MATH3821 Assignment 2 - Report

Submitted 24th July 2022

Student Declaration

We declare that this assessment item is our own work, except where acknowledged, and has not been submitted for academic credit elsewhere. We acknowledge that the assessor of this item may, for the purpose of assessing this item reproduce this assessment item and provide a copy to another member of the University; and/or communicate a copy of this assessment item to plagiarism checking service (which may then retain a copy of the assessment on its database for the purpose of future plagiarism checking). We certify that we have read and understood the University rules in respect of Student Academic Misconduct.

Kirat Kounsal (z5163354),

Samuel McLeod (z5061746), S. McLeod

Martin Tran (z5330510),

Goal of Statistical Analysis

The New York City housing market is uniquely characterised by its disproportionate level of renters compared to homeowners. Due to diminished housing supply, both rent and housing prices have experienced great increases over time, a reflection of the housing crises seen in many major cities worldwide, including some in Australia, such as Sydney and Melbourne. Due to its ongoing housing crisis, historically stemming back to lack of buildable land, many New York properties have been subject to rent control or rent stabilisation measures. As of 2017, roughly half of all apartments are subject to such policies. Intended to protect eligible those with low socioeconomic status, some argue that rent policies put upwards pressure on regular rental prices due to constricted supply, and discourage construction of affordable housing, thus failing to address the central issue of housing shortages. Through analysis of the New York housing dataset, we aim to uncover the physical, demographic, and economic factors contributing to rent and housing prices, and determine whether rent control policies have had an observable effect on the New York rental market as a whole. We will do this through a statistical framework to predict housing prices for rental apartments, and observe whether or not they have stayed in line with rent over time.

Data Collection and Exploration

The dataset used in our analysis was sourced from the New York City Housing and Vacancy Survey (NYCHPD, 2017), which comprises of answers to questionnaires directed to the greater New York City area's housing population between the years 1991 and 2017. With 35 unique variables and 102218 unique participants, the dataset used thoroughly captures the nuanced and diverse paradigms of New York's housing population. Several methods of data cleaning were employed. The first phase of cleaning involved appropriately naming each variable, recording categorical variables as numeric via binary and dummy variables, and omission of NA (missing) values. After this, to ease readability and smoothen the analysis, several variables such as number of stories and length of lease were divided into ranges and tagged with a numeric median value to mitigate high variance within specific outlying ranges. Lastly, a noticeable issue with the data was the presence of placeholder values of 9999999 and 999999 for household value and rent respectively, denoting whether the property was rented or owned. As such, the data was also split into a dataset for renters and a separate one for homeowners.

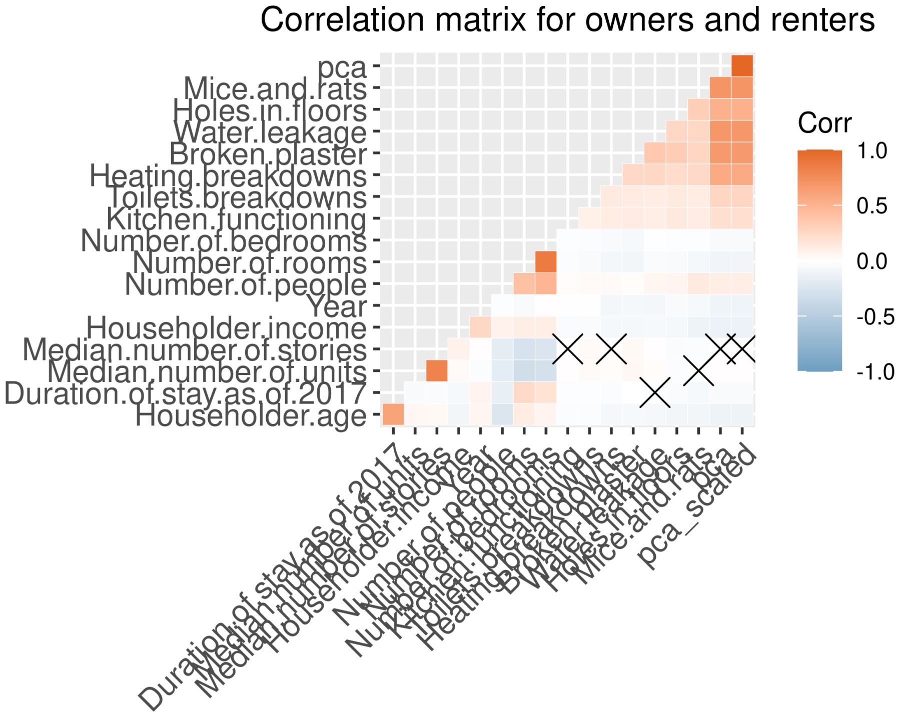
The given variables in the dataset could be categorised into groups that reflected their nature. We were able to sort them into 3 main themes:

- Social factors: Householder sex, age, and race
- Fixed physical factors: Number of units, stories, rooms, bedrooms, plumbing and kitchen facilities, general building condition and location.
- Short term physical factors (these generally related to the wear and tear of the property: Various factors such as the severity of damage to walls, floors, and other issues such as mice/rats and plumbing issues, as well as current resident rating.

