Nekilnojamojo turto objektų kainų analizė Lietuvoje

Statistikos laboratorinis darbas Nr. 2

VU

2025-04-17

Table of Contents

# Įvadas

Šiame tyrime analizuojami Lietuvos nekilnojamojo turto rinkos duomenys, siekiant nustatyti įvairius dėsningumus ir statistines priklausomybes.

# Duomenų aprašymas

Analizei naudojami duomenys buvo atsisiųsti iš [Lithuanian Real Estate Listings](https://github.com/valdas-v1/lithuanian-real-estate-listings) GitHub repozitorijos. Duomenys buvo surinkti 2024 m. vasarį iš [Aruodas.lt](https://www.aruodas.lt/) puslapio. Duomenų rinkinyje yra informacija apie parduodamus ir nuomojamus butus, garažus, namus, sklypus ir patalpas.

Pasirinktus naudojamui duomenis apima:

* Kaina (price) - pardavimo arba nuomos kaina
* Įrengimas (equipment) - būsto ar pastato įrengimo lygis
* Peržiūrų skaicius (views total) - bendras peržiūrų skaičius, rodo kiek dėmesio sulaukia patalpos
* Plotas (area) - nurodytas patalpų plotas
* Pastatų tipas (building type) - pastato rūšis pagal jo paskirtį ir struktūrą
* Privatūs pardavėjai (private seller) - asmenys parduodantys turtą be tarpininkų
* Parduoti arba išnomuoti pastatai (sold\_or\_rented)

## Duomenų nuskaitymas

data\_dir <- "C:/Users/zabit/Documents/GitHub/Statistikos-lab-2/data"  
  
folders <- list.dirs(data\_dir, full.names = FALSE, recursive = FALSE)  
  
kable(data.frame(Kategorijos = folders),   
 caption = "Nekilnojamojo turto duomenų kategorijos") %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Nekilnojamojo turto duomenų kategorijos

Kategorijos

apartments

apartments\_rent

garages\_parking

garages\_parking\_rent

house\_rent

houses

land

land\_rent

premises

premises\_rent

csv\_data\_list <- list()  
  
for (folder in folders) {  
 file\_path <- file.path(data\_dir, folder, "all\_cities\_20240214.csv")  
 if (file.exists(file\_path)) {  
 df <- read.csv(file\_path)  
 csv\_data\_list[[folder]] <- df  
 }  
}

## Duomenų patikrinimas ir išskirčių šalinimas

Prieš pradedant statistinę analizę, būtina identifikuoti ir pašalinti galimai klaidingas ar nekorektiškas reikšmes duomenyse. Nekilnojamojo turto rinkoje egzistuoja neįprastai didelių ar mažų kainų, kurios gali atsirasti dėl duomenų įvedimo klaidų, klaidingo formato ar kitų priežasčių. Tokios išskirtys gali reikšmingai paveikti statistinės analizės rezultatus.

# Apibrėžiame kainų ribas išskirčių identifikavimui  
min\_threshold <- 20 # Minimali kaina eurais  
max\_threshold <- 25000000 # Maksimali kaina eurais  
  
# Sukuriame rezultatų lentelę  
removal\_results <- data.frame(  
 Kategorija = character(),  
 Pašalinta\_eilučių = integer(),  
 Per\_didelės\_kainos = integer(),  
 Per\_mažos\_kainos = integer(),  
 stringsAsFactors = FALSE  
)  
  
# Tikriname ir šaliname išskirtis kiekviename duomenų rinkinyje  
for (type in names(csv\_data\_list)) {  
 if (!is.null(csv\_data\_list[[type]]) && "price" %in% colnames(csv\_data\_list[[type]])) {  
 extreme\_high <- sum(csv\_data\_list[[type]]$price > max\_threshold, na.rm = TRUE)  
 extreme\_low <- sum(csv\_data\_list[[type]]$price < min\_threshold, na.rm = TRUE)  
 extreme\_total <- extreme\_high + extreme\_low  
   
 if (extreme\_total > 0) {  
 # Išsaugome pradinį eilučių skaičių  
 original\_count <- nrow(csv\_data\_list[[type]])  
   
 # Filtruojame duomenis, išlaikydami tik patikimas kainas arba NA reikšmes  
 csv\_data\_list[[type]] <- csv\_data\_list[[type]][  
 (csv\_data\_list[[type]]$price >= min\_threshold &   
 csv\_data\_list[[type]]$price <= max\_threshold) |   
 is.na(csv\_data\_list[[type]]$price), ]  
   
 # Fiksuojame rezultatus  
 new\_count <- nrow(csv\_data\_list[[type]])  
 removed\_count <- original\_count - new\_count  
   
 # Pridedame rezultatus į suvestinę  
 removal\_results <- rbind(removal\_results, data.frame(  
 Kategorija = type,  
 Pašalinta\_eilučių = removed\_count,  
 Per\_didelės\_kainos = extreme\_high,  
 Per\_mažos\_kainos = extreme\_low  
 ))  
 }  
 }  
}  
  
# Atvaizduojame išskirčių šalinimo rezultatus  
kable(removal\_results,   
 caption = "Išskirčių šalinimo rezultatų suvestinė") %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Išskirčių šalinimo rezultatų suvestinė

Kategorija

Pašalinta\_eilučių

Per\_didelės\_kainos

Per\_mažos\_kainos

land\_rent

2

0

2

premises

65

64

1

premises\_rent

192

159

33

# Patikriname duomenų rinkinių dydžius po valymo  
data\_sizes <- data.frame(  
 Eilučių\_skaičius = sapply(csv\_data\_list, nrow),  
 Stulpelių\_skaičius = sapply(csv\_data\_list, ncol)  
)  
  
kable(data\_sizes,   
 caption = "Duomenų rinkinių dydžiai po išskirčių šalinimo") %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Duomenų rinkinių dydžiai po išskirčių šalinimo

Eilučių\_skaičius

Stulpelių\_skaičius

apartments

7721

38

apartments\_rent

3208

38

garages\_parking

497

28

garages\_parking\_rent

307

27

house\_rent

310

40

houses

7284

39

land

6322

27

land\_rent

102

27

premises

1491

37

premises\_rent

2547

37

# Randame unikalius stulpelių pavadinimus visuose duomenų rinkiniuose  
all\_columns <- unique(unlist(lapply(csv\_data\_list, colnames)))  
unique\_columns <- sort(all\_columns)  
  
# Analizuojame stulpelių pasikartojimą skirtinguose duomenų rinkiniuose  
column\_presence <- data.frame(  
 Stulpelis = unique\_columns,  
 Pasikartojimų\_skaičius = sapply(unique\_columns, function(col) {  
 sum(sapply(csv\_data\_list, function(df) col %in% colnames(df)))  
 }),  
 stringsAsFactors = FALSE  
)  
  
# Rikiuojame pagal pasikartojimų skaičių mažėjimo tvarka  
column\_presence <- column\_presence[order(column\_presence$Pasikartojimų\_skaičius, decreasing = TRUE),]  
  
# Atvaizduojame unikalių stulpelių analizę  
kable(column\_presence,   
 caption = paste("Unikalių stulpelių pasikartojimas duomenų rinkiniuose (iš viso:",  
 nrow(column\_presence), "stulpeliai)"),  
 row.names = FALSE) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Unikalių stulpelių pasikartojimas duomenų rinkiniuose (iš viso: 52 stulpeliai)

Stulpelis

Pasikartojimų\_skaičius

add\_date

10

call\_forwarding

10

coordinates

10

description

10

images

10

link

10

listing\_id

10

microdistrict

10

modified

10

phone\_number

10

price

10

private\_seller

10

region

10

reserved

10

selected

10

sold\_or\_rented

10

street

10

type\_id

10

unique\_item\_number

10

valid\_till

10

views\_today

10

views\_total

10

description\_tags

9

area

8

additional\_equipment

6

build\_year

6

building\_energy\_efficiency\_class

6

equipment

6

heating\_system

6

house\_no.

6

no.\_of\_floors

6

security

6

additional\_premises

4

building\_type

4

floor

4

number\_of\_rooms

4

purpose

4

type

4

water\_system

4

price\_per\_month

3

accommodates\_no.\_of\_cars

2

area\_.a.

2

closest\_body\_of\_water

2

distance\_from\_body\_of\_water

2

features

2

flat\_no.

2

lot\_no.

2

number

2

plot\_area

2

premises\_nr.

2

object

1

premises\_sum

1

# Duomenų vizualizacija

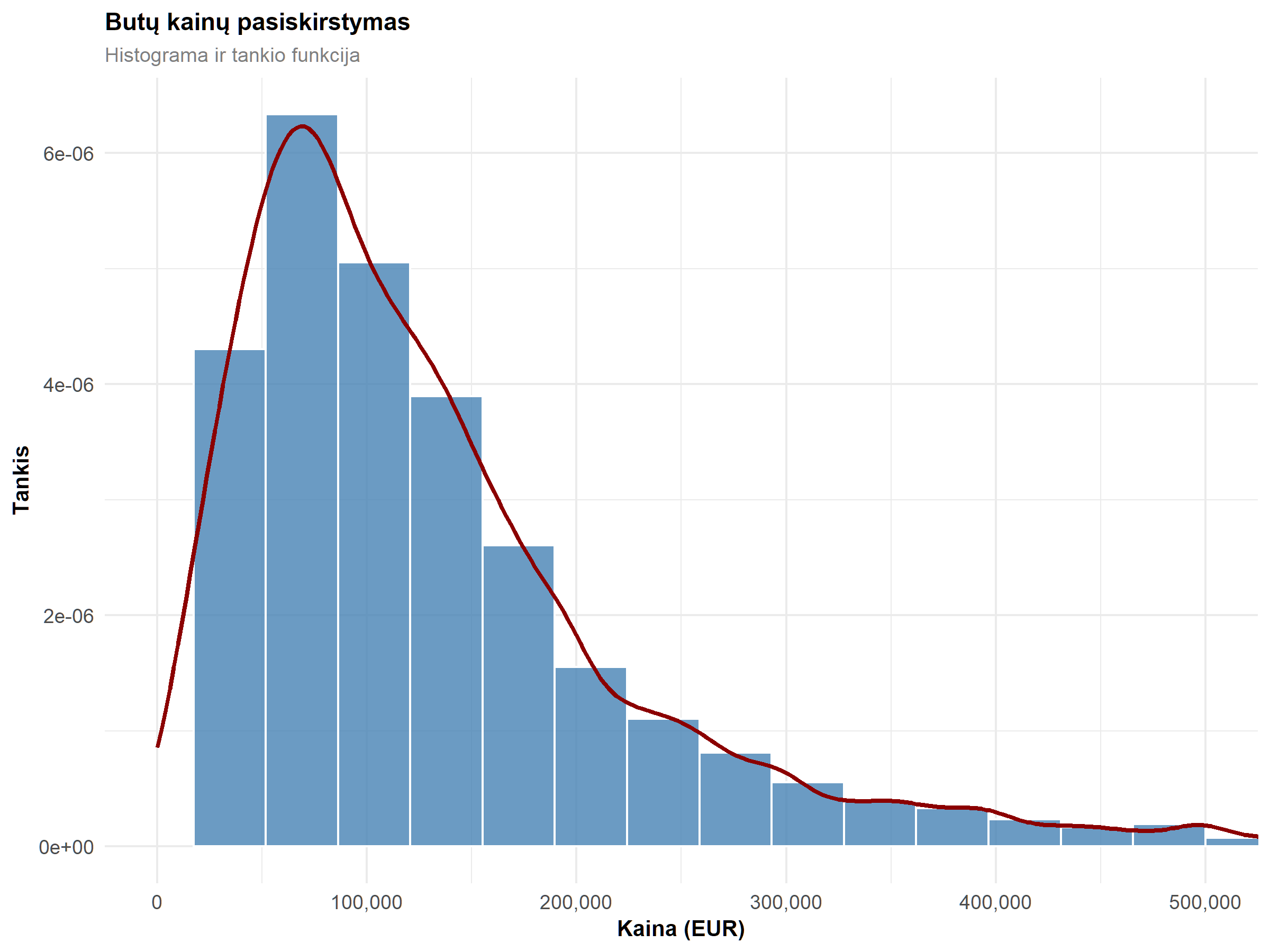
Grafikai padės geriau suprasti Lietuvos nekilnojamojo turto rinkos ypatybes.

