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

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

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" – 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" - gyventojų, lankančių pradinį mokslą skaičius 100 gyventojų.

"education_tertiary_gross_enrol_ratio" – - gyventojų, lankančių aukštajį (ar kitą trečio lygio) mokslą 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 skirtingiems gyventojų prieaugio tipams 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))
```

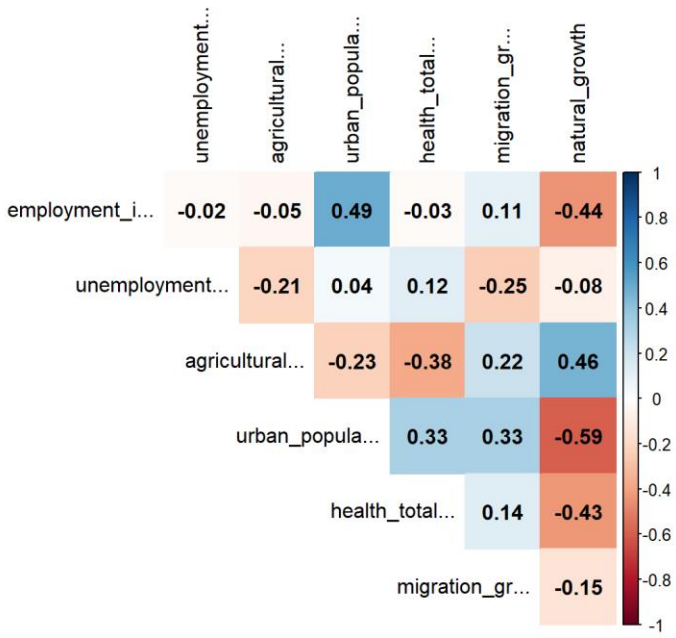
```r
x <- reduce(list(pop_natural, pop_total, country_stats, happiness), inner_join, by =
"country")


# 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",
```

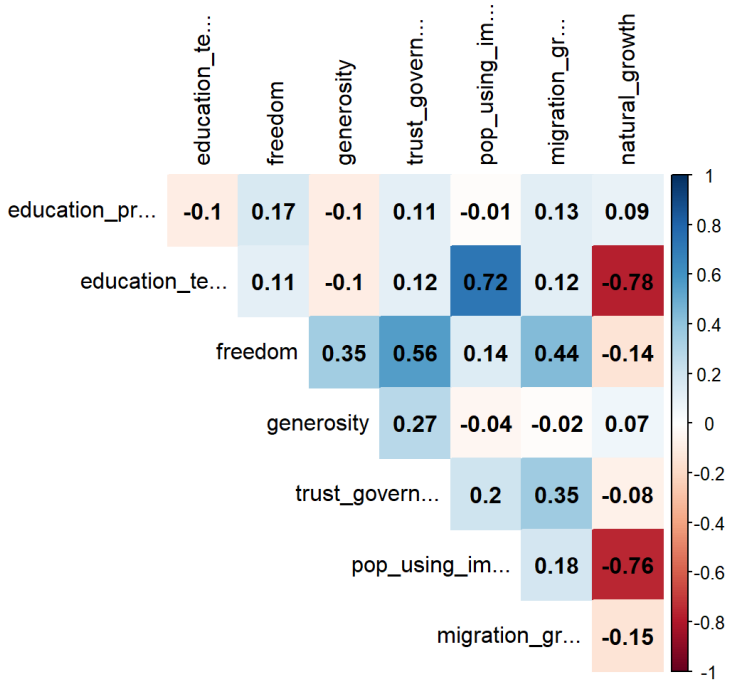
```
    migration_growth >= 0 & natural_growth < 0 ~ 1, # "P migation, N natural",
    migration_growth < 0 & natural_growth >= 0 ~ 2, # "N migation, P natural",
    TRUE ~ 3 # "N migration, N natural"
  ))
)


# 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.

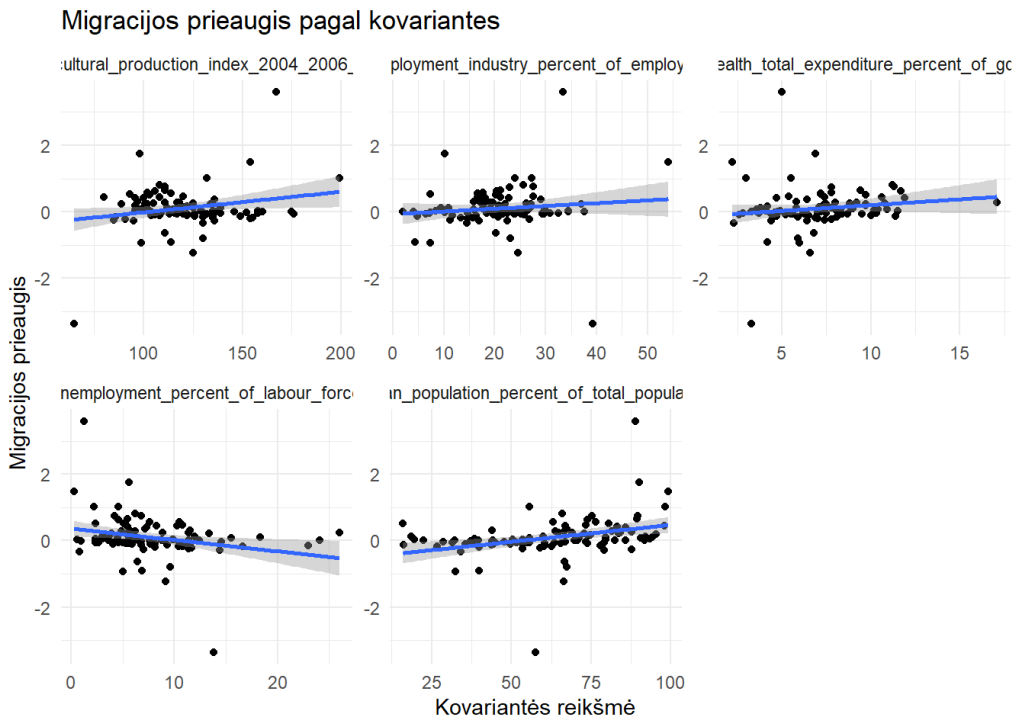
```
library(corrplot)

# 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 %>%
```

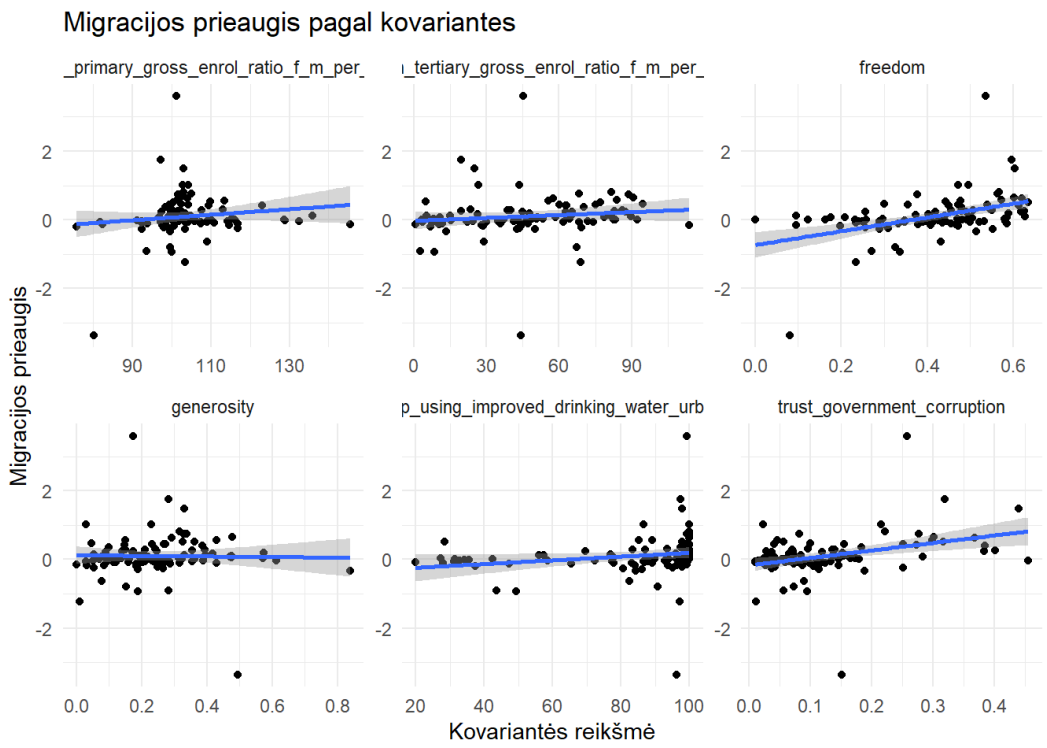
```
    dplyr::select(6:length(regression_train), {{ name }}, {{ name2 }}) %>%
    set_names(., str_trunc(names(.), 15)) %>%
    cor()

  corrplot(correlation_matrix, order = "original", method = "color", type = "upper", diag =
FALSE, tl.col = "black", addCoef.col = "black")
}


correlation(migration_growth, natural_growth)
```



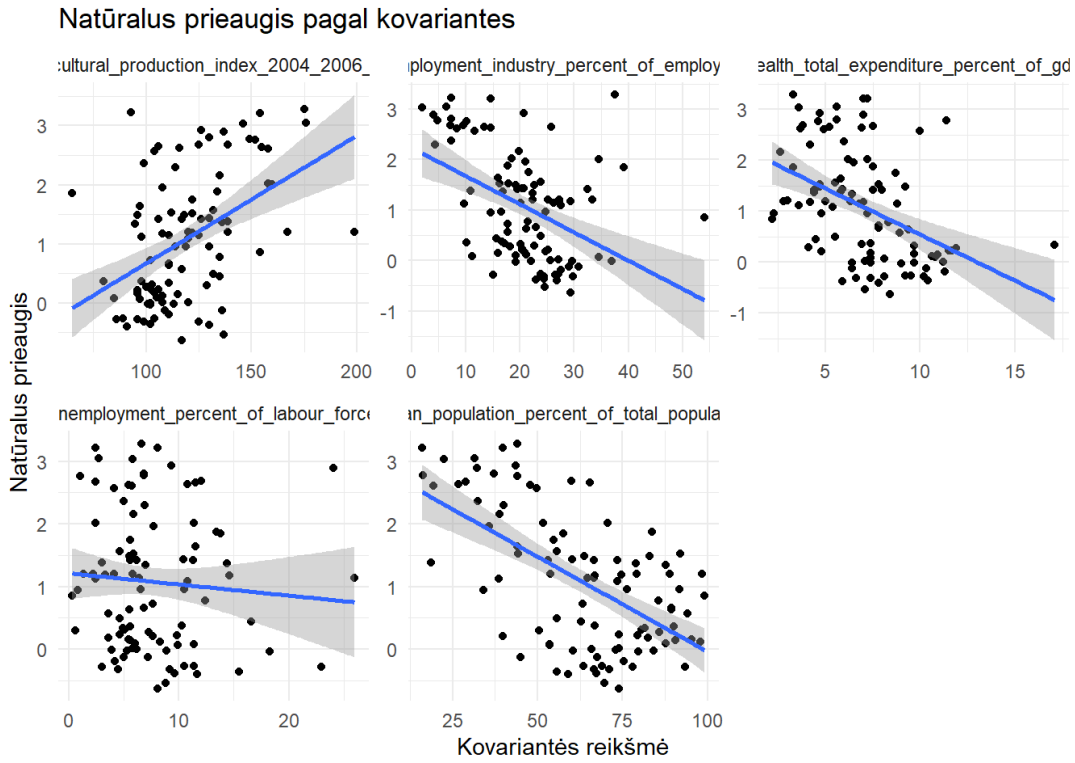


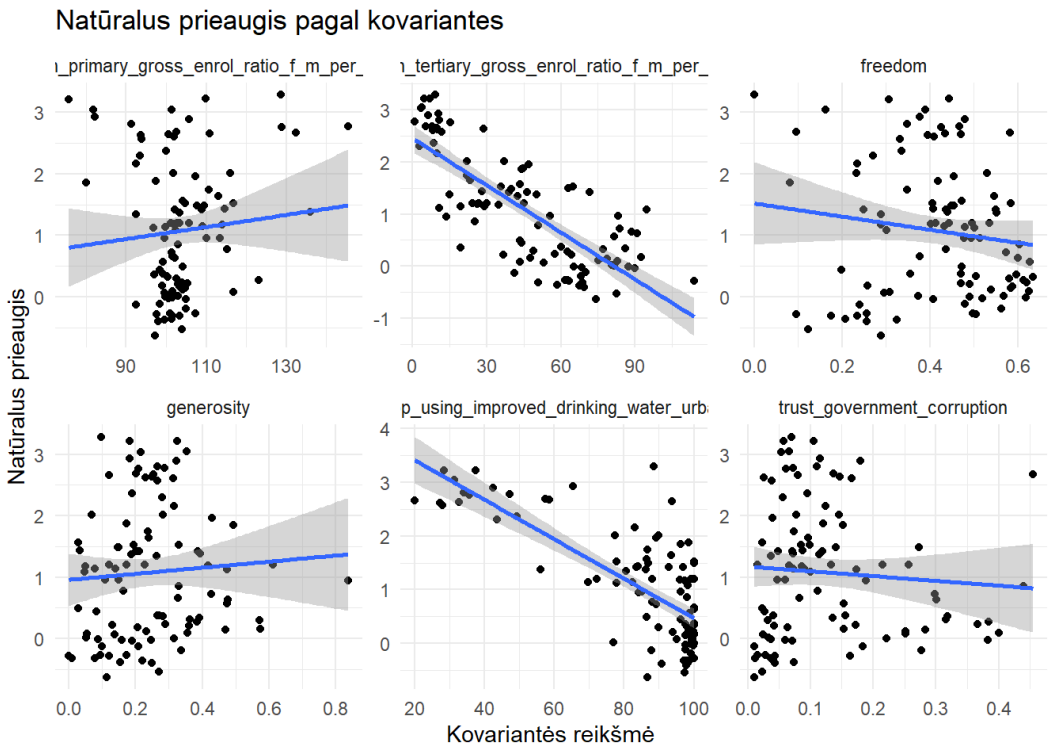
```r
# sklaidos diagramos su kiekviena kovariante
scatterplot <- function(name, name2, main, ylab) {
  a <- regression_train %>%
    dplyr::select(1:5, {{ name }}, -{{ name2 }}) %>%
    pivot_longer(-{{ name }}) %>%
    ggplot(aes(x = value, y = {{ name }})) +
    facet_wrap(vars(name), scales = "free") +
    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")
```



