4 Linear Regression

Immanuel Klein

This task is about finding linear regression models to predict house prices in King County using various covariates. We begin by fitting a linear model on `price` and `log(price)` and then extended it by adding non-linear squared terms for `yr_built` and `sqft_living`, in order to try to improve the model's fit. We compare the prediction accuracy of these models by splitting the dataset into training and test sets, calculating the MSE for each model. We try to further enhance prediction accuracy, and extend the model by adding interaction terms and additional polynomial terms.

```
library("ggplot2")
library("tidyverse")
library("car")
```

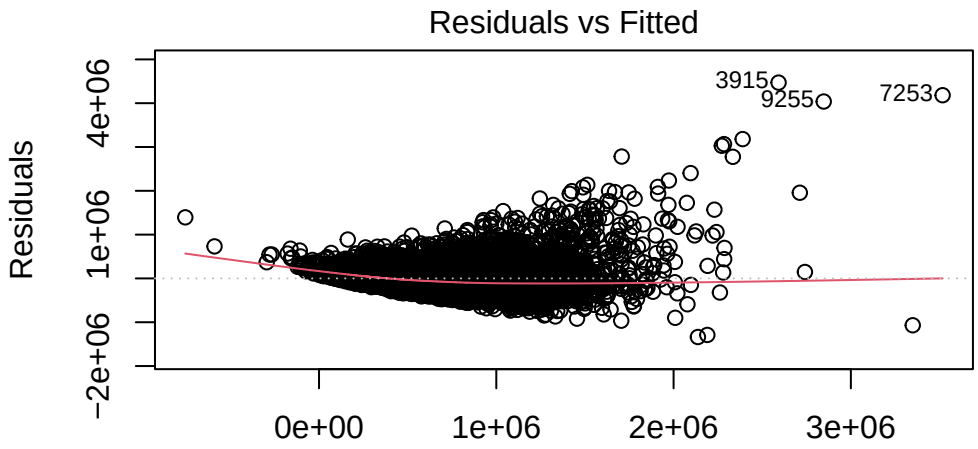
Exercise (a)

```
# Read data and use wanted variables only
data <- read.csv("kc_house_data.csv")[, c("price",
                                          "bedrooms",
                                          "bathrooms",
                                          "sqft_living",
                                          "floors",
                                          "view",
                                          "condition",
                                          "grade",
                                          "yr_built")]
# Fit linear model
model <- lm(price ~ bedrooms +
              bathrooms +
              sqft_living +
              floors +
              view +
```

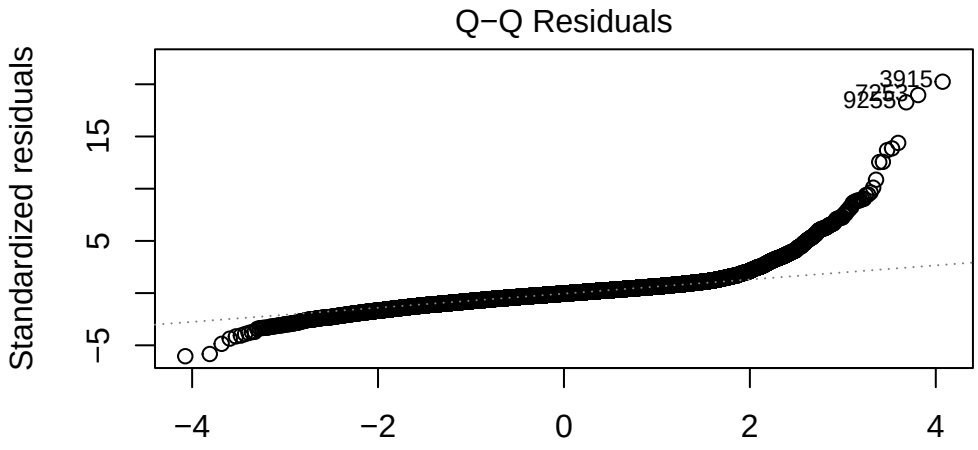
```
              condition +
              grade +
              yr_built,
            data = data)
summary(model)
```

```
Call:
lm(formula = price ~ bedrooms + bathrooms + sqft_living + floors +
    view + condition + grade + yr_built, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-1337280  -111873   -10359    90133  4470268

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.356e+06  1.326e+05  47.950  < 2e-16 ***
bedrooms    -4.065e+04  2.066e+03 -19.671  < 2e-16 ***
bathrooms    4.769e+04  3.487e+03  13.677  < 2e-16 ***
sqft_living  1.693e+02  3.307e+00  51.199  < 2e-16 ***
floors       2.832e+04  3.496e+03   8.101 5.72e-16 ***
view         7.138e+04  2.107e+03  33.885  < 2e-16 ***
condition    1.815e+04  2.519e+03   7.205 6.01e-13 ***
grade        1.228e+05  2.187e+03  56.160  < 2e-16 ***
yr_built    -3.650e+03  6.824e+01 -53.484  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 221600 on 21604 degrees of freedom
Multiple R-squared:  0.6359,    Adjusted R-squared:  0.6358
F-statistic:  4717 on 8 and 21604 DF,  p-value: < 2.2e-16
```

```
plot(model, which = 1)
```




```r
plot(model, which = 2)
```