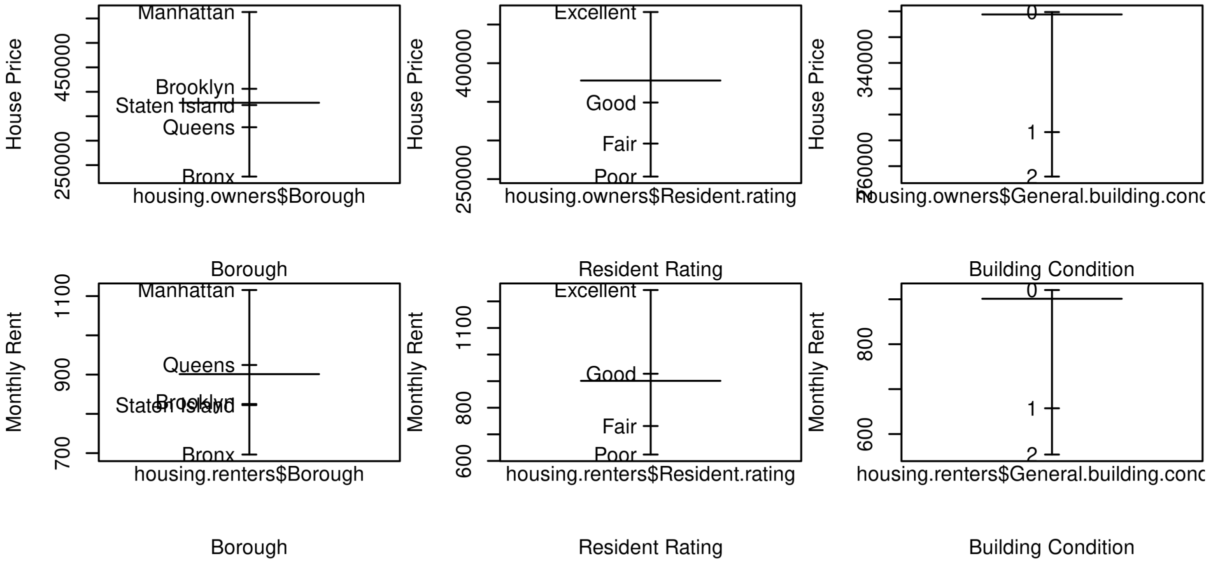
There was also a "pca" variable in the dataset which indexed the general building quality and short term features. This was created using principal component analysis which is used to aid dimensional reduction.

Although there were too many variables to consider pairwise plots, we were able to construct a correlation matrix for the non-factor variables.

From the correlation matrix below, it is evident all of the short term physical features are positively correlated with the pca index, rendering it an appropriate candidate predictor to use for simplify our model. Naturally, some of the other variables are strongly correlated with each other, as they are quite similar. For example, the number of rooms and bedrooms in a house are strongly linked, as well as the number of units and stories in a building. This information will help us to choose predictors that will avoid multicollinearity in the model.



