



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

BENAS SKIRIUS

Studijų modulio

P160B131 Programavimas duomenų tvarkymui ir vizualizavimui

2 laboratorinio darbo ataskaita

Kaunas, 2023

TURINYS

DARBO TIKSLAS	3
1. UŽDUOTIS	3
1.1. PROGRAMOS KODAS.....	3
2.2. REZULTATAI IR IŠVADOS	3
2. UŽDUOTIS.	4
2.1. PROGRAMOS KODAS.....	4
2.2. REZULTATAI IR IŠVADOS	4
3. UŽDUOTIS	4
3.1. PROGRAMOS KODAS.....	4
4.2. REZULTATAI IR IŠVADOS	5
5. UŽDUOTIS.	5
5.1. PROGRAMOS KODAS.....	5
5.2. REZULTATAI IR IŠVADOS	5
Literatūra	8

DARBO TIKSLAS

Pasitikrinti žinias, įgytas šio modulio metu

1. UŽDUOTIS

Importuoti duomenis ir juos išfiltruoti.

1.1. R PROGRAMOS KODAS

```
library(tidyverse)
library(dplyr)
library(ggplot2)

lab<-read.csv("D:/Downloads/lab_sodra.csv")

#Isrenkami tik reikiami duomenys
filtered_df <- subset(lab, lab[,5] == 452000)

#Isimami stulpeliai su daugiau nei 50% trukstamu reiksmiu
remove_columns_with_many_nas <- function(df, threshold = 0.5) {
  max_nas <- nrow(df) * threshold
  df_filtered <- df[, colSums(is.na(df)) <= max_nas]
  return(df_filtered)
}
df_filtered <- remove_columns_with_many_nas(filtered_df, threshold = 0.5)

#avgWage stulpelyje trukstamos vertes uzpildomos vidutinemis vertemis pagal imone
df_filtered <- df_filtered %>%
  group_by(name) %>%
  filter(!all(is.na(avgWage)))
  mutate(avgWage = ifelse(is.na(avgWage), mean(avgWage, na.rm = TRUE),
avgWage)) %>%
  ungroup()

#tax stulpelyje trukstamos vertes uzpildomos vidutinemis vertemis pagal imone
df_filtered <- df_filtered %>%
  group_by(name) %>%
  filter(!all(is.na(tax)))
  mutate(tax = ifelse(is.na(tax), mean(tax, na.rm = TRUE), tax)) %>%
  ungroup()
  filter(!all(is.na(tax)))

lab <- df_filtered
```

2.2. REZULTATAI IR IŠVADOS

Importuojami duomenis. Iš duomenų atrenkame duomenis su ecoActCode 452000. Išimame stulpelius su daugiau, nei 50% trūkstamų verčių, o stulpelius turinčius mažiau, nei 50% trūkstamų verčių užpildome vidutinėmis vertėmis. Imanes, kurios neturi duomenų bet kuriame stulpelyje išimame.

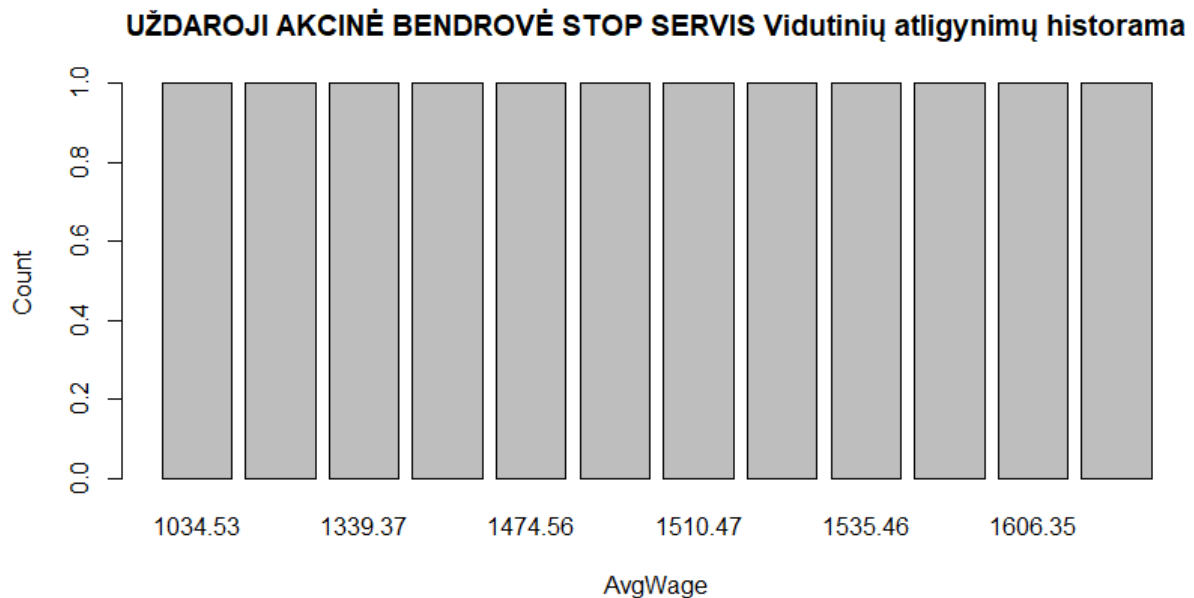
2. UŽDUOTIS.

