Nekilnojamojo turto objektų kainų analizė Lietuvoje

Statistikos laboratorinis darbas Nr. 2

Temile Danylaite, Martynas Zabitis

2025-04-17

Turinys

1 Įvadas

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

2 Duomenų aprašymas

Analizei naudojami duomenys buvo atsisiųsti iš Lithuanian Real Estate Listings GitHub repozitorijos. Duomenys buvo surinkti 2024 m. vasarį iš Aruodas.lt puslapio. Duomenų rinkinyje yra informacija apie parduodamus ir nuomojamus butus, garažus, namus, sklypus ir patalpas.

Pasirinktus naudojamui duomenis apima namų, butų bei komercinių patalpų:

- 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)

2.1 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")
```

Table 1: 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
  }
}
```

2.2 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")
```

Table 2: 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

3 Duomenų vizualizacija

Grafikai padės geriau suprasti Lietuvos nekilnojamojo turto rinkos ypatybes.

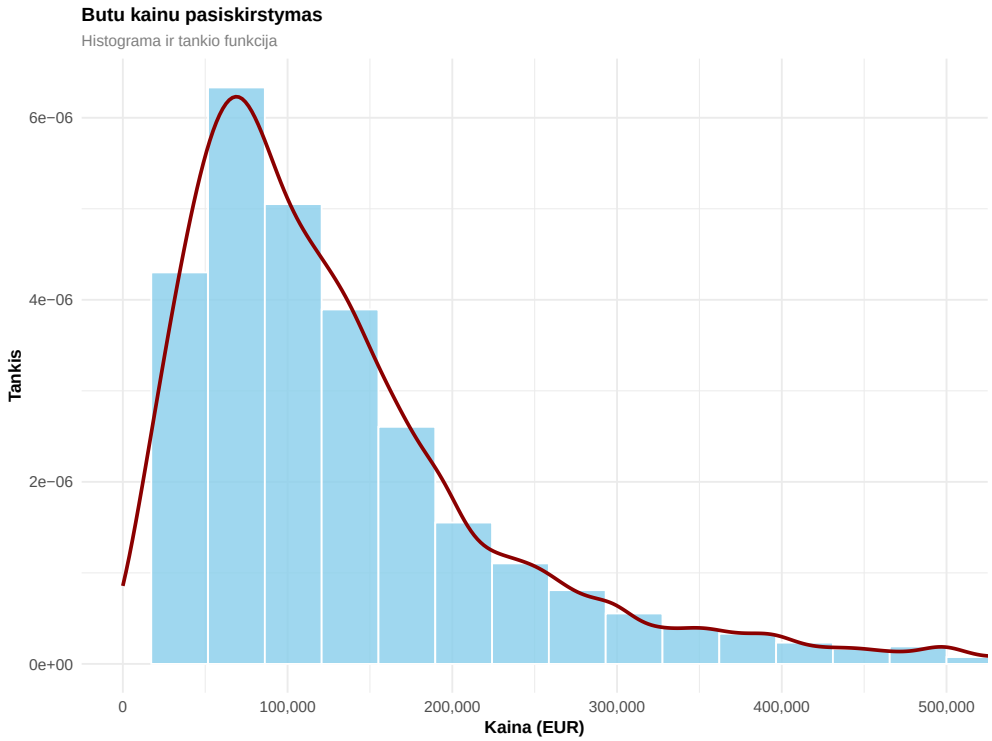
```
# Nustatome bendrą grafikų stilių
theme_scientific <- function() {
  theme_minimal() +
    theme(
      plot.title = element_text(face = "bold", size = 11),
      plot.subtitle = element_text(size = 9, color = "gray50"),
      axis.title = element_text(face = "bold", size = 10),
      axis.text = element_text(size = 9),
      legend.title = element_text(face = "bold", size = 9),
      legend.text = element_text(size = 8)
    )
}
```

3.1 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 = "skyblue",
```

```
                   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)
}
```