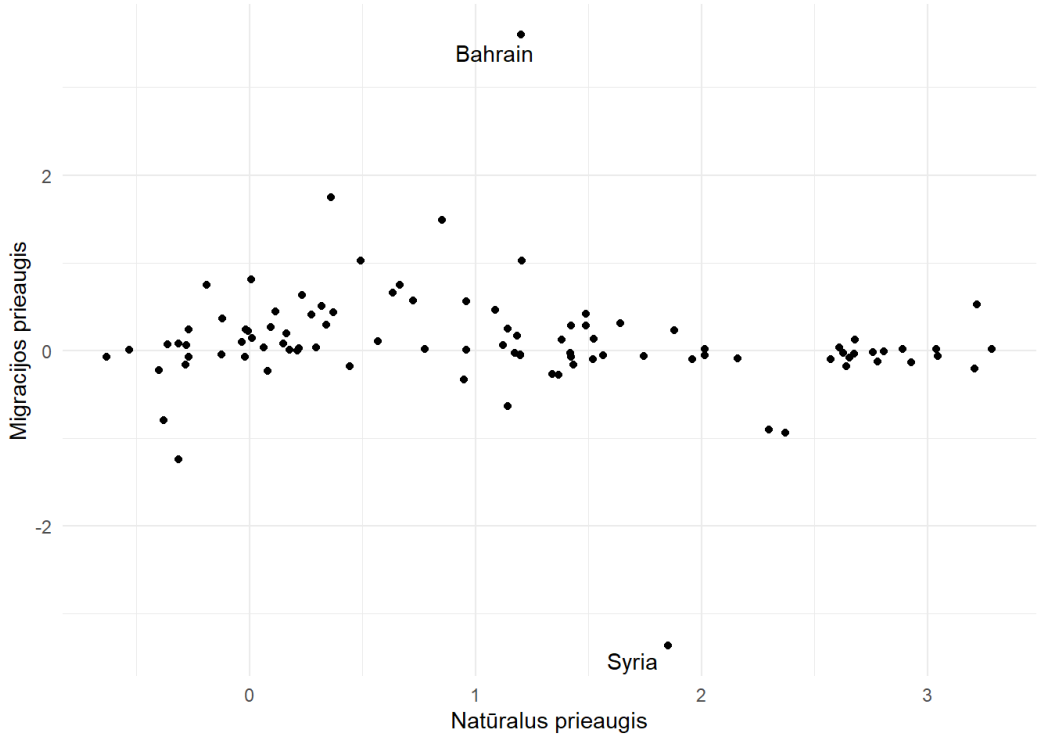



```
scatterplot(natural_growth, migration_growth, "Natūralus prieaugis pagal
kovariantes","Natūralus prieaugis")
```






```
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 prieaugiui.

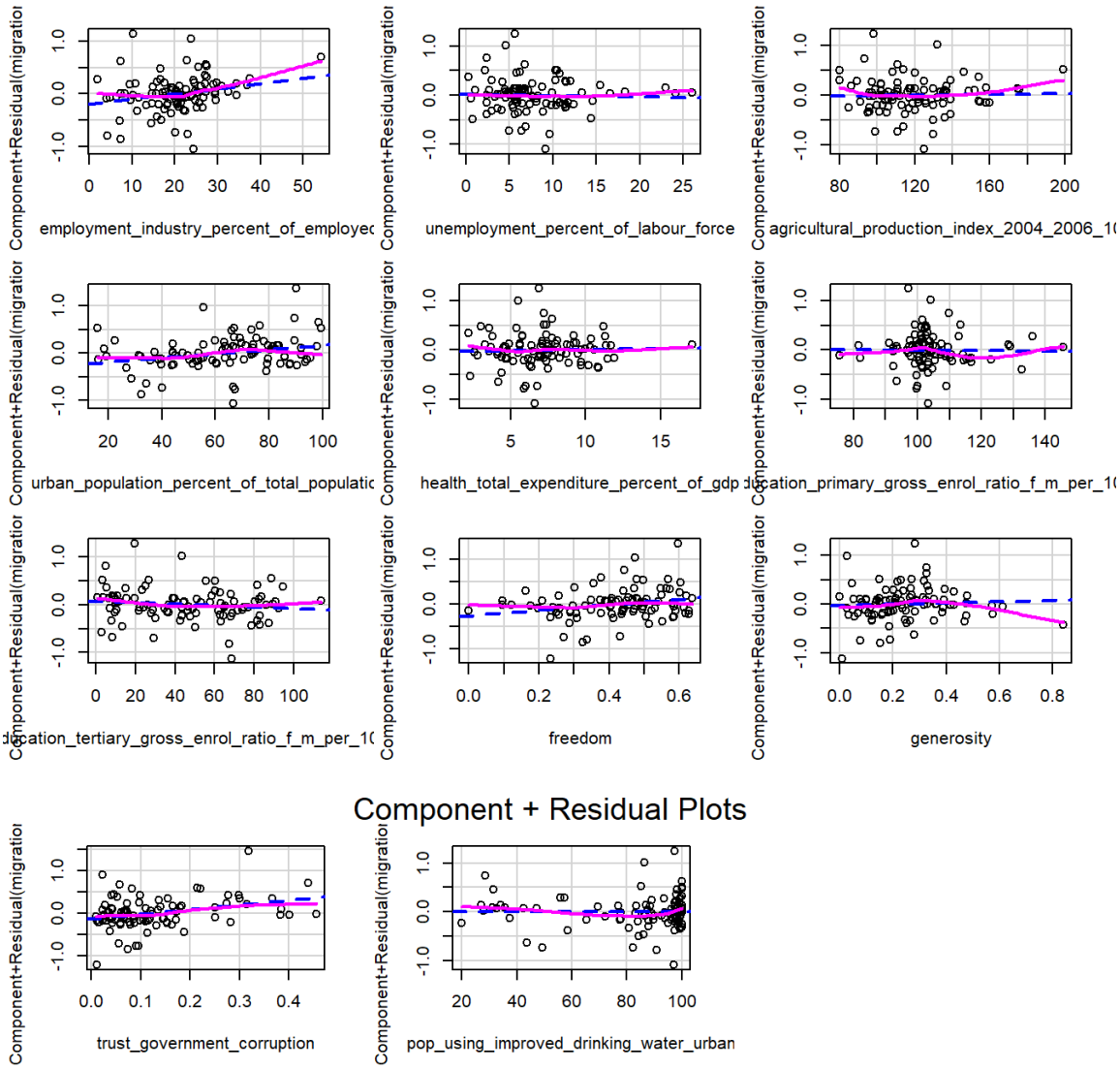
Migracijos prieaugiui pradžiai sudarytas modelis, naudojantis visas kovariantes. Pažingsnine regresija sumažintas modelis statistiškai reikšmingai nesiskyrė nuo pilno, naudojančio visas kovariantes ($p = 0.99$). Sumažintą modelį sudaro kovariantės “employment_industry_percent_of_employed“ ($\beta = 0.010$, $p = 0.07$), “urban_population_percent_of_total_population“ ($\beta = 0.003$, $p = 0.16$), “freedom” ($\beta = 0.70$, $p = 0.03$) ir “trust_government_corruption” ($\beta = 1.18$, $p < 0.01$). Šį modelį interpretuojant galima teigti, kad didesnė dalis dirbanti industrijos sektoriuje ir gyvenanti miestuose teigiamai įtakoja migraciją į šalį. Teigiamai migraciją taip pat įtakoja didesnė asmeninė laisvė šalyje, didesnis pasitikėjimas vyriausybe.

Papildomai pateikti standartizuoti koeficientai, kurie atsižvelgia į kovariančių matavimo skalių skirtumus. Iš gautų rezultatų matome, kad didžiausia įtaką teigiamai migracijai daro kovariantės “freedom” ir “trust_government_corruption”.

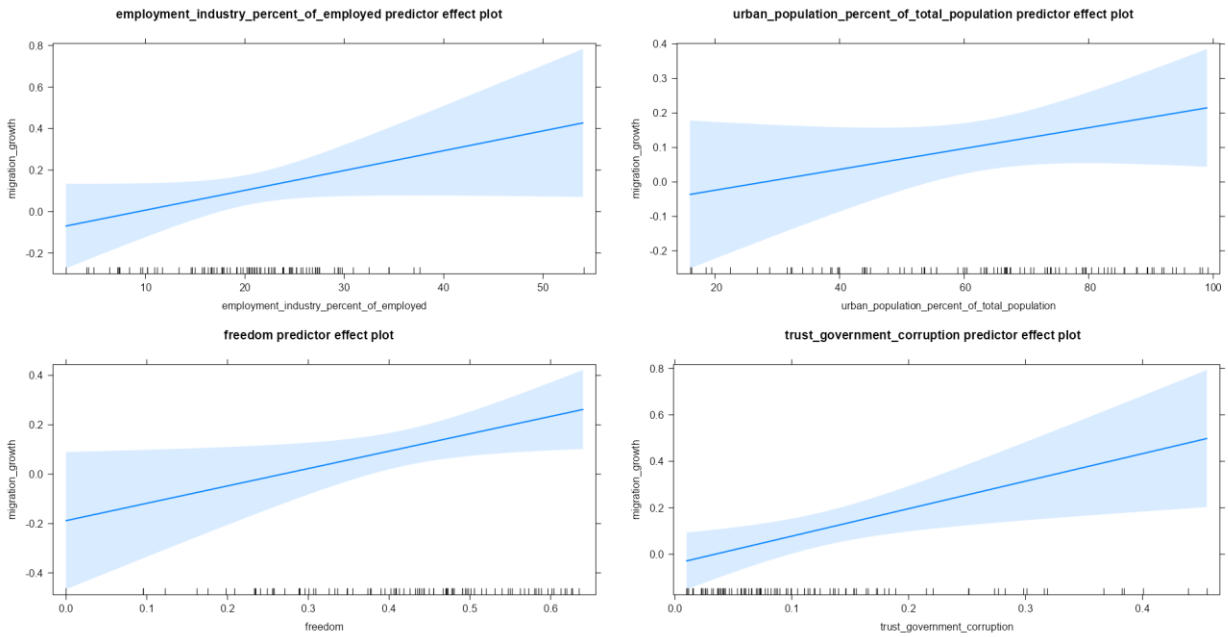
Tiesa, šis modelis paaiškina tik maža dalį atsako dispersijos (R-adj = 0.31), todėl abejotina aš šis modelis tinkamas prognozuoti migracijos prieaugį.

```r
# 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
}
```

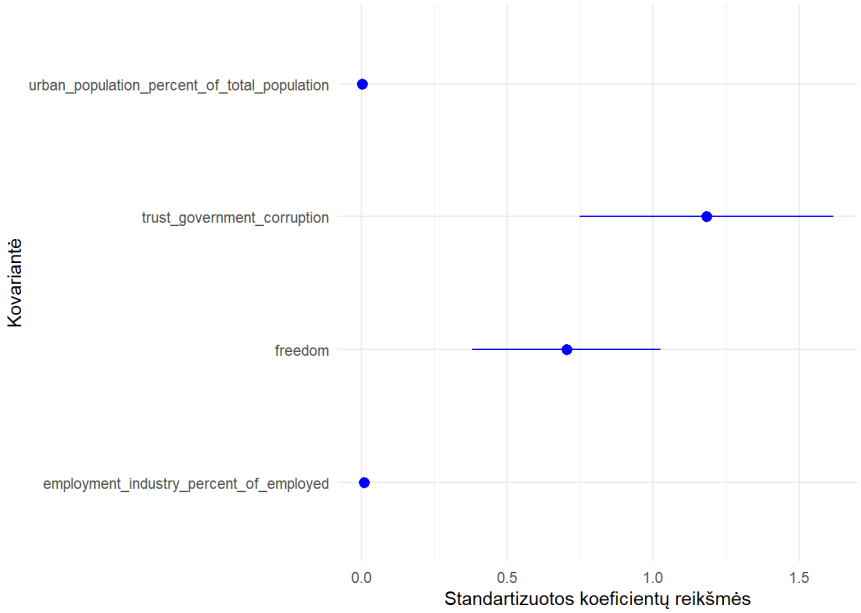
```r
# Atskirai apmokomi modeliai migracijos ir natūraliam prieaugiui
print("Tiesinės regresijos modelis migracijos prieaugiui")

## [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
```




```
##
## 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
```

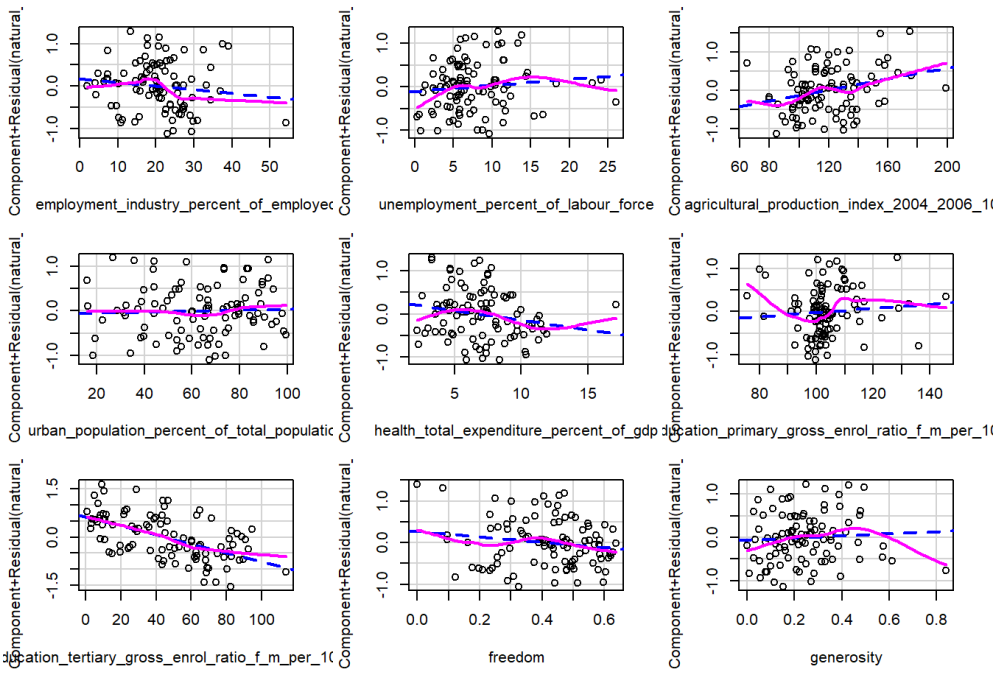


Naudojant tą pačią procedurą, kaip aprašyta prieš tai, sudarytas modelis natūraliam populiacijos prieaugiui. Pažingsnine regresija gautas modelis statistiškai reikšmingai nesiskyrė nuo naudojančio visas kovariantes (p = 0.533).

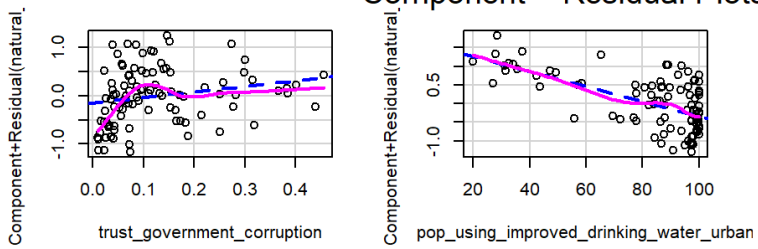
Gautą modelį sudaro kovariantės “agricultural_production_index“ (β = 0.007, p = 0.01), “education_tertiary_gross_enrol_ratio“ (β = -0.016, p < 0.01) ir “pop_using_improved_drinking_water_urban“ (β = -0.20, p < 0.01). Interpretuodami galime teigti, kad šalys su mažesne prieiga prie geros kokybės geriamojo vandens, mažesniu trečio lygmens mokslo lankymu vidutiniškai pasižymi didesniu natūraliu gyventoju prieaugiu. Teigiamai šį prieaugį veikia ir agrikultūrinės produkcijos kiekis. Pagal standartizuotus koeficientus, visų 3 kovariančių įtaka panaši.

Tiek pagal diagnostinius grafikus, tiek pagal paaiškintą dispersijos dalį (R-adj = 0.70), šis modelis tikėtina labiau tinka prognozuoti natūralaus prieaugio reikšmes, negu prieš tai sudarytas modelis prognozuoti migracijos prieaugį.

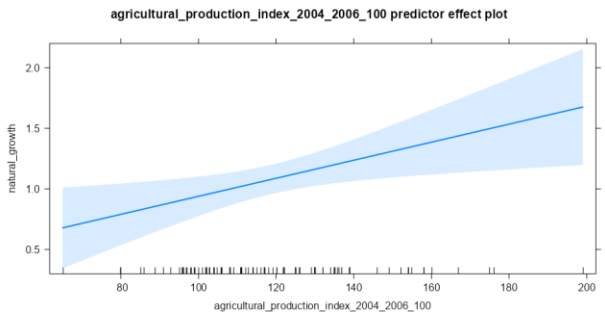
```
print("Tiesinės regresijos modelis natūraliam prieaugiui")