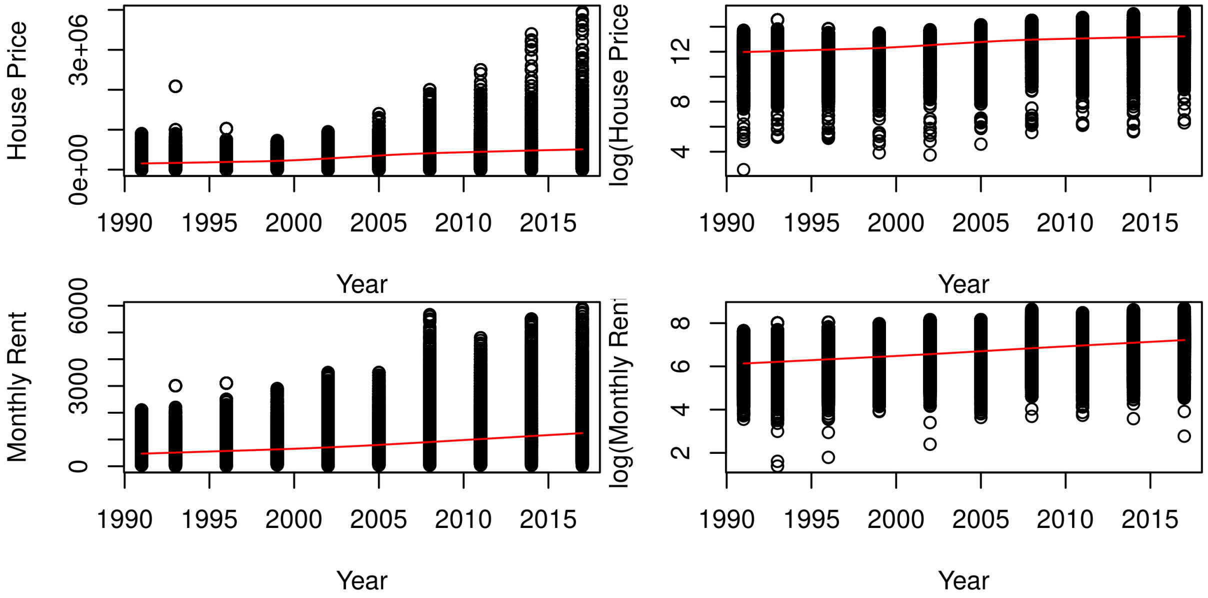
We can also analyse factor box plots to determine whether they may be significant in predicting house prices. Of interest are the Borough (the location), Resident rating, and general building condition.



Since there are clearly differences between the rent/housing prices for each of the factors, they may be useful for our model.

Model Choice

Since we aim to predict housing prices, which are continuous variables, a linear model is an intuitive and in our opinion appropriate approach. Yet, it was also clear from residual plots for the standard linear model that transformations may be necessary. We know housing and rent prices are expected to increase with time through natural inflation, and as such, plotting house and rent prices against years will give a clear indication of which transformation is most appropriate.



From the above plots, it is clear that taking the log of both prices allows for a sounder model than the standard specification as indicated by both the roughly constant variance over time and more linear smoothed trend line, which was achieved via loess smoothing. However, there were more benefits.

- Whilst the variance of the log models do not appear to strictly follow a normal distribution, it exceeded the fan shape of the standard model.
- Residual plots transformation suggested it upheld the assumptions of a linear model best, which will be further elaborated further in the Diagnostics section.
- Additionally, the linear model with log transformation achieved a larger Adjusted R squared value than several other transformations, including square root and inverse.
- Furthermore, the log-linear specification significantly increases the ease of interpretation for estimated coefficients.
- Whilst a GLM of gamma family is also a viable model for our dataset, we tended towards simplicity, both due to depth constraints and to allow for ease of interpretability.

As such, we chose to use a linear model, with log(House Price) and log(Monthly Rent) as the response variables.

Model Fitting

The next step is fitting the model to our dataset and determining the best subset of variables to use.

As such, model fitting was carried out via the Akaike Information Criterion (AIC), where we employed both forward and backward selection. The AIC method is derived from Information theory and estimates information loss, eventually choosing the model with least information loss based on the given variables. It is appropriate here because it balances both goodness of fit and parsimony whilst maintain asymptotic validity. Forward selection of the homeowner data linear model omitted 16 erroneous variables from the base model, with a final AIC score of -6732.85. This was supported by a subsequent backward selection on the same model and dataset, which both ordered the variables by degree of influence on the model, and through inspection of AIC scores omitted the same 16 variables.

A similar method was employed on the renters dataset and linear model, with forward selection omitting 8 variables and backward selection ordering the variables by degree of influence. An interesting observation here is that with only 8 omitted variables whilst the model based on homeowners omitted 16, the rent based model has more predictors than the homeowner model. Intuitively, this makes sense since the rental framework is more heavily influenced by micro-factors not entirely considered by homeowners.

A search for the models which had the highest adjusted R-squared values was then performed. Adjusted R-squared is a measure of the goodness-of-fit of a model that penalizes having more predictors. The best models in terms of adjusted R-squared for both the owners and renters were very similar to the models selected above. In terms of adjusted R-squared, the selected models were nearly identical to the models

Stepwise Model Selection			
log(Household.value) Model			
Variables		Criterion	
Year	General.building.condition	Adj-R2: 0.43	
Borough	Broken.plaster	AIC: -6732.85	
Number.of.rooms	Mice.and.rats		
Number.of.units	Kitchen.functioning		
Number.of.stories	Holes.in.floor		
Resident.rating	Heating.breakdowns		
log(Monthly.rent) Model			
Variables			Criterion
Year	General.building.condition	Severity.stairways	Adj-R2: 0.34
Borough	Broken.plaster	Toilets.breakdowns	AIC: -36599.63
Number.of.rooms	Mice.and.rats	Water.leakage	
Number.of.units	Kitchen.functioning	Plumbing.facilities	
Number.of.stories	Severity.windows	Kitchen.facilities	
Resident.rating	Severity.floors		

