Projektinis darbas

Vilniaus universitetas
Matematikos ir Informatikos fakultetas
Duomenų mokslas, II kursas

Lukas Janušauskas

Europos švietimo rodiklių analizė

Projektinis duomenų vizualizavimo projektas

2020 gegužė

Tikslas

Ištirti švietimo rodiklius Europos šalyse: skirtumus tarp Europos regionų, mokytojų darbo krūvio ir algų, skirtingų pasiskirstymų.

Užduotys

1. Pavaizduoti duomenis, pasitelkiant PCA.
2. Ištirti PCA komponenčių scree plot.
3. Ištirti laiko, praleisto mokantis, pasiskirtymą Europoje.
4. Ištirti studentų stojimo procento pasiskirtymą Europoje.

Duomenys

Duomenų šaltiniai:

1. **EUROSTAT**. Naudotas mokinių, tenkančių vienam mokytojui rodiklis.
2. **OECD**. Naudotas mokytojų algų rodiklis ir stojimo procentas.
3. **ourworldindata**. Naudotas vidutinis metų skaičius, skirtas mokslui.

Stulpelis	Reikšmė	Tipas
`geo`	ISO 3 simbolių šalies kodas	`chr`
`student_teach_ratio`	Studentų, tenkančių vienam mokytojui vidurkis	`numeric`
`schooling`	Vidutinis skaičius metų, kuriuos skiria mokslui	`numeric`
`enrollment`	Procentas stoajnčiųjų į bakalauro programas	`numeric`
`teacher_pa`	Mokytojų alga	`numeric`
`subregion`	Europos regionas	`chr`

```
schooling <- read.csv('data/expected-years-of-schooling.csv')
schooling <- schooling %>%
  rename(schooling = 'Expected.years.of.schooling',
         geo = 'Code') %>%
  filter(Year == 2022, geo != "") %>%
  select(c('geo', 'schooling'))

ratio_st <- get_eurostat('educ_uoe_perp04')
ratio_st <- ratio_st %>%
  filter(isced11 == 'ED2',
         TIME_PERIOD == '2022-01-01') %>%
  mutate(geo = countrycode(geo, 'eurostat', 'iso3c')) %>%
  rename('student_teach_ratio' = 'values') %>%
  select(c('geo', 'student_teach_ratio'))


students <- read.csv('data/oecd_students.csv')
students <- students %>%
  filter(INST_TYPE_EDU == 'INST_EDU_PUB',
```

```
         EDUCATION_LEV == 'ISCED11_6') %>%
  rename("enrollment" = "OBS_VALUE",
         "geo" = 'REF_AREA') %>%
  select(c('geo', 'enrollment'))

teach_pay <- read.csv('data/oecd_teachers.csv')
teach_pay <- teach_pay %>%
  rename('geo' = 'REF_AREA',
         'teacher_pay' = 'OBS_VALUE') %>%
  filter(PERS_TYPE == 'TE',
         EDUCATION_LEV == 'ISCED11_24',
         REF_PERIOD == 2022) %>%
  select(c('geo', 'teacher_pay'))

df <- ratio_st %>%
  inner_join(schooling, by=join_by('geo')) %>%
  inner_join(students, by=join_by('geo')) %>%
  inner_join(teach_pay, by=join_by('geo'))

cols <- c(colnames(df), 'subregion')

df <- df %>%
  inner_join(ne_countries(), by=join_by("geo" == "iso_a3")) %>%
  select(cols)
```

```
kable(head(df))
```

geo	student_teach_ratio	schooling	enrollment	teacher_pay	subregion
AUT	8.6	16.36746	73.44809	74796.17	Western Europe
BGR	10.6	13.86803	86.35822	NA	Eastern Europe
DEU	12.8	17.34335	76.13349	90235.42	Western Europe
DNK	10.9	18.77403	99.73496	63678.88	Northern Europe
EST	10.1	15.94298	91.45513	32373.76	Northern Europe
GRC	8.1	20.02638	100.00000	29193.91	Southern Europe

PCA

SVD ir scree plot

Trumpai prsitatysiu PCA, kad galėtumėte suprasti sekančius punktus. Pakete, kurį naudojau, pagrindinių komponenčių analizė buvo implementuota, naudojant singular value decomposition. Kiti, modernesni, pasirinkimai buvo t-SNE ir UMAP, tačiau duomenų kiekis nebuvo didelis, todėl naudojau SVD.