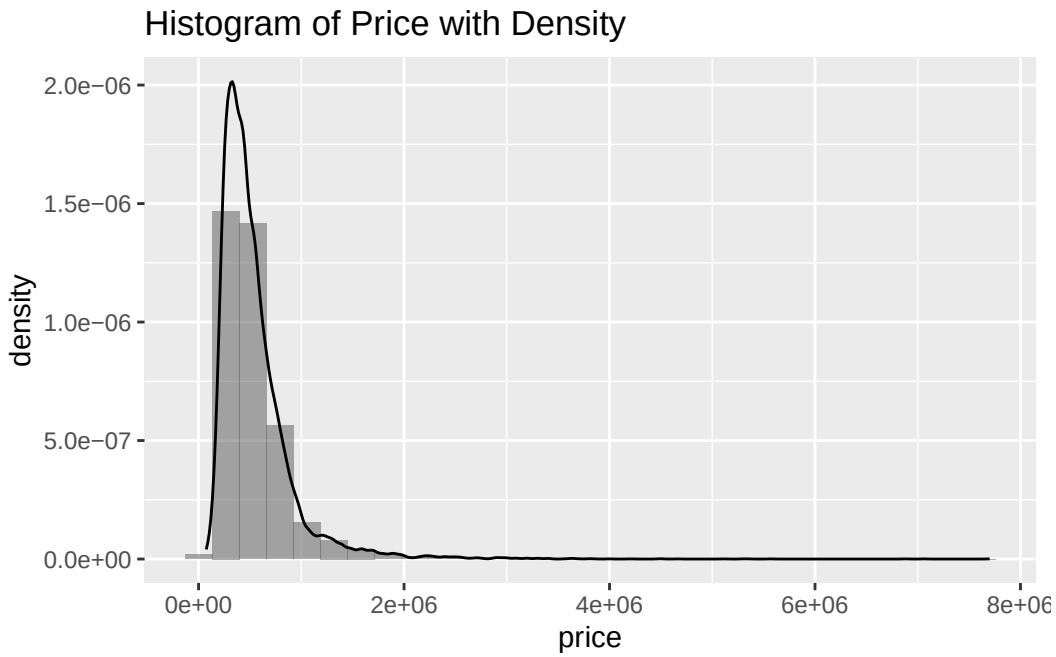


- All variables have p-values that are much lower than 0.05, meaning that all variables are statistically significant in predicting the house price.

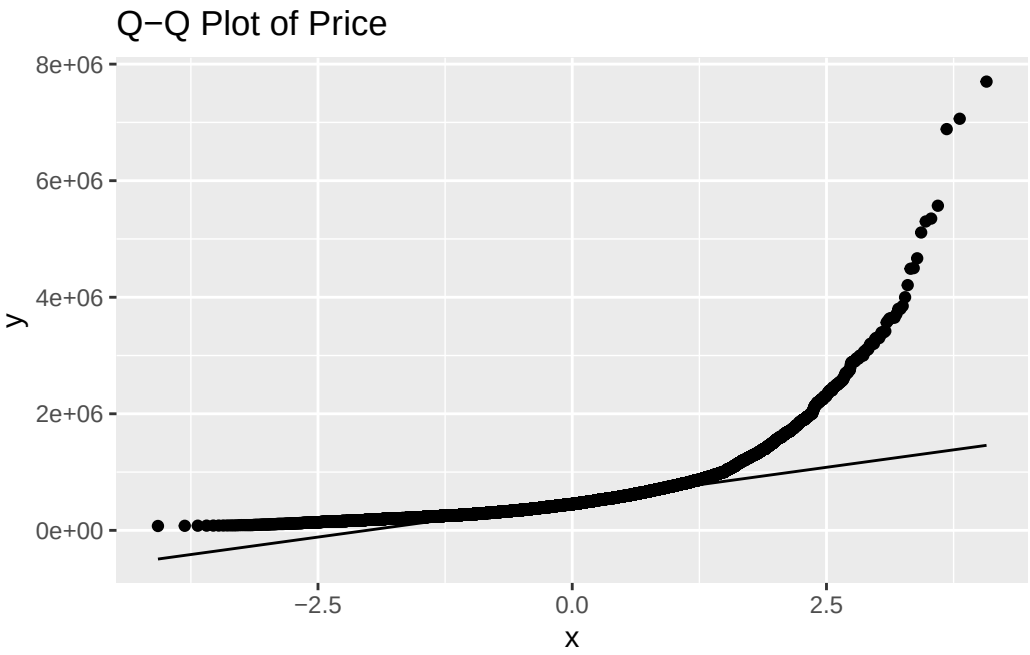
- The R-squared value of 0.6359 shows that rougly 64% of the variability in house prices is explained by the model, which means that the model has a somewhat moderate fit. The adjusted R-squared is very close to the R-squared value, so adding additional variables did not significantly worsen the model fit.
- We do the residual analysis by looking at the Residuals vs. Fitted and Q-Q Residuals plots of the model:
  - Residuals vs. Fitted: The red line is not completely horizontal, so there could be a possible non-linear relationship that our model is not capturing well. Also: as fitted values increase, the variance of the residuals also increases, which indicates heteroscedasticity.
  - Q-Q Residuals: There are deviations from the 45-degree line at both tails, indicating that the residuals are not perfectly normally distributed.