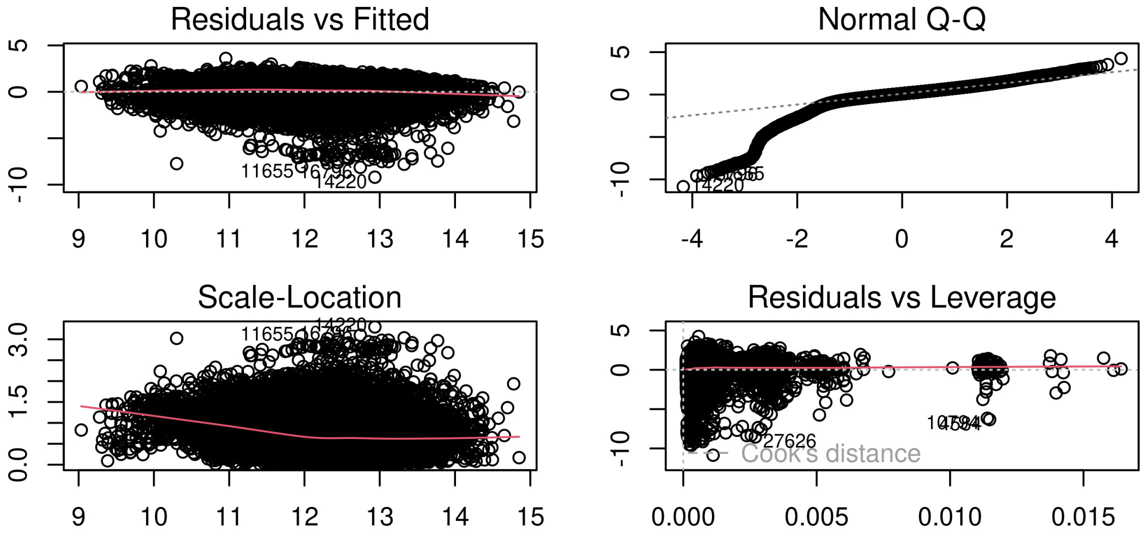
with the best adjusted R-squared, clearly supporting our choices of models.

With these new linear models, not only have we attained a higher level of simplicity, but also a better overall fit to the data.

Diagnostics

An essential component in regression analysis is verifying the diagnostics, both the assumptions used or violated, and any statistical limitations.

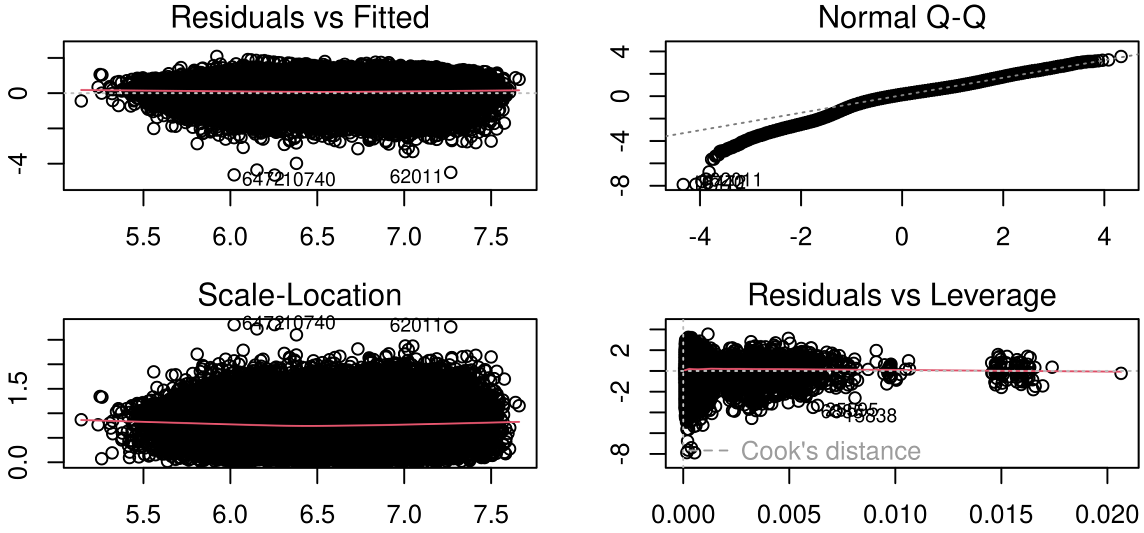
For the new log-transformed Linear Model to predict household value:



- Residuals vs Fitted suggest a fairly linear relationship due to the horizontal line.
- Normal Q-Q plot suggests normality is an inaccurate assumption for the residuals since the tail tends greatly away from the suggested line. This suggests that the distribution of the residuals is skewed.
- Scale-Location plot suggests that the variance of residuals is fairly homogeneous but does experience a pivot point that is undesirable.
- Residuals vs Leverage plot suggest a high number of outliers (plethora of data points with standardized residuals greater than absolute value 3. A sizable number of values are far beyond the Cook's distance regression line, meaning a large chunk of data points will have an impact on the

regression slope parameter if removed. This was expected as the housing data covers a very wide range of properties, so there are bound to be outliers.