Nubrėžkite histogramą vidutiniam atlyginimui.

2.1. R PROGRAMOS KODAS

```
alga <- subset(lab, lab[,3] == "UŽDAROJI AKCINĖ BENDROVĖ STOP SERVIS")  
  
my_bar<-barplot(table(alga[,8]),xlab="AvgWage",ylab="Count")  
title(main = "UŽDAROJI AKCINĖ BENDROVĖ STOP SERVIS Vidutinių atligynimų  
historama")
```

2.2. REZULTATAI IR IŠVADOS



3. UŽDUOTIS

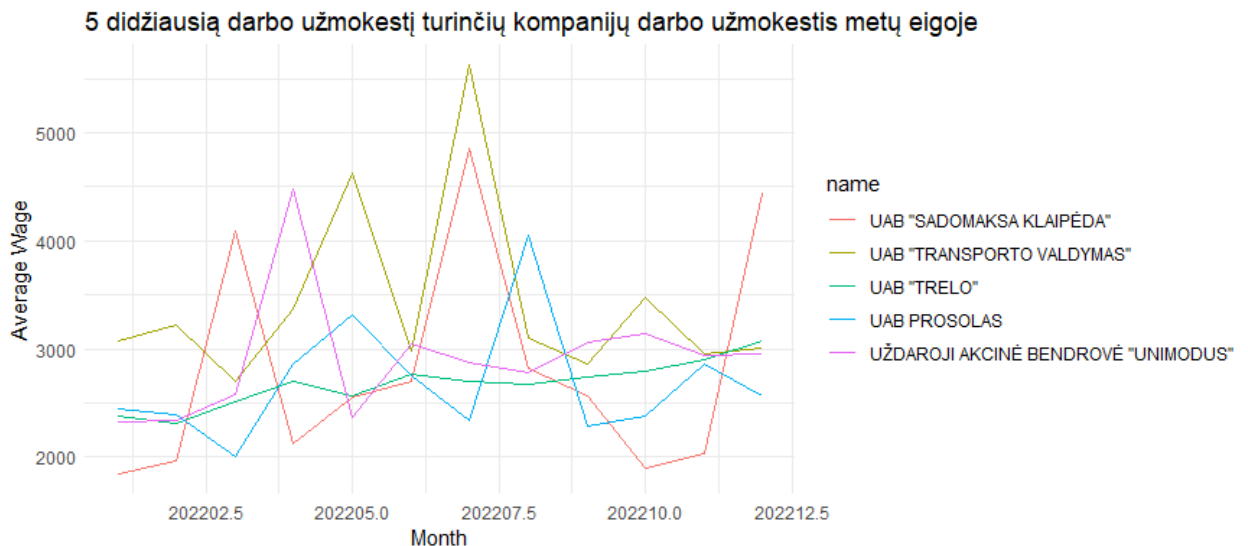
Išrinkite 5 įmones, kurių vidutinio darbo užmokestis buvo didžiausias pagal nurodytą veiklos sritį. Atvaizduokite šių įmonių vidutinio atlyginimo kitimo dinamiką metų eigoje.