Exercise (b)

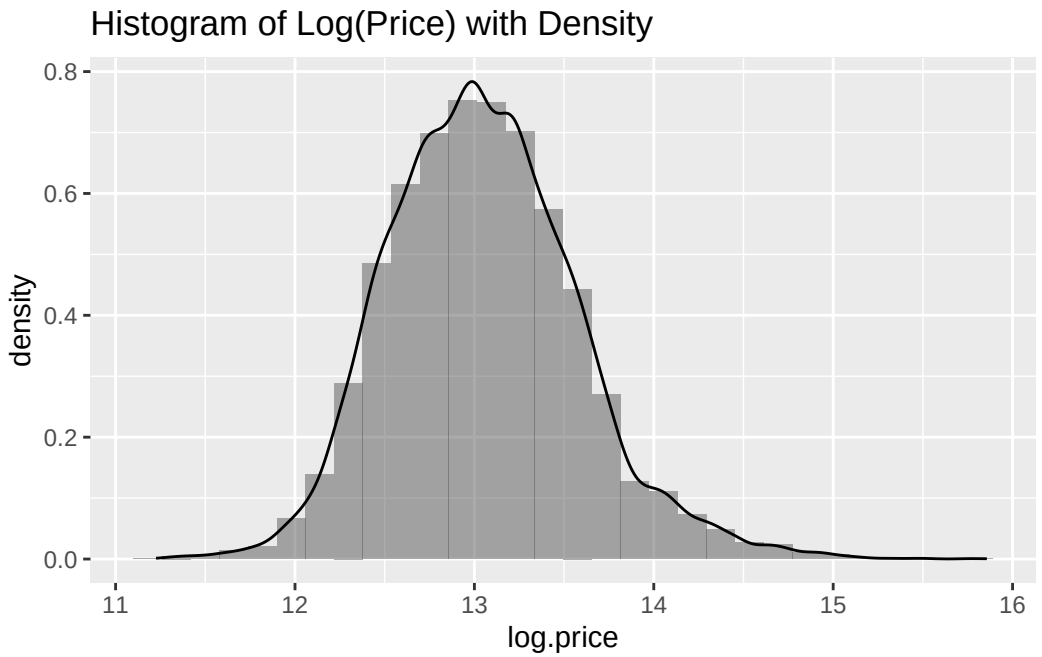
```
# Histogram and Q-Q plot for price
ggplot(data, aes(x = price)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,
                 alpha = 0.5) +
  geom_density() +
  ggtitle("Histogram of Price with Density")
```