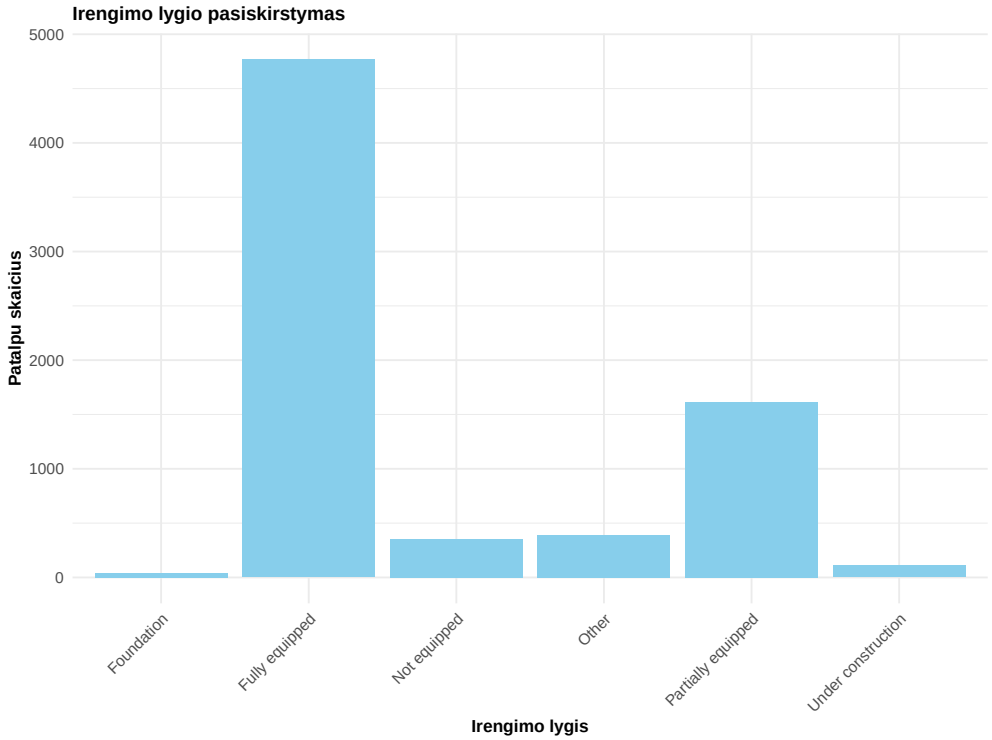


3.2 Įrengimo lygio pasiskirstymo analizė

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))
```

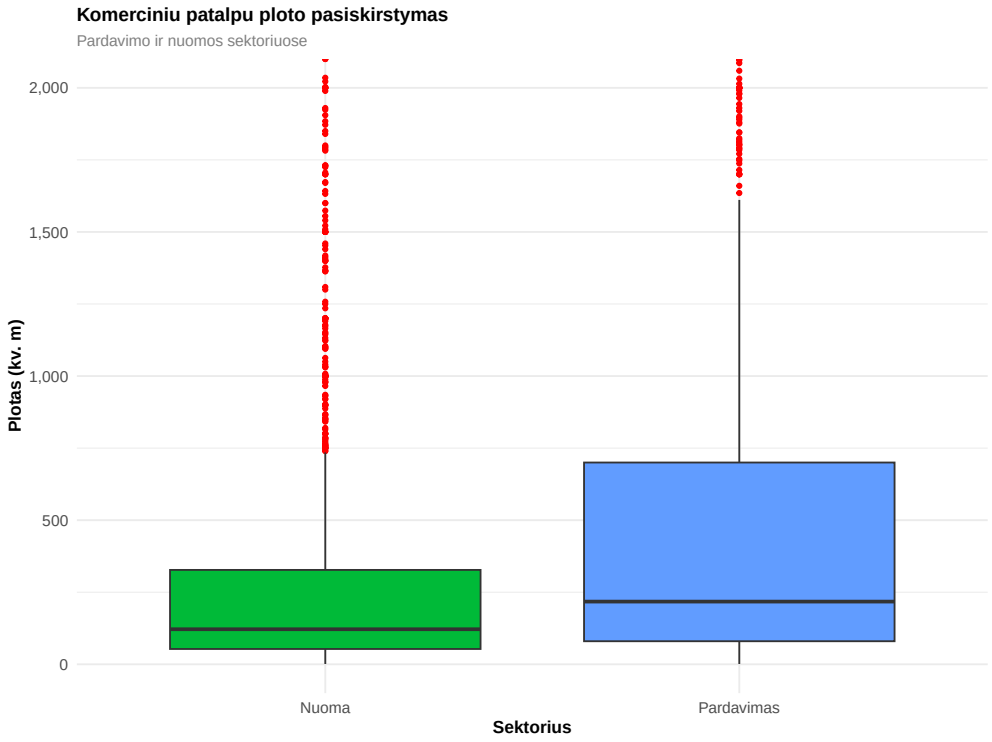


3.3 Komercinių patalpų ploto analizė

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)
```