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

data <- regression_train %>% dplyr::select(-migration_growth)
model_linear_natural <- linear_fit(natural_growth ~ .)
```

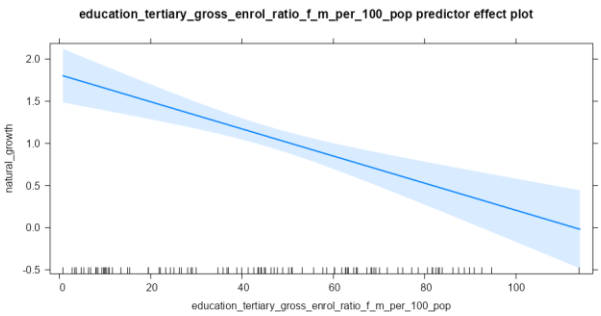


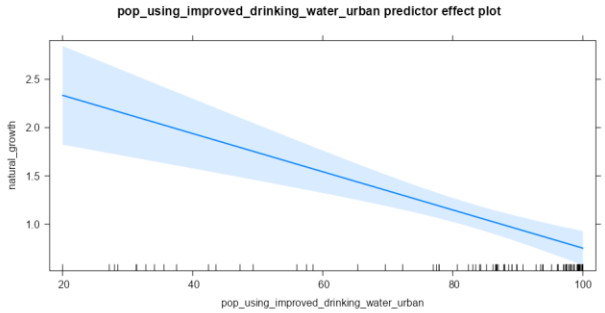
Component + Residual Plots



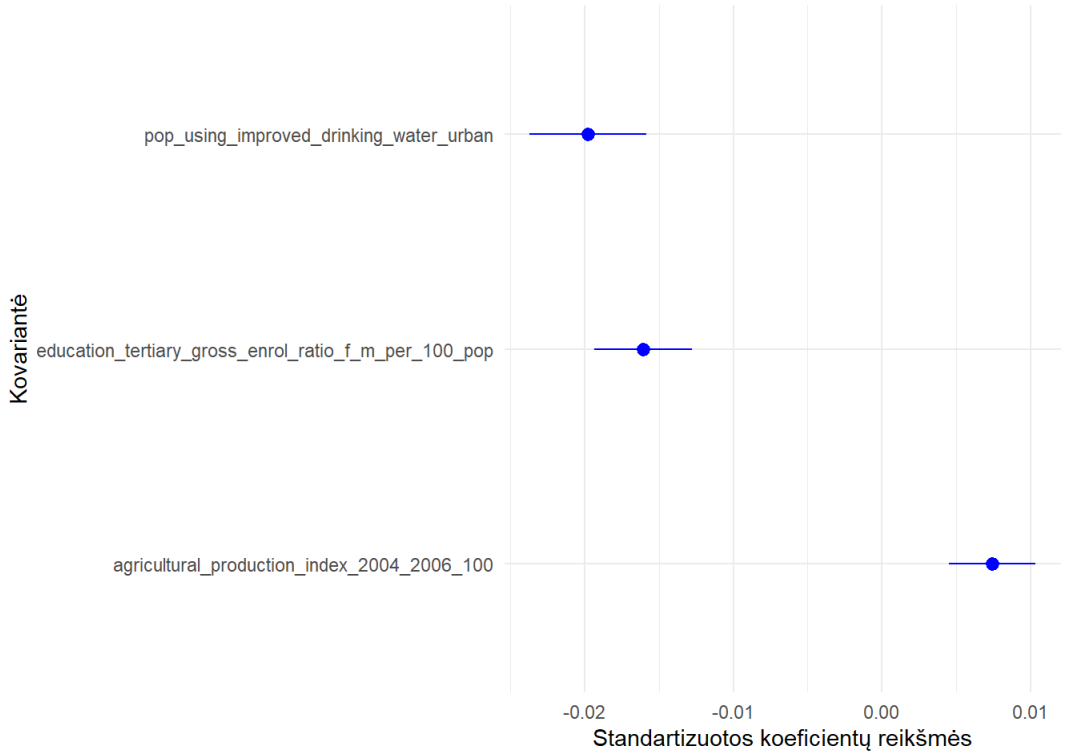

```
## Analysis of Variance Table
##
## Model 1: natural_growth ~ employment_industry_percent_of_employed +
unemployment_percent_of_labour_force +
##     agricultural_production_index_2004_2006_100 +
urban_population_percent_of_total_population +
##     health_total_expenditure_percent_of_gdp +
education_primary_gross_enrol_ratio_f_m_per_100_pop +
##     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
```



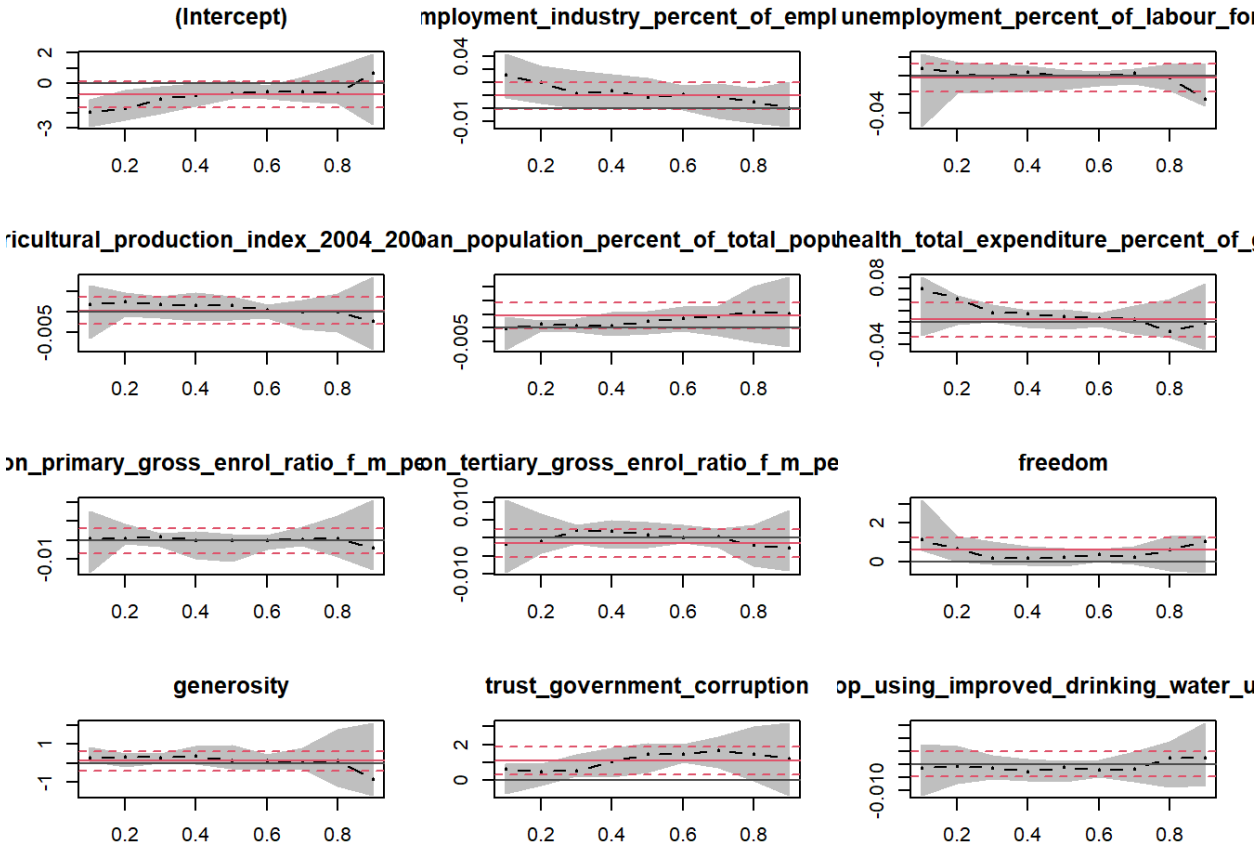
Nors prieš tai abiems modeliams buvo įvertinta kovariančių įtaka gyventojų prieaugio vidurkiui, tikėtina, kad kovariančių įtaka nėra pastovi lyginant didžiausią ir mažiausią prieaugį turėjusioms šalis, todėl papildomai sudaryti kvantilių regresijos modeliai. Kvantilių modeliai sudaryti kvantiliams nuo 0.1 iki 0.9, keičiant kvantilius po 0.1.

Kaip ir prieš tai pirmiausia sudarytas modelis migracijos gyventojų prieaugiui. Pagal Wald testą su nei viena kovariante negautas statistiškai reikšmingas skirtumas tarp ją atitinkančių koeficientų reikšmių imant skirtingus kvantilius.

Kiekvieno kvantilio modeliams koeficientai $\beta(\tau)$ parodo, kiek pasikeičia tam tikras atsako kvantilis, kai tą koeficientą atitinkanti kovariantė padidėja vienetu pvz. prieš tai sudaryto modelio atveju 0.5 kvantiliui (medianai) vieno procento išlaidų sveikatos apsaugai (kovariantė “health_total_expenditure_percent_of_gdp”) padidėjimas migracijos prieaugio medianą padidina 0.01.

```
# 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))
```

```r
  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
```

Analogiškai sudaryti kvantilių regresijos modeliai natūraliam gyventojų prieaugiui. Imant 0.1 reikšmingumo lygmenį, kovariantėms „employment_industry_percent_of_employed“ (p=0.092),

„urban_population_percent_of_total_population“ (p=0.09), „freedom“ (p=0.09) ir „generosity“ (p=0.06) rastas statistiškai reikšmingas koeficientų reikšmių skirtumas skirtingiems atsako kvantiliams. Apibendrindami galime teigti, kad mažą gyventojų prieaugį turinčioms šalims „generosity“ ir „„employment_industry_percent_of_employed“ padidėjimas mažina natūralų gyventojų prieaugį, tačiau dideles natūralaus prieaugio reikšmes turinčioms šalims šių kovariančių įtaka atsakui teigiama.

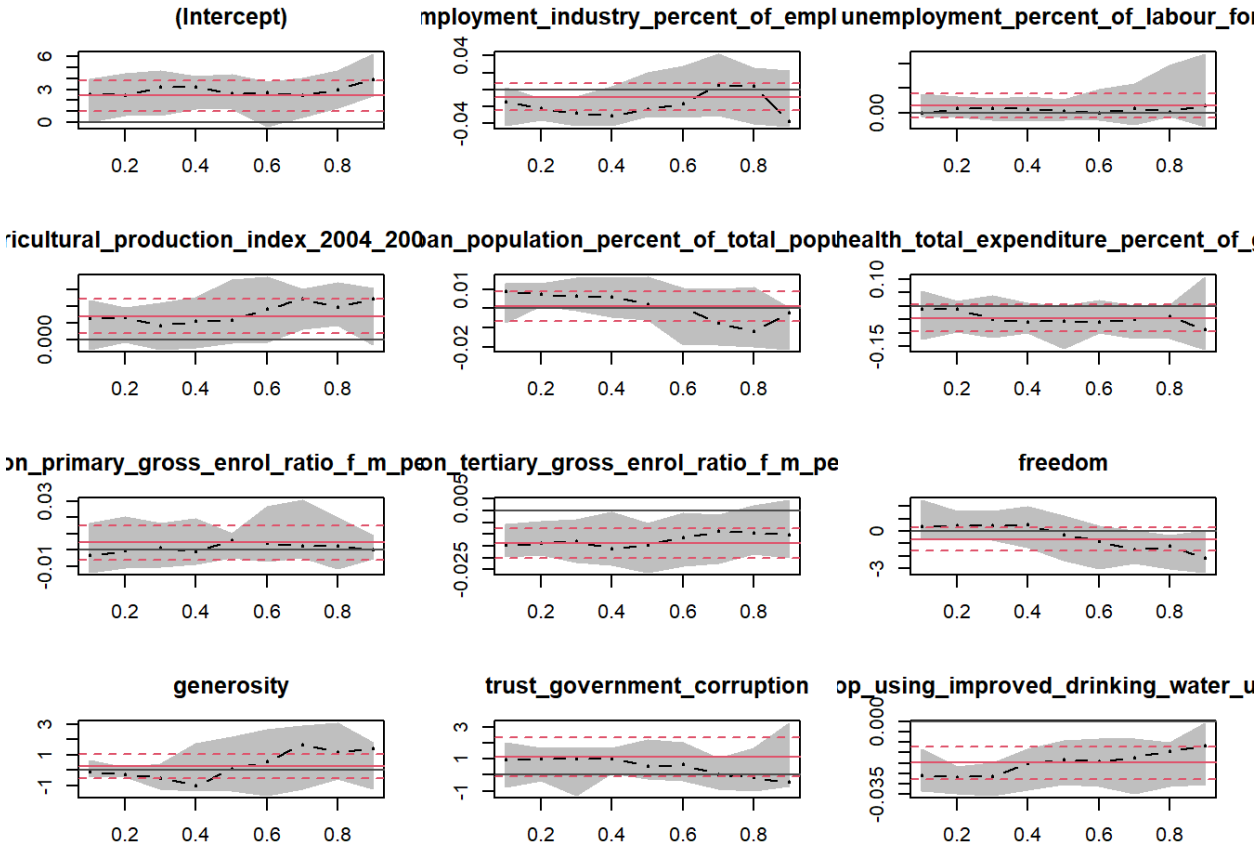
Priešingi rezultatai gaunami su kovariantėmis „freedom“ ir „urban_population_percent_of_total_population“: mažas natūralaus prieaugio reikšmes turinčioms šalims jų įtaka atsakui teigiama, tuo tarpu dideles reikšmes turinčioms šalims – neigiama.

0.5 kvantilio modelyje, naudojančiame visas kovariantes, gauta statistiškai reikšminga požymių „education_tertiary_gross_enrol_ratio“ (p=0.03), ‚pop_using_improved_drinking_water“ (p=0.04) įtaka. Šie rezultatai labai panašūs į rezultatus, gautus naudojant tiesinį regresijos modelį.