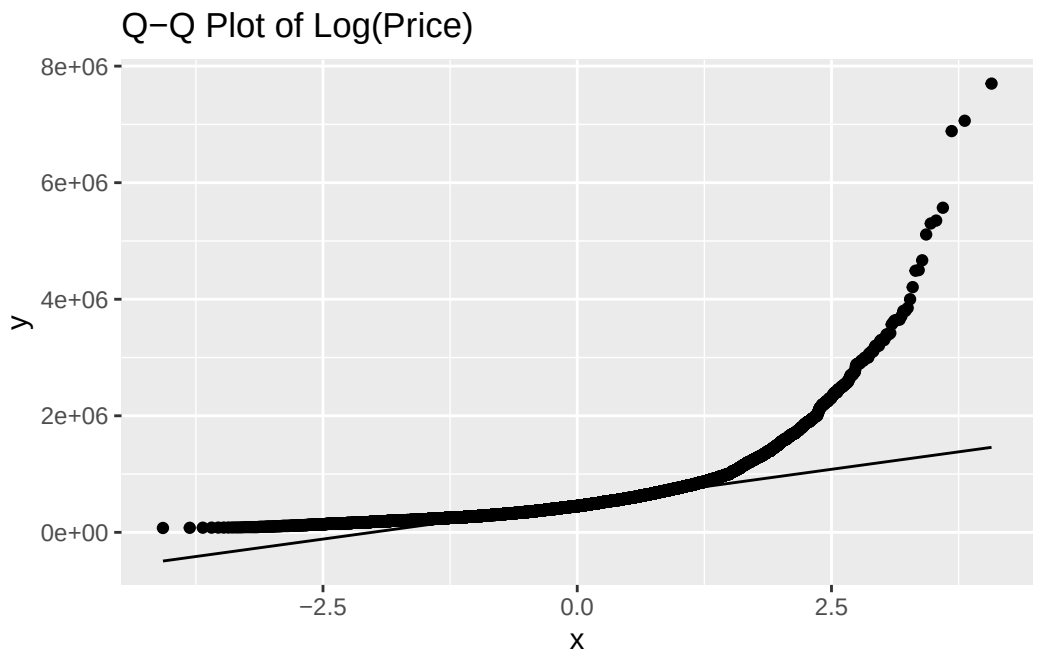

```r
ggplot(data, aes(sample = price)) +
  stat_qq() +
  stat_qq_line() +
  ggtitle("Q-Q Plot of Price")
```




```r
# Histogram and Q-Q plot for log(price)
data$log.price <- log(data$price)

ggplot(data, aes(x = log.price)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,
                 alpha = 0.5) +
  geom_density() +
  ggtitle("Histogram of Log(Price) with Density")
```




```r
ggplot(data, aes(sample = price)) +
  stat_qq() +
  stat_qq_line() +
  ggtitle("Q-Q Plot of Log(Price)")
```

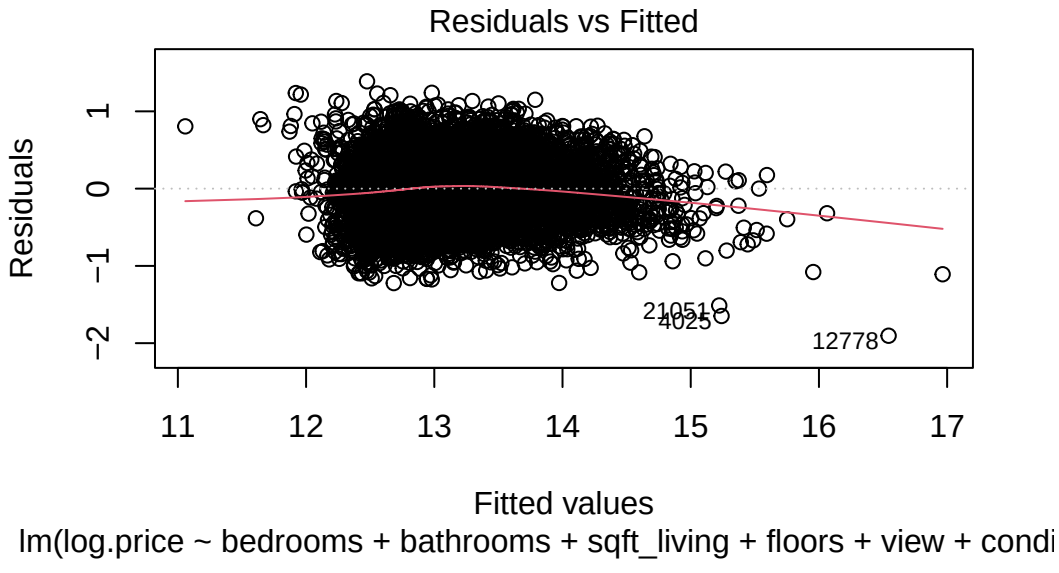



```
# Linear model with log(price)
model.log <- lm(log.price ~ bedrooms +
                  bathrooms +
                  sqft_living +
                  floors +
                  view +
                  condition +
                  grade +
                  yr_built,
                data = data)
summary(model.log)
```