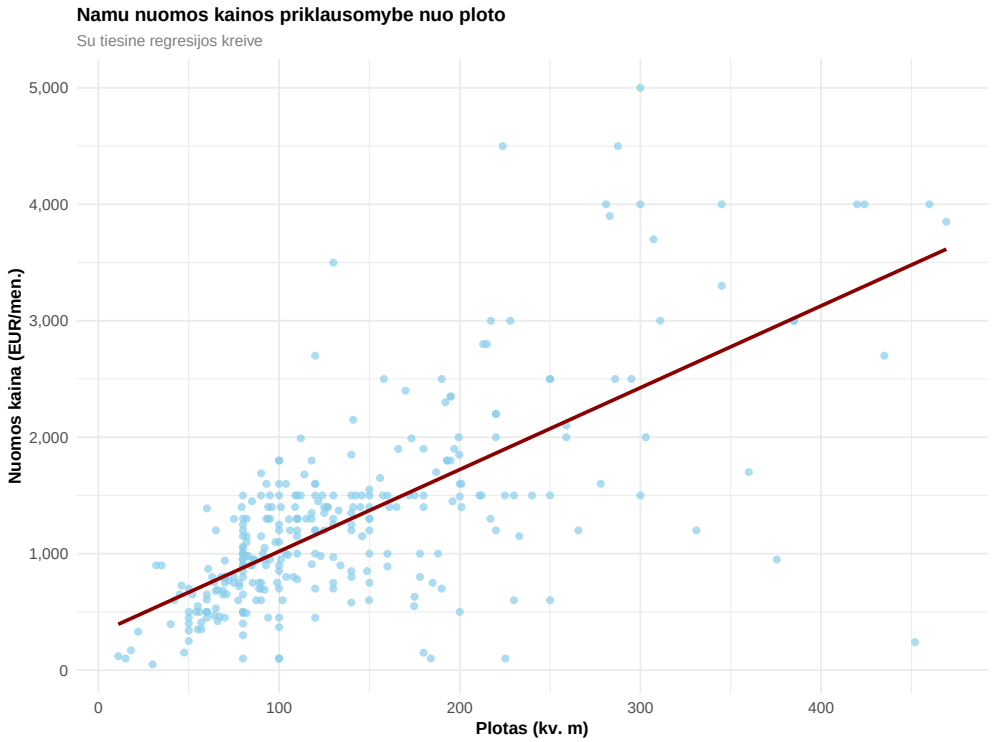


3.4 Namų nuomos kainos ryšys su plotu

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 = "skyblue") +
  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
```

4 Pagrindinės skaitinės charakteristikos

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)
```

Table 3: 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)
```

Table 4: 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)
```

Table 5: 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)
```

Table 6: 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)
```

Table 7: 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

5 Dažnių lentelės parinktiems kategoriniams kintamiesiems.

```
Kategoriniai_kintamieji <- c("sold_or_rented", "reserved", "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   reserved      FALSE   7136
## 2   reserved       TRUE    148
##   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
```

6 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.

6.1 Tyrimas apie namų kainas

Tyrimo hipotezė: vidutinė namų kaina regionuose yra 100000 Eur

Statistinė hipotezė:

Nulinė hipotezė(H_0): vidutinė namų kaina yra 100000 Eur

Alternatyvioji hipotezė (H_1): vidutinė namų kaina nėra lygo 100000 Eur

$$H_0 : \mu = 100000$$

$$H_1 : \mu \neq 100000$$

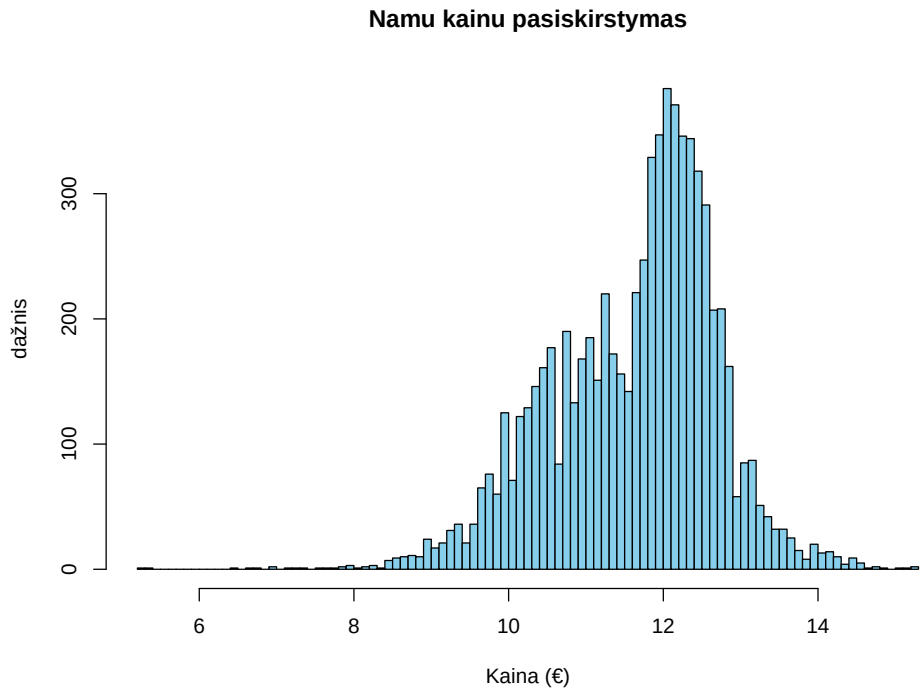
Statistinis testas:

Vienos imties t-testas, kai dispersija nežinoma:

$$t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}}, \quad t \sim t(n-1)$$

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 p-reikšmė ($< 2.2 \times 10^{-16}$) mažesnė už reikšmingumo lygmenį ($\alpha = 0.05$), 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.