We can conduct a similar analysis for our rent prediction model:



- Both plots of residuals against fitted and standardised residuals against fitted unveil a horizontal trend line, indicative of both linearity and homoscedasticity.
- Normality, whilst still not an accurate assumption, is slightly improved, as indicated by the tails that are less skewed. This again was expected.
- Just like above, the existence of outliers with heavy influence is shown via the clear divergence of many data points from the Cook's distance trend line.

It should also be noted that both sets of residual plots were stark improvements on base models, and even models with other transformations.

Model Assessment

Whilst we didn't do any assessment on the predictive power of our chosen model, there are many techniques that can be utilised. Most rely on the notion of cross-validation, where the model is run on a subset (training set) of the data, and then its predictions generated are compared to the actual observed values. With such a large dataset, the leave-one-out CV method would have been too computationally expensive, so a K-fold CV method would be more appropriate to conduct predictive assessment. Being derived from a diverse and varied dataset, our model's limitation would be predictive accuracy, as there are many unobserved variables that our model would not able to capture.

Conclusion and Insights

From the models that we ran, we can conclude rental prices are determined by more micro-level physical factors than housing prices, however, both are determined by economic and fixed physical factors. We can use the models that we developed to provide more insight into our goal, which was to determine whether rent control measures have had a significant effect on the New York rental market as compared to its housing market. When running a simple linear regression on House Value, and the predicted rent derived from our model, for the years of 1991 and 2017, we were able to show that on average, in 1991, house value is roughly 463 times the monthly rent (s.e. 23.8), and in 2017, house value is around 768 times the monthly rent (s.e. 39.9). Since the 95% confidence intervals for these ratios do not overlap at all, we can conclude that New York's rent control policies have had a significant impact on dampening the effects of housing booms on renters.