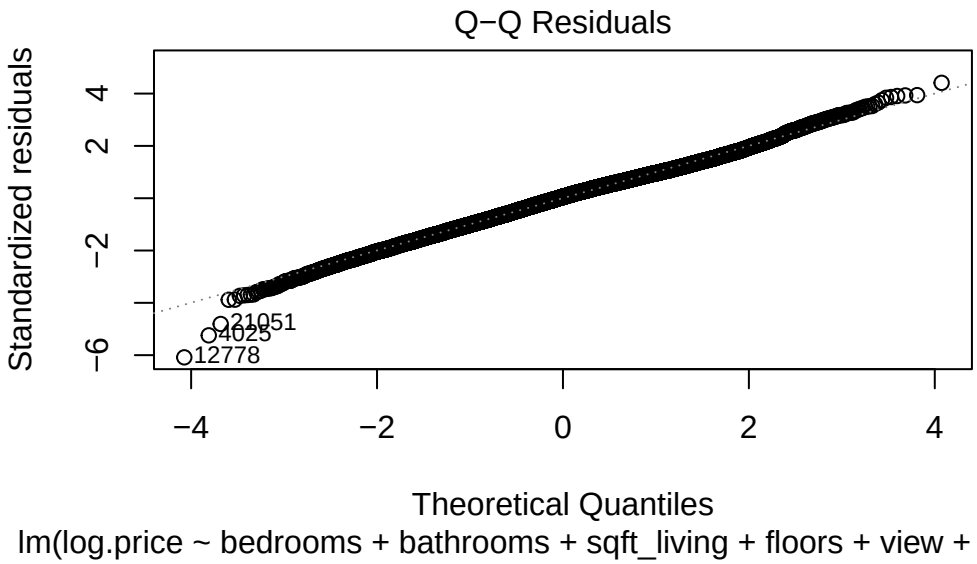
```
Call:
lm(formula = log.price ~ bedrooms + bathrooms + sqft_living +
    floors + view + condition + grade + yr_built, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-1.90374 -0.21157  0.01624  0.21288  1.38880

Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.151e+01  1.884e-01 114.155  < 2e-16 ***
bedrooms    -2.366e-02  2.937e-03  -8.056 8.28e-16 ***
bathrooms    8.500e-02  4.956e-03  17.152  < 2e-16 ***
sqft_living  1.664e-04  4.701e-06  35.402  < 2e-16 ***
floors       8.569e-02  4.968e-03  17.246  < 2e-16 ***
view         6.740e-02  2.994e-03  22.510  < 2e-16 ***
condition    4.226e-02  3.580e-03  11.803  < 2e-16 ***
grade        2.218e-01  3.108e-03  71.359  < 2e-16 ***
yr_built    -5.526e-03  9.699e-05 -56.980  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3149 on 21604 degrees of freedom
Multiple R-squared:  0.6426,    Adjusted R-squared:  0.6425
F-statistic:  4856 on 8 and 21604 DF,  p-value: < 2.2e-16
```

```r
# Same plots for model.log
plot(model.log, which = 1)
```