```
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_natural <- 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
```

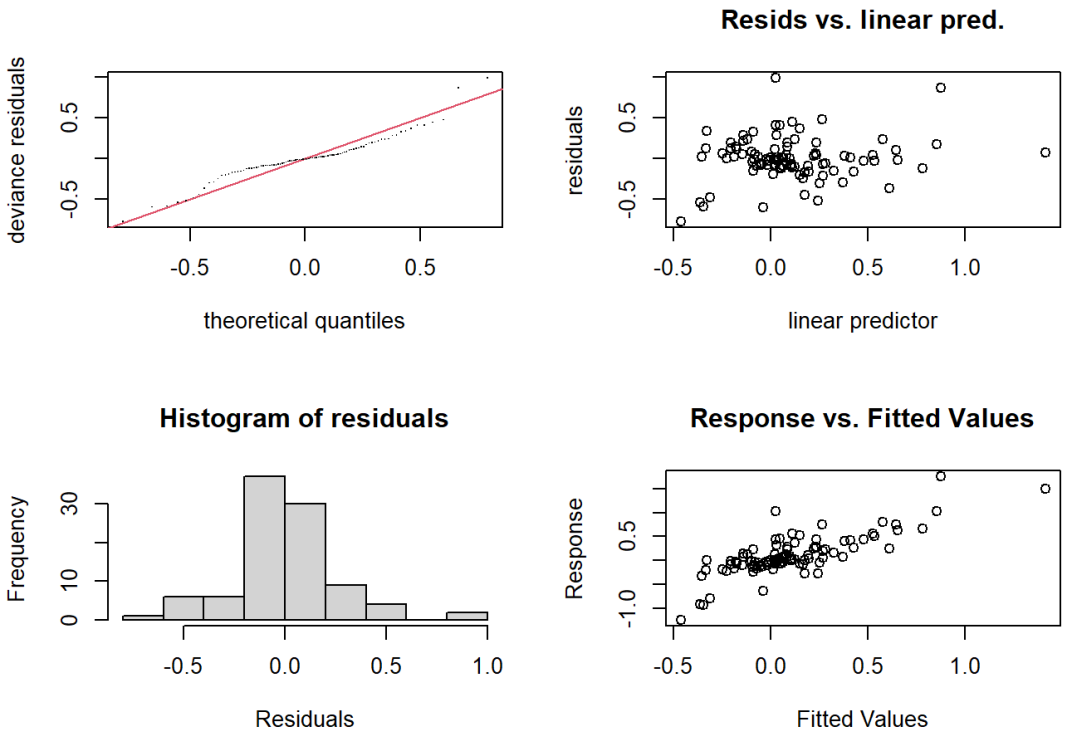
```
## education_primary_gross_enrol_ratio_f_m_per_100_pop  0.974933
## education_tertiary_gross_enrol_ratio_f_m_per_100_pop 0.791577
## freedom                                              0.088479 .
## generosity                                           0.005959 **
## trust_government_corruption                          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. Šiam tikslui pasitelkta *mgcv* biblioteka. Naudojant *gam* funkciją iš šios bibliotekos, parametras λ parenkamas automatiškai, naudojant generalized cross validation, taip apsaugant modelį nuo persimokymo.

Migracijos prieaugiui iš pradžių glodniaisiais splainais modeliuotos visos kovariantės, tačiau dalis jų modelyje gautos supaprastintos iki paprasto tiesinio sąryšio, todėl atsižvelgus į šiuos rezultatus glodniaisiais 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“, ir “trust_government_corruption“. Šių kovariančių įtakos pavaizduotos grafiškai. Kaip ir prieš tai sudarytame tiesiniame grafike matoma, kad šiuo modeliu paaiškinama sąlyginai nedidelė dalis atsako dispersijos (R-adj = 0.472), todėl šis modelis taip pat gali netikti reikšmių prognozavimui.

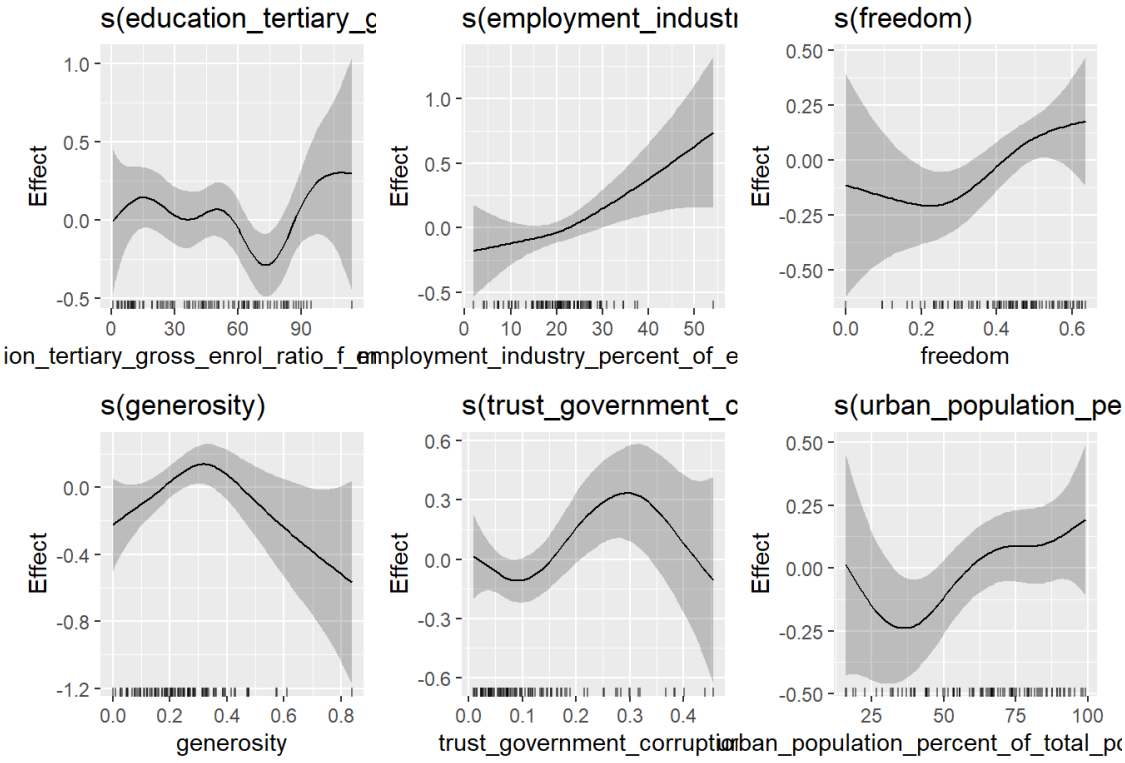
```
library(mgcv)
library(gratia)

# 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)
```



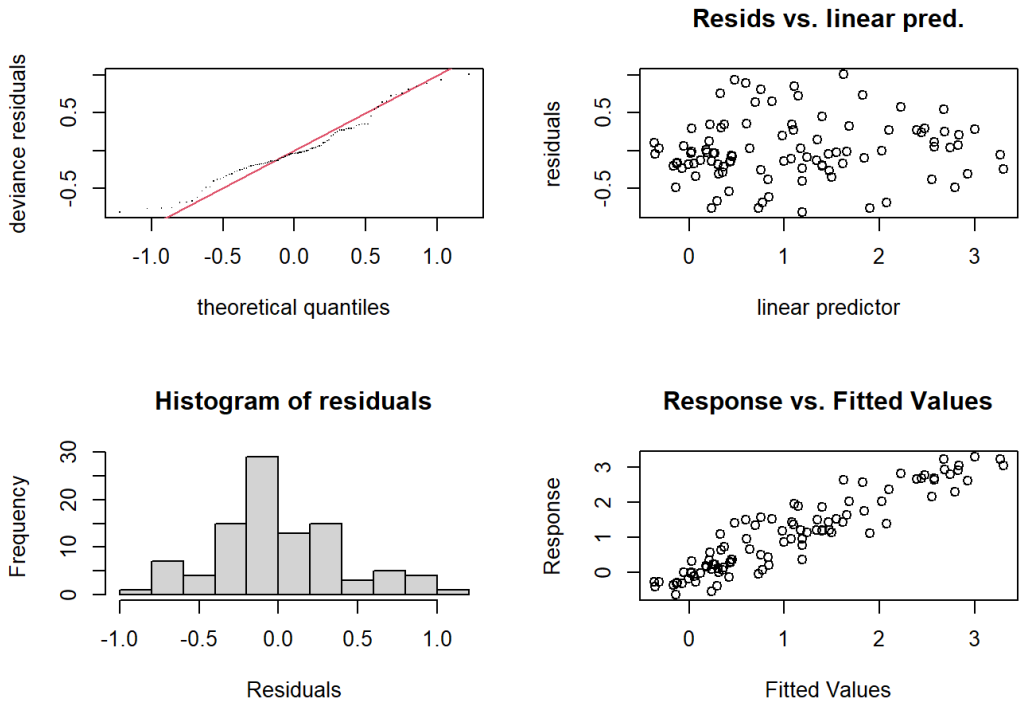

```
k.check(model_gam_migration)

##                                                          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
```

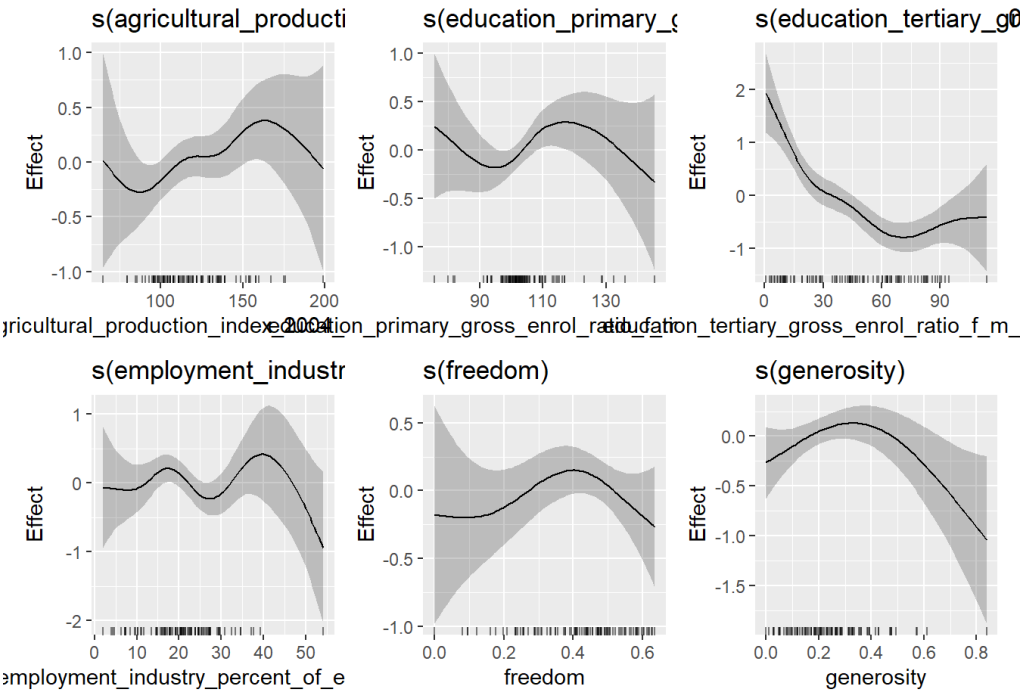
Analogiškai sudarytas modelis ir natūraliam migracijos priaugiui. Glodnaisiais splainais modeliuojamos kovariantės “employment_industry_percent_of_employed”, “agricultural_production_index”, “education_primary_gross_enrol_ratio”, “education_tertiary_gross_enrol_ratio”, “freedom” ir “generosity”. Pagal diagnostinius grafikus ir paaiškintos dispersijos dalį (R-adj = 0.81) tikėtina, kad šis modelis gana tiksliai geba prognozuoti natūralų šalies gyventojų prieaugį.

```
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
```

Visi prieš tai sudaryti modeliai tarpusavyje palyginti siekiant palyginti siekiant surasti, kuris modelis labiausiai tinka prognozuoti migracijos gyventojų prieaugį (iš kvantilių regresijos naudotas medianos regresijos modelis). Tiesinis ir glodniųjų splainų modeliai palyingti pagal AIC. Įvertintas prieš tai sudarytų modelių gebėjimas prognozuoti gyventojų prieaugio reikšmes. Tiek mokymo, tiek testavimo duomenims apskaičiuotos vidutinės absoliučios paklaidos (angl. Mean Absolute Error), vidutinės kvadratinės paklaidos šaknies (angl. Root Mean Square Error) metrikos. Grafiškai pavaizduotos tikrų ir prognozuotų reikšmių sklaidos diagramos.

Pagal AIC statistiką glodniųjų splainų modeliu gaunami geresni rezultatai (tiesiniam ir glodniųjų splainų modeliams atitinkamai 79 ir 71). Pagerėjimas lyginant su tiesiniu modeliu matomas ir prognozuojant reikšmes testavimo aibėje (MAE atitinkamai 0.30 ir 0.24, RMSE – 0.58 ir 0.53). Medianos regresijos atveju pagal abi metrikas gauti prasčiausi rezultatai.

Prognozavimui panaudojus testavimo aibę gauti priešingi rezultatai: geriausi rezultatai pagal abi metrikas gauti naudojant medianos regresiją, prasčiausi – glodniųjų splainų modelį. Iš sklaidos diagramų matome, kad didelė dalis testavimo aibėje esančių stebėjimų visų modelių buvo prognozuojami visiškai klaidingai.