# Nustatome bendrą grafikų stilių  
theme\_scientific <- function() {  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(face = "bold", size = 11),  
 plot.subtitle = element\_text(size = 9, color = "gray50"),  
 axis.title = element\_text(face = "bold", size = 10),  
 axis.text = element\_text(size = 9),  
 legend.title = element\_text(face = "bold", size = 9),  
 legend.text = element\_text(size = 8)  
 )  
}

## Kainų pasiskirstymo analizė

Analizuojame butų kainų pasiskirstymą, siekdami nustatyti kainų tendencijas ir išsibarstymo charakteristikas.

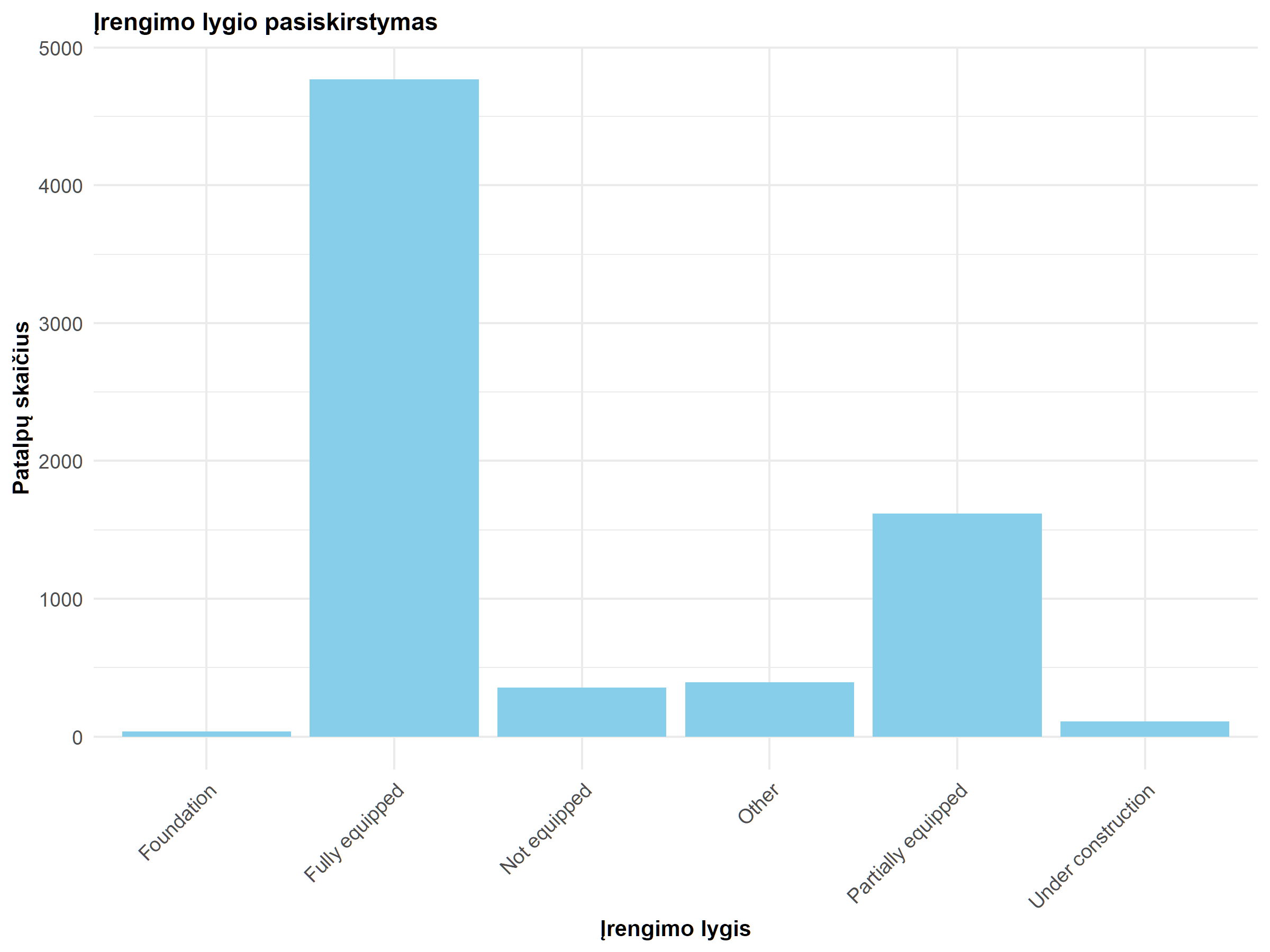
# Butų kainų pasiskirstymo vizualizacija  
if ("apartments" %in% names(csv\_data\_list) && "price" %in% colnames(csv\_data\_list[["apartments"]])) {  
  
 df <- data.frame(price = csv\_data\_list[["apartments"]]$price)  
   
 # Braižome histogramą su tankio kreive  
 price\_hist <- ggplot(df, aes(x = price)) +  
 geom\_histogram(aes(y = after\_stat(density)),  
 bins = 30,   
 fill = "steelblue",   
 color = "white",   
 alpha = 0.8) +  
 geom\_density(color = "darkred", linewidth = 1) +  
 labs(title = "Butų kainų pasiskirstymas",  
 subtitle = "Histograma ir tankio funkcija",  
 x = "Kaina (EUR)",  
 y = "Tankis") +  
 theme\_scientific() +  
 scale\_x\_continuous(labels = comma, limits = c(0, 1000000)) +  
 coord\_cartesian(xlim = c(0, 500000))  
   
 print(price\_hist)  
}



## Įrengimo lygio pasiskirstymo analizė

Analizuojame, kokie įrengimo lygiai yra duomenų rinkinyje.

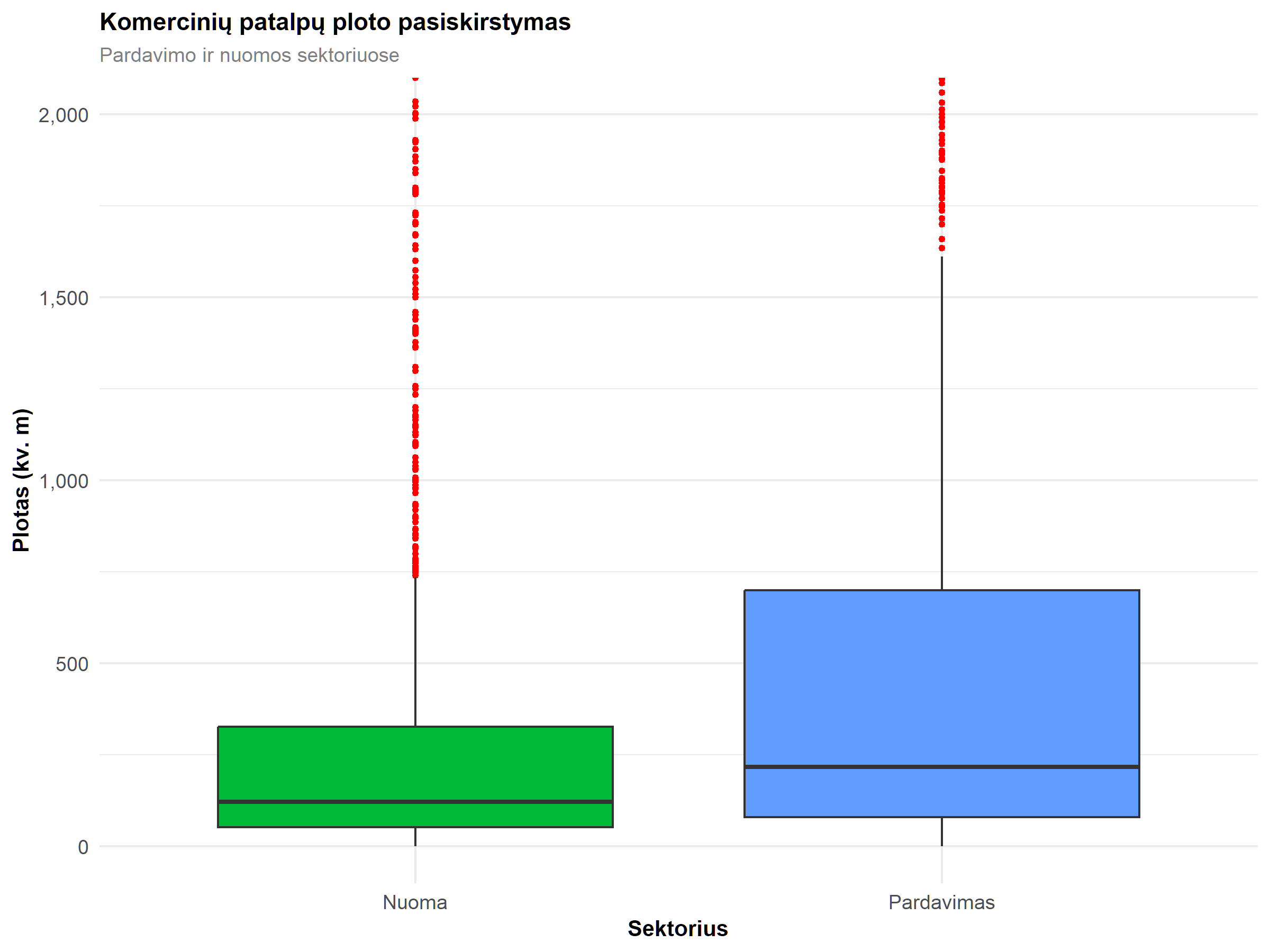
library(readxl)  
duom <- read\_excel("duomenys.xlsx")  
  
equipment\_data <- data.frame(  
 equipment = names(table(duom$equipment)),  
 count = as.vector(table(duom$equipment))  
)  
  
ggplot(equipment\_data, aes(x = equipment, y = count)) +  
 geom\_bar(stat = "identity", fill = "skyblue") +  
 labs(title = "Įrengimo lygio pasiskirstymas",  
 x = "Įrengimo lygis",  
 y = "Patalpų skaičius") +  
 theme\_scientific() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



## Komercinių patalpų ploto analizė

Analizuojame komercinių patalpų ploto pasiskirstymą skirtinguose segmentuose (pardavimas ir nuoma).

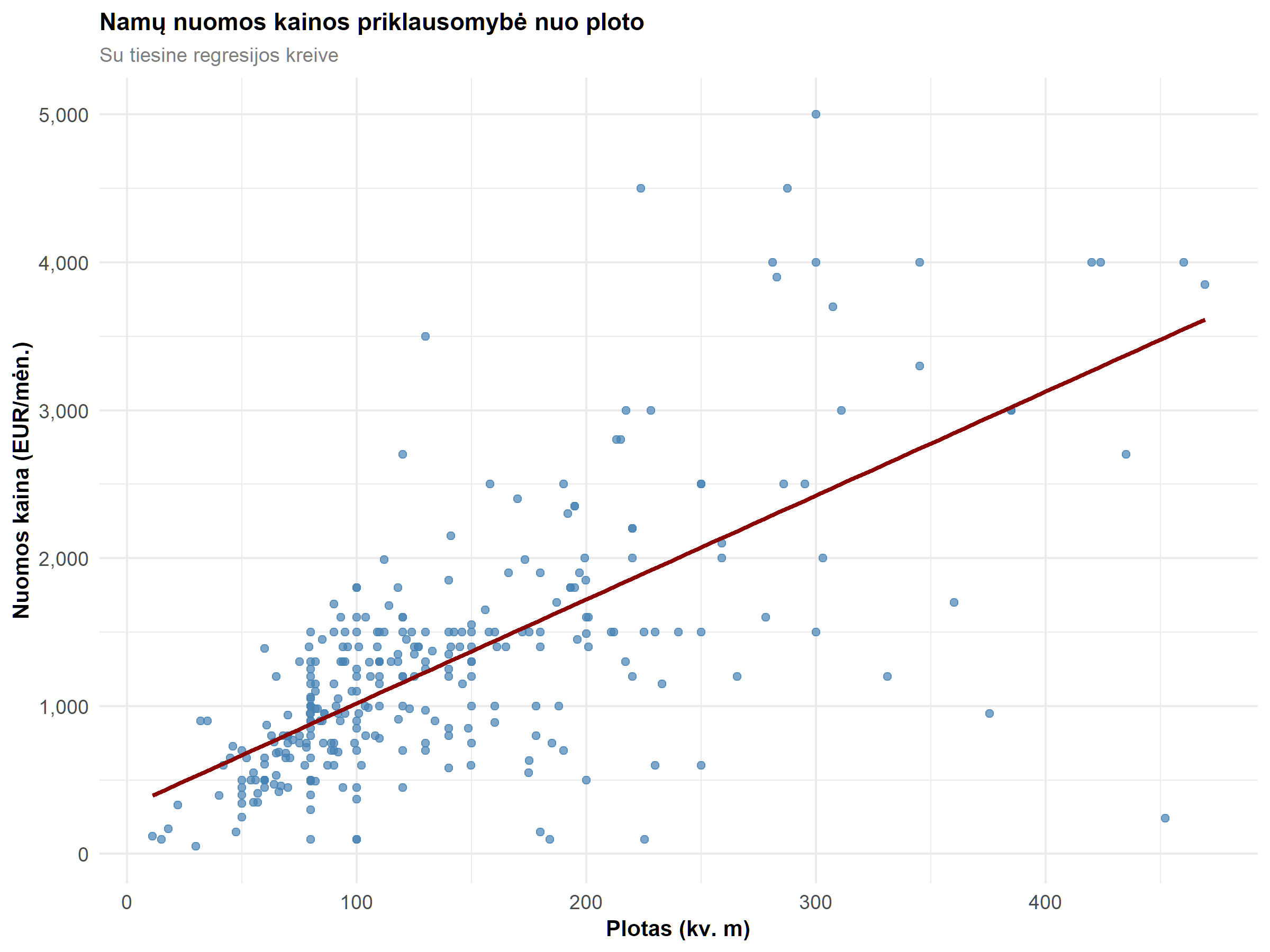
# Komercinių patalpų ploto analizė  
premises\_types <- c("premises", "premises\_rent")  
premises\_data <- list()  
  