6.2 Tyrimas apie namų įrengimą

Tyrimo hipotezė: 60 proc. namų yra įrengti pilnai arba dalinai

Statistinė hipotezė:

Nulinė hipotezė(H_0): 60 proc. namų yra įrengti pilnai arba dalina

Alternatyvioji hipotezė (H_1): proporcija namų, kurie yra įrengti pilnai arba dalinai, skiriasi nuo 60 proc.

$$H_0 : p = 0.6$$
$$H_1 : p \neq 0.6$$

Statistinis testas:

Vienos imties proporcijų testas:

$$z = \frac{m - na}{\sqrt{na(1-a)}} = \frac{\hat{p} - a}{\sqrt{\frac{a(1-a)}{n}}}$$

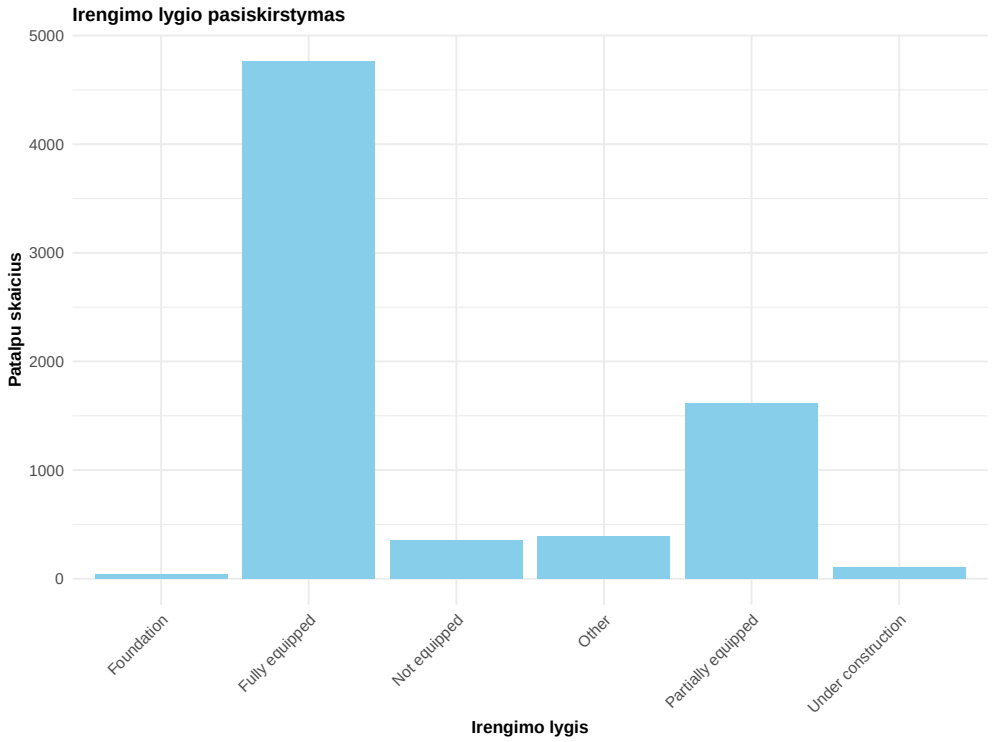
Čia $\hat{p} = \frac{m}{n}$, kur: m – įrengtų objektų skaičius, n – bendras objektų skaičius.

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 p-reikšmė ($< 2.2 \times 10^{-16}$) mažesnė už reikšmingumo lygmenį ($\alpha = 0.05$), tai darome išvadą, kad rastas statistiškai reikšmingas skirtumas, todėl atmetame nulinę hipotezę.

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

6.3 Tyrimas apie kainas ir peržiūrų skaičių

Tyrimo hipotezė: egzistuoja ryšys tarp namų kainos ir jų peržiūrų skaičiaus.

Statistinė hipotezė:

Nulinė hipotezė(H_0): nėra jokios koreliacijos tarp namų kainos ir peržiūrų skaičiaus.

Alternatyvioji hipotezė (H_1): Egzistuoja reikšminga koreliacija tarp namų kainos ir peržiūrų skaičiaus.

$$H_0 : \rho = 0$$

$$H_1 : \rho \neq 0$$

Statistinis testas:

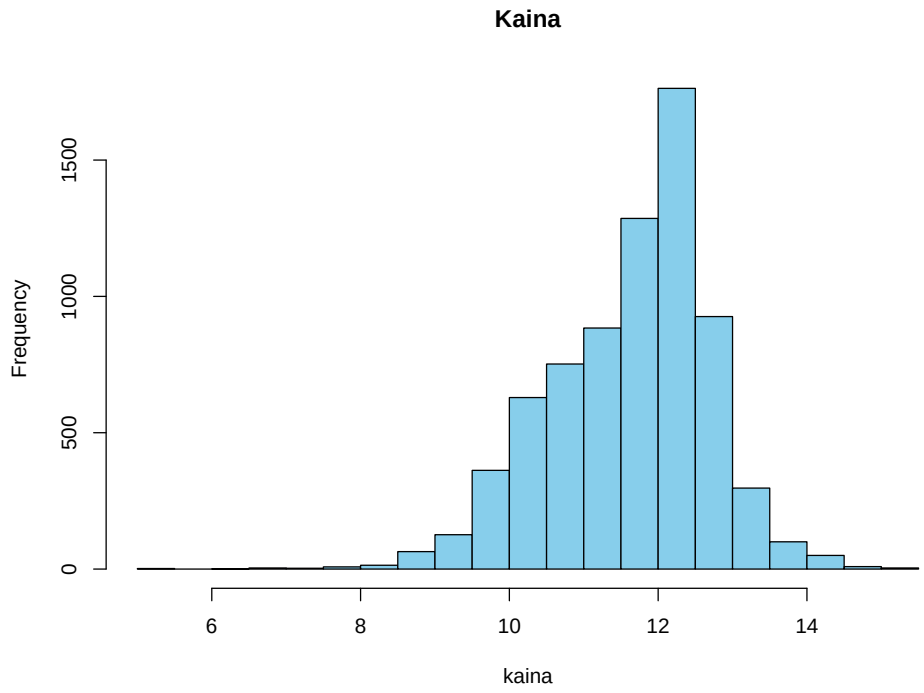
Koreliacijos lygybės nuliui testas:

$$T = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$

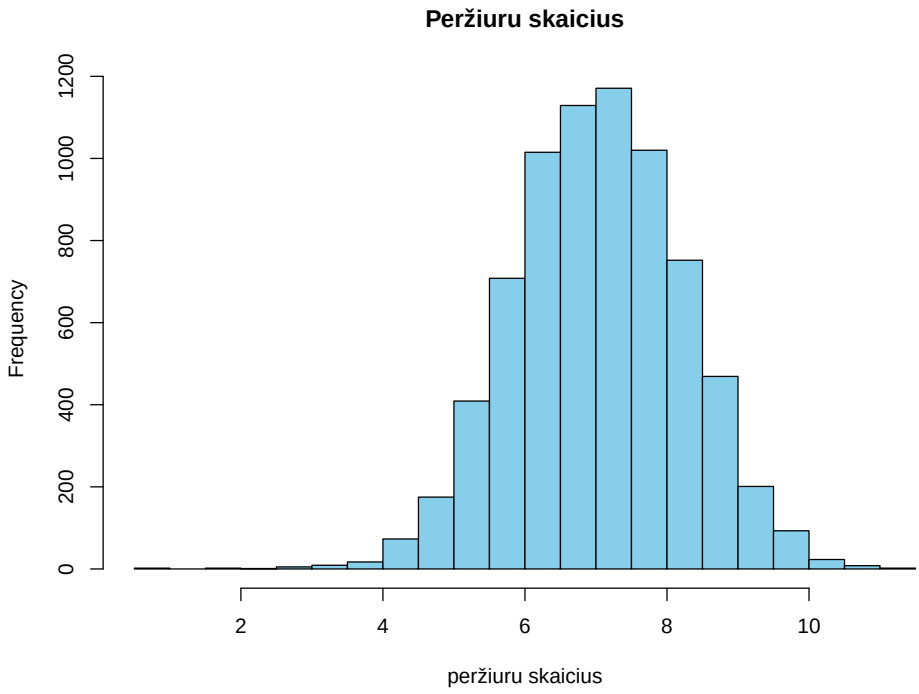
čia imties koreliacijos koeficientas, - imties dydis.

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

```
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")
```



```r
hist(duom$log_views_total, breaks = 30, main = "Peržiūrų skaičius", col = "skyblue", xlab = "peržiūrų sk
```