Tegu mūsų reikšmės bus matrica X, tada ši matrica faktorizuojama pagal formulę:

$$X = U\Sigma V^T \tag{1}$$

Čia U, V - matricos sudarytos iš kairiųjų ir dešiniųjų tikrinių vektorių (angl. left and right singtular vectors), Σ - diagonalinė matrica, sudaryta iš kairiųjų ir dešiniųjų tikrinių reikšmių (angl. eigenvalues). Šis metodas yra geresnis už kanoninę formą $U\Lambda U^{-1}$ tuo, kad reikalavimai matricai X yra žymiai lankstesni.

Svarbus faktas - Σ pagrindinėje įstrižainėje esančios tikrinės reikšmės(jų šaknys) atitinka paaiškintą dispersiją. Tiriant PCA, dažnai šios dispersijos yra tiriamos, norint nustatyti, kiek komponenčių užteks. Tai pavaizduosiu vadinamu scree plot.

Duomenų paruošimui reikia imputuoti praleistus duomenis. Kadangi duomenyse negali būti praleistų reikšmių, naudosiu CART algoritmą (medžiais paremtą ML-modelį), iš paketo `mice`.

```
raw_data <- select(df, -c('geo', 'subregion'))

imp <- mice(raw_data, method="cart")
```

```
##
##  iter imp variable
##   1   1  student_teach_ratio  enrollment  teacher_pay
##   1   2  student_teach_ratio  enrollment  teacher_pay
##   1   3  student_teach_ratio  enrollment  teacher_pay
##   1   4  student_teach_ratio  enrollment  teacher_pay
##   1   5  student_teach_ratio  enrollment  teacher_pay
##   2   1  student_teach_ratio  enrollment  teacher_pay
##   2   2  student_teach_ratio  enrollment  teacher_pay
##   2   3  student_teach_ratio  enrollment  teacher_pay
##   2   4  student_teach_ratio  enrollment  teacher_pay
##   2   5  student_teach_ratio  enrollment  teacher_pay
##   3   1  student_teach_ratio  enrollment  teacher_pay
##   3   2  student_teach_ratio  enrollment  teacher_pay
##   3   3  student_teach_ratio  enrollment  teacher_pay
##   3   4  student_teach_ratio  enrollment  teacher_pay
##   3   5  student_teach_ratio  enrollment  teacher_pay
##   4   1  student_teach_ratio  enrollment  teacher_pay
##   4   2  student_teach_ratio  enrollment  teacher_pay
##   4   3  student_teach_ratio  enrollment  teacher_pay
##   4   4  student_teach_ratio  enrollment  teacher_pay
##   4   5  student_teach_ratio  enrollment  teacher_pay
##   5   1  student_teach_ratio  enrollment  teacher_pay
##   5   2  student_teach_ratio  enrollment  teacher_pay
##   5   3  student_teach_ratio  enrollment  teacher_pay
##   5   4  student_teach_ratio  enrollment  teacher_pay
##   5   5  student_teach_ratio  enrollment  teacher_pay
```

```
pr_comp_data <- complete(imp)

pr_comp_res <- prcomp(pr_comp_data)
```