3.1. R PROGRAMOS KODAS

```
company_avg_wages <- lab %>%  
  group_by(name) %>%  
  summarize(AverageAvgWage = mean(avgWage, na.rm = TRUE)) %>%  
  ungroup() %>%  
  arrange(desc(AverageAvgWage))  
  
top_5_companies <- head(company_avg_wages, 5)  
filtered_df <- lab %>%  
  filter(name %in% top_5_companies$name)  
  
ggplot(filtered_df, aes(x = month, y = avgWage, color = name)) +  
  geom_line() +  
  labs(title = "5 didžiausią darbo užmokestį turinčių kompanijų darbo  
užmokestis metų eigoje",
```

```
x = "Month",
y = "Average Wage") +
theme_minimal()
```

3.2. REZULTATAI IR IŠVADOS



4. UŽDUOTIS.

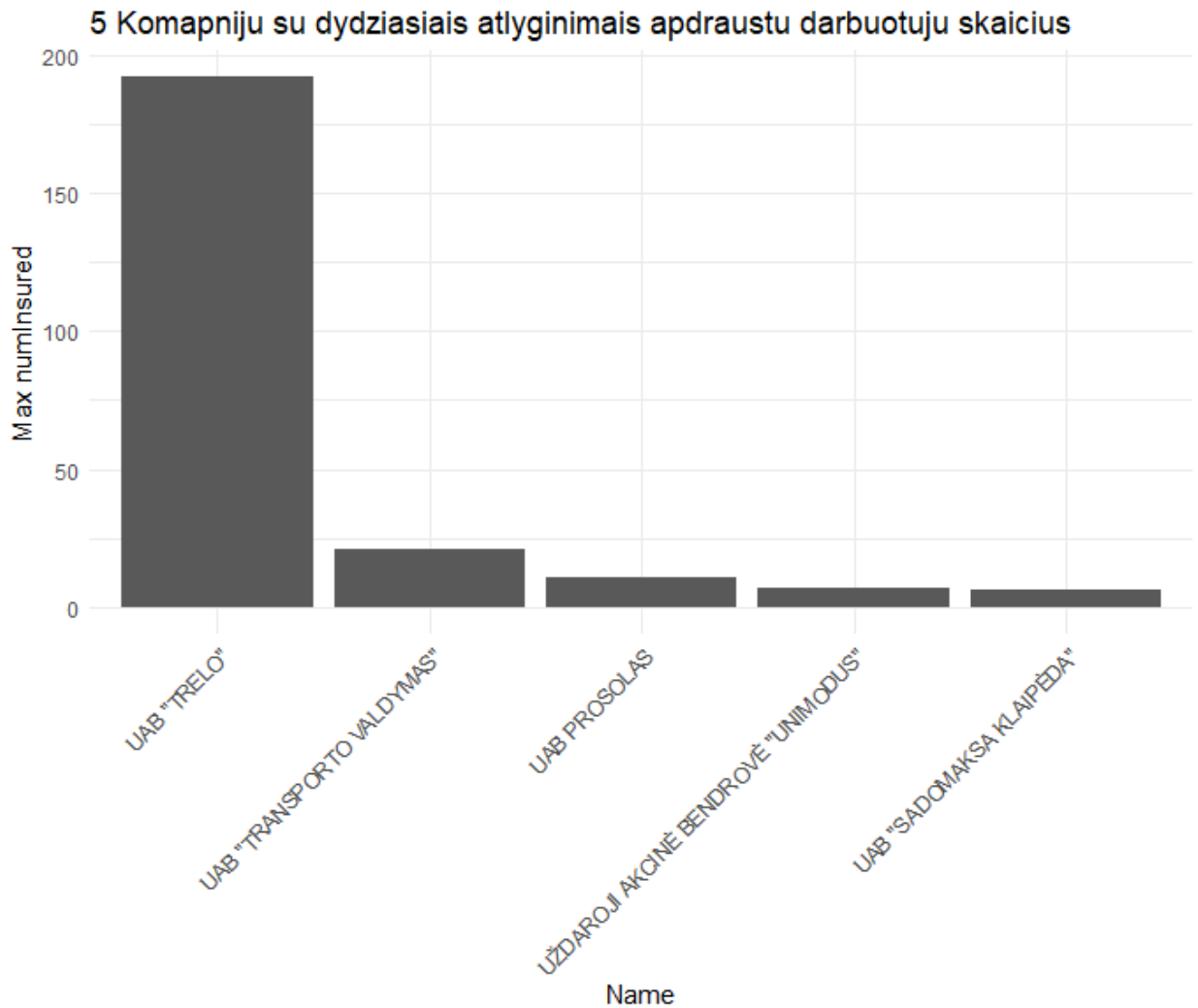
Iš anksčiau išrinktų 5 įmonių, išrinkite maksimalų apdraustų darbuotojų skaičių per šiuos metus. Atvaizduokite stulpelinėje diagrama mažėjimo tvarka.