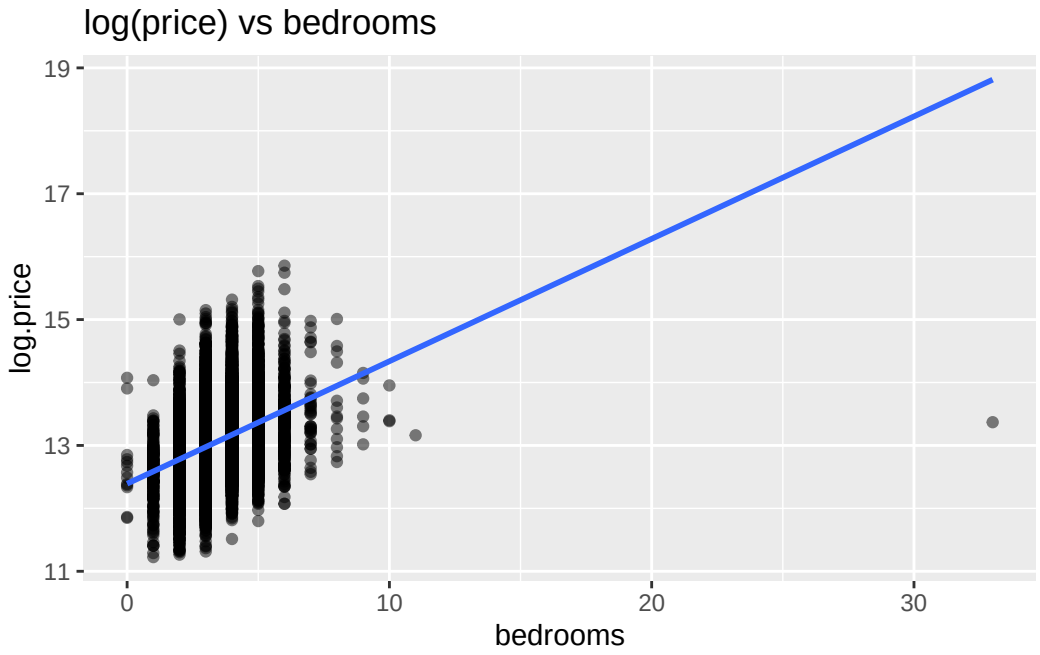

```r
plot(model.log, which = 2)
```

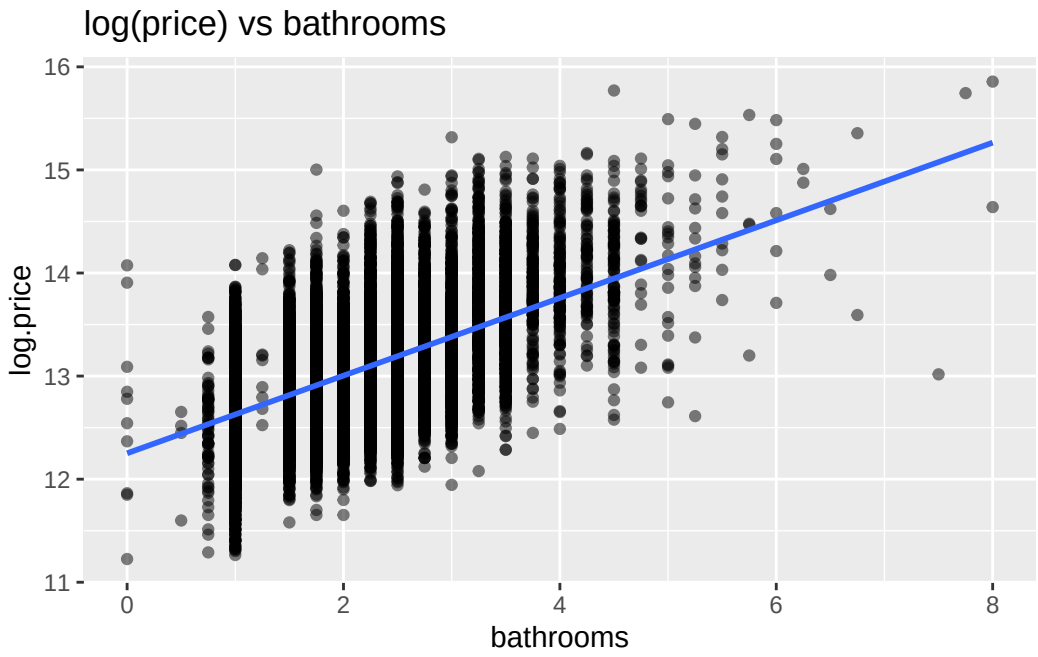


- The original `price` distribution is heavily skewed and far from normal. This is evident in both the histogram and the Q-Q plot. The long right tail indicates that a small number of houses have very high prices, which makes sense in a real estate scenario but it is problematic for linear regression assumptions.
- The log-transformed distribution is much closer to a normal distribution. The histogram is more symmetric, and the Q-Q plot aligns better with the 45-degree line. This suggests that using `log(price)` as the response variable of the linear model may lead to better model performance and more reliable results.
- The R-squared value for the log-transformed model is a bit higher than that of the original model. This suggests that the log-transformed model explains a slightly larger proportion of the variance in the response variable. The improvement is small however. Regarding the covariates, there is no loss in significance with the log transformation.
- The Residuals vs. Fitted plot for the log-transformed model has less heteroscedasticity. The residuals are more evenly spread around the line. The Q-Q plot for the log-transformed model is much closer to a straight line, showing that the residuals are more normally distributed in comparison to the original model.
- The log-transformed model is more adequate because of its slightly better R-squared value, similar significance of covariates, and large improvements in the residual analysis.