# Apjungiame duomenis iš abiejų šaltinių  
for (type in premises\_types) {  
 if (type %in% names(csv\_data\_list) && "area" %in% colnames(csv\_data\_list[[type]])) {  
 df <- csv\_data\_list[[type]]  
 df$type <- ifelse(type == "premises", "Pardavimas", "Nuoma") # Lietuviškas žymėjimas  
   
 # Užtikriname, kad plotas būtų skaitinis  
 df$area <- as.numeric(gsub(",", ".", as.character(df$area)))  
   
 # Atmetame nelogiškus ploto dydžius (pvz., neigiamus ar per didelius)  
 df <- df[!is.na(df$area) & df$area > 0 & df$area < 10000, ]  
   
 # Užtikriname, kad visi stulpeliai būtų vienodi abiem šaltiniam (premises ir premises\_rent)  
 if (length(premises\_data) > 0) {  
 # Nustatome bendrus stulpelius tarp esamo ir pridedamo duomenų rinkinių  
 common\_cols <- intersect(colnames(df), colnames(premises\_data[[1]]))  
 # Paliekame tik bendrus stulpelius  
 df <- df[, common\_cols, drop = FALSE]  
 }  
   
 premises\_data[[type]] <- df  
 }  
}  
  
# Sujungiame duomenis, užtikrindami stulpelių suderinamumą  
if (length(premises\_data) == 2) {  
 # Užtikriname, kad stulpeliai abiem šaltiniuose būtų identiški  
 common\_cols <- intersect(colnames(premises\_data[[1]]), colnames(premises\_data[[2]]))  
 premises\_data[[1]] <- premises\_data[[1]][, common\_cols, drop = FALSE]  
 premises\_data[[2]] <- premises\_data[[2]][, common\_cols, drop = FALSE]  
}  
  
# Sujungiame duomenis  
combined\_premises <- do.call(rbind, premises\_data)  
  
# Braižome boxplot  
area\_boxplot <- ggplot(combined\_premises, aes(x = type, y = area, fill = type)) +  
 geom\_boxplot(outlier.color = "red", outlier.size = 1) +  
 labs(title = "Komercinių patalpų ploto pasiskirstymas",  
 subtitle = "Pardavimo ir nuomos sektoriuose",  
 x = "Sektorius",  
 y = "Plotas (kv. m)") +  
 theme\_scientific() +  
 theme(legend.position = "none") +  
 scale\_fill\_manual(values = c("Pardavimas" = "#619CFF", "Nuoma" = "#00BA38")) +  
 scale\_y\_continuous(labels = comma) +  
 coord\_cartesian(ylim = c(0, 2000))  
  
print(area\_boxplot)



## Namų nuomos kainos ryšys su plotu

Analizuojame, kaip namų nuomos kainų dydis priklauso nuo ploto.

df <- csv\_data\_list[["house\_rent"]]  
  
# Standartizuojame ploto stulpelį: pakeičiame kablelius taškais ir konvertuojame į skaičius  
df$area <- as.numeric(gsub(",", ".", as.character(df$area)))  
  
# Atmetame nelogiškas reikšmes  
df <- df[!is.na(df$area) & !is.na(df$price) &   
 df$area > 0 & df$area < 500 &   
 df$price > 0 & df$price < 6000, ]  
  
# Apskaičiuojame kainą už kvadratinį metrą  
df$price\_per\_sqm <- df$price / df$area  
 # Braižome sklaidos diagramą su regresijos linija  
scatter\_plot <- ggplot(df, aes(x = area, y = price)) +  
 geom\_point(alpha = 0.7, color = "steelblue") +  
 geom\_smooth(method = "lm", color = "darkred", se = FALSE) +  
 labs(title = "Namų nuomos kainos priklausomybė nuo ploto",  
 subtitle = "Su tiesine regresijos kreive",  
 x = "Plotas (kv. m)",  
 y = "Nuomos kaina (EUR/mėn.)") +  
 theme\_scientific() +  
 scale\_color\_viridis\_c() +  
 scale\_y\_continuous(labels = comma) +  
 scale\_x\_continuous(labels = comma)  
  
print(scatter\_plot)



# Pridedame koreliacijos koeficientą  
correlation <- cor(df$area, df$price, use = "complete.obs")  
cat("Koreliacijos koeficientas tarp namų ploto ir nuomos kainos:", round(correlation, 3), "\n")

## Koreliacijos koeficientas tarp namų ploto ir nuomos kainos: 0.692

# Pagrindinės skaitinės charakteristikos

## Pasirinktų kiekybinių kintamųjų aprašomoji statistika

Pateikiame pagrindinės skaitines charakteristikas kiekybiniams kintamiesiems.

filter\_datasets\_by\_column <- function(data\_list, column\_name) {  
 filtered <- data\_list[sapply(data\_list, function(df) column\_name %in% colnames(df))]  
 return(filtered)  
}  
  
# Statistikų skaičiavimas kintamajam  
calculate\_summary <- function(data\_list, variable\_name, target\_datasets) {  
  
 results <- data.frame(  
 Duomenų\_rinkinys = character(),  
 Vidurkis = numeric(),  
 Mediana = numeric(),  
 Moda = character(),  
 Stand\_nuokr = numeric(),  
 Q1 = numeric(),  
 Q3 = numeric(),  
 Minimumas = numeric(),  
 Maksimumas = numeric(),  
 stringsAsFactors = FALSE  
 )  
   
 for (df\_name in target\_datasets) {  
 if (df\_name %in% names(data\_list) && variable\_name %in% colnames(data\_list[[df\_name]])) {  
 # Išskiriame reikšmes ir konvertuojame į skaitinius duomenis  
 values <- data\_list[[df\_name]][[variable\_name]]  
 numeric\_values <- as.numeric(gsub(",", ".", as.character(values)))  
   
 # Pašaliname NA reikšmes skaičiavimams  
 clean\_values <- numeric\_values[!is.na(numeric\_values)]  
   
 if (length(clean\_values) > 0) {  
   
 # Apskaičiuojame papildomas statistikas  
 mean\_val <- mean(clean\_values)  
 median\_val <- median(clean\_values)  
 sd\_val <- sd(clean\_values)  
 quant\_vals <- quantile(clean\_values, probs = c(0.25, 0.5, 0.75))  
 min\_val <- min(clean\_values)  
 max\_val <- max(clean\_values)  
   
 # Pridedame rezultatus į lentelę  
 results <- rbind(results, data.frame(  
 Duomenų\_rinkinys = df\_name,  
 Vidurkis = mean\_val,  
 Mediana = median\_val,  
 Stand\_nuokr = sd\_val,  
 Q1 = quant\_vals[1],  
 Q3 = quant\_vals[3],  
 Minimumas = min\_val,  
 Maksimumas = max\_val  
 ))  
 }  
 }  
 }  
   
 return(results)  
}  
  
# Apibrėžiame analizuojamus kiekybinius kintamuosius  
columns\_to\_check <- c(  
 "price", "price\_per\_month", "views\_total", "area", "area\_.a.",   
 "build\_year", "no.\_of\_floors", "floor", "number\_of\_rooms", "plot\_area"  
)  
  
# Sukuriame sąrašą rezultatams saugoti  
column\_results <- list()  
  
# Apdorojame kiekvieną stulpelį ir saugome rezultatus  
for (col in columns\_to\_check) {  
 column\_results[[col]] <- filter\_datasets\_by\_column(csv\_data\_list, col)  
}

# Apibrėžiame duomenų rinkinio grupes  
sale\_datasets <- c("apartments", "houses", "premises")  
rent\_datasets <- c("apartments\_rent", "house\_rent", "premises\_rent")  
all\_datasets <- c("apartments", "apartments\_rent",  
 "house\_rent", "houses", "premises", "premises\_rent")  
  
sale\_price\_stats <- calculate\_summary(csv\_data\_list, "price", sale\_datasets)  
rent\_price\_stats <- calculate\_summary(csv\_data\_list, "price", rent\_datasets)  
views\_stats <- calculate\_summary(csv\_data\_list, "views\_total", all\_datasets)  
floors\_stats <- calculate\_summary(csv\_data\_list, "no.\_of\_floors", all\_datasets)  
rooms\_stats <- calculate\_summary(csv\_data\_list, "number\_of\_rooms", all\_datasets)  
  
# Atvaizduojame rezultatus lentelėse  
kable(sale\_price\_stats,   
 caption = "Pardavimų kainų statistika pagal nekilnojamojo turto tipą",   
 digits = 2,  
 row.names = FALSE) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Pardavimų kainų statistika pagal nekilnojamojo turto tipą

Duomenų\_rinkinys

Vidurkis

Mediana

Stand\_nuokr

Q1

Q3

Minimumas

Maksimumas

apartments

143718.1

107558

146129.7

64000

172000

43

2.5e+06

houses

183734.4

140000

223884.9

55000

235000

200

4.2e+06

premises

413170.4

165000

762212.4

70000

399850

490

1.0e+07

kable(rent\_price\_stats,   
 caption = "Nuomos kainų statistika pagal nekilnojamojo turto tipą",   
 digits = 2,  
 row.names = FALSE) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Nuomos kainų statistika pagal nekilnojamojo turto tipą

Duomenų\_rinkinys

Vidurkis

Mediana

Stand\_nuokr

Q1

Q3

Minimumas

Maksimumas

apartments\_rent

609.95

525

1529.12

380

690.0

20

84900

house\_rent

1428.76

1200

1327.40

750

1500.0

50

13000

premises\_rent

886472.97

1300

3213628.37

500

5268.5

22

24045000

kable(views\_stats,   
 caption = "Peržiūrų skaičiaus statistika pagal nekilnojamojo turto tipą",   
 digits = 0,  
 row.names = FALSE) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Peržiūrų skaičiaus statistika pagal nekilnojamojo turto tipą

Duomenų\_rinkinys

Vidurkis

Mediana

Stand\_nuokr

Q1

Q3

Minimumas

Maksimumas

apartments

1573

892

2244

425

1860

0

56297

apartments\_rent

1806

606

9703

286

1315

2

355786

house\_rent

1275

582

2332

262

1411

20

24014

houses

2247

1133

3549

501

2612

2

71418

premises

647

310

1296

132

710

0

21298

premises\_rent

742

257

2341

106

607

1

46715

kable(floors\_stats,   
 caption = "Aukštų skaičiaus statistika pagal nekilnojamojo turto tipą",   
 digits = 1,  
 row.names = FALSE) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Aukštų skaičiaus statistika pagal nekilnojamojo turto tipą