4.1. R PROGRAMOS KODAS

```
max_numInsured <- filtered_df %>%
  group_by(name) %>%
  summarize(MaxNumInsured = max(numInsured, na.rm = TRUE)) %>%
  ungroup()

ggplot(max_numInsured, aes(x = reorder(name, -MaxNumInsured), y =
MaxNumInsured)) +
  geom_bar(stat = "identity") +
  labs(title = "5 Komapniju su dydziasiais atlyginimais apdraustu darbuotuju
skaicius",
x = "Name",
y = "Max numInsured") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

4.2.REZULTATAI IR IŠVADOS



5. UŽDUOTIS.

Sukurti web aplikaciją pagal nurodytą veiklos sritį.

Aplikacijos tikslas – vizualizuoti atlyginimų dinamiką (plotOutput), įvedus įmonės kodą.

5.1. R PROGRAMOS KODAS

```
library(shiny)
library(tidyverse)

ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      selectizeInput(inputId = "imones_kodas", label = "Imones Kodas",
choices = NULL, selected = NULL)
    ),
    mainPanel(tabsetPanel(
      tabPanel("grafikas", plotOutput("plot")),
      tabPanel("lentele", tableOutput("table"))
    )
  )
)
```

```

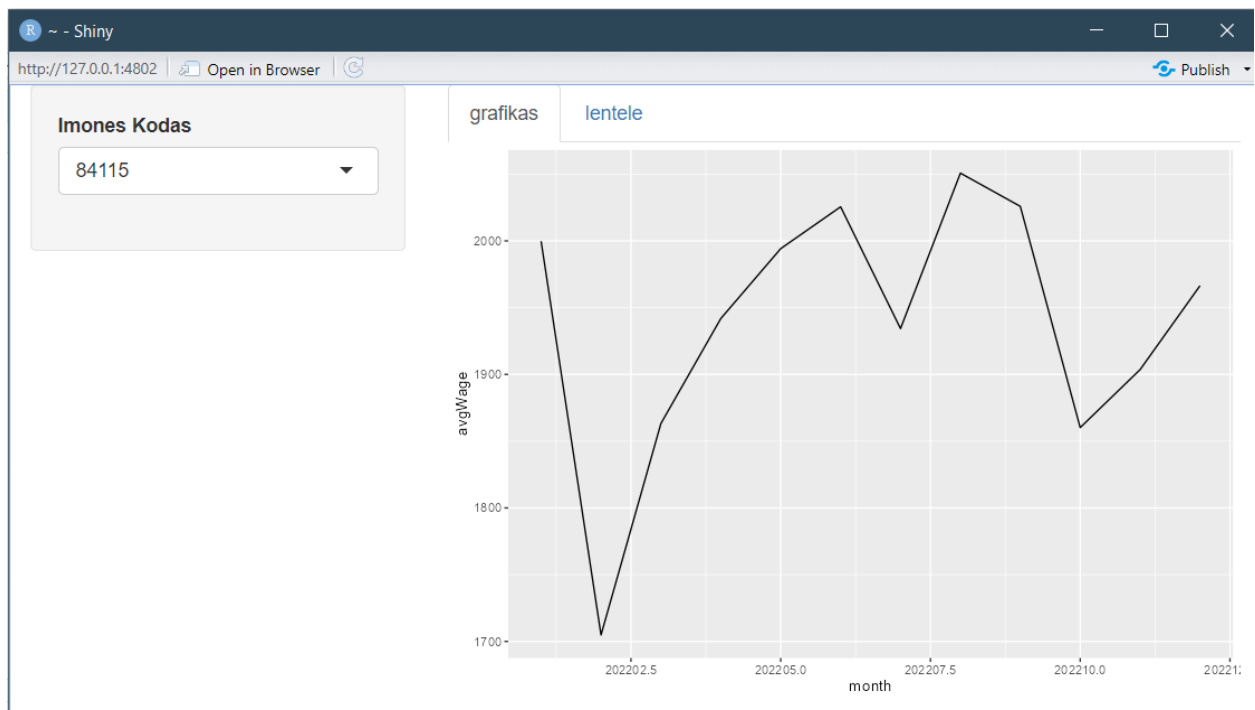
    )
  }
  server <- function(input, output, session) {
    data <- lab
    updateSelectizeInput(session, "imones_kodas", choices = data$code, server =
TRUE)

    output$table <- renderTable(
      data %>%
        filter(code == input$imones_kodas) , digits = 0
    )

