

Nekilnojamojo turto objektų kainų analizė Lietuvoje

Statistikos laboratorinis darbas Nr. 2

VU

2025-04-17

Contents

1 Įvadas	1
2 Duomenų aprašymas	1
2.1 Duomenų nuskaitymas	1
2.2 Duomenų patikrinimas ir išskirčių šalinimas	2
2.3 Duomenų vizualizacija	7
2.3.1 Kainų pasiskirstymo analizė	8
2.3.2 Komercinių patalpų ploto analizė	9
2.3.3 Namų nuomos kainos ryšys su plotu	11
3 Pagrindinės skaitinės charakteristikos	12
3.1 Kiekybinių kintamųjų aprašomoji statistika	12
4 Sudarykite dažnių lenteles kategoriniams kintamiesiems.	16
5 Suformuluokite bent 6 tyrimo hipotezes iš savo duomenų rinkinio	16
5.1 Namų ir butų dydžių palyginimas	16
5.2 Privačių pardavėjų proporcijos palyginimas tarp butų ir namų rinkų	16
6 Užrašykite kokius testus parinkote savo tyrimo hipotezėms. Hipotezės turi būti skirtos skirtingų testų naudojimui. Jei reikia susikurkite naujus kintamuosius iš turimų duomen.	16
7 Patikrinkite, ar kintamieji tenkina būtinas sąlygas testų taikymui. Jei netenkina, atlikite duomenų transformacijas.	16
7.1 Namų ir butų dydžių palyginimo duomenų paruošimas	16
7.2 Pardavėjų proporcija	18
8 Atlikite statistinį tyrimą savo suformuluotoms hipotezėms.	19
8.1 Namų ir butų dydžių statistinis tyrimas	19
8.2 Pardavėjų proporcija	19

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. Tyrime naudojami duomenys apima kainų, ploto, vietos ir kitų svarbių charakteristikų informaciją.

2.1 Duomenų nuskaitymas

```
# Duomenų vieta
data_dir <- "C:/Users/zabit/Documents/GitHub/Statistikos-lab-2/data"

# Gauname aplankų pavadinimus
folders <- list.dirs(data_dir, full.names = FALSE, recursive = FALSE)

# Atspausdiname visų aplankų pavadinimus
kable(data.frame(Kategorijos = folders),
       caption = "Nekilnojamojo turto duomenų kategorijos") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

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 failų nuskaitymas į sąrašą
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 pradėdant statistinę analizę, būtina identifikuoti ir pašalinti galimai klaidingas ar nekorektiškas reikšmes duomenyse. Nėkilnojamojo 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]])) {
    # Identifikuojame kraštutines reikšmes
    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
if (nrow(removal_results) > 0) {
  kable(removal_results,
        caption = "Išskirčių šalinimo rezultatų suvestinė") %>%
    kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
}

```

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

```

# 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"))

```

Table 3: 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

Pašalintos ekstremalios kainos, kurios galėjo iškreipti vidutines reikšmes ir kitas statistines charakteristikas.

```

# Sukuriame lentelę su stulpelių srašais kiekvienam duomenų rinkiniui
columns_by_dataset <- data.frame(

```

```

Duomenų_rinkinys = character(),
Stulpelių_skaičius = integer(),
Stulpelių_pavadinimai = character(),
stringsAsFactors = FALSE
)

# Pildome lentelę informacija apie stulpelius
for (folder_name in names(csv_data_list)) {
  columns_by_dataset <- rbind(columns_by_dataset, data.frame(
    Duomenų_rinkinys = folder_name,
    Stulpelių_skaičius = ncol(csv_data_list[[folder_name]]),
    Stulpelių_pavadinimai = paste(colnames(csv_data_list[[folder_name]]), collapse = ", ")
  ))
}

# Atvaizduojame lentelę su stulpelių informacija
kable(columns_by_dataset,
  caption = "Kiekvieno duomenų rinkinio stulpelių struktūra" %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
    latex_options = c("scale_down", "hold_position"),
    font_size = 8) %>%
  column_spec(1, width = "8em") %>%
  column_spec(2, width = "8em") %>%
  column_spec(3, width = "32em")

