagų, todėl laikyta, kad paprastas tiesinės regresijos modelis gali būti netinkamas prognozuoti betono stiprį.

Siekiant turėti palyginamąjį modelį sudarytas paprastas tiesinės regresijos modelis. Kaip vienas iš būdų pagerinti modelį buvo pasirinkta polinominė regresija. Į modelį buvo įtraukti antrieji laipsniai tų kintamųjų, kuriuose pagal tiesinės regresijos diagnostinius grafikus rasta neišnaudotos informacijos. Pasitelkiant grafines modelių diagnostikas ir statistinius testus laikyta, kad antru modeliu gaunami geresni rezultatai.

Siekiant dar labiau pagerinti rezultatus (ir parodyti, kad aukštesnių laipsnių įtraukimas į paprasta tiesinį modelį nėra pakankamas tikslioms prognozėms gauti) sudarytas apibendrintas adityvus regresijos modelis, naudojantis glodniuosius splainus. Pirmiausia naudotas numatytasis mazgų skaičius, tačiau naudojant modelio diagnostikas pasirinkta šį skaičių padidinti. Parinkus didesnį mazgų skaičių modelio diagnostikose beveik nerasta nukrypimų. Visos modelyje naudotos kovariantės modelyje statistiškai reikšmingos.

Testavimo aibė panaudota įvertinti visų šių modelių gebėjimą prognozuoti reikšmes. Kaip ir tikėtasi, geriausi rezultatai gauti naudojant glodniųjų splainų modelį (paklaidos pagal dvi skirtingas matavimo metrikas daugiau nei dvigubai mažesnės už tiesinį ir apie 50% mažesnės už polinominį modelį). Dėl šios priežasties daroma išvada, kad sudarytas glodniųjų splainų apibendrintas adityvus tinkamas prognozuoti betono stiprį, be to šiuo modeliu gaunami ryškiai geresni rezultatai negu naudojant paprastesnį metodą netiesiniams sąryšiams tarp atsako ir kovariančių modeliuoti, kuriuo paprastas tiesinės regresijos modelis papildomas kovariančių aukštesniais laipsniais.

# 1 Priedas. Tiesinio modelio migracijos prieaugiui diagnostiniai grafikai.



# 2 Priedas. Tiesinio modelio natūraliam prieaugiui diagnostiniai grafikai.