Statistinis tyrimas:

```r
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 (= 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.

6.4 Tyrimas apie namų ir butų dydžius

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

Statistinė hipotezė:

Statistinė hipotezė: (H_0): vidutiniškai namų ir butų dydžiai yra lygūs

Alternatyvioji hipotezė (H_1): vidutinis namų dydis yra didesnis nei butų dydis

$$H_0 : \mu_{houses} = \mu_{apartments}$$

$$H_1 : \mu_{houses} > \mu_{apartments}$$

kur:

- μ_{houses} - vidutinis namų plotas
- $\mu_{apartments}$ - vidutinis butų plotas

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

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n} + \frac{s_2^2}{m}}}, \quad t \sim t(k)$$

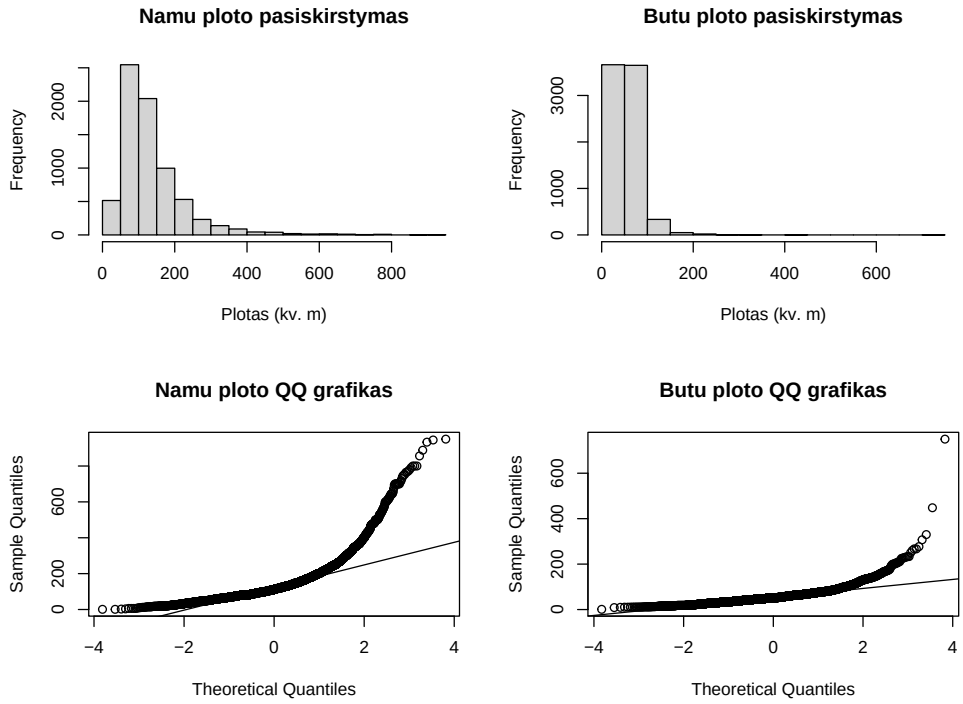
$$\text{čia } k \leq \frac{\left(\frac{s_1^2}{n} + \frac{s_2^2}{m}\right)^2}{\frac{s_1^4}{(n-1)n^2} + \frac{s_2^4}{(m-1)m^2}}$$

kur:

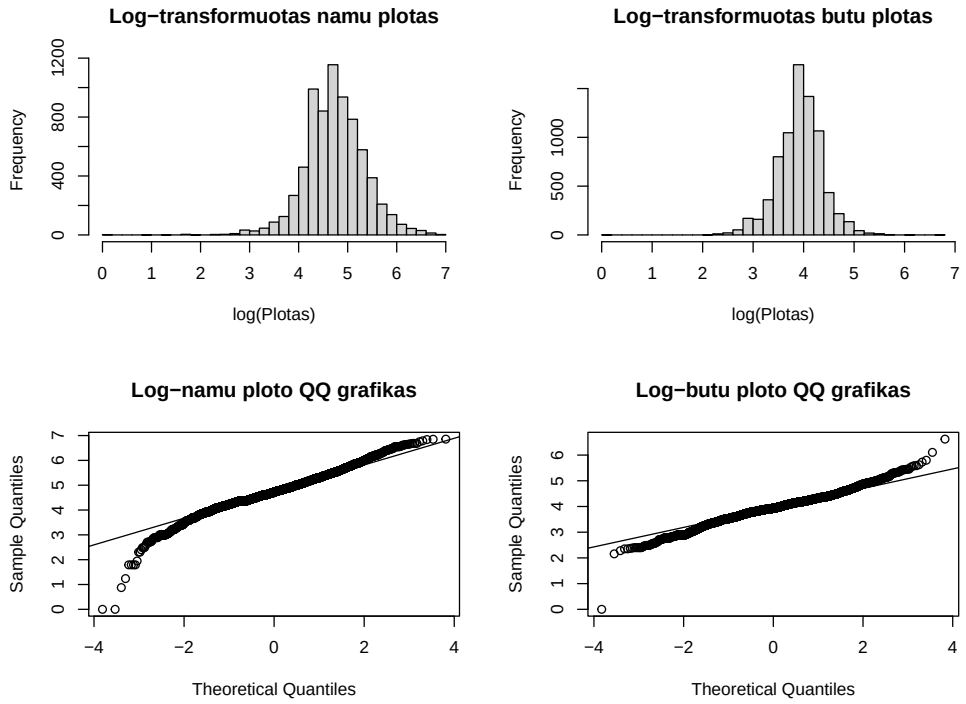
$\bar{X}_1$, $\bar{X}_2$ – pirmosios ir antrosios imties vidurkiai, s_1^2, s_2^2 – pirmosios ir antrosios imties dispersijos, n_1, n_2 – pirmosios ir antrosios imties dydžiai.

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 < 10

# 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)
```




```r
# 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 $t = 92.856 > 1.645 = t_{0.05(13262)}$, tai darome išvadą, kad namų ir butų plotų vidurkių (t.y. 137.188 ir 56.292) skirtumas yra statistiškai reikšmingas (H_0 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.

6.5 Tyrimas apie pardavėjus ir jų butų bei namų rinkas

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

Statistinė hipotezė:

Statistinė hipotezė: (H_0): privačių pardavėjų proporcijos butų ir namų rinkose yra vienodos.

Alternatyvioji hipotezė (H_1): privačių pardavėjų proporcijos butų ir namų rinkose skiriasi.

$$H_0 : p_{apartments} = p_{houses}$$
$$H_1 : p_{apartments} \neq p_{houses}$$

kur:

$p_{apartments}$ - privačių pardavėjų proporcija butų rinkoje p_{houses} - privačių pardavėjų proporcija namų rinkoje

Statistinis testas: Dviejų imčių proporcijų testas:

$z = \frac{\hat{p}_1 - \hat{p}_2 - a}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n} + \frac{\hat{p}_2(1-\hat{p}_2)}{m}}}$

kur:

$\hat{p}_1 = \frac{k_1}{n}, \quad \hat{p}_2 = \frac{k_2}{m}$ – imčių proporcijos,
n ir m – atitinkamų imčių dydžiai

Duomenys neatitiko testo taikymui reikalingų sąlygų, todėl jie buvo paruošiami, pašalinamos NA reikšmės.