```

Table 4: Kiekvieno duomenų rinkinio stulpelių struktūra

Duomenų_rinkinys	Stulpelių_skaičius	Stulpelių_pavadinimai
apartments	38	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, number_of_rooms, area, floor, no._of_floors, build_year, equipment, building_type, heating_system, link, add_date, modified, selected, views_total, views_today, house_no., flat_no., building_energy_efficiency_class, description_tags, additional_premises, security, additional_equipment, valid_till, unique_item_number, object
apartments_rent	38	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, price_per_month, house_no., number_of_rooms, area, floor, no._of_floors, build_year, equipment, building_type, heating_system, description_tags, additional_premises, additional_equipment, security, link, add_date, modified, selected, views_total, views_today, valid_till, flat_no., building_energy_efficiency_class, unique_item_number
garages_parking	28	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, area, type, accommodates_no._of_cars, features, link, add_date, modified, valid_till, selected, views_total, views_today, number, unique_item_number, description_tags
garages_parking_rent	27	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, number, area, type, accommodates_no._of_cars, features, link, add_date, modified, valid_till, selected, views_total, views_today, unique_item_number

house_rent	40	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, price_per_month, plot_area, area, no._of_floors, build_year, equipment, building_type, heating_system, link, add_date, modified, valid_till, selected, views_total, views_today, number_of_rooms, water_system, closest_body_of_water, distance_from_body_of_water, building_energy_efficiency_class, description_tags, additional_premises, additional_equipment, security, house_no., unique_item_number
houses	39	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, plot_area, area, no._of_floors, build_year, equipment, building_type, heating_system, link, add_date, modified, selected, views_total, views_today, house_no., number_of_rooms, water_system, closest_body_of_water, distance_from_body_of_water, description_tags, additional_premises, additional_equipment, security, valid_till, building_energy_efficiency_class, unique_item_number
land	27	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, area_a., purpose, type, link, add_date, modified, views_total, views_today, description_tags, valid_till, selected, unique_item_number, lot_no.
land_rent	27	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, lot_no., area_a., purpose, type, link, add_date, modified, valid_till, selected, views_total, views_today, description_tags, unique_item_number
premises	37	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, house_no., area, floor, no._of_floors, build_year, equipment, premises_sum, purpose, heating_system, water_system, description_tags, additional_equipment, link, add_date, modified, selected, views_total, views_today, unique_item_number, premises_nr., valid_till, security, building_energy_efficiency_class
premises_rent	37	listing_id, type_id, price, region, microdistrict, street, coordinates, images, description, phone_number, private_seller, call_forwarding, reserved, sold_or_rented, price_per_month, house_no., area, floor, no._of_floors, equipment, purpose, building_energy_efficiency_class, link, add_date, modified, valid_till, selected, views_total, views_today, heating_system, additional_equipment, security, water_system, description_tags, premises_nr., build_year, unique_item_number

```

# 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 unikalų stulpelių analizę
kable(column_presence,
       caption = paste("Unikalų stulpelių pasikartojimas duomenų rinkiniuose (iš viso:",
                        nrow(column_presence), "stulpeliai)"),
       row.names = FALSE) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed")) %>%
scroll_box(width = "100%", height = "300px")

```

Table 5: 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

2.3 Duomenų vizualizacija

Grafikai padės geriau suprasti Lietuvos nekilnojamojo turto rinkos ypatybes.

```
# Papildomų vizualizacijai reikalingų paketų įkėlimas
library(ggplot2)
library(scales)
library(gridExtra)
library(ggExtra)

# 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)
    )
}
```