Iš šių rezultatų galime teigti, kad nei vienas modelis nėra tinkamas prognozuoti migracijos prieaugį, jie gali būti naudingi tik įvertinti kovariančių įtaką. Tokie rezultatai visai natūralus – migracijos prieaugis kiekvienai šaliai labai stipriai priklauso nuo kiekvienos šalies politinės situacijos specifikos, todėl prognozuoti migracijos prieaugį vienu modeliu yra sudėtinga.

```
library(yardstick)

# regresijos modelių įvertinimas
regression_test <- function(column, model_linear, model_gam, model_quantile, 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),
      predicted_quantile = predict(model_quantile,data)[,5]
    )

  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))
  print("Kvantilių regresijos modelis")
  print(set(regression_test, {{ column }}, predicted_quantile))

  regression_test %>%
    pivot_longer(c(predicted_gam, predicted_linear,predicted_quantile)) %>%
    mutate(name = factor(name, levels = c("predicted_linear",
"predicted_gam","predicted_quantile"))) %>%
    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)
```

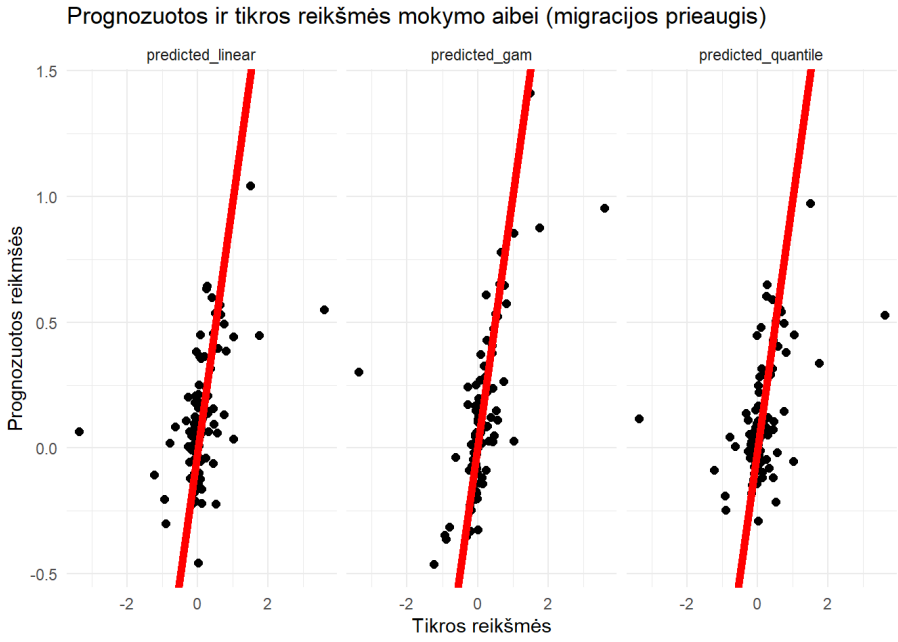
```
## [1] 70.74101
```

```
resid(model_quantile_migration) %>% abs() %>% apply(2,sum)
```

```
## tau= 0.1 tau= 0.2 tau= 0.3 tau= 0.4 tau= 0.5 tau= 0.6 tau= 0.7 tau= 0.8
## 35.38392 28.66851 24.45724 22.35572 21.90891 22.34460 23.48362 29.32433
## tau= 0.9
## 46.10688
```

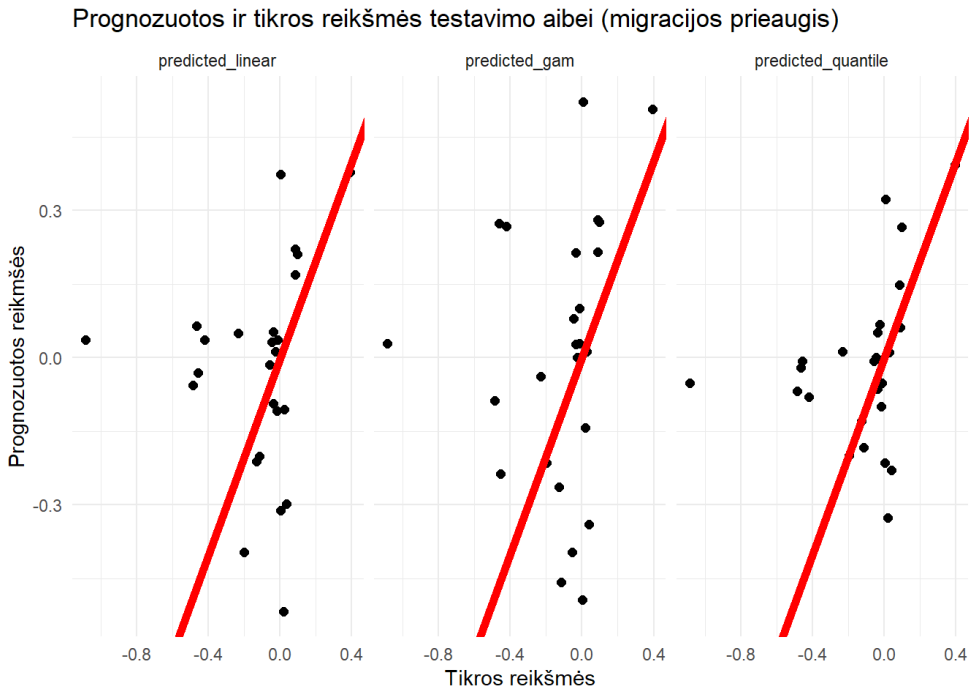
```
regression_test(
  migration_growth, model_linear_migration, model_gam_migration, model_quantile_migration,
regression_train,
  "Prognozuotos ir tikros reikšmės mokymo aibei (migracijos prieaugis)"
)
```

```
## [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
## [1] "Kvantilių regresijos modelis"
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.586
## 2 mae     standard       0.293
```



```
regression_test(
  migration_growth, model_linear_migration, model_gam_migration, model_quantile_migration,
test,
  "Prognozuotos ir tikros reikšmės testavimo aibei (migracijos prieaugis)"
)
```

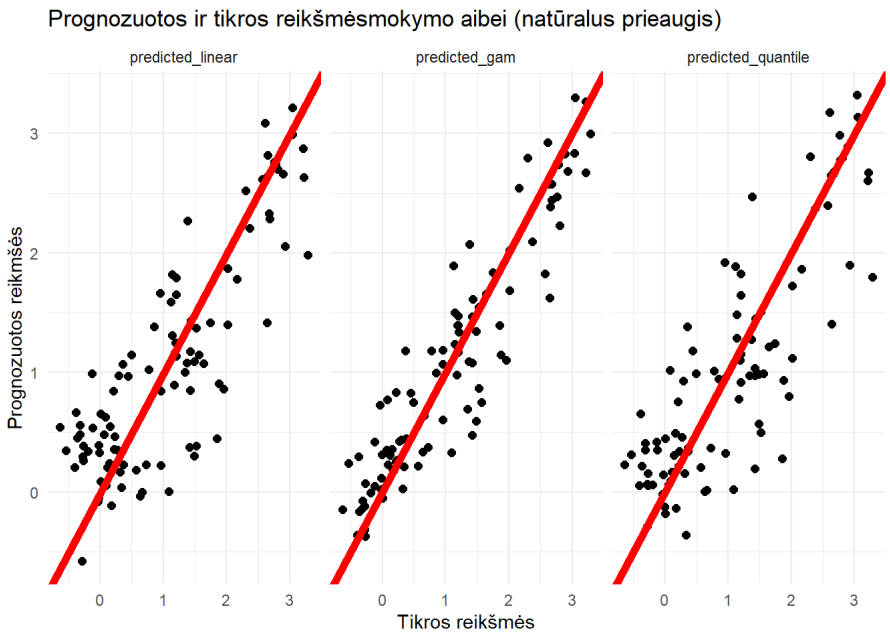
```
## [1] "Tiesinis modelis"
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.345
## 2 mae     standard       0.244
## [1] "GAM modelis"
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.381
## 2 mae     standard       0.279
## [1] "Kvantilių regresijos modelis"
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.298
## 2 mae     standard       0.194
```



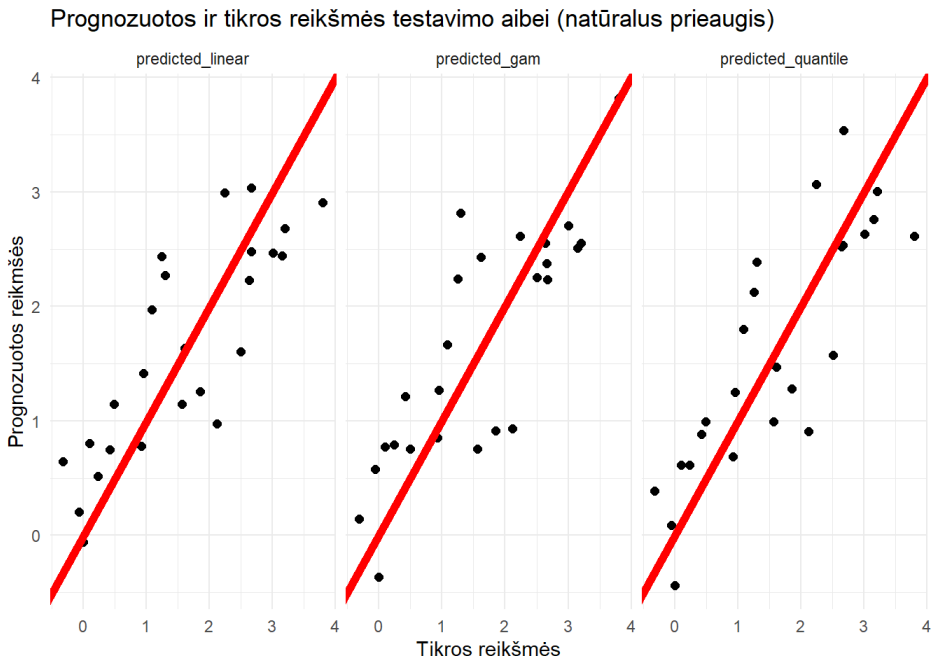
Toks pat palyginimas atliktas ir natūraliam prieaugiui. Natūraliam prieaugiui naudojant modelį su glodniaisiais splainais gaunamas pagerėjimas lyginant su tiesiniu modeliu pagal AIC (AIC reikšmės atitinkamai 181 ir 156). Mokymo aibėje lyginant prognozuotų ir tikrų reikšmių sklaidos diagramas matomas glodniųjų splainų modeliu gautas rezultatų pagerėjimas lyginant su tiesiniu (MAE atitinkamai 0.47 ir 0.31). Medianos regresijos modeliu gauti panašūs rezultatai kaip ir naudojant tiesinį modelį.

Nepaisant šių rezultatų, panaudojus testavimo aibę skirtumai tarp modelio tampa tik minimalūs (MAE tiesiniam, glodniųjų splainų ir medianos regresijos modeliams atitinkamai 0.57, 0.56 ir 0.56).

```
# 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

resid(model_quantile_natural) %>% abs() %>% apply(2,sum)

## tau= 0.1 tau= 0.2 tau= 0.3 tau= 0.4 tau= 0.5 tau= 0.6 tau= 0.7 tau= 0.8
## 62.71193 53.97719 47.10880 44.00954 42.48561 43.90943 52.47930 59.29189
## tau= 0.9
## 76.99340

regression_test(
  natural_growth, model_linear_natural, model_gam_natural, model_quantile_natural,
regression_train,
  "Prognozuotos ir tikros reikšmėsmokymo aibei (natūralus prieaugis)"
)


## [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
## [1] "Kvantilių regresijos modelis"
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.580
## 2 mae     standard       0.438
```



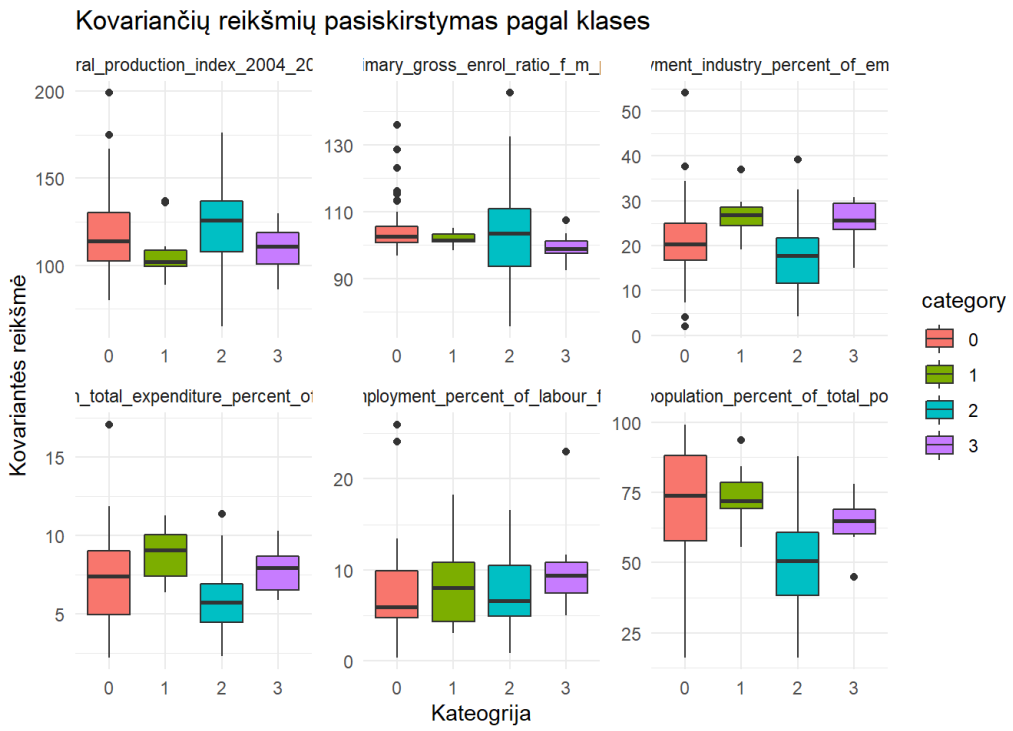

```
regression_test(
  natural_growth, model_linear_natural, model_gam_natural, model_quantile_natural, test,
  "Prognozuotos ir tikros reikšmės testavimo aibei (natūralus prieaugis)"
)
## [1] "Tiesinis modelis"
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 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
## [1] "Kvantilių regresijos modelis"
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.645
## 2 mae     standard       0.555
```



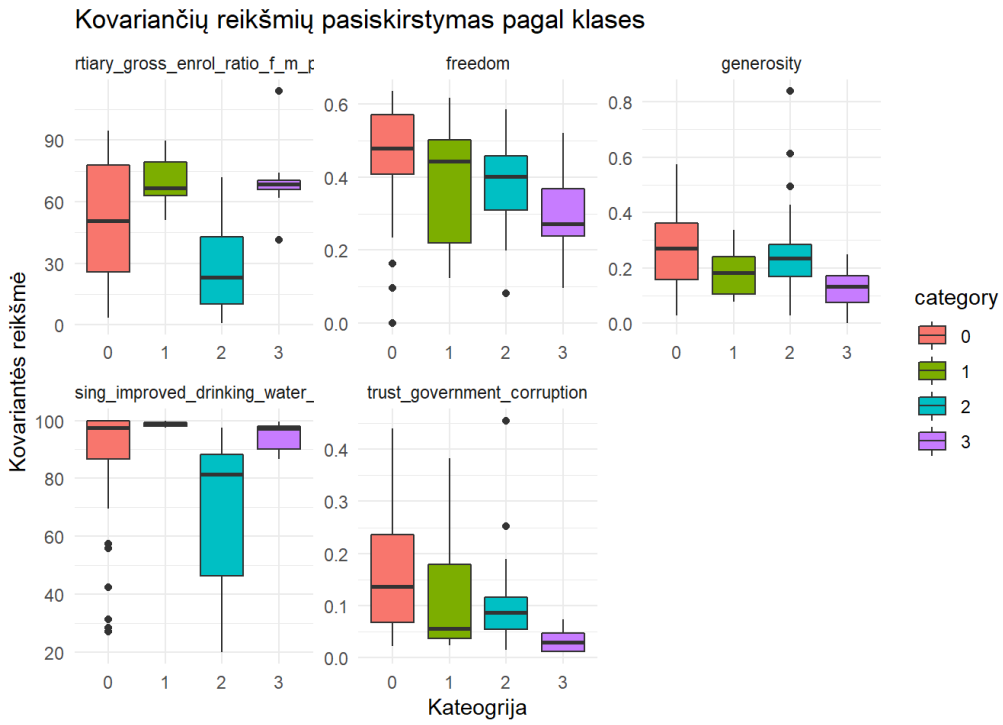
Atsižvelgiant į tai, kad sudaryti regresijos modeliai nebuvo tinkami prognozuoti migracijos prieaugį, pasirinkta sudaryti multinominės logistinės regresijos modelį, kuriuo siekiama supaprastinti uždavinį ir gauti geresnius rezultatus negu prieš tai sudarytais regresijos modeliais, kai siekiama sužinoti tik kokiai klasei priklauso šalis (ar šalies natūralus/migrantų prieaugiai teigiami ar neigiamai). Akivaizdu, kad kitas šio modelio privalumas yra, kad gauti prognozei apie tai, kokio tipo yra šalies demografinis pokytis, reikalingas tik vienas modelis vietoje dviejų.