Scree plot

Toliau, paprasčiausiai naudojantis paketu **stats** pasinaudojau **prcomp**. Prieš jungiant pagrinines komponentes su duomenimis, nusibrėžkime scree plot ištirti komponenčių tikrines reikšmes, kurios kaip žinome atstoja paaiškintą dispersiją.

```
eigenvalues <- pr_comp_res$sdev ^ 2

expl_var <- eigenvalues / sum(eigenvalues) * 100

plot_data <-   data.frame(
    Komponente = 1:length(eigenvalues),
    `Tikrine reiksme` = eigenvalues,
    expl_var = expl_var
)

ggplot(plot_data) +
    geom_line(aes(x = Komponente, y = expl_var),
              size=2, color="#78003F") +
    geom_point(aes(x = Komponente, y = expl_var),
              size=5, color="#E64164") +
    labs(
      title = "Scree plot",
      x = "Komponentės nr.",
      y = "Paaiškinta dispersija (%)"
    ) +
    theme_minimal() +
    theme(
      plot.title = element_text(hjust = 0.5, size = 20, face = "bold"),
      axis.title = element_text(size = 12),
      axis.text = element_text(size = 10),
      panel.grid.major.x = element_blank(),
      panel.grid.minor = element_blank(),
      legend.position = "bottom"
    )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

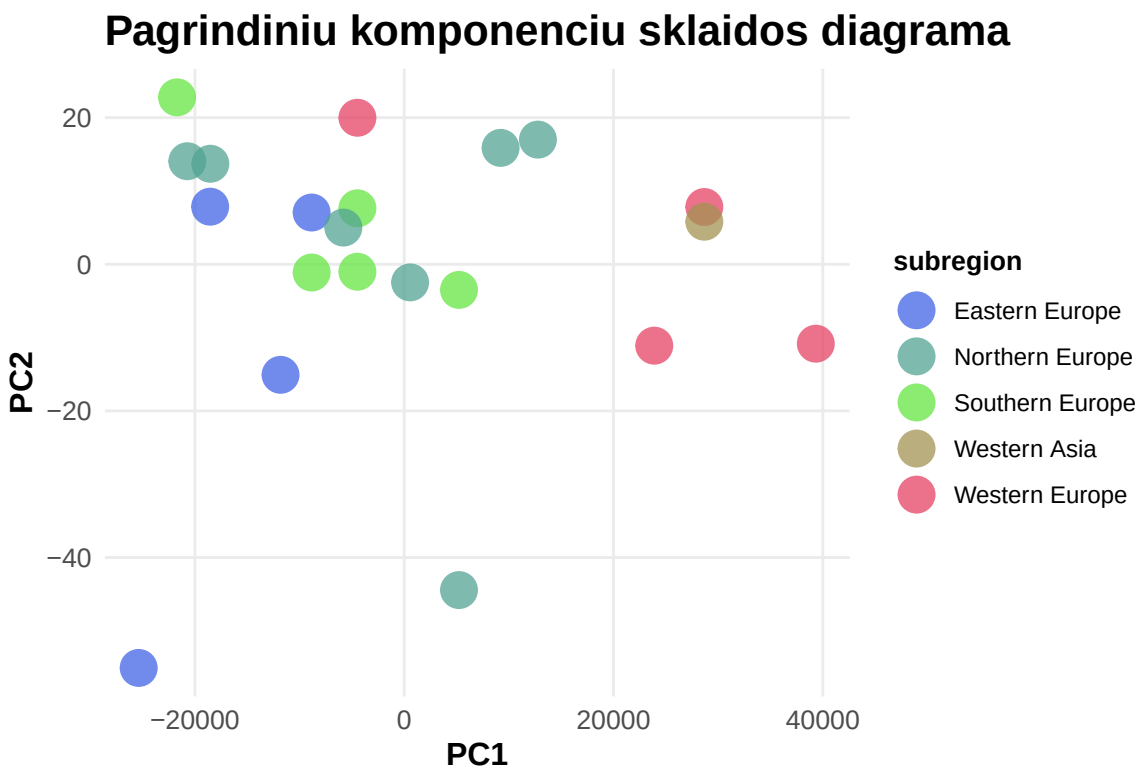
PCA iliustracija

```
pr_comp_decomp <- pr_comp_res[['x']]

df_pca <- cbind(df, pr_comp_decomp)

df_pca %>%
  ggplot() +
    geom_point(aes(x=PC1, y=PC2, color=subregion),
               size=6, alpha=0.75) +
    scale_color_manual(values=colorRampPalette(c("#4164e6", "#64e641", "#e64164"))(5)) +
    theme_minimal() +
    ggtitle("Pagrindinių komponenčių sklaidos diagrama") +
    theme(
      text = element_text(family = "sans"),
      plot.title = element_text(size = 16, face = "bold", hjust = 0),
      plot.subtitle = element_text(size = 12, hjust = 0, margin = margin(b = 20)),
      axis.title = element_text(size = 12, face = "bold"),
      axis.text = element_text(size = 10),
      legend.title = element_text(size = 10, face = "bold"),
      legend.text = element_text(size = 9),
      legend.position = "right",
      plot.margin = margin(t = 20, r = 20, b = 20, l = 20),
      panel.grid.minor = element_blank()
```

```
  )
```



Kaip matome, Vakarų ir Rytų Europa išsiskiria, tačiau skirtumo tarp Šiaurės ir Pietų Europos pagrindines kopmonentės iš mano išrinktų duomenų neatskiria.

Sklaidos diagrama su grotelėmis

Toliau, patikrinau, ar pasikeitė koreliacija tarp mokytojų algos ir mokytojui tenkančio mokinio skaičiaus 2020, 2021 ir 2022 metais. Tam panaudojau paprasčiausią sklaidos diagramą su grotelėmis.

```
teach_pay_all_years <- read.csv('data/oecd_teachers.csv')
teach_pay_all_years <- teach_pay_all_years %>%
  rename('geo' = 'REF_AREA',
         'teacher_pay' = 'OBS_VALUE',
         'year' = 'REF_PERIOD') %>%
  filter(PERS_TYPE == 'TE',
         EDUCATION_LEV == 'ISCED11_24') %>%
  select(c('geo', 'year', 'teacher_pay'))

ratio_st_all <- get_eurostat('educ_uoe_perp04')
```

```
## Dataset query already saved in cache_list.json...