2.3.1 Kainų pasiskirstymo analizė

Pirmiausiai 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"]])) {
  # Pasiruošiamo duomenis
  df <- data.frame(price = csv_data_list[["apartments"]]$price)
```

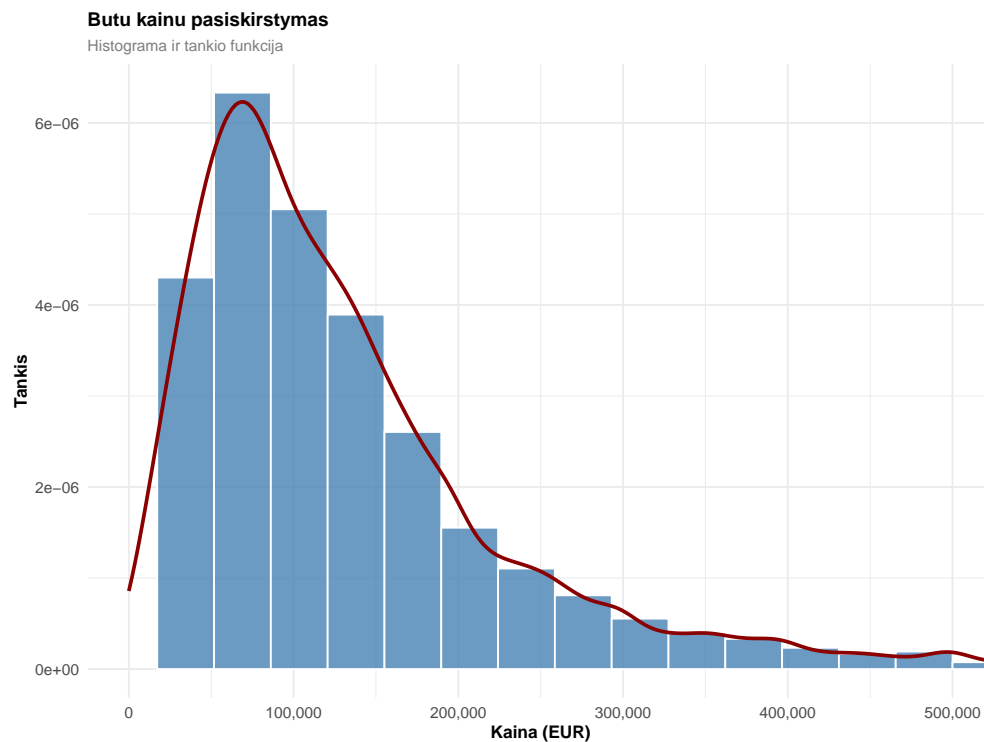