Appendix

Data Cleaning

```r
# Packages ----------------------------------------------------------------------------
library(tidyverse)
library(ggforce)
library(ggcorrplot)
library(MASS)

# Load in data ------------------------------------------------------------------------
housing.df <- read.delim("dataset_processed.txt", header = TRUE, sep = ",")

# Clean data --------------------------------------------------------------------------
housing.df.cleaned <- housing.df %>%
  as_tibble() %>%
  # Make values NA
  mutate(Household.value = na_if(Household.value, 9999999),
         Monthly.rent = na_if(Monthly.rent, 99999)) %>%
  # Replace 100+ with 100-199 in number of units
  mutate(Number.of.units = replace(
    Number.of.units,
    Number.of.units == '100+ units',
    '100-199 units')) %>%
  # Replace negative income with 0
  mutate(Householder.income = replace(
    Householder.income,
    Householder.income < 0,
    0
  )) %>%
  # Get median number of units in each apartment
  mutate(Median.number.of.units = factor(
    Number.of.units,
    c('1 unit', '2 units', '3-5 units', '6-9 units', '10-19 units',
      '20-49 units', '50-99 units', '100-199 units', '200+ units'),
    c(1, 2, 4, 7.5, 14.5, 34.5, 74.5, 149.5, 249.5)
  ) %>% as.character() %>% as.numeric()) %>%
  # Create column for median number of stories
  mutate(Median.number.of.stories = factor(
    Number.of.stories,
    c("1 story", "1-2 stories", "2 stories", "3-5 stories", "6-10 stories",
      "11-20 stories", "21+ stories"),
    c(1, 1.5, 2, 4, 8, 15.5, 25.5)
  ) %>% as.character() %>% as.numeric()) %>%
  # Create column for median length of lease
  mutate(Median.length.of.lease = factor(
    Length.of.lease,
    c("Less than 1 year", "1 year", "Between 1 and 2 years",
      "2 years", "More than 2 years", "No lease", "Owner-occupied"),
    c(0.5, 1, 1.5, 2, 3.5, 0, NA)
  ) %>% as.character() %>% as.numeric()) %>%
  # Replace "8" with NA in some of the columns
  mutate(across(
    c(Householder.sex, Householder.hispanic.origin, Plumbing.facilities,
      Kitchen.facilities),
    ~ na_if(.x, '8')
  )) %>%
  # Replace blank cell with "Unknown" in race column
```

```
mutate(Householder.race = replace(
  Householder.race, Householder.race == '', 'Unknown')) %>%
# Change some columns to factors
mutate(across(
  c(Householder.sex, Householder.hispanic.origin, Householder.race,
    Number.of.units:Number.of.stories, Plumbing.facilities:Length.of.lease,
    Resident.rating, Borough, Status:General.building.condition),
  ~ as.factor(.))) %>%
# Re-order some of the factor levels
mutate(
  Number.of.units = factor(
    Number.of.units,
    levels = c("1 unit", "2 units", "3-5 units", "6-9 units", "10-19 units",
               "20-49 units", "50-99 units", "100-199 units", "200+ units")
  ),
  Number.of.stories = factor(
    Number.of.stories,
    levels = c("1 story", "1-2 stories", "2 stories", "3-5 stories",
               "6-10 stories", "11-20 stories", "21+ stories")
  ),
  Length.of.lease = factor(
    Length.of.lease,
    levels = c("No lease", "Less than 1 year", "1 year", "Between 1 and 2 years",
               "2 years", "More than 2 years", "Owner-occupied")
  ),
  Resident.rating = factor(
    Resident.rating,
    levels = c("Poor", "Fair", "Good", "Excellent")
  ),
  Severity.walls = factor(
    Severity.walls,
    levels = c("0", "1", "2")
  ),
  Severity.windows = factor(
    Severity.windows,
    levels = c("0", "1", "2", "3")
  ),
  Severity.stairways = factor(
    Severity.stairways,
    levels = c("0", "1", "2")
  ),
  Severity.floors = factor(
    Severity.floors,
    levels = c("0", "1", "2")
  ),
  General.building.condition = factor(
    General.building.condition,
    levels = c("0", "1", "2")
  )
) %>%
# Change binary factor variables to integers
mutate(
  Householder.female = factor(
    Householder.sex,
    c('Female', 'Male'),
    c(1, 0)
  ) %>% as.character() %>% as.integer(),
    Householder.hispanic.origin = factor(
      Householder.hispanic.origin,
      c("Hispanic", "Not hispanic"),
      c(1, 0)
    ) %>% as.character() %>% as.integer(),
    Owner.in.building = factor(
      Owner.in.building,
      c("Yes", "No"),
      c(1, 0)
    ) %>% as.character() %>% as.integer(),
    Plumbing.facilities = factor(
      Plumbing.facilities,
      c("Yes", "No"),
      c(1, 0)
    ) %>% as.character() %>% as.integer(),
    Kitchen.facilities = factor(
      Kitchen.facilities,
      c("Yes", "No"),
      c(1, 0)
    ) %>% as.character() %>% as.integer(),
    Status.Owner = factor(
      Status,
      c("Owner", "Renter"),
      c(1, 0)
    ) %>% as.character() %>% as.integer()
  ) %>%
  dplyr::select(-Householder.sex, -Status) %>%
  # Change some columns to double
  mutate(across(c(Household.value, Householder.age, Duration.of.stay.as.of.2017,
                  Monthly.rent, Householder.income),
                ~ as.double(.))) %>%
  # Change Number.of.people column to integer
  mutate(Number.of.people = as.integer(Number.of.people))

# View cleaned data frame
housing.df.cleaned %>% glimpse()

# View unique values for each variable
housing.df.cleaned %>%
  dplyr::select(!where(is.double)) %>%
  sapply(unique)

housing.df.cleaned %>%
  summary()

housing.df.cleaned %>%
  is.na() %>%
  colSums()

# Get data frames for renters and owners ----------------------------------

housing.df.cleaned.renters <-
  housing.df.cleaned %>%
  filter(Status.Owner == 0) %>%
  dplyr::select(-Status.Owner, -Household.value)

housing.df.cleaned.owners <-
```

```
  housing.df.cleaned %>%
  filter(Status.Owner == 1) %>%
  dplyr::select(-Status.Owner, -Monthly.rent, -Median.length.of.lease,
                -Length.of.lease, -Owner.in.building)
# -Household.value
housing.df.cleaned.owners.and.rents <-
  housing.df.cleaned %>%
  dplyr::select(-Monthly.rent, -Median.length.of.lease, -Household.value)

# Converting into Base R once cleaned ------------------------------------
housing <- data.frame(housing.df.cleaned)
housing.owners <- data.frame(housing.df.cleaned.owners)
housing.renters <- data.frame(housing.df.cleaned.renters)
```