Stačiakampėmis diagramomis kiekvienai kovariantei pavaizduotas jos pasiskirstymas pagal klases. Naudojamos tokios pat kovariantės kaip ir prieš tai.

```
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")
```



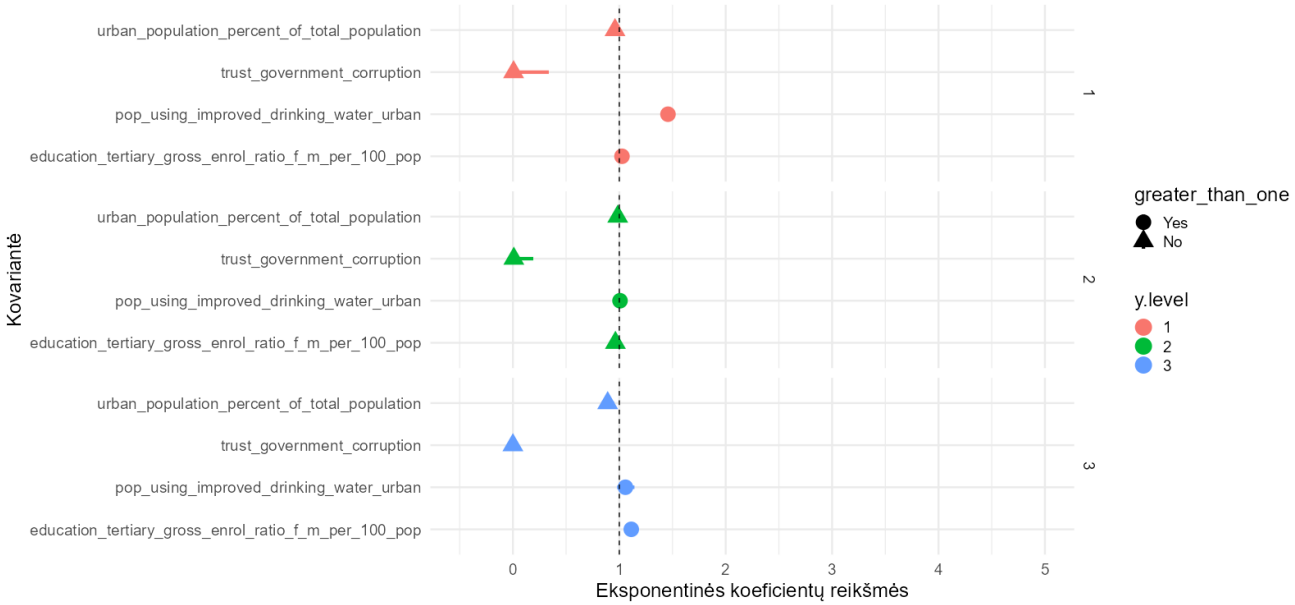

```
classification_train %>%
  dplyr::select(7:length(classification_train), category) %>%
  pivot_longer(-category) %>%
  ggplot(aes(x = category, y = value, fill = category)) +
  facet_wrap(vars(name), scales = "free") +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Kovariančių reikšmių pasiskirstymas pagal klases") +
  xlab("Kateogrija") + ylab("Kovariantės reikšmė")
```



Sudarytas multinomiės logistinės regresijos modelis naudojantis visas kovariantes. Pažingsnine regresija sumažintas modelis reikšmingai nesiskyrė nuo pilno (p = 0.34). Lyginamąja kategorija pasirinkta kategorija “0”, todėl modelio koeficientai interpretuojami jos atžvilgiu pvz. vieno procento padidėjimas mieste gyvenančios gyventojų dalies 4% sumažina galimybę įvykti kategorijai “1” kategorijos “0” atžvilgiu, 2% sumažina galimybę įvykti kategorijai “2” kategorijos “0” atžvilgiu ir 11% sumažina galimybę įvykti kategorijai “3” kategorijos “0” atžvilgiu.

```
# 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)
```

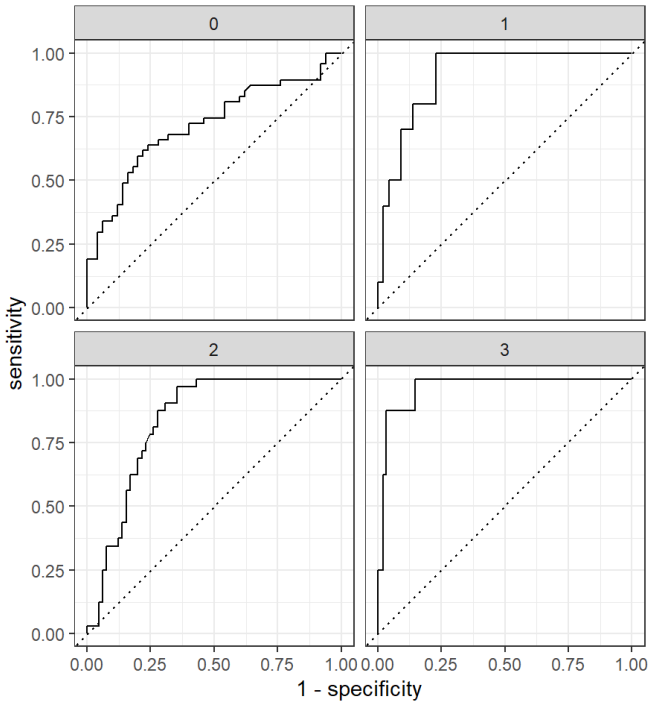


Klasifikavimo modelio kokybė vertinta naudojant maišos matricas (angl. confusion matrices), bendrą tikslumą (angl. accuracy), F-score, J-index. Kadangi turimas daugelio klasių (multiclass) uždavinys paskutinių dviejų minėtų modelio kokybės vertinimo metrikų bendros reikšmės gautos naudojant „macro“ vidurkinimą imant metrikos reikšmių visoms 4 klasėms vidurkį (taip kiekvienai klasei priskiriant lygų svorį). Papildomai sudaryta metrika, kuri „pataiso“ bendrą tikslumą priskirdama 0.5 stebėjimui jeigu teisingai buvo prognozuojamas vieno tipo gyventojų prieaugis. Galiausiai nubraižytos ROC kreivės ir apskaičiuota AUC statistika (daugelio klasės uždaviniui AUC pritaikytas naudojant Hand, Till metodą).

Modeliu gauta tikslumas lygus 0.67, „pataisytas“ tikslumas lygus 0.82. Pagal maišos matricą matoma, kad modeliui sunkiai sekasi atskirti klases „0“ (teigiami abiejų tipų prieaugiai) ir „2“ (neigiamas migracijos ir teigiamas natūralus prieaugiai).

```
# 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
```



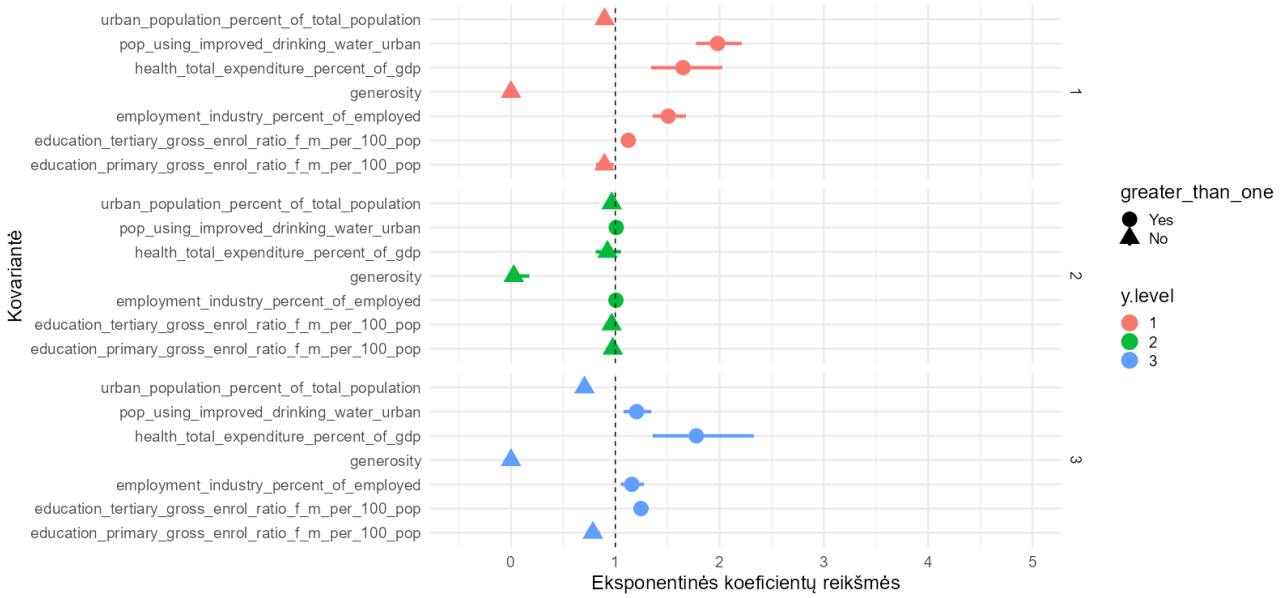
Kadangi duomenų rinkinyje turimos gana stipriai išbalansuotos klasės (klasės “0” - teigiamas migracijos prieaugis, teigiamas natūralus stebėjimų beveik 6 kartus daugiau už klasės „3“ - neigiami abiejų tipų prieaugiai stebėjimų kiekį) pasirinkta šią problemą spręsti (ir tikėtina pagerinti modeliu gaunamus rezultatus) sugeneruojant dirbtinų stebėjimų mažumos klasėms naudojant SMOTE algoritmą.

Analogiškai praėjusiam modeliui, sudarytas multinominės logistinės regresijos modelis, mokymui naudojantis SMOTE sugeneruotus dirbtinius papildomus stebėjimus. Pažingsnine regresija gautas modelis statistiškai reikšmingai nesiskyrė nuo pilno (p = 0.60). Koeficientų interpretacija taip pat analogiška praėjusiam modeliui.

```
# 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)
```

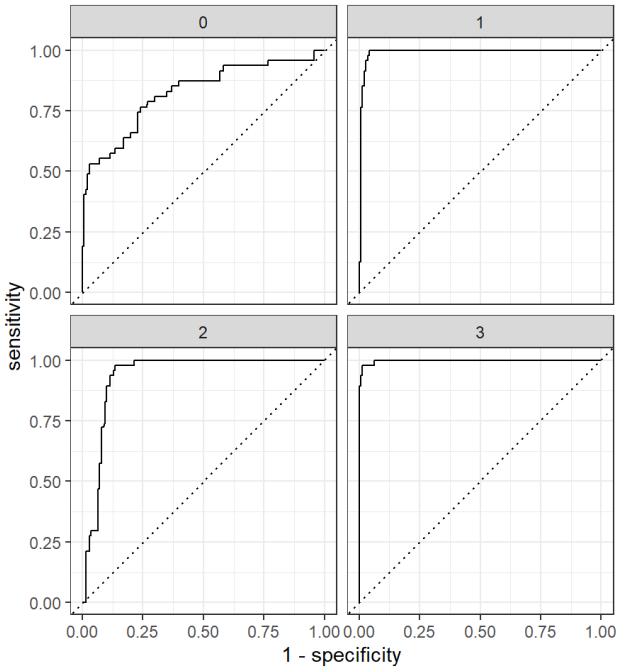
```r
smote_recipe <- prep(smote_recipe, training = classification_train)

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   -78.04724                             0.451424798
## 2     5.48757                            -0.006804232
## 3    72.40568                            -0.178757218
##   urban_population_percent_of_total_population
## 1                                   -0.08797960
## 2                                   -0.02502958
## 3                                   -0.45658276
##   health_total_expenditure_percent_of_gdp
## 1                              0.78370798
## 2                             -0.05718385
## 3                              0.11190683
##   education_primary_gross_enrol_ratio_f_m_per_100_pop
## 1                                         -0.01023564
## 2                                         -0.01370988
## 3                                         -0.85938214
##   education_tertiary_gross_enrol_ratio_f_m_per_100_pop   freedom generosity
## 1                                          0.13119162 -13.79913 -19.607247
## 2                                         -0.04115902  -1.31000  -2.897483
```

```
## 3                                          0.34508959  13.68574 -30.026863
##   trust_government_corruption pop_using_improved_drinking_water_urban
## 1                   13.593242                                0.67752809
## 2                   -3.378663                                0.01374586
## 3                  -70.597268                                0.31405698
##
## Std. Errors:
##   (Intercept) employment_industry_percent_of_employed
## 1    1.023145                             0.11970460
## 2    3.008252                             0.04503703
## 3    2.535592                             0.11119875
##   urban_population_percent_of_total_population
## 1                                   0.05021285
## 2                                   0.01974761
## 3                                   0.09355286
##   health_total_expenditure_percent_of_gdp
## 1                               0.2771792
## 2                               0.1376968
## 3                               0.3893734
##   education_primary_gross_enrol_ratio_f_m_per_100_pop
## 1                                          0.08360149
## 2                                          0.02435611
## 3                                          0.10017359
##   education_tertiary_gross_enrol_ratio_f_m_per_100_pop  freedom generosity
## 1                                          0.04422784 6.075000   6.032812
## 2                                          0.01960441 2.566533   1.975537
## 3                                          0.06706318 6.035300   8.744504
##   trust_government_corruption pop_using_improved_drinking_water_urban
## 1                   7.632834                                0.11627297
## 2                   3.271378                                0.01832338
## 3                   1.929764                                0.11536189
##
## Residual Deviance: 180.0295
## AIC: 240.0295

plot_coefficients(model_logistic2_small)
```



Modeliu gauta tikslumas lygus 0.83, „pataisytas“ tikslumas lygus 0.90. Abi šios metrikos ryškiai pagerina praeitu modeliu gautus rezultatus. Iš maišos matricos matomas pagerėjimas tarpusavyje atskiriant klases „0“ ir „2“.

```
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 25  0  9  1
##          1  6 47  0  0
##          2 14  0 38  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      0.821463479467331
## 4 custom_metric multiclass 0.906914893617021
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc macro          0.936
```



Dirbtinių stebėjimų generavimas su SMOTE algoritmu galėjo sukelti persimokymą, todėl modelių rezultatus reikia patikrinti ir naudojant testavimo aibe. Pirmuoju modeliu gautas bendras tikslumas lygus 0.6, “pataisytas” tikslumas - 0.8. Abiejų tipų gyventojų prieaugis buvo teisingai prognozuotas 15 kartų, vien tik migracijos - 2 kartus ir vien tik natūralus – 8 kartus.