```

# 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)
}

```



2.3.2 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 (puz., neigiamus ar per didelius)
    df <- df[!is.na(df$area) & df$area > 0 & df$area < 10000, ]

    # Užtikriname, kad visi stulpeliai būtų vienodi abiem šaltiniams (premises ir premises_rent)
    if (length(premises_data) > 0) {
      # Nustatome bendrus stulpelius tarp esamo ir pridiamo 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
  }
}

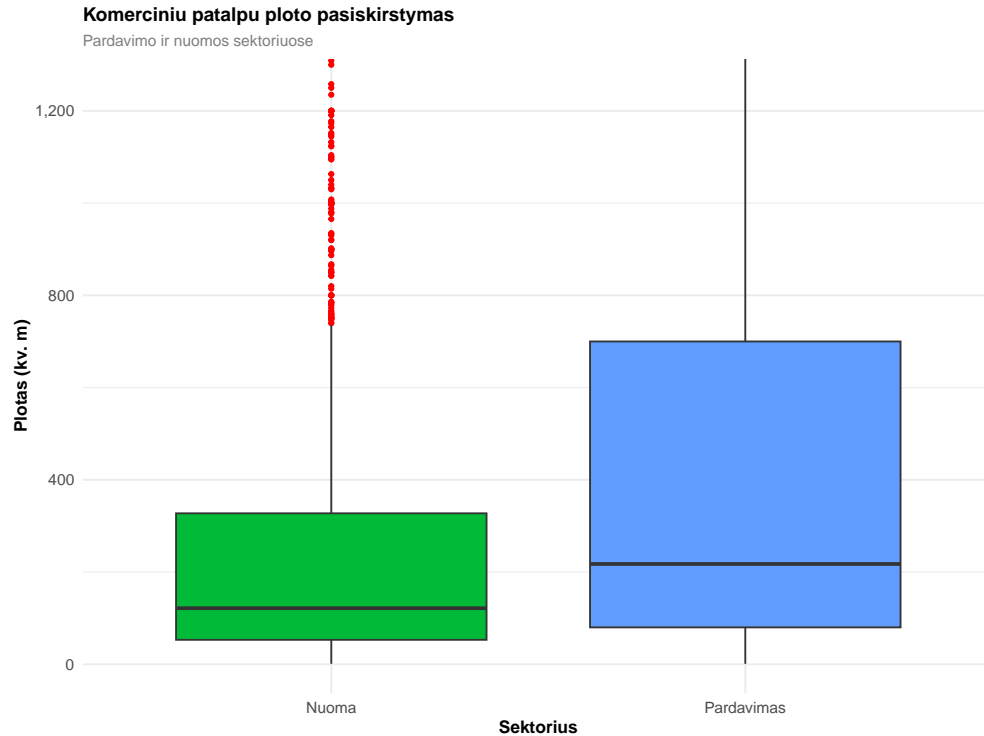
# Sujungiame duomenis, užtikrindami stulpelių suderinamumą
if (length(premises_data) == 2) {
  # Užtikriname, kad stulpeliai abiem šaltiniuose būtų identiški
  common_cols <- intersect(colnames(premises_data[[1]]), colnames(premises_data[[2]]))
  premises_data[[1]] <- premises_data[[1]][, common_cols, 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 = "Komerinių 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, 1250))

print(area_boxplot)

```



2.3.3 Namų nuomos kainos ryšys su plotu

Analizuojame, kaip namų nuomos kainų dydis priklauso nuo ploto.

```
# Namų nuomos kainos ir ploto priklausomybės analizė
if ("house_rent" %in% names(csv_data_list) &&
    all(c("price", "area") %in% colnames(csv_data_list[["house_rent"]])) {

  # Pasiruošiame duomenis
  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
```

3 Pagrindinės skaitinės charakteristikos

3.1 Kiekybinių kintamųjų aprašomoji statistika

Šiame skyriuje pateikiamos pagrindinės skaitinės charakteristikos kiekybiniais kintamiesiems.

```

# Duomenų rinkinių filtravimas pagal stulpelio pavadinimą

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) {
  # Sukuriame tuščią rezultatų lentelę su lietuviškais pavadinimais
  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", "garages_parking", "houses", "land", "premises")
rent_datasets <- c("apartments_rent", "house_rent", "premises_rent")
all_datasets <- c("apartments", "apartments_rent", "garages_parking", "garages_parking_rent",
  "house_rent", "houses", "land", "land_rent", "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"))

```

Table 6: Pardavimų kainų statistika pagal nekilnojamojo turto tipą

Duomenų rinkinys	Vidurkis	Mediana	Stand. nuokr	Q1	Q3	Minimumas	Maksimumas
apartments	143718.13	107558	146129.71	64000	172000	43	2500000
garages_parking	19015.55	15000	19477.64	10000	22499	500	248000
houses	183734.43	140000	223884.94	55000	235000	200	4200000
land	115388.60	35000	386437.38	18000	79900	100	12000000
premises	413170.38	165000	762212.43	70000	399850	490	10000000

```

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"))