Duomenų\_rinkinys

Vidurkis

Mediana

Stand\_nuokr

Q1

Q3

Minimumas

Maksimumas

apartments

5.1

5

3.0

3

5

1

34

apartments\_rent

5.3

5

3.0

4

6

1

34

house\_rent

1.8

2

0.6

1

2

1

4

houses

1.6

2

0.6

1

2

1

15

premises

2.4

2

1.9

1

3

1

18

premises\_rent

2.8

2

2.9

1

3

1

31

kable(rooms\_stats,   
 caption = "Kambarių skaičiaus statistika pagal nekilnojamojo turto tipą",   
 digits = 1,  
 row.names = FALSE) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"))

Kambarių skaičiaus statistika pagal nekilnojamojo turto tipą

Duomenų\_rinkinys

Vidurkis

Mediana

Stand\_nuokr

Q1

Q3

Minimumas

Maksimumas

apartments

2.4

2

1.0

2

3

1

13

apartments\_rent

2.0

2

0.8

1

2

1

10

house\_rent

4.2

4

1.7

3

5

1

13

houses

4.2

4

2.0

3

5

1

54

# Dažnių lentelės parinktiems kategoriniams kintamiesiems.

Kategoriniai\_kintamieji <- c("sold\_or\_rented", "region", "equipment",  
 "building\_type", "private\_seller")  
  
for (kintamasis in Kategoriniai\_kintamieji) {  
 # Dažniai  
 dažniai <- table(duom[[kintamasis]])  
   
   
 lentelė <- data.frame(Kintamasis = rep(kintamasis, length(dažniai)),  
 Kategorija = names(dažniai),  
 Dažnis = as.integer(dažniai))  
   
  
 print(lentelė)  
}

## Kintamasis Kategorija Dažnis  
## 1 sold\_or\_rented FALSE 7272  
## 2 sold\_or\_rented TRUE 12  
## Kintamasis Kategorija Dažnis  
## 1 region Akmenės m. 4  
## 2 region Akmenės r. sav. 28  
## 3 region Alytaus k. 5  
## 4 region Alytaus r. sav. 100  
## 5 region Alytus 28  
## 6 region Anykščių m. 16  
## 7 region Anykščių r. sav. 58  
## 8 region Birštono m. 8  
## 9 region Birštono sav. 8  
## 10 region Biržų m. 23  
## 11 region Biržų r. sav. 22  
## 12 region Druskininkų m. 31  
## 13 region Druskininkų sav. 31  
## 14 region Elektrėnų m. 9  
## 15 region Elektrėnų sav. 58  
## 16 region Ignalinos m. 9  
## 17 region Ignalinos r. sav. 63  
## 18 region Jonavos m. 7  
## 19 region Jonavos r. sav. 66  
## 20 region Joniškio m. 17  
## 21 region Joniškio r. sav. 33  
## 22 region Jurbarko m. 13  
## 23 region Jurbarko r. sav. 44  
## 24 region Kaišiadorių m. 11  
## 25 region Kaišiadorių r. sav. 61  
## 26 region Kalvarijos m. 2  
## 27 region Kalvarijos sav. 15  
## 28 region Kaunas 419  
## 29 region Kauno r. sav. 591  
## 30 region Kazlų Rūdos m. 5  
## 31 region Kazlų Rūdos sav. 17  
## 32 region Kėdainių m. 28  
## 33 region Kėdainių r. sav. 52  
## 34 region Kelmės m. 22  
## 35 region Kelmės r. sav. 68  
## 36 region Klaipėda 127  
## 37 region Klaipėdos r. sav. 651  
## 38 region Kretingos m. 54  
## 39 region Kretingos r. sav. 79  
## 40 region Kupiškio m. 14  
## 41 region Kupiškio r. sav. 27  
## 42 region Lazdijų k. 1  
## 43 region Lazdijų m. 7  
## 44 region Lazdijų r. sav. 48  
## 45 region Marijampolės m. 26  
## 46 region Marijampolės sav. 56  
## 47 region Mažeikių m. 32  
## 48 region Mažeikių r. sav. 49  
## 49 region Molėtų m. 12  
## 50 region Molėtų r. sav. 77  
## 51 region Neringos m. 10  
## 52 region Pagėgių m. 2  
## 53 region Pagėgių sav. 7  
## 54 region Pakruojo k. 1  
## 55 region Pakruojo m. 5  
## 56 region Pakruojo r. sav. 39  
## 57 region Palanga 231  
## 58 region Panevėžio r. sav. 120  
## 59 region Panevėžys 100  
## 60 region Pasvalio m. 14  
## 61 region Pasvalio r. sav. 35  
## 62 region Plungės m. 26  
## 63 region Plungės r. sav. 41  
## 64 region Prienų m. 10  
## 65 region Prienų r. sav. 35  
## 66 region Radviliškio m. 41  
## 67 region Radviliškio r. sav. 46  
## 68 region Raseinių m. 11  
## 69 region Raseinių r. sav. 42  
## 70 region Rietavo m. 3  
## 71 region Rietavo sav. 7  
## 72 region Rokiškio m. 6  
## 73 region Rokiškio r. sav. 36  
## 74 region Skuodo m. 14  
## 75 region Skuodo r. sav. 24  
## 76 region Šakių m. 4  
## 77 region Šakių r. sav. 42  
## 78 region Šalčininkų m. 7  
## 79 region Šalčininkų r. sav. 46  
## 80 region Šiauliai 205  
## 81 region Šiaulių m. sav. 2  
## 82 region Šiaulių r. sav. 216  
## 83 region Šilalės m. 4  
## 84 region Šilalės r. sav. 24  
## 85 region Šilutės m. 11  
## 86 region Šilutės r. sav. 44  
## 87 region Širvintų m. 6  
## 88 region Širvintų r. sav. 54  
## 89 region Švenčionių m. 9  
## 90 region Švenčionių r. sav. 62  
## 91 region Tauragės m. 33  
## 92 region Tauragės r. sav. 34  
## 93 region Telšių m. 38  
## 94 region Telšių r. sav. 38  
## 95 region Trakų m. 17  
## 96 region Trakų r. sav. 171  
## 97 region Ukmergės m. 37  
## 98 region Ukmergės r. sav. 51  
## 99 region Utenos m. 35  
## 100 region Utenos r. sav. 63  
## 101 region Varėnos m. 9  
## 102 region Varėnos r. sav. 49  
## 103 region Vilkaviškio m. 18  
## 104 region Vilkaviškio r. sav. 42  
## 105 region Vilniaus m. sav. 4  
## 106 region Vilniaus r. sav. 791  
## 107 region Vilnius 906  
## 108 region Visagino sav. 3  
## 109 region Zarasų m. 9  
## 110 region Zarasų r. sav. 62  
## Kintamasis Kategorija Dažnis  
## 1 equipment Foundation 37  
## 2 equipment Fully equipped 4768  
## 3 equipment Not equipped 356  
## 4 equipment Other 394  
## 5 equipment Partially equipped 1619  
## 6 equipment Under construction 110  
## Kintamasis Kategorija Dažnis  
## 1 building\_type Blocked house 1246  
## 2 building\_type Farmstead 712  
## 3 building\_type Garden house 510  
## 4 building\_type House 4353  
## 5 building\_type Other 113  
## 6 building\_type Part of the house 350  
## Kintamasis Kategorija Dažnis  
## 1 private\_seller FALSE 5199  
## 2 private\_seller TRUE 2085

# Atlikti tyrimai

Toliau tyrimui atlikti bus remiamasi 5 - 9 užduočių punktais:

* Bus suformuluojamos 6 tyrimo hipotezės iš duomenų rinkinio;
* Užrašomi,kokie testai pasirinkti tyrimo hipotezėms.
* Patikrinama, ar kintamieji tenkina būtinas sąlygas testų taikymui. (Jei netenkina, atliekamos duomenų transformacijos)
* Atliekamas statistinis tyrimas suformuluotoms hipotezėms.
* Pateikiamas tyrimo atsakymas.

## Tyrimas

Tyrimo hipotezė: vidutinė namų kaina regionuose yra 100000 Eur

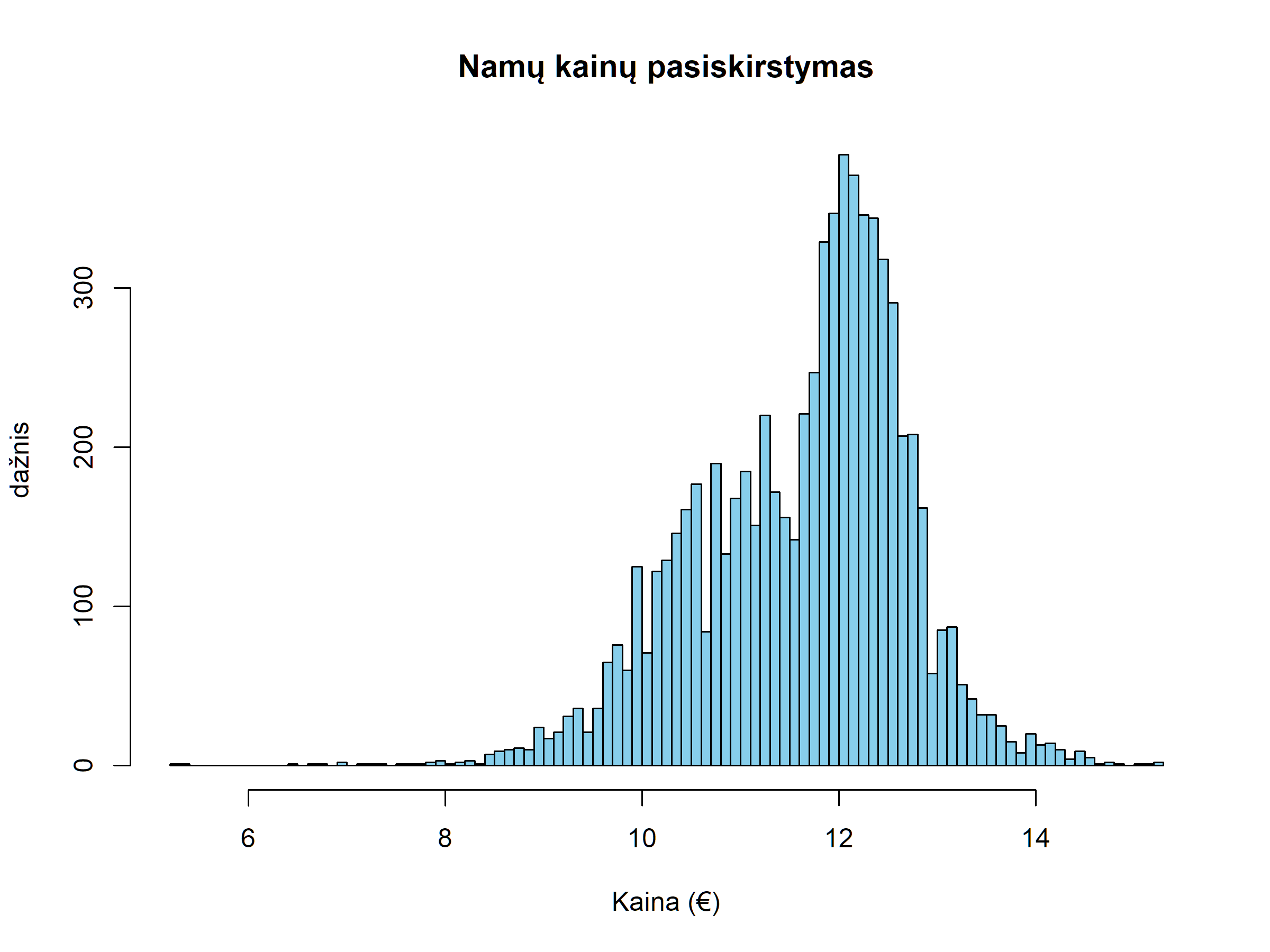
Statistinė hipotezė:

Statistinis testas:

Vienos imties t-testas, kai dispersija nežinoma:

Kadangi kintamasis netenkino būtinų sąlygų testo taikymui, buvo atlikta logaritminė transformacija, siekiant pagerinti normalumą ir taip užtikrinti, kad duomenys atitiktų normalųjį pasiskirstymą, kas yra būtina norint atlikti vienos imties t-testą, kai dispersija nežinoma.

duom$log\_price <- log(duom$price)  
  
hist(duom$log\_price,  
 breaks = 100,   
 col = "skyblue",   
 main = "Namų kainų pasiskirstymas",   
 xlab = "Kaina (€)",  
 ylab = "dažnis")



Statistinis tyrimas:

t.test(duom$log\_price, mu = 100000, paired = FALSE, var.equal = TRUE)

##   
## One Sample t-test  
##   
## data: duom$log\_price  
## t = -7992124, df = 7283, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 1e+05  
## 95 percent confidence interval:  
## 11.60473 11.65378  
## sample estimates:  
## mean of x   
## 11.62926

Statistinė išvada: kadangi -reikšmė () mažesnė už reikšmingumo lygmenį (), tai darome išvadą, kad rastas statistiškai reikšmingas skirtumas, todėl atmetame nulinę hipotezę.