```
# 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:

```r
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
```

```r
alpha <- 0.05
df <- 1
critical_chi_sq <- qchisq(1 - alpha, df)
critical_chi_sq
```

```
## [1] 3.841459
```

Statistinė išvada: kadangi $\chi^2 = 1.6181 < 3.841 = \chi^2_{0.05(1)}$, tai darome išvadą, kad privačių pardavėjų proporcijų skirtumas tarp butų ir namų rinkų nėra statistiškai reikšmingas (H_0 neatmetame).

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

6.6 Tyrimas apie komercines patalpas ir jų plotą

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

Statistinė hipotezė:

Statistinė hipotezė: (H_0): komercinių patalpų ploto dispersijos pardavimo ir nuomos sektoriuose yra vienodos.

Alternatyvioji hipotezė (H_1): komercinių patalpų ploto dispersijos pardavimo ir nuomos sektoriuose skiriasi.

$$H_0 : \sigma^2_{premises} = \sigma^2_{premises\_rent}$$

$$H_1 : \sigma^2_{premises} \neq \sigma^2_{premises\_rent}$$

kur:

- $\sigma^2_{premises}$ - komercinių patalpų ploto dispersija pardavimo sektoriuje

- $\sigma^2_{premises\_rent}$ - komercinių patalpų ploto dispersija nuomos sektoriuje

Statistinis testas:

Dviejų imčių dispersijų palyginimo testas:

$$F = \frac{s_1^2}{s_2^2}$$

kur:

s_1^2 - pirmosios imties dispersija s_2^2 – antrosios imties dispersija

```
# 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
```

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 $F = 2.0921 > 1.09879 = F_{0.025}(1475, 2059)$, tai darome išvadą, kad komercinių patalpų ploto dispersijos pardavimo ir nuomos sektoriuose skiriasi statistiškai reikšmingai (H_0 atmetame).

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

6.7 Tyrimas apie renovuotas ir nerenovuotas nuomuojamų butų kainas

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

Statistinė hipotezė:

Statistinė hipotezė: (H_0): renovuotų ir nerenovuotų butų nuomos kainos vidurkiai yra lygūs.

Alternatyvioji hipotezė (H_1): renovuotų butų nuomos kainų vidurkis yra didesnis nei nerenovuotų.

$$H_0 : \mu_{renovated} = \mu_{non_renovated}$$
$$H_1 : \mu_{renovated} > \mu_{non_renovated}$$

kur:

$\mu_{renovated}$ - renovuotų butų nuomos kainos vidurkis $\mu_{non_renovated}$ - nerenovuotų butų nuomos kainos vidurkis

Statistinis testas:

Dviejų priklausomų imčių (porinis) t-testas:

$t = \frac{\bar{d}}{sd/\sqrt{n}}, \quad t \sim t(n-1)$

kur:

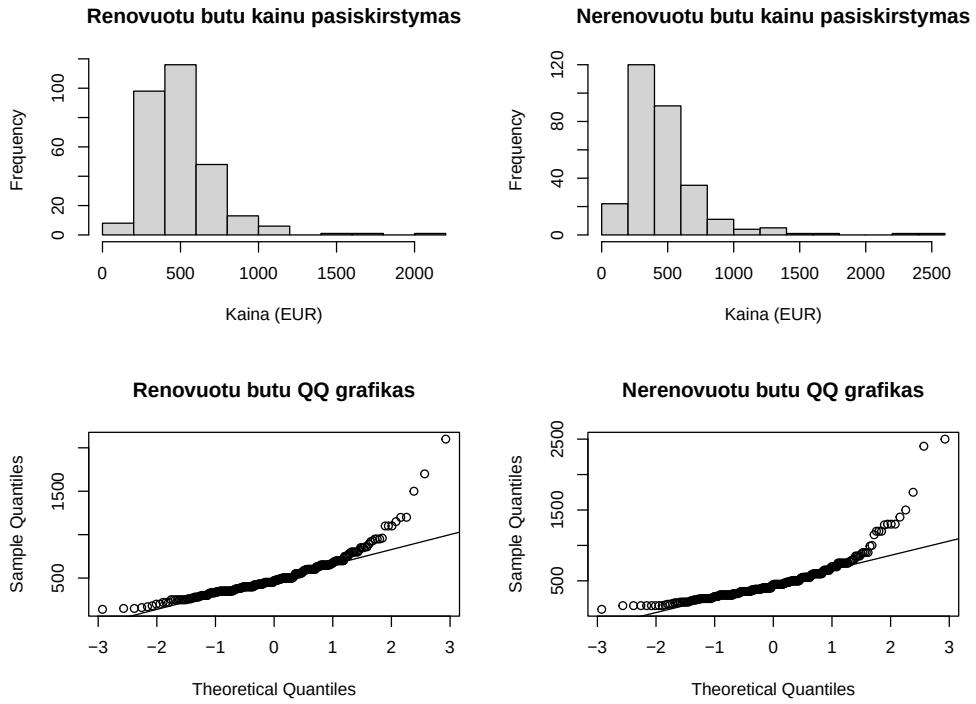
$\bar{d}$ – porinių skirtumų vidurkis, s_d – porinių skirtumų standartinis nuokrypis, n – porų skaičius

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.