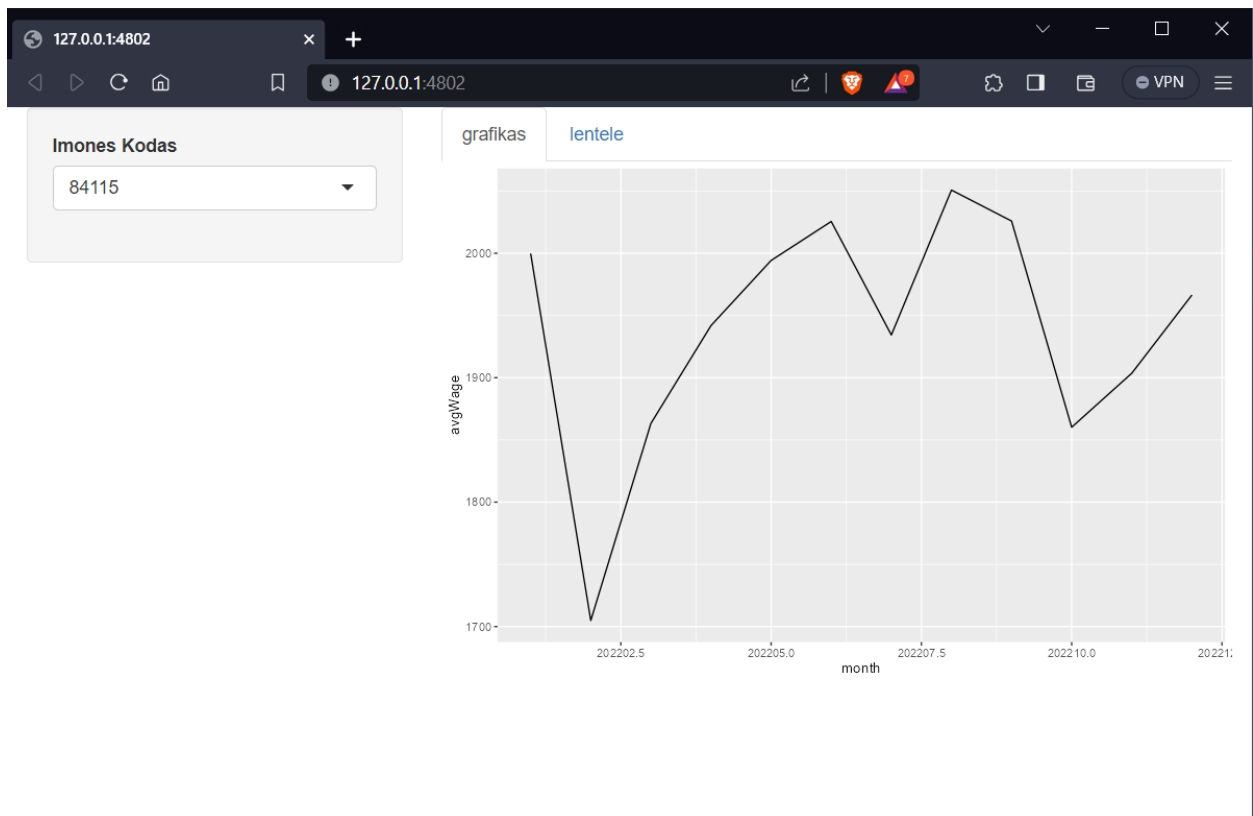
    output$plot <- renderPlot(
      data %>%
        filter(code == input$imones_kodas) %>%
        ggplot(aes(x = month, y = avgWage)) +
        geom_line()
    )
  }
  shinyApp(ui, server)

```

5.2.REZULTATAI IR IŠVADOS



1 pav. Shiny aplikacija veikianti R lange.



2 pav. Shiny aplikacija veikianti naršyklėje.

LITERATŪRA

1. Paskaitų skaidrės.
2. R dokumentacija.