Tyrimo išvada: tyrimo duomenys parodė, kad vidutinė namų kaina nėra lygi 100 000 Eur.

## Tyrimas

Tyrimo hipotezė: 60 procentų namų yra įrengti pilnai arba dalinai

Statistinė hipotezė:

Statistinis testas:

Vienos imties proporcijų testas:

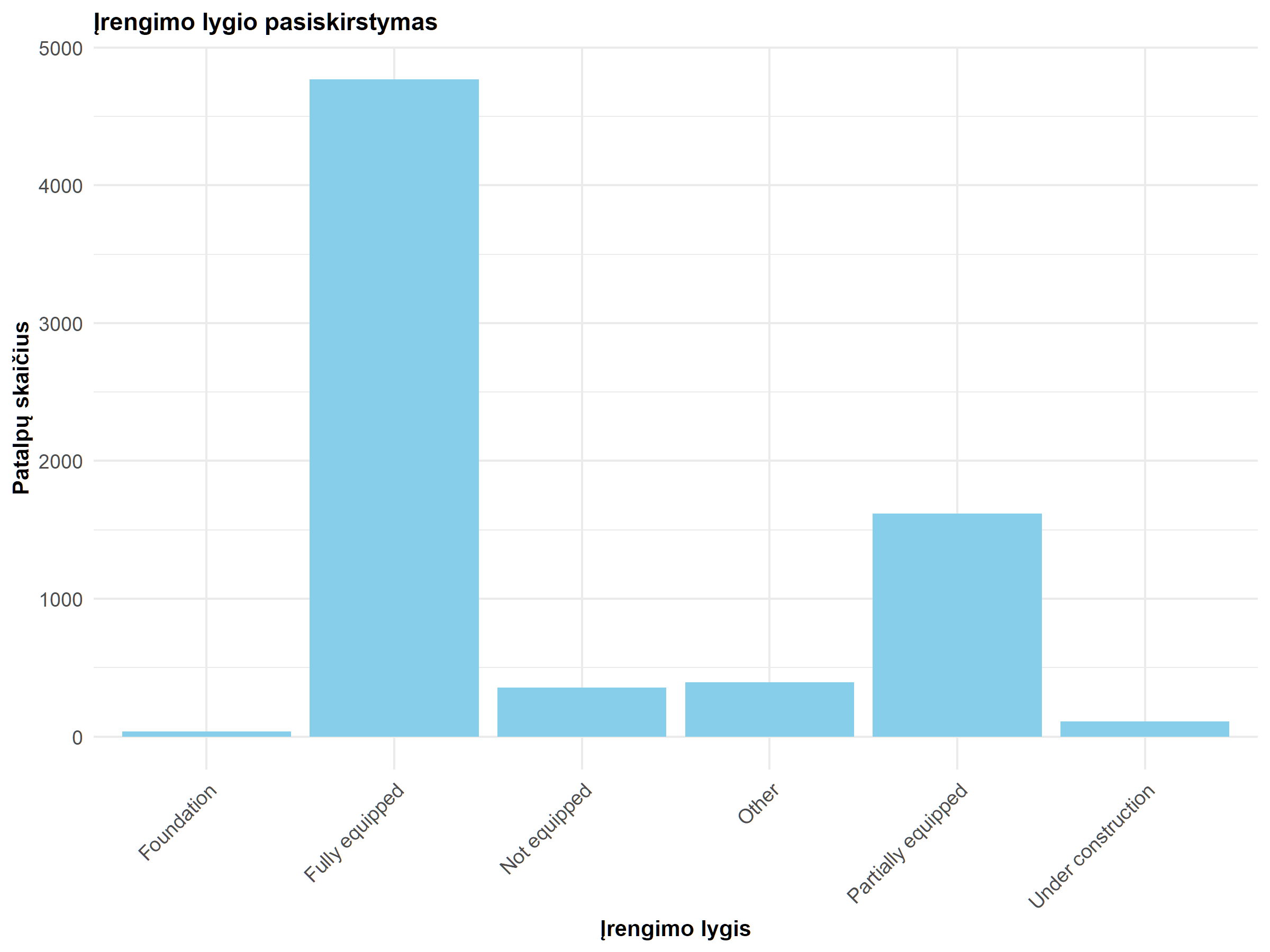
Čia , kur: – įrengtų objektų skaičius, – bendras objektų skaičius.

Kintamasis equipment yra kategorinis ir turi kelias reikšmes (pvz., “Fully equipped”, “Partially equipped”, “Not equipped” ir pan.), vadinasi nėra dvinaris ir netenkina testo taikymo sąlygos. Todėl šį kintamąjį transformuojame į binarinį, kad galėtume tikrinti proporciją įrengtų(pilnai arba dalinai) patalpų.

table(duom$equipment)

##   
## Foundation Fully equipped Not equipped Other   
## 37 4768 356 394   
## Partially equipped Under construction   
## 1619 110

ggplot(equipment\_data, aes(x = equipment, y = count)) +  
 geom\_bar(stat = "identity", fill = "skyblue") +  
 labs(title = "Įrengimo lygio pasiskirstymas",  
 x = "Įrengimo lygis",  
 y = "Patalpų skaičius") +  
 theme\_scientific() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



# Filtruojame tik įrengtus/dalinai įrengtus būstus:  
equipped\_count <- sum(duom$equipment %in% c("Fully equipped", "Partially equipped"))  
  
total\_count <- nrow(duom)

Statistinis tyrimas:

prop.test(equipped\_count, total\_count, p = 0.6, alternative = "two.sided")

##   
## 1-sample proportions test with continuity correction  
##   
## data: equipped\_count out of total\_count, null probability 0.6  
## X-squared = 2325.1, df = 1, p-value < 2.2e-16  
## alternative hypothesis: true p is not equal to 0.6  
## 95 percent confidence interval:  
## 0.8690373 0.8842685  
## sample estimates:  
## p   
## 0.8768534

Statistinė išvada: kadangi -reikšmė () mažesnė už reikšmingumo lygmenį (), tai darome išvadą, kad rastas statistiškai reikšmingas skirtumas, todėl atmetame nulinę hipotezę.

Tyrimo išvada: 87.7% namų yra pilnai arba dalinai įrengta.

## Tyrimas

Tyrimo hipotezė: tarp namų kainos ir peržiūrų skaičiaus yra reikšminga koreliacija.

Statistinė hipotezė:

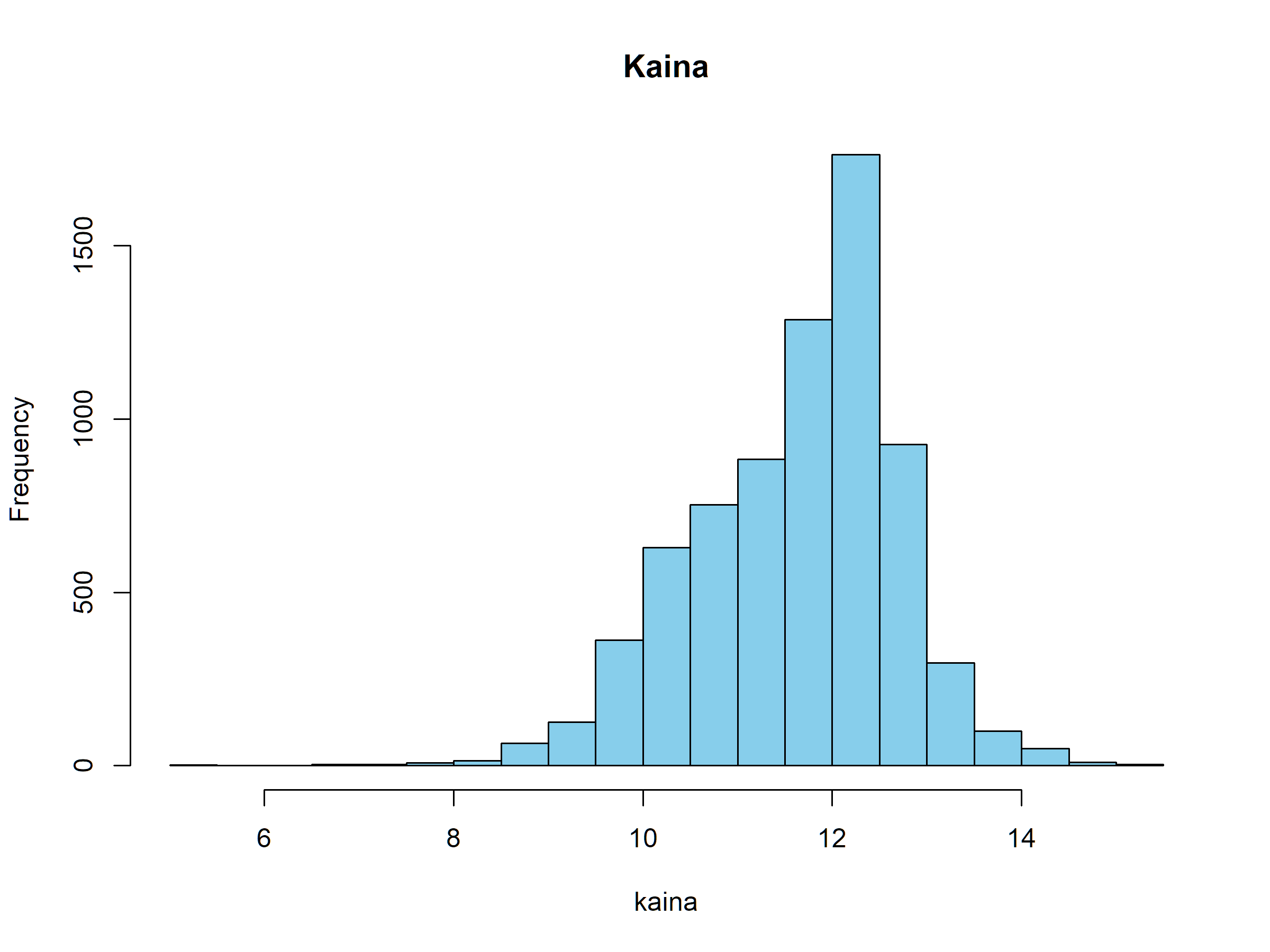
Statistinis testas:

Koreliacijos lygybės nuliui testas:

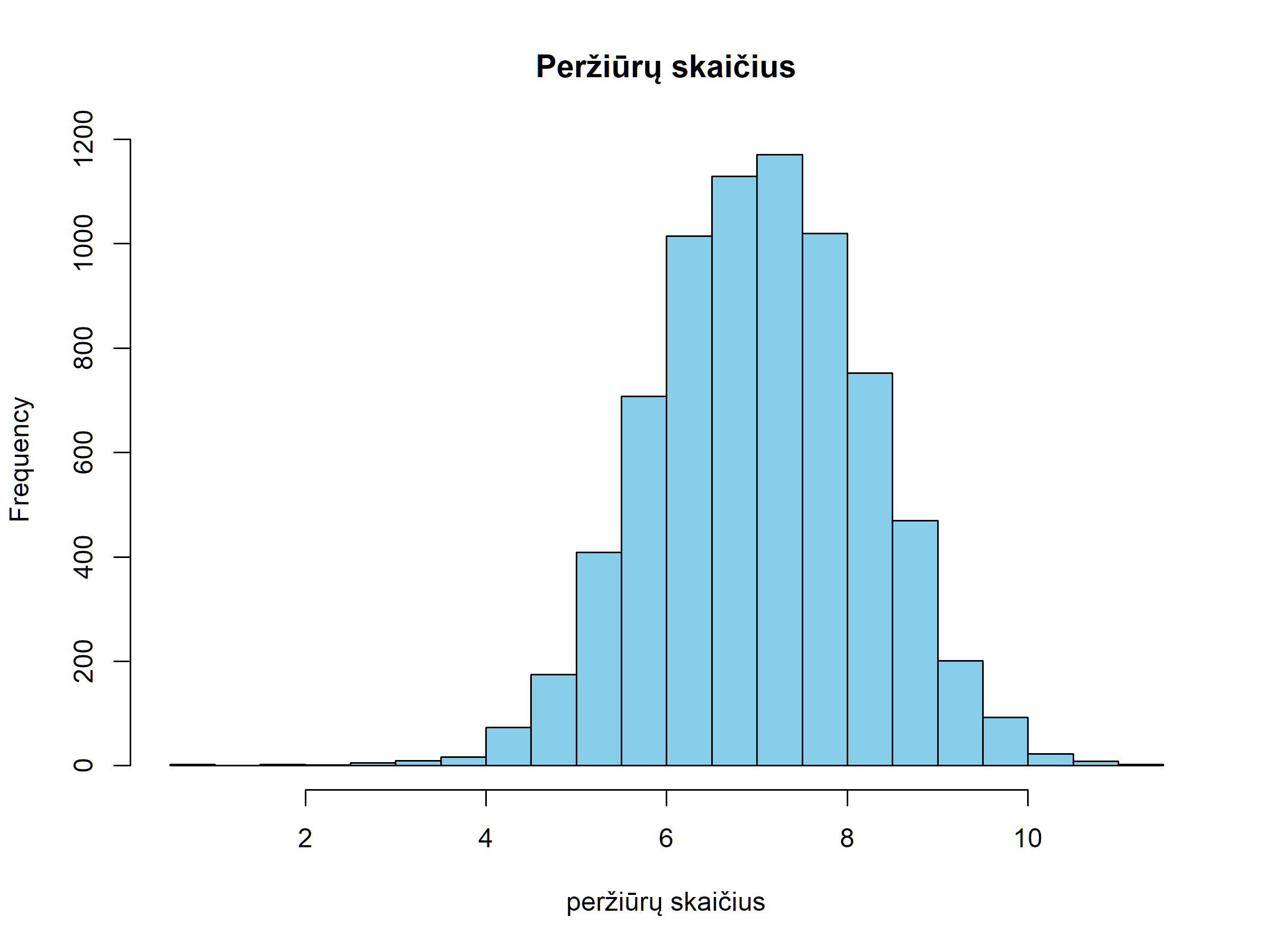
čia 𝑟 imties koreliacijos koeficientas, 𝑛 - imties dydis.

Kintamųjų (peržiūrų skaičius ir kaina) duomenys netenkino testo sąlygos, todėl buvo logaritmuojami, kad labiau atitiktų normalųjį pasisikirstymą.

duom$log\_price <- log(duom$price)  
  
duom$log\_views\_total <- log(duom$views\_total)  
  
  
hist(duom$log\_price, breaks = 30, main = "Kaina", col = "skyblue", xlab = "kaina")



hist(duom$log\_views\_total, breaks = 30, main = "Peržiūrų skaičius", col = "skyblue", xlab = "peržiūrų skaičius")



Statistinis tyrimas:

cor.test(duom$log\_price, duom$log\_views\_total)

##   
## Pearson's product-moment correlation  
##   
## data: duom$log\_price and duom$log\_views\_total  
## t = -11.554, df = 7282, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.1566494 -0.1115447  
## sample estimates:  
## cor   
## -0.1341666

Statistinė išvada: kadangi p-reikšmė(p < 2.2e-16) mažesnė už reikmšingumo lygmenį (𝛼 = 0.05), tai atmetame nuline hipoteze ir teigiame, kad rasta koreliacija statistiskai reiksminga, tačiau silpna ir neigiama.

Tyrimo išvada: namų kaina ir peržiūrų skaičius yra susiję.

## Tyrimas

Tyrimo hipotezė: vidutinės namų ir butų dydžių dispersijos yra lygios.

Statistinė hipotezė: (): vidutiniškai namų ir butų dydžiai yra lygūs

Alternatyvioji hipotezė (): vidutinis namų dydis yra didesnis nei butų dydis

kur:

* - vidutinis namų plotas
* - vidutinis butų plotas

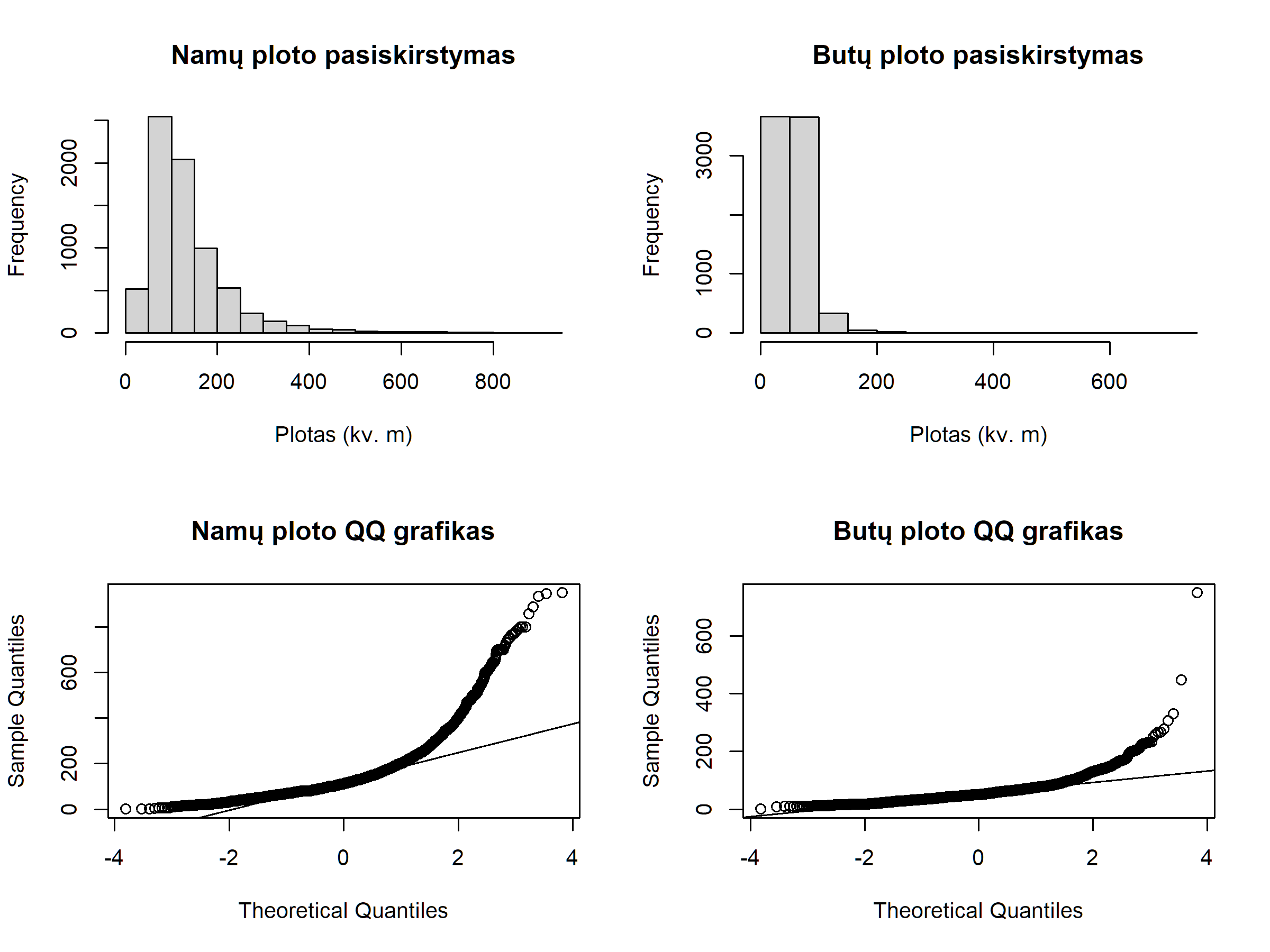
Statistinis testas: dviejų nepriklausomų imčių t-testas: nelygios dispersijos:

kur:

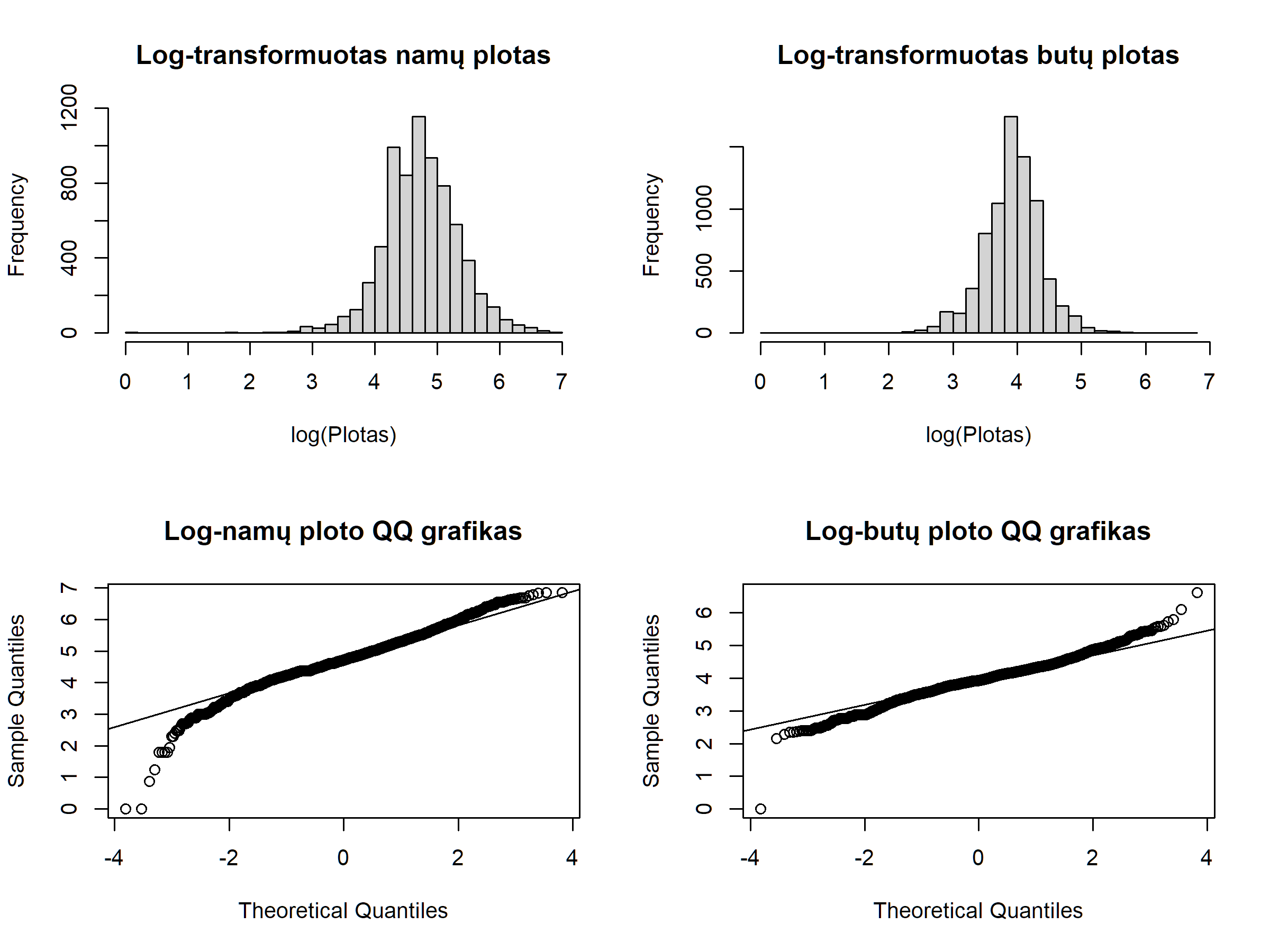
, – pirmosios ir antrosios imties vidurkiai , – pirmosios ir antrosios imties dispersijos , – pirmosios ir antrosios imties dydžiai