```

Table 7: 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"))
```

Table 8: 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
garages_parking	727	433	1017	194	876	13	12209
garages_parking_rent	374	173	728	80	404	6	7521
house_rent	1275	582	2332	262	1411	20	24014
houses	2247	1133	3549	501	2612	2	71418
land	869	346	2965	140	872	1	191374
land_rent	477	256	560	100	619	11	2658
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"))
```

Table 9: 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"))
```

Table 10: 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

4 Sudarykite dažnių lentelės kategoriniams kintamiesiems.

5 Suformuluokite bent 6 tyrimo hipotezes iš savo duomenų rinkinio

5.1 Namų ir butų dydžių palyginimas

Nulinė hipotezė (H_0): vidutinės namų ir butų dydžiai yra lygūs

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

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

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

kur:

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

5.2 Privačių pardavėjų proporcijos palyginimas tarp butų ir namų rinkų

Nulinė hipotezė (H_0): Privačių pardavėjų proporcija butų ir namų rinkose yra vienoda

$$H_0 : p_{apartments} = p_{houses}$$

Alternatyvioji hipotezė (H_1): Privačių pardavėjų proporcija butų rinkoje skiriasi nuo privačių pardavėjų proporcijos namų rinkoje

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

kur:

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

6 Užrašykite kokius testus parinkote savo tyrimo hipotezėms. Hipotezės turi būti skirtos skirtingų testų naudojimui. Jei reikia susikurkite naujus kintamuosius iš turimų duomen.

Namų ir butų dydžių hipotezei naudosime dviejų nepriklausomų imčių t-testą. Privačių pardavėjų proporcijų palyginimui naudosime dviejų proporcijų testą (z-testas proporcijoms),

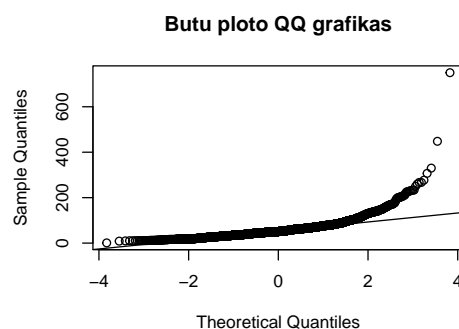
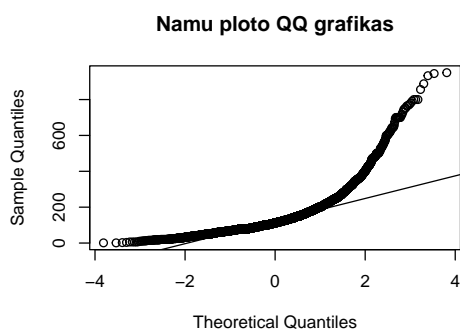
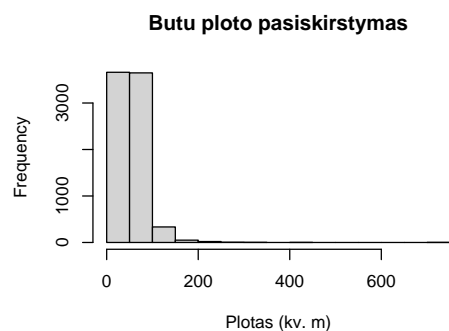
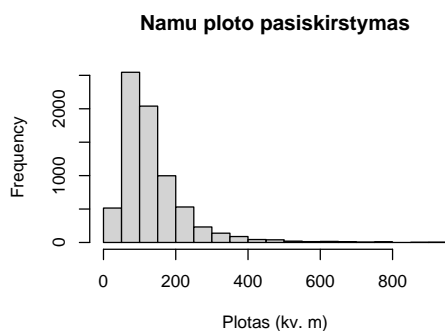
7 Patikrinkite, ar kintamieji tenkina būtinas sąlygas testų taikymui. Jei netenkina, atlikite duomenų transformacijas.

7.1 Namų ir butų dydžių palyginimo duomenų paruošimas