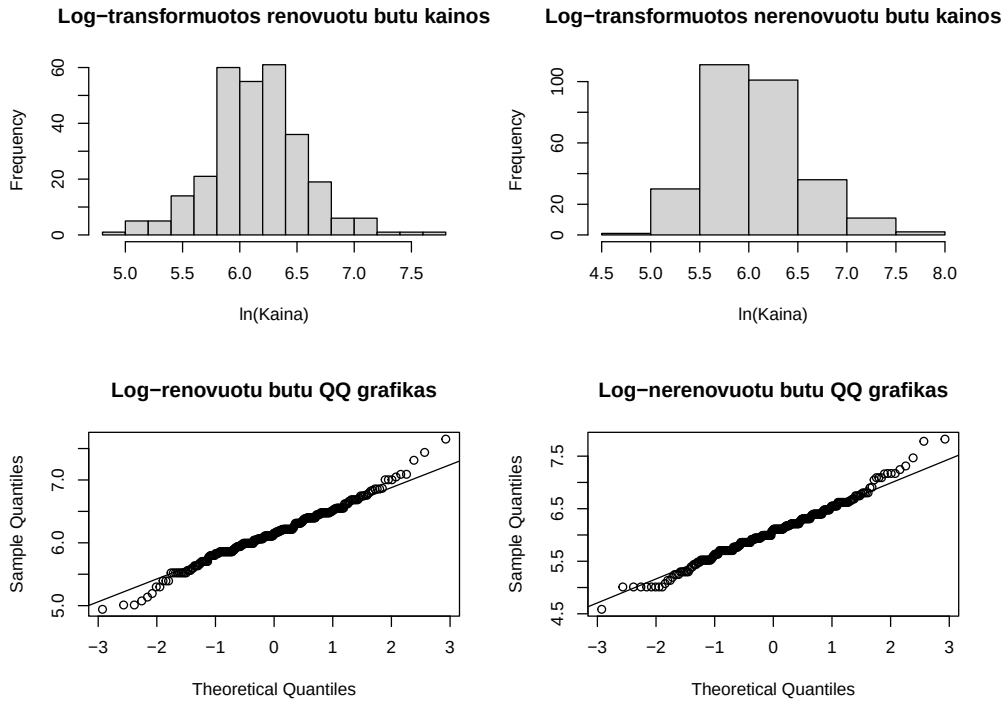
```
# Filtruojame ir atspausdiname renovuotus ir nerenovuotus butus su jų kainomis
build_year_data <- csv_data_list[["apartments_rent"]]$build_year
price_data <- csv_data_list[["apartments_rent"]]$price

# 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$build_year[i])
  if (grepl("construction", x) && grepl("renovation", x)) {
    construction_year <- as.numeric(substr(x, 1, 4))
    renovation_year <- as.numeric(substr(x, regexpr("renovation", x) - 5, regexpr("renovation", x) - 2))
    if (!is.na(construction_year) && !is.na(renovation_year) && construction_year < 2000 && renovation_y
      renovated_data <- rbind(renovated_data, data.frame(
        build_year = x,
        price = full_data$price[i],
        construction_year = construction_year
      ))
    }
  }
}

# Pervardijame renovuotų butų stulpelius
if (nrow(renovated_data) > 0) {
```

```r
  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))
)
```

```r
# 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[rando
  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)
```




```r
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 = 2.7235, df = 291, p-value = 0.003425
## alternative hypothesis: true mean difference is greater than 0
## 95 percent confidence interval:
##  0.0341075       Inf
## sample estimates:
## mean difference
##      0.08653854
```

```
alpha <- 0.05
df <- nrow(combined_data_clean) - 1
critical_t <- qt(1 - alpha, df)
critical_t
```

```
## [1] 1.650107
```

Statistinė išvada: kadangi $t = 2.9697 > 1.650107 = t_{0.05}(290)$, tai darome išvadą, kad renovuotų butų nuomos kainos vidurkis yra statistiškai reikšmingai didesnis nei nerenovuotų butų nuomos kainos vidurkis (H_0 atmetame).

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

7 Šalitiniai

https://github.com/valdas-v1/lithuanian-real-estate-listings

Github teikiami duomenys apie parduodamus ir nuomojamus butus bei kitas patalpas, prieiga per internetą: https://github.com/valdas-v1/lithuanian-real-estate-listings;

Prof., Dr. Jurgitos Markevičiūtės medžiaga DM statistikos kursui;