Antruoju (su SMOTE generuotais stebėjimais apmokytu) modeliu gautas tikslumas lygus 0.68, “pataisytas” tikslumas - 0.84. Testavimo aibėje 17 kartų teisingai prognozuoti abiejų tipų prieaugiai, 8 kartus – tik natūralus prieaugis.

Jeigu vietoje multinominės logistinės regresijos modelių būtų panaudojami 2 prieš tai sudaryti apibendrinti adityvieji regresijos glodniųjų splainų modeliai iš jų gautų skaitinių reikšmių priskiriant klases, testavimo aibėje gautas tikslumas lygus 0.48, “pataisytas” vidurkis lygus 0.72. 12 stebėjimų buvo teisingai prognozuotos abi klasės, 11 stebėjimų – tik natūralus prieaugis, 1 kartą – tik migracijos prieaugis ir 1 kartą klaidingai buvo prognozuoti abiejų tipų migracijos prieaugiai.

Iš šių rezultatų matoma, kad multinominės regresijos modeliais gaunami geresni rezultatai negu šiam tikslui pritaikius prieš tai sudarytus 2 regresijos modelius. Panaudojus dirbtinių stebėjimų generavimą SMOTE algoritmu gautas rezultatų pagerėjimas. Visais atvėjais modeliai dažniau klydo prognozuojami kokio tipo buvo migracijos prieaugis, negu prognozuodami kokio tipo buvo natūralus prieaugis.

```
# 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  5  0 13  0
##          3  0  1  0  1
## # A tibble: 2 x 2
##   results             n
##   <chr>           <int>
## 1 Correct both       17
## 2 Correct natural     8
## [1] "Modelio kokybės metrikos"
## # A tibble: 4 x 3
##   .metric       .estimator .estimate
##   <chr>         <chr>      <chr>
## 1 accuracy      multiclass 0.68
## 2 j_index       macro      0.499900793650794
## 3 f_meas        macro      0.621212121212121
## 4 custom_metric multiclass 0.84

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

Atlikta regresinę analizę natūraliam ir migracijos gyventojų prieaugiui pagal ekonominius ir socialinius šalių indikatorius.

Sudarytas tiesinis modelis prognozuoti migracijos prieaugį. Siekiant sumažinti modelį naudota pažingsninė regresija. Pagal gautą modelį migracijos prieaugį teigiamai įtakoja dalis darbuotojų, dirbančių industrijos sektoriuje (kovariantė “employment_industry_percent_of_employed“, $p = 0.07$), gyventojų dalis miestuose (kovariantė “urban_population_percent_of_total_population“, $p = 0.16$), asmeninės laisvės (požymis “freedom”, $p = 0.03$) ir pasitikėjimo vyriausybe (požymis “trust_government_corruption”, $p < 0.01$) įvertinimai.

Analogiškai sudarytas tiesinis modelis natūraliam populiacijos prieaugiui. Pagal gautą modelį natūralų gyventojų prieaugis teigiamai susijęs su šalies agrikultūrinės produkcijos kiekiu (požymis “agricultural_production_index“, $p = 0.01$) , neigiamai susijęs su trečio lygmens mokslo lankomumu (požymis “education_tertiary_gross_enrol_ratio“, $p < 0.01$) ir miesto gyventojų dalimi, turinčios prieigą prie geros kokybės geriamo vandens (požymis “pop_using_improved_drinking_water_urban“, $p < 0.01$).

Siekta įvertinti ar kovariančių įtaka yra pastovi lyginant didžiausią ir mažiausią prieaugį turėjusioms šalis, todėl papildomai sudaryti kvantilių regresijos modeliai.

Migracijos prieaugiui pagal Wald testą su nei viena kovariante negautas statistiškai reikšmingas skirtumas tarp ją atitinkančių koeficientų reikšmių imant skirtingus kvantilius. Tuo tarpu natūraliam gyventojų prieaugiui kovariantėms „employment_industry_percent_of_employed“ ($p=0.092$), „urban_population_percent_of_total_population“ ($p=0.09$), „freedom“ ($p=0.09$) ir „generosity“ ($p=0.06$) rastas statistiškai reikšmingas skirtumas tarp jų koeficientų skirtingiems atsako kvantiliams.

Siekiant geriau prognozuoti gyventojų prieaugio reikšmes, abiejų tipų prieaugiams prognozuoti sudaryti apibendrintieji adityvūs modeliai, naudojantys glodniosius splainus, kuriais tarp kovariančių ir atsako modeliuojamas netiesinis sąryšis.