```
# Paruošiame duomenis testui kaip ir anksčiau
houses_area <- as.numeric(gsub(",", ".", as.character(csv_data_list[["houses"]][extract_itex]area)))
apartments_area <- as.numeric(gsub(",", ".", as.character(csv_data_list[["apartments"]][extract_itex]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]

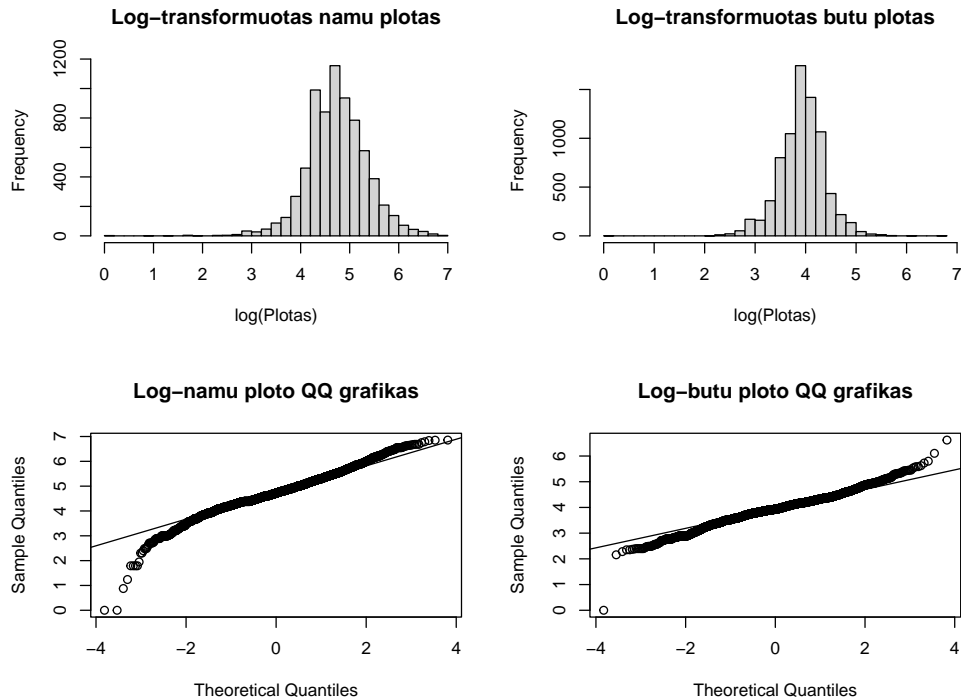
# Skirstiniai su histogramomis ir QQ grafikais
par(mfrow=c(2,2))
hist(houses_area, main="Namų ploto pasiskirstymas", xlab="Plotas (kv. m)", breaks=25)
hist(apartments_area, main="Butų ploto pasiskirstymas", xlab="Plotas (kv. m)", breaks=25)
qqnorm(houses_area, main="Namų ploto QQ grafikas")
qqline(houses_area)
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))
```

7.2 Pardavėjų proporcija

```
# 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)]

# Konvertuojame į logines reikšmes (TRUE/FALSE), jei dar nėra tokios
if (!is.logical(apartments_private)) {
  apartments_private <- apartments_private == "True"
}

if (!is.logical(houses_private)) {
  houses_private <- houses_private == "True"
}
```

```

}

# Skaiciuojame 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

```

8 Atlikite statistinį tyrimą savo suformuluotoms hipotezėms.

8.1 Namų ir butų dydžių statistinis tyrimas

```

# Log-transformuoti duomenys
log_t_test_rezultatas <- t.test(log_houses_area, log_apartments_area,
                                alternative = "greater",
                                var.equal = FALSE)

print(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

```

8.2 Pardavėjų proporcija

```

# Dviejų proporcijų testo atlikimas
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
```

9 Pateikite tyrimo atsakymą.