## Reading cache file C:\Users\lukas\AppData\Local\Temp\RtmpKevueg/eurostat/e2d3d9ef4ef34fae284654343bb
```

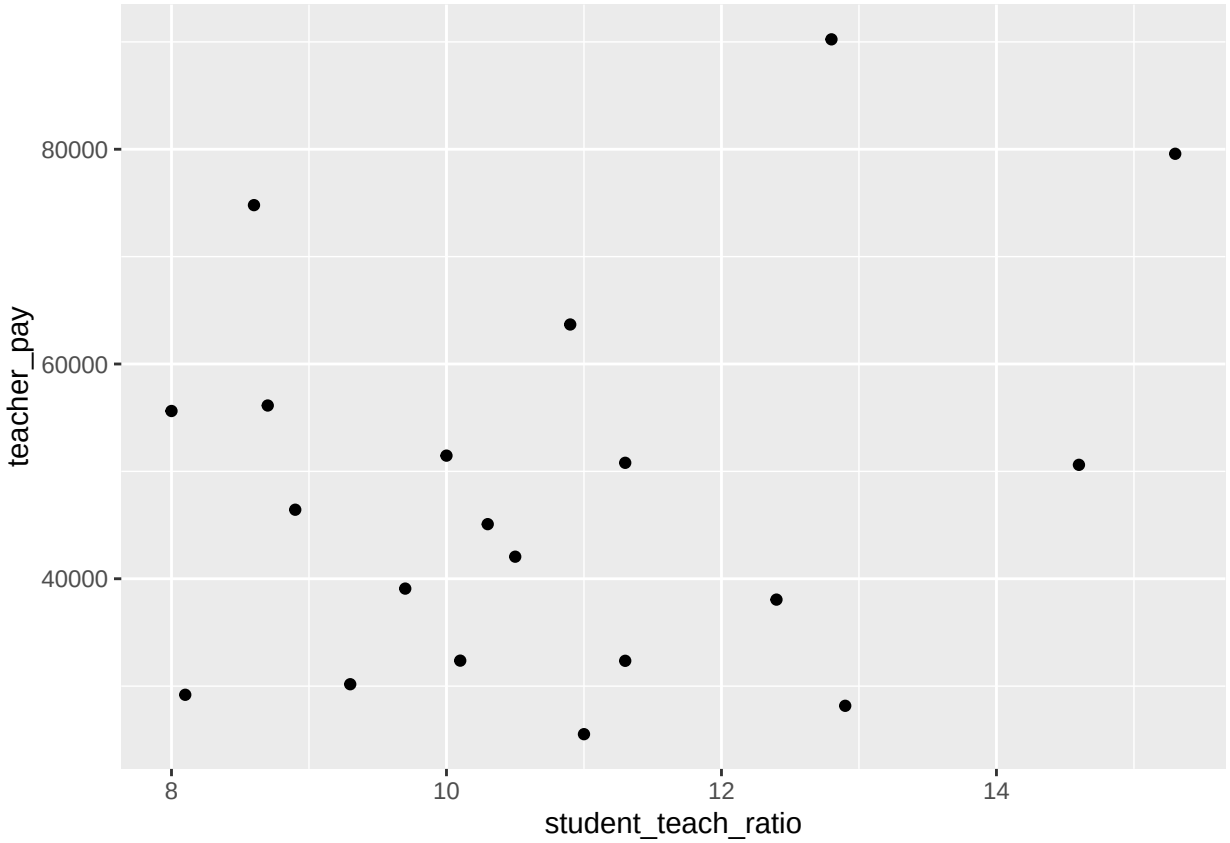
```
## Table  educ_uoe_perp04  read from cache file:  C:\Users\lukas\AppData\Local\Temp\RtmpKevueg/eurostat/
```

```
ratio_st_all <- ratio_st_all %>%
  filter(isced11 == 'ED2',
         ((TIME_PERIOD == '2022-01-01') |
          (TIME_PERIOD == '2021-01-01') |
          (TIME_PERIOD == '2020-01-01'))  )%>%
  mutate(geo = countrycode(geo, 'eurostat', 'iso3c'),
         year = year(TIME_PERIOD)) %>%
  rename('student_teach_ratio' = 'values') %>%
  select(c('geo', 'year', 'student_teach_ratio'))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `geo = countrycode(geo, "eurostat", "iso3c")`.
## Caused by warning:
## ! Some values were not matched unambiguously: EU27_2020
```

```
teach_pay_all_years %>%
  inner_join(ratio_st_all, by = join_by(geo, year)) %>%
  ggplot() +
    geom_point(aes(x=student_teach_ratio, y=teacher_pay))
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Interaktyvus žemėlapis

Bibliografija

https://math.mit.edu/classes/18.095/2016IAP/lec2/SVD_Notes.pdf