Plots

```
# Correlation Matrices -------------------------------------------------
housing.df.cleaned.renters %>%
  dplyr::select(where(is.numeric)) %>%
  cor(use = 'complete.obs') %>%
  ggcorrplot(type = 'lower', hc.order = TRUE,
             outline.color = 'white',
             colors = c("#6D9EC1", "white", "#E46726"),
             p.mat = cor_pmat(dplyr::select(housing.df.cleaned.renters,
                                            where(is.numeric))),
             ggtheme = ggplot2::theme_gray) +
  labs(title = 'Correlation matrix for renters') +
  theme(plot.title = element_text(hjust = 0.5))

housing.df.cleaned.owners %>%
  dplyr::select(where(is.numeric)) %>%
  cor(use = 'complete.obs') %>%
  ggcorrplot(type = 'lower',
             hc.order = TRUE,
             outline.color = 'white',
             colors = c("#6D9EC1", "white", "#E46726"),
             p.mat = cor_pmat(dplyr::select(housing.df.cleaned.owners,
                                            where(is.numeric))),
             ggtheme = ggplot2::theme_gray) +
  labs(title = 'Correlation matrix for owners') +
  theme(plot.title = element_text(hjust = 0.5))

housing.df.cleaned.owners.and.rents %>%
  dplyr::select(where(is.numeric)) %>%
  cor(use = 'complete.obs') %>%
  ggcorrplot(type = 'lower', hc.order = TRUE,
             outline.color = 'white',
             colors = c("#6D9EC1", "white", "#E46726"),
             p.mat = cor_pmat(dplyr::select(housing.df.cleaned.owners.and.rents,
                                            where(is.numeric))),
             ggtheme = ggplot2::theme_gray) +
  labs(title = 'Correlation matrix for owners and renters') +
  theme(plot.title = element_text(hjust = 0.5))

# Factor Plots ----------------------------------------------------------
par(mfrow=c(2,3))
plot.design(housing.owners$Borough~housing.owners$Household.value,
            xlab = "Borough", ylab = "House Price")
plot.design(housing.owners$Resident.rating~housing.owners$Household.value,
            xlab = "Resident Rating", ylab = "House Price")
plot.design(housing.owners$General.building.condition~housing.owners$Household.value,
            xlab = "Building Condition", ylab = "House Price")

plot.design(housing.renters$Borough~housing.renters$Monthly.rent,
            xlab = "Borough", ylab = "Monthly Rent")
plot.design(housing.renters$Resident.rating~housing.renters$Monthly.rent,
            xlab = "Resident Rating", ylab = "Monthly Rent")
plot.design(housing.renters$General.building.condition~housing.renters$Monthly.rent,
            xlab = "Building Condition", ylab = "Monthly Rent")

# Price-Year Plots -----------------------------------------------------
plot(housing.owners$Year, housing.owners$Household.value, xlab = "Year",
     ylab = "House Price")
lines(loess.smooth(housing.owners$Year, housing.owners$Household.value), col = "red")

plot(housing.owners$Year, log(housing.owners$Household.value), xlab = "Year",
     ylab = "log(House Price)")
lines(loess.smooth(housing.owners$Year, log(housing.owners$Household.value)), col = "red")

plot(housing.renters$Year, housing.renters$Monthly.rent, xlab = "Year",
     ylab = "Monthly Rent")
lines(loess.smooth(housing.renters$Year, housing.renters$Monthly.rent), col = "red")

plot(housing.renters$Year, log(housing.renters$Monthly.rent), xlab = "Year",
     ylab = "Monthly Rent")
lines(loess.smooth(housing.renters$Year, log(housing.renters$Monthly.rent)), col = "red")
```