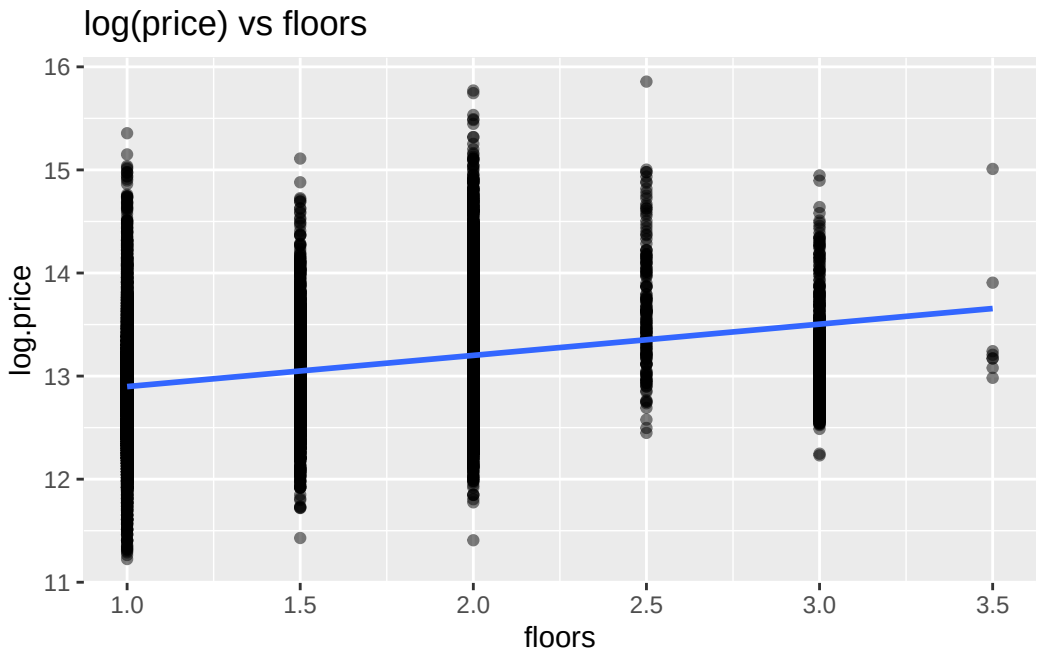
Exercise (c)

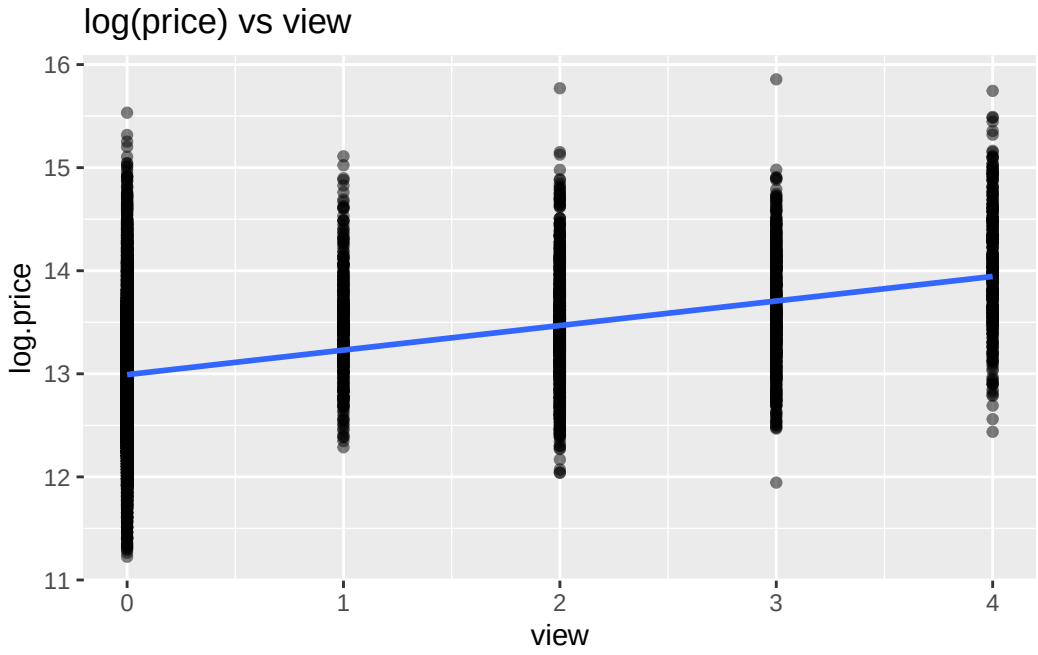
```
# Plotting each covariate against log(price)
for (var in c("bedrooms",
              "bathrooms",
              "sqft_living",
              "floors",
              "view",
              "condition",
              "grade",
              "yr_built")) {
  print(ggplot(data, aes(x = .data[[var]], y = log.price)) +
    geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) +
    ggtitle(paste("log(price) vs", var)))
}
```