Abiejų gyventojų prieaugių tipams tarpusavyje lyginti tiesinis, glodniųjų splainų ir medianos regresijos modeliai, siekiant surasti, kuris modelis labiausiai tinkamas naudoti prognozėms. Panaudojus migracijos prieaugio modelius testavimo aibei prognozuoti glodniųjų splainų modeliu gauti geriausi rezultatai pagerėjimas (tiesiniam, glodniųjų splainų ir medianos regresijos modeliams MAE reikšmės atitinkamai 0.30, 0.24 ir 0.29). Nepaisant to, testavimo aibėje gauti priešingi rezultatai. Pagal prognozuotų ir tikrų reikšmių sklaidos diagramą pastebėta, kad prognozuojant didelę dalį stebėjimų buvo daromos stiprios klaidos. Daryta išvada, kad nei vienas modelis nebuvo tinkamas prognozuoti migracijos gyventojų prieaugį.

Natūraliam prieaugiui tiek pagal prognozuotų ir tikrų reikšmių sklaidos diagramas, tiek pagal skaitines metrikas glodniųjų splainų modeliu gauti geriausi rezultatai (tiesiniam, glodniųjų splainų ir medianos regresijos modeliams MAE reikšmės atitinkamai 0.47, 0.31 ir 0.44). Panaudojus testavimo aibę skirtumai tarp modelių gauti visiškai minimalūs. Apskritai gauta, kad šie modeliai labiau tinkami prognozuoti natūralų gyventojų prieaugį negu atitinkami modeliai migracijos prieaugiui.

Atsižvelgiant į tai, kad sudaryti regresijos modeliai nebuvo tinkami prognozuoti migracijos prieaugį, pasirinkta sudaryti multinominės logistinės regresijos modelį, kuriuo siekiama supaprastinti uždavinį ir gauti geresnius rezultatus negu prieš tai sudarytais glodniųjų splainų regresijos modeliais, kai siekiama sužinoti tik kokiai klasei priklauso šalis (ar šalies natūralus/migrantų prieaugiai teigiami ar neigiamai).

Kadangi duomenų rinkinyje turimos gana stipriai išbalansuotos klasės pasirinkta šią problemą spręsti sugeneruojant dirbtinų stebėjimų mažumos klasėms naudojant SMOTE algoritmą ir apmokyti antrą multinominės regresijos modelį.

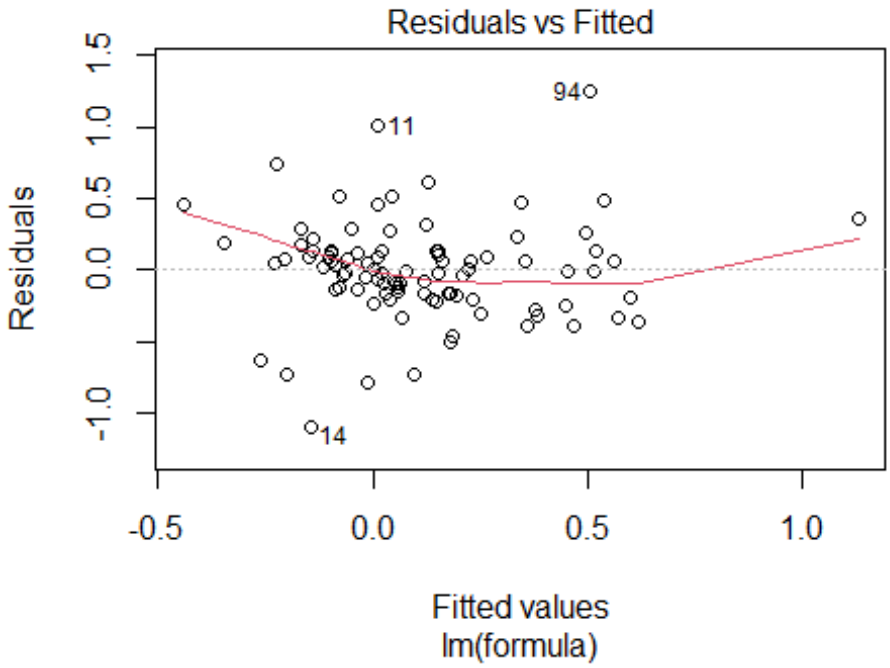
Pirmuoju modeliu gautas ROCAUC lygus 0.86. Bendras tikslumas testavimo aibėje lygus 0.6, “pataisytas” tikslumas - 0.8. Abiejų tipų gyventojų prieaugis buvo teisingai prognozuotas 15 kartų, vien tik migracijos - 2 kartus ir vien tik natūralus – 8 kartus.

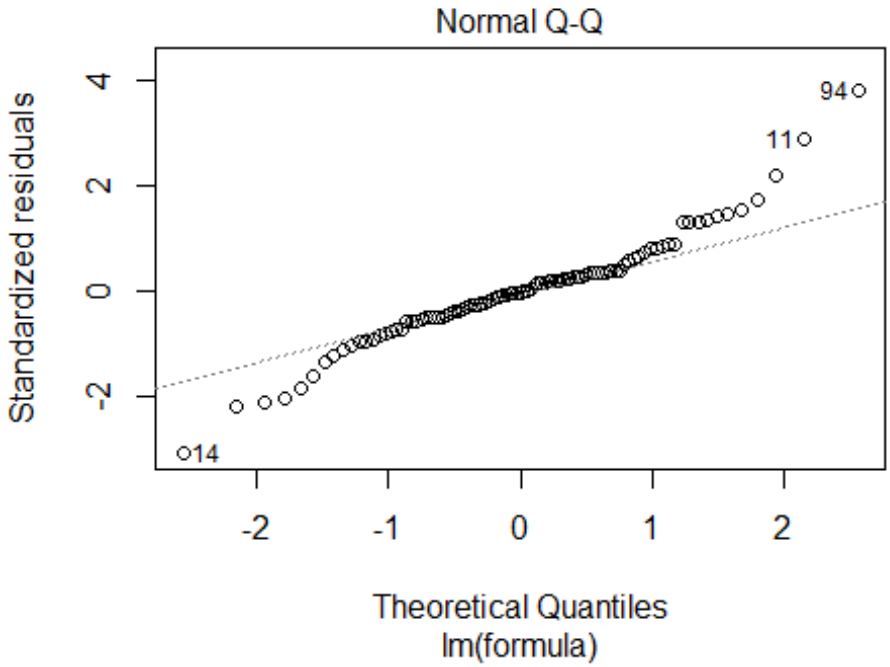
Antruoju (su SMOTE generuotais stebėjimais apmokytu) modeliu gautas rezultatų pagerėjimas: ROCAUC lygus 0.94, bendras tikslumas testavimo aibėje lygus 0.68, “pataisytas” tikslumas - 0.84. Testavimo aibėje 17 kartų teisingai prognozuoti abiejų tipų prieaugiai, 8 kartus – tik natūralus prieaugis.

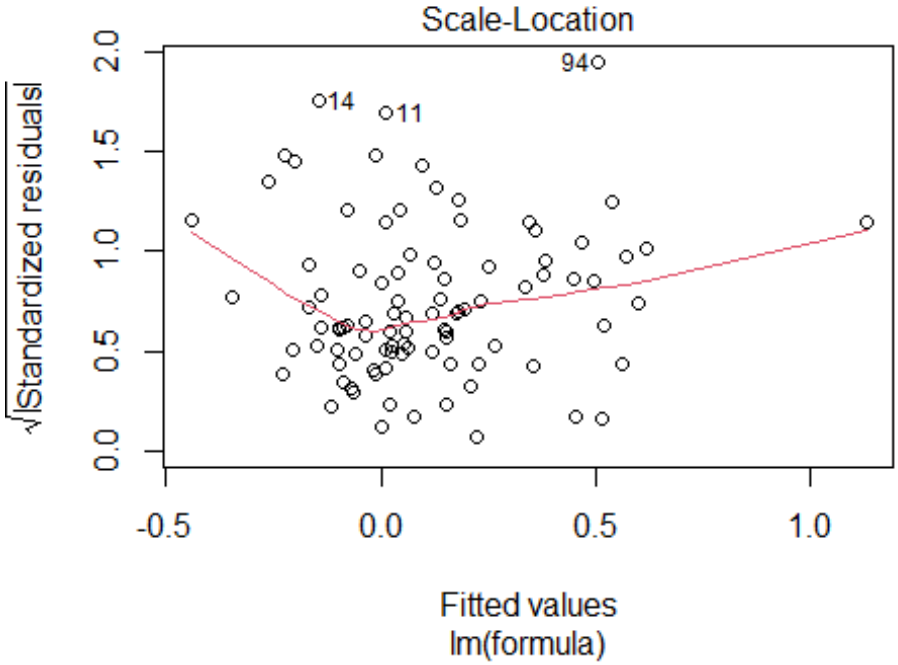
Multinominės logistinės regresijos naudojimas pagerina rezultatus, gaunamus rankiniu būdu iš 2 prieš tai sudarytų glodniųjų splainų regresijos modelių prognozuotų skaitinių reikšmių priskiriant klases: šiuo metodu bendras tikslumas lygus tik 0.48, “pataisytas” vidurkis - 0.72.

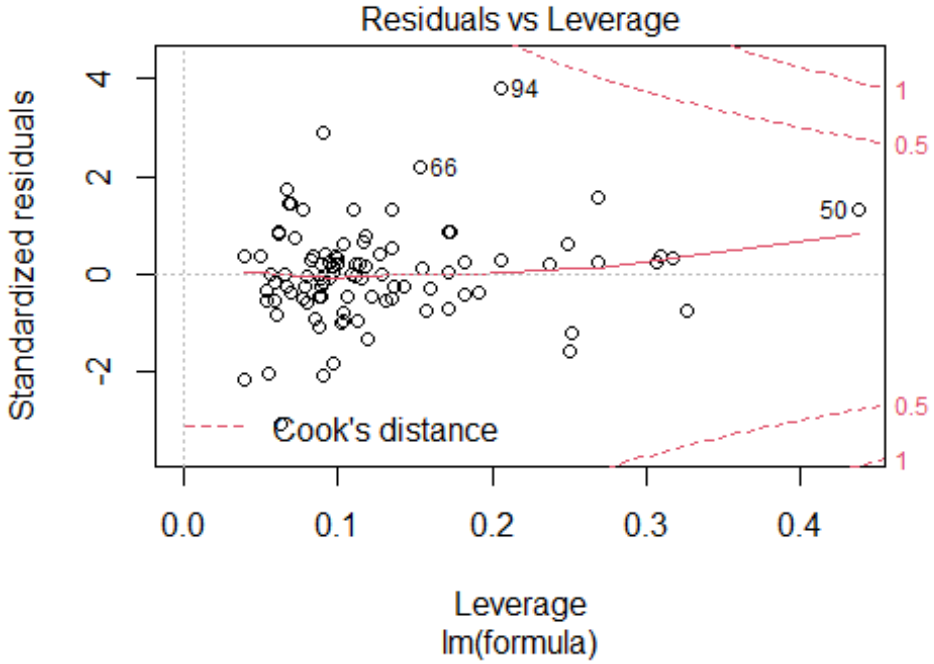
Daroma išvada, kad kaip ir prieš tai sudaryti atskiri regresijos modeliai, multinominės logistinės regresijos modelis dažniau klysta prognozuodamas kokio tipo migracijos prieaugis yra šalyje, negu prognozuodamas kokio tipo yra natūralus prieaugis. SMOTE algoritmo panaudojimas pagerino gautus rezultatus. Abiejų multinominės regresijos modelių (nenaudojant SMOTE ir naudojant) rezultatai geresni negu gaunami pritaikant prieš tai sudarytus atskirus glodniųjų splainų modelius priskirti klasėms.

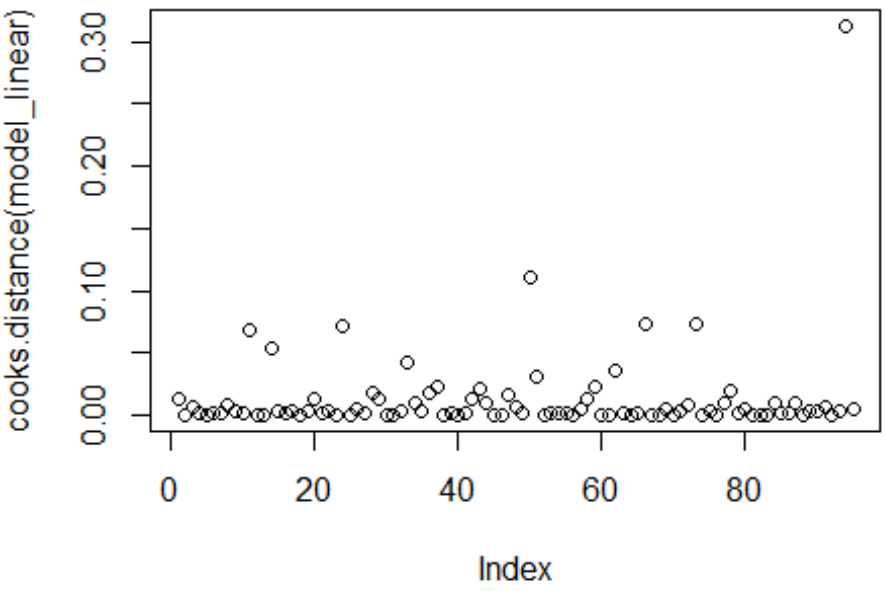
1 Priedas. Tiesinio modelio migracijos prieaugiui diagnostiniai grafikai.



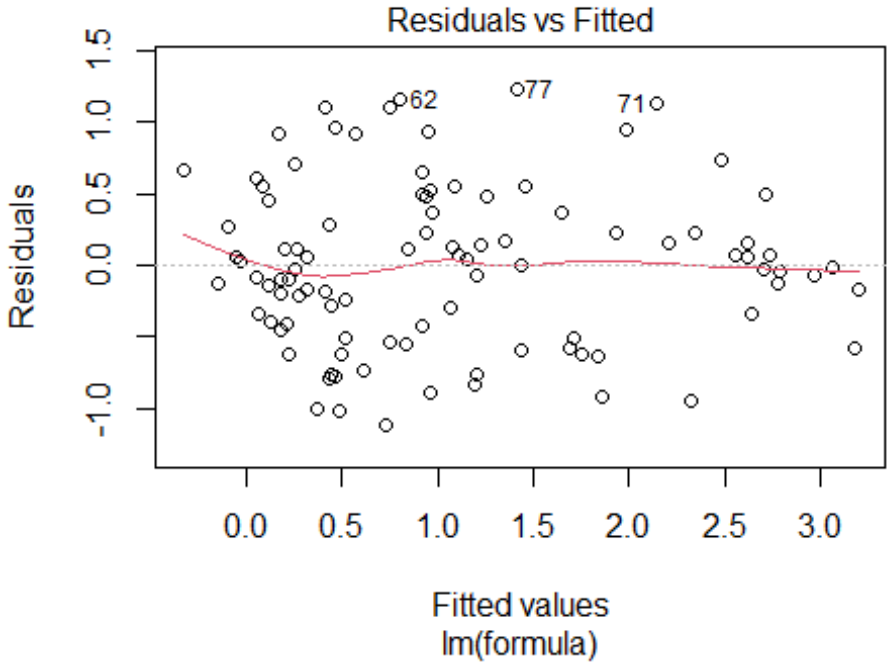


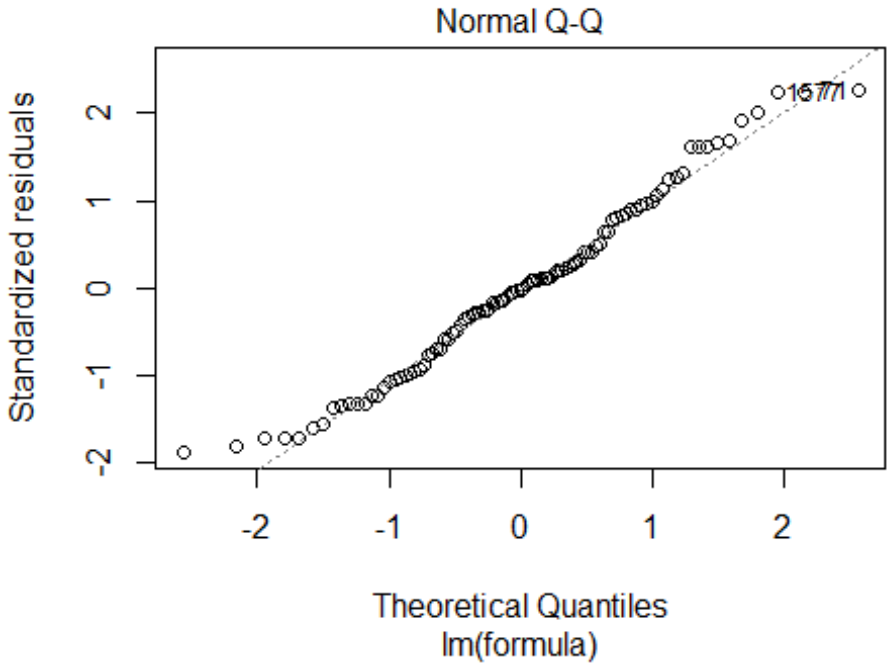

Scale-Location
94
14
11
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
-0.5
0.0
0.5
1.0
Fitted values
lm(formula)

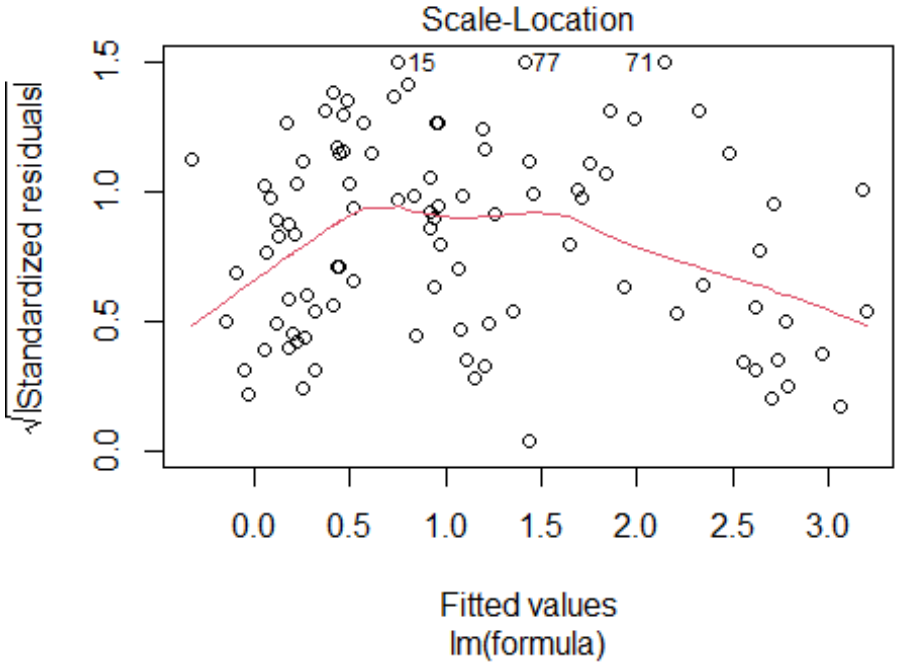

Residuals vs Leverage
94
66
50
Standardized residuals
4
2
0
-2
Cook's distance
1
0.5
0.5
1
0.0
0.1
0.2
0.3
0.4
Leverage
lm(formula)

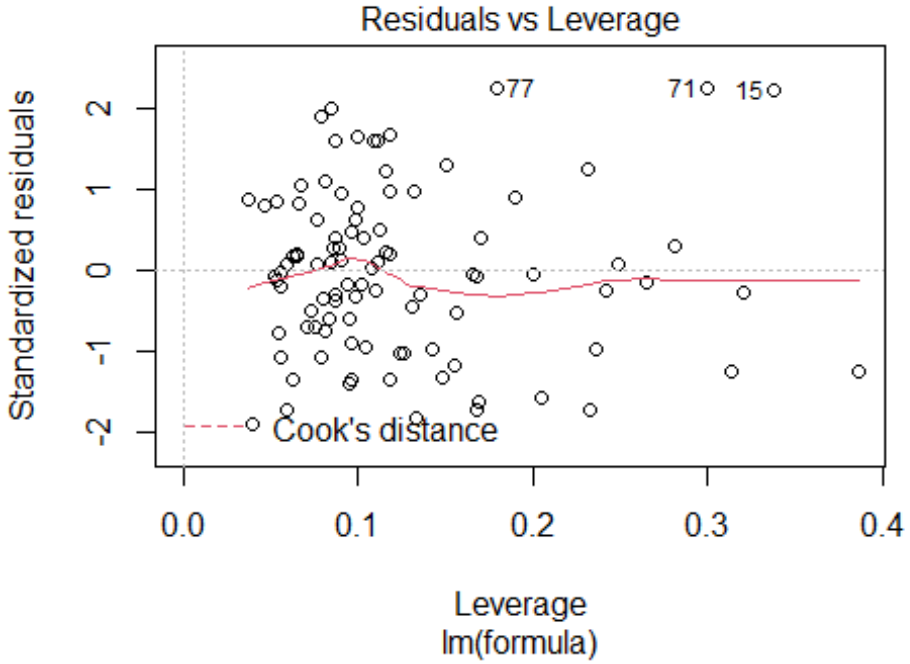

cooks.distance(model_linear)
0.00
0.10
0.20
0.30
0
20
40
60
80
Index

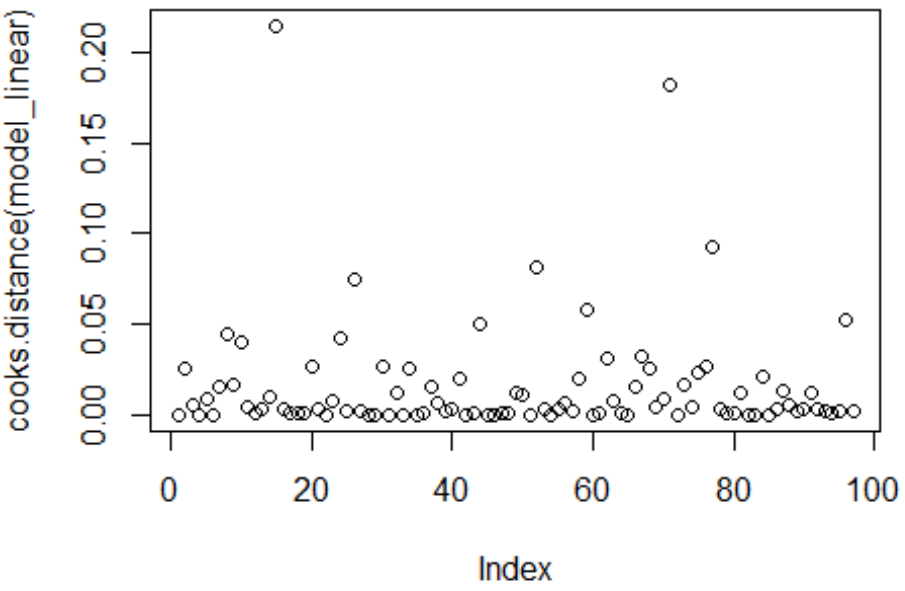
2 Priedas. Tiesinio modelio natūraliam prieaugiui diagnostiniai grafikai.






Scale-Location
15
77
71
√|Standardized residuals|
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Fitted values
lm(formula)


Residuals vs Leverage
77
71
15
Standardized residuals
-2
-1
0
1
2
Cook's distance
0.0
0.1
0.2
0.3
0.4
Leverage
lm(formula)


cooks.distance(model_linear)
0.00
0.05
0.10
0.15
0.20
0
20
40
60
80
100
Index