Model Choice and Selection

```
# Linear Models and Stepwise Selection

own.full.lm <- lm(log(Household.value) ~ .,
                  data = housing.owners)

summary(own.full.lm)
anova(own.full.lm)

own.intercept.lm <- lm(log(Household.value) ~ 1, data = housing.owners)
summary(own.intercept.lm)

rent.full.lm <- lm(log(Monthly.rent) ~ .,
                   data = housing.renters)

summary(rent.full.lm)

rent.intercept.lm <- lm(log(Monthly.rent) ~ 1, data = housing.renters)
summary(rent.intercept.lm)

forward.AIC.own <- stepAIC(own.intercept.lm,
                           scope = list(lower = own.intercept.lm,
                                        upper = own.full.lm),
                           direction = 'forward')
```

```
backward.AIC.own <- stepAIC(own.full.lm, direction = 'backward')

forward.AIC.own$call
backward.AIC.own$call

own.selected.lm <-
  lm(formula = log(Household.value) ~ Year + Median.number.of.units +
       Borough + Number.of.rooms + Resident.rating + Median.number.of.stories +
       Mice.and.rats + Broken.plaster + General.building.condition +
       Heating.breakdowns + Kitchen.functioning + Holes.in.floors,
     data = housing.owners)

summary(own.selected.lm)
anova(own.selected.lm)

par(mfrow = c(2, 2))
plot(own.selected.lm)
par(mfrow = c(1, 1))

rent.selected.lm <-
  lm(formula = log(Monthly.rent) ~ Year + Resident.rating + Borough +
       Number.of.rooms + Mice.and.rats + General.building.condition +
       Median.number.of.units + Broken.plaster + Median.number.of.stories +
       Severity.windows + Toilets.breakdowns + Severity.stairways +
       Kitchen.functioning + Water.leakage + Severity.floors + Plumbing.facilities +
       Kitchen.facilities, data = housing.renters)

summary(rent.selected.lm)
anova(rent.selected.lm)

par(mfrow = c(2, 2))
plot(rent.selected.lm)
par(mfrow = c(1, 1))

# Getting AdjR2 for models using regsubsets  ------------------------------
# Regsubsets for owners models to get best adjusted r-squared

own_best_subsets <- regsubsets(log(Household.value) ~ .,
                               data = housing.owners, nvmax = 100)
own_best_subsets_summary <- summary(own_best_subsets)

which.max(own_best_subsets_summary$adjr2)

# Model with best adjusted r^2
Number.of.parameters.owners <- which.max(own_best_subsets_summary$adjr2)
Adj.R2.owners <- own_best_subsets_summary$adjr2[Number.of.parameters.owners]
Parameters.included.owners <- own_best_subsets$xnames[
  own_best_subsets_summary$which[Number.of.parameters.owners, ]
]

Parameters.included.owners <- paste(Parameters.included.owners[-1], collapse = ', ')

best.adj.r2.owners <- data.frame(
  Number.of.parameters = Number.of.parameters.owners,
  Adj.R2 = Adj.R2.owners
)
```

```
best.adj.r2.owners
names(Parameters.included.owners) <-
  'Paramaters included in house owners model with best adjusted R-squared'

# Compare with selected model
summary(own.selected.lm)$adj.r.squared

Adj.R2.owners.comparison <- data.frame(
  'Adj.R2 from best regsubsets' = Adj.R2.owners,
  'Adj.R2 from selected model' = summary(own.selected.lm)$adj.r.squared
)
Adj.R2.owners.comparison

# Regsubsets for renters models to get best adjusted r-squared
rent_best_subsets <- regsubsets(log(Monthly.rent) ~ .,
                                data = housing.renters, nvmax = 100)
rent_best_subsets_summary <- summary(rent_best_subsets)

which.max(rent_best_subsets_summary$adjr2)

# Model with best adjusted r^2
Number.of.parameters.renters <- which.max(rent_best_subsets_summary$adjr2)
Adj.R2.renters <- rent_best_subsets_summary$adjr2[Number.of.parameters.renters]
Parameters.included.renters <- rent_best_subsets$xnames[
  rent_best_subsets_summary$which[Number.of.parameters.renters, ]
]

Parameters.included.renters <- paste(Parameters.included.renters[-1], collapse = ', ')

best.adj.r2.renters <- data.frame(
  Number.of.parameters = Number.of.parameters.renters,
  Adj.R2 = Adj.R2.renters
)

best.adj.r2.renters
names(Parameters.included.renters) <-
  'Paramaters included in house renters model with best adjusted R-square'
Parameters.included.renters

# Compare with selected model
summary(rent.selected.lm)$adj.r.squared

Adj.R2.renters.comparison <- data.frame(
  'Adj.R2 from best regsubsets' = Adj.R2.renters,
  'Adj.R2 from selected model' = summary(rent.selected.lm)$adj.r.squared
)
Adj.R2.renters.comparison

# The adjusted R-squared is very slightly smaller in both of the selected
# models. However, it is so close that it actually supports our choices.
```

Prediction and Evaluation

```
# Predicting ---------------------------------------------------------------
# Predicting rental prices for owned property data
owned.new.data <- housing.owners
owned.new.data$Predicted.monthly.rent <-
  exp(predict(rent.selected.lm,
              newdata =dplyr::select(owned.new.data, -Household.value)))

owned.new.data.2017 = owned.new.data[owned.new.data$Year == 2017,]
owned.new.data.1991 = owned.new.data[owned.new.data$Year == 1991,]

# Finding relationship between rent prices and house prices in 1991 and 2017.
summary(lm(Household.value~Predicted.monthly.rent, data = owned.new.data.2017))
summary(lm(Household.value~Predicted.monthly.rent, data = owned.new.data.1991))
```