Patirkinimas ir tranformavimas kodel?trumpas aprašiukas

# Paruošiame duomenis testui kaip ir anksčiau  
houses\_area <- as.numeric(gsub(",", ".", as.character(csv\_data\_list[["houses"]]$area)))  
apartments\_area <- as.numeric(gsub(",", ".", as.character(csv\_data\_list[["apartments"]]$area)))  
  
# Pašaliname NA ir galimai neteisingus dydžius  
houses\_area <- houses\_area[!is.na(houses\_area) & houses\_area > 0 & houses\_area < 1000]  
apartments\_area <- apartments\_area[!is.na(apartments\_area) & apartments\_area > 0 & apartments\_area < 1000]  
  
# Skirstiniai su histogramomis ir QQ grafikais  
par(mfrow=c(2,2))  
hist(houses\_area, main="Namų ploto pasiskirstymas", xlab="Plotas (kv. m)", breaks=25)  
hist(apartments\_area, main="Butų ploto pasiskirstymas", xlab="Plotas (kv. m)", breaks=25)  
qqnorm(houses\_area, main="Namų ploto QQ grafikas")  
qqline(houses\_area)  
qqnorm(apartments\_area, main="Butų ploto QQ grafikas")  
qqline(apartments\_area)



# Logaritminė transformacija  
log\_houses\_area <- log(houses\_area)  
log\_apartments\_area <- log(apartments\_area)  
  
# Logaritmuotų duomenų patikrinimas  
hist(log\_houses\_area, main="Log-transformuotas namų plotas", xlab="log(Plotas)", breaks=25)  
hist(log\_apartments\_area, main="Log-transformuotas butų plotas", xlab="log(Plotas)", breaks=25)  
qqnorm(log\_houses\_area, main="Log-namų ploto QQ grafikas")  
qqline(log\_houses\_area)  
qqnorm(log\_apartments\_area, main="Log-butų ploto QQ grafikas")  
qqline(log\_apartments\_area)



par(mfrow=c(1,1))

Statistinis tyrimas:

log\_t\_test\_rezultatas <- t.test(log\_houses\_area, log\_apartments\_area, alternative = "greater",  
 var.equal = FALSE)  
  
log\_t\_test\_rezultatas

##   
## Welch Two Sample t-test  
##   
## data: log\_houses\_area and log\_apartments\_area  
## t = 92.856, df = 13262, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## 0.792307 Inf  
## sample estimates:  
## mean of x mean of y   
## 4.739410 3.932813

Statistinė išvada: kadangi , tai darome išvadą, kad namų ir butų plotų vidurkių (t.y. ir ) skirtumas yra statistiškai reikšmingas ( atmetame). Namų vidutinis plotas yra reikšmingai didesnis nei butų vidutinis plotas. Tai rodo, kad namai paprastai yra didesni nei butai.

Tyrimo išvada: tyrimas parodė, kad namai paprastai yra didesni nei butai.

## Tyrimas Pardavėjų proporcijos palyginimas tarp butų ir namų rinkų

Tyrimo hipotezė: privačių pardavėjų proporcija butų ir namų rinkose yra vienoda

Statistinė hipotezė:

kur:

- privačių pardavėjų proporcija butų rinkoje - privačių pardavėjų proporcija namų rinkoje

Statistinis testas: Dviejų imčių proporcijų testas:

kur:

– pirmosios imties proporcija, – antrosios imties proporcija, – bendroji proporcija, ir – pirmosios ir antrosios imties dydžiai, ir – sėkmių skaičiai pirmojoje ir antrojoje imtyse.

Patikrinimas ir transformaimas kodel? trumpas apršas

# Duomenų paruošimas  
apartments\_private <- csv\_data\_list[["apartments"]]$private\_seller  
houses\_private <- csv\_data\_list[["houses"]]$private\_seller  
  
# Pašaliname NA reikšmes  
apartments\_private <- apartments\_private[!is.na(apartments\_private)]  
houses\_private <- houses\_private[!is.na(houses\_private)]  
  
if (!is.logical(apartments\_private)) {  
 apartments\_private <- apartments\_private == "True"  
}  
  
if (!is.logical(houses\_private)) {  
 houses\_private <- houses\_private == "True"  
}  
  
# Skaičiuojame privačių pardavėjų kiekį kiekviename rinkos segmente  
apartments\_private\_count <- sum(apartments\_private)  
houses\_private\_count <- sum(houses\_private)  
  
# Bendras kiekvieno segmento dydis  
apartments\_total <- length(apartments\_private)  
houses\_total <- length(houses\_private)  
  
# Proporcijų apskaičiavimas  
apartments\_prop <- apartments\_private\_count / apartments\_total  
houses\_prop <- houses\_private\_count / houses\_total

Staristinis tyrimas:

prop\_test\_results <- prop.test(  
 x = c(apartments\_private\_count, houses\_private\_count),  
 n = c(apartments\_total, houses\_total),  
 alternative = "two.sided",  
 correct = TRUE # Taikoma Yates pataisa  
)  
  
print(prop\_test\_results)

##   
## 2-sample test for equality of proportions with continuity correction  
##   
## data: c(apartments\_private\_count, houses\_private\_count) out of c(apartments\_total, houses\_total)  
## X-squared = 1.6181, df = 1, p-value = 0.2034  
## alternative hypothesis: two.sided  
## 95 percent confidence interval:  
## -0.005099863 0.024245427  
## sample estimates:  
## prop 1 prop 2   
## 0.2958166 0.2862438

alpha <- 0.05  
df <- 1   
critical\_chi\_sq <- qchisq(1 - alpha, df)  
critical\_chi\_sq

## [1] 3.841459

Statistinė išvada: kadangi , tai darome išvadą, kad privačių pardavėjų proporcijų skirtumas tarp butų ir namų rinkų nėra statistiškai reikšmingas ( neatmetame).

Tyrimo išvada: tyrimas parodė, kad privačių pardavėjų proporcijos abiejose rinkose yra panašios.

## Tyrimas Komercinių patalpų ploto dispersijų palyginimas

Tyrimo hipotezė: komercinių patalpų ploto dispersijos pardavimo ir nuomos sektoriuose yra lygios

Statistinė hipotezė:

kur:

* - komercinių patalpų ploto dispersija pardavimo sektoriuje
* - komercinių patalpų ploto dispersija nuomos sektoriuje

Statistinis testas:

Dviejų imčių dispersijų palyginimo testas:

kur:

- pirmosios imties dispersija – antrosios imties dispersija

Aprašiukas

# Ištraukiame reikalingus duomenis  
premises\_area <- as.numeric(gsub(",", ".", as.character(csv\_data\_list[["premises"]]$area)))  
premises\_rent\_area <- as.numeric(gsub(",", ".", as.character(csv\_data\_list[["premises\_rent"]]$area)))  
  
# Pašaliname NA ir nelogiškas reikšmes  
premises\_area <- premises\_area[!is.na(premises\_area) & premises\_area > 0 & premises\_area < 10000]  
premises\_rent\_area <- premises\_rent\_area[!is.na(premises\_rent\_area) & premises\_rent\_area > 0 & premises\_rent\_area < 10000]

Statistinis tyrimas:

f\_test\_results <- var.test(  
 premises\_area, premises\_rent\_area, alternative = "two.sided")  
  
print(f\_test\_results)

##   
## F test to compare two variances  
##   
## data: premises\_area and premises\_rent\_area  
## F = 2.0921, num df = 1475, denom df = 2059, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 1.903967 2.300458  
## sample estimates:  
## ratio of variances   
## 2.09206

Statistinė išvada: kadangi tai darome išvadą, kad komercinių patalpų ploto dispersijos pardavimo ir nuomos sektoriuose skiriasi statistiškai reikšmingai ( atmetame).

Tyrimo išvada: tyrimas parodė, kad pardavimo ir nuomos sektoriai yra nevienodai homogeniški ploto atžvilgiu.

## Tyrimas Renovuotų ir nerenovuotų nuomuojamų butų kaina

Tyrimo hipotezė: renovuotų ir nerenovuotų butų nuomos kainos vidurkiai yra vienodi.

Statistinė hipotezė:

kur:

- renovuotų butų nuomos kainos vidurkis - nerenovuotų butų nuomos kainos vidurkis

Statistinis testas:

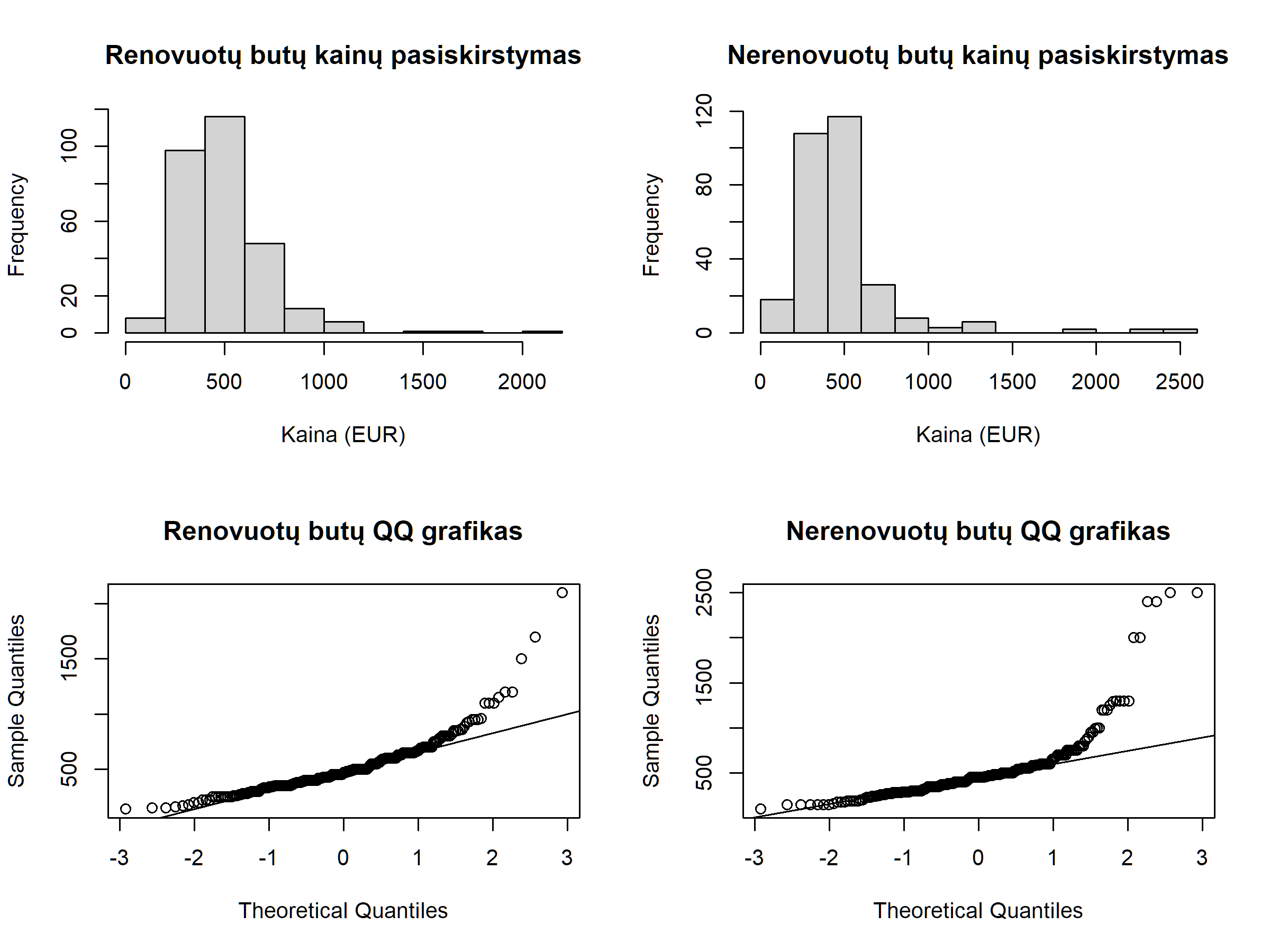
Dviejų priklausomų imčių (porinis) t-testas: Porinio t-testo formulė LaTeX formatu:

kur:

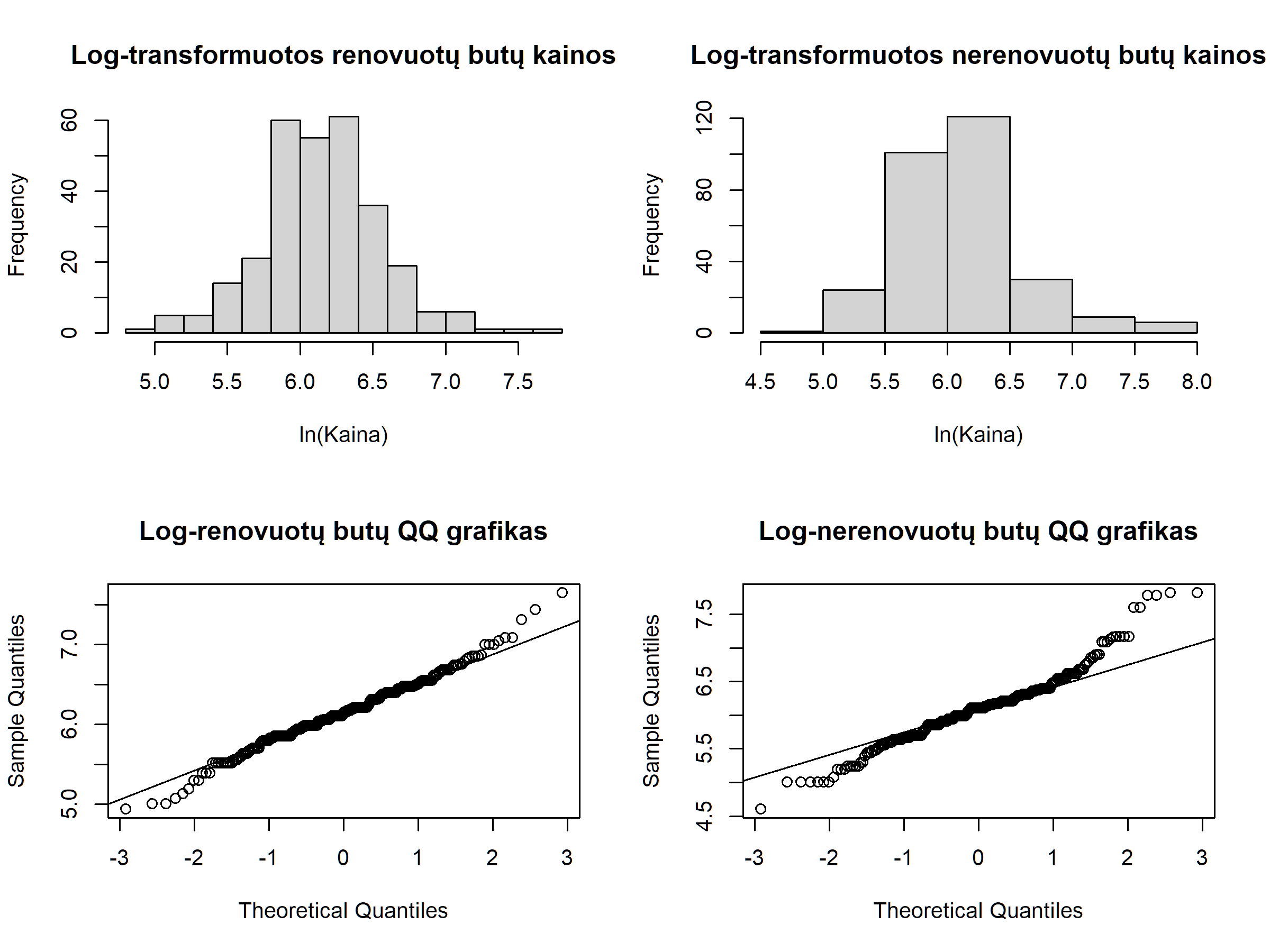
– porinių skirtumų vidurkis – porinių skirtumų standartinis nuokrypis – porų skaičius

Mūsų duomenų rinkinys neturėjo tinkamų duomenų poriniui testui. Mes iš esamų duomenų susikūrėme tokį duomenų rinkinį, su kuriuo būtų galima atlikti porinį t testą, kad pamatytume kaip atlikti šios statistikos testą. Buvo atlikta logaritminė transformacija, siekiant pagerinti normalumą ir taip užtikrinti, kad duomenys atitiktų normalųjį pasiskirstymą.

# Filtruojame ir atspausdiname renovuotus ir nerenovuotų butų su jų kainomis  
build\_year\_data <- csv\_data\_list[["apartments\_rent"]]$build\_year  
price\_data <- csv\_data\_list[["apartments\_rent"]]$price  
  
# Sukuriame pilną duomenų rinkinį  
full\_data <- data.frame(  
 build\_year = build\_year\_data,  
 price = price\_data  
)  
  
renovated\_data <- data.frame(  
 build\_year = character(0),  
 price = numeric(0),  
 construction\_year = numeric(0)  
)  
  
for (i in 1:nrow(full\_data)) {  
 x <- as.character(full\_data$build\_year[i])  
 if (grepl("construction", x) && grepl("renovation", x)) {  
 construction\_year <- as.numeric(substr(x, 1, 4))  
 renovation\_year <- as.numeric(substr(x, regexpr("renovation", x) - 5, regexpr("renovation", x) - 2))  
 if (!is.na(construction\_year) && !is.na(renovation\_year) && construction\_year < 2000 && renovation\_year > 2010) {  
 renovated\_data <- rbind(renovated\_data, data.frame(  
 build\_year = x,  
 price = full\_data$price[i],  
 construction\_year = construction\_year  
 ))  
 }  
 }  
}  
  
# Pervardijame renovuotų butų stulpelius  
if (nrow(renovated\_data) > 0) {  
 colnames(renovated\_data)[1:2] <- c("build\_year\_renovated", "price\_renovated")  
}  
  
# Identifikuojame nerenovuotus butus ir ištraukiame jų statybos metus  
non\_renovated\_data <- data.frame(  
 build\_year = character(0),  
 price = numeric(0),  
 construction\_year = numeric(0)  
)  
  
for (i in 1:nrow(full\_data)) {  
 x <- as.character(full\_data$build\_year[i])  
 if (!grepl("renovation", x)) {  
 # Jei statybos metai pateikti kaip skaičius  
 if (grepl("^\\d{4}$", x)) {  
 construction\_year <- as.numeric(x)  
 if (!is.na(construction\_year)) {  
 non\_renovated\_data <- rbind(non\_renovated\_data, data.frame(  
 build\_year = x,  
 price = full\_data$price[i],  
 construction\_year = construction\_year  
 ))  
 }  
 } else if (grepl("construction", x)) {  
 # Jei yra "construction" formatas  
 construction\_year <- as.numeric(substr(x, 1, 4))  
 if (!is.na(construction\_year)) {  
 non\_renovated\_data <- rbind(non\_renovated\_data, data.frame(  
 build\_year = x,  
 price = full\_data$price[i],  
 construction\_year = construction\_year  
 ))  
 }  
 }  
 }  
}  
  
# Pervardijame nerenovuotų butų stulpelius  
if (nrow(non\_renovated\_data) > 0) {  
 colnames(non\_renovated\_data)[1:2] <- c("build\_year\_non\_renovated", "price\_non\_renovated")  
}  
  
# Sukuriame lentelę rezultatams  
combined\_data <- data.frame(  
 ID\_Renovuoto = numeric(nrow(renovated\_data)),  
 Statybos\_Metai\_Renovuoto = character(nrow(renovated\_data)),  
 Statybos\_Metai\_Skaicius\_Renovuoto = numeric(nrow(renovated\_data)),  
 Kaina\_Renovuoto = numeric(nrow(renovated\_data)),  
 ID\_Nerenovuoto = numeric(nrow(renovated\_data)),  
 Statybos\_Metai\_Skaicius\_Nerenovuoto = numeric(nrow(renovated\_data)),  
 Kaina\_Nerenovuoto = numeric(nrow(renovated\_data))  
)  
  
# Užpildome lentelę duomenimis  
for (i in 1:nrow(renovated\_data)) {  
 target\_year <- renovated\_data$construction\_year[i]  
   
 # Randame nerenovuotus butus su tokiais pačiais statybos metais  
 matching\_indices <- which(non\_renovated\_data$construction\_year == target\_year)  
 if (length(matching\_indices) > 0) {  
 # Jei yra sutampančių statybos metų, parenkame atsitiktinį butą iš jų  
 random\_idx <- sample(matching\_indices, 1)  
 } else {  
 next  
 }  
   
 # Užpildome duomenis  
 combined\_data[i, "ID\_Renovuoto"] <- i  
 combined\_data[i, "Statybos\_Metai\_Renovuoto"] <- as.character(renovated\_data[i, "build\_year\_renovated"])  
 combined\_data[i, "Statybos\_Metai\_Skaicius\_Renovuoto"] <- renovated\_data$construction\_year[i]  
 combined\_data[i, "Kaina\_Renovuoto"] <- renovated\_data[i, "price\_renovated"]  
   
 combined\_data[i, "ID\_Nerenovuoto"] <- random\_idx  
 combined\_data[i, "Statybos\_Metai\_Skaicius\_Nerenovuoto"] <- non\_renovated\_data$construction\_year[random\_idx]  
 combined\_data[i, "Kaina\_Nerenovuoto"] <- non\_renovated\_data[random\_idx, "price\_non\_renovated"]  
}  
  
# Pašaliname eilutes su NA reikšmėmis  
combined\_data\_clean <- combined\_data[complete.cases(combined\_data), ]  
  
# Papildomai pašaliname eilutes, kur Kaina\_Renovuoto yra 0  
combined\_data\_clean <- combined\_data\_clean[combined\_data\_clean$Kaina\_Renovuoto > 0, ]  
  
par(mfrow=c(2,2))  
hist(combined\_data\_clean$Kaina\_Renovuoto, main="Renovuotų butų kainų pasiskirstymas",   
 xlab="Kaina (EUR)")  
hist(combined\_data\_clean$Kaina\_Nerenovuoto, main="Nerenovuotų butų kainų pasiskirstymas",   
 xlab="Kaina (EUR)")  
  
# QQ grafikai  
qqnorm(combined\_data\_clean$Kaina\_Renovuoto, main="Renovuotų butų QQ grafikas")  
qqline(combined\_data\_clean$Kaina\_Renovuoto)  
qqnorm(combined\_data\_clean$Kaina\_Nerenovuoto, main="Nerenovuotų butų QQ grafikas")  
qqline(combined\_data\_clean$Kaina\_Nerenovuoto)



log\_kaina\_renovuoto <- log(combined\_data\_clean$Kaina\_Renovuoto)  
log\_kaina\_nerenovuoto <- log(combined\_data\_clean$Kaina\_Nerenovuoto)  
  
hist(log\_kaina\_renovuoto, main="Log-transformuotos renovuotų butų kainos",   
 xlab="ln(Kaina)")  
hist(log\_kaina\_nerenovuoto, main="Log-transformuotos nerenovuotų butų kainos",   
 xlab="ln(Kaina)")  
  
qqnorm(log\_kaina\_renovuoto, main="Log-renovuotų butų QQ grafikas")  
qqline(log\_kaina\_renovuoto)  
qqnorm(log\_kaina\_nerenovuoto, main="Log-nerenovuotų butų QQ grafikas")  
qqline(log\_kaina\_nerenovuoto)



Statistinis tyrimas:

t\_test\_result <- t.test(  
 log\_kaina\_renovuoto,   
 log\_kaina\_nerenovuoto,   
 alternative = "greater",  
 paired = TRUE  
)  
  
print(t\_test\_result)

##   
## Paired t-test  
##   
## data: log\_kaina\_renovuoto and log\_kaina\_nerenovuoto  
## t = 1.717, df = 291, p-value = 0.04352  
## alternative hypothesis: true mean difference is greater than 0  
## 95 percent confidence interval:  
## 0.002169347 Inf  
## sample estimates:  
## mean difference   
## 0.05569547

alpha <- 0.05   
df <- nrow(combined\_data\_clean) - 1  
critical\_t <- qt(1 - alpha, df)  
critical\_t

## [1] 1.650107

Statistinė išvada: kadangi , tai darome išvadą, kad renovuotų butų nuomos kainos vidurkis yra statistiškai reikšmingai didesnis nei nerenovuotų butų nuomos kainos vidurkis ( atmetame).

Tyrimo išvada: tyrimas parodė, kad renovuoti butai nuomojami už statistiškai reikšmingai didesnę kainą nei nerenovuoti tos pačios statybos metų butai.

# Šalitiniai