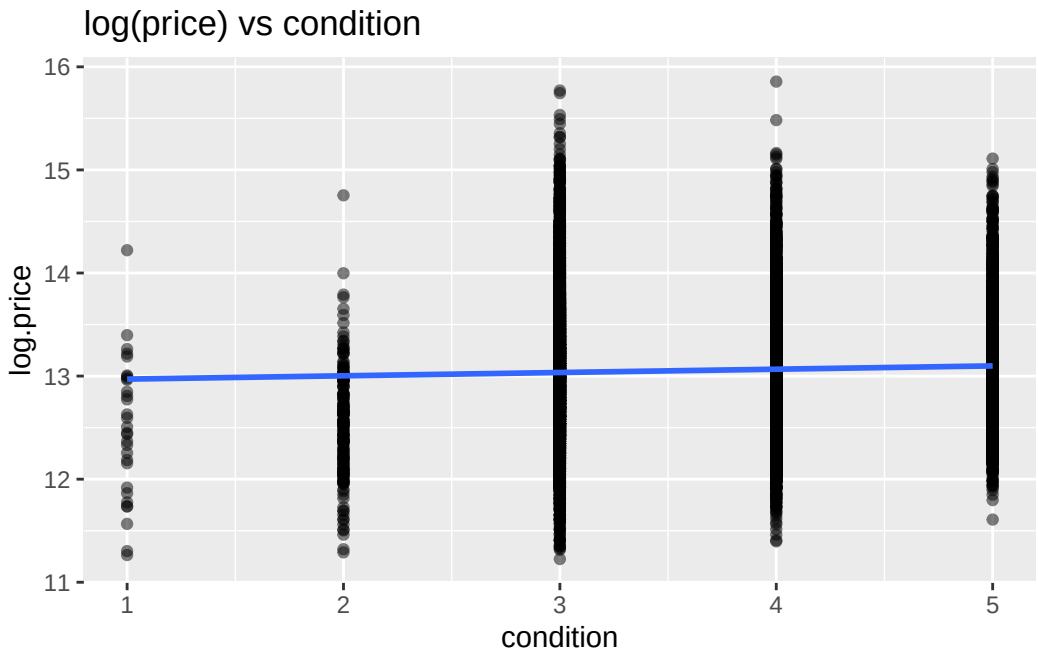


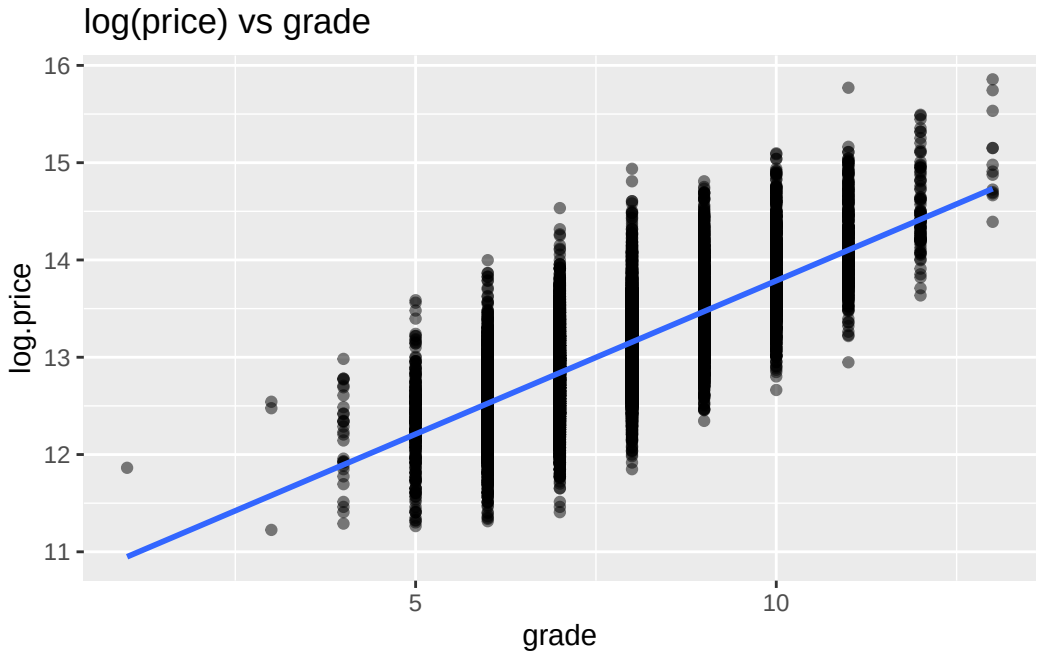

log(price) vs bathrooms
log.price
bathrooms
16
15
14
13
12
11
0
2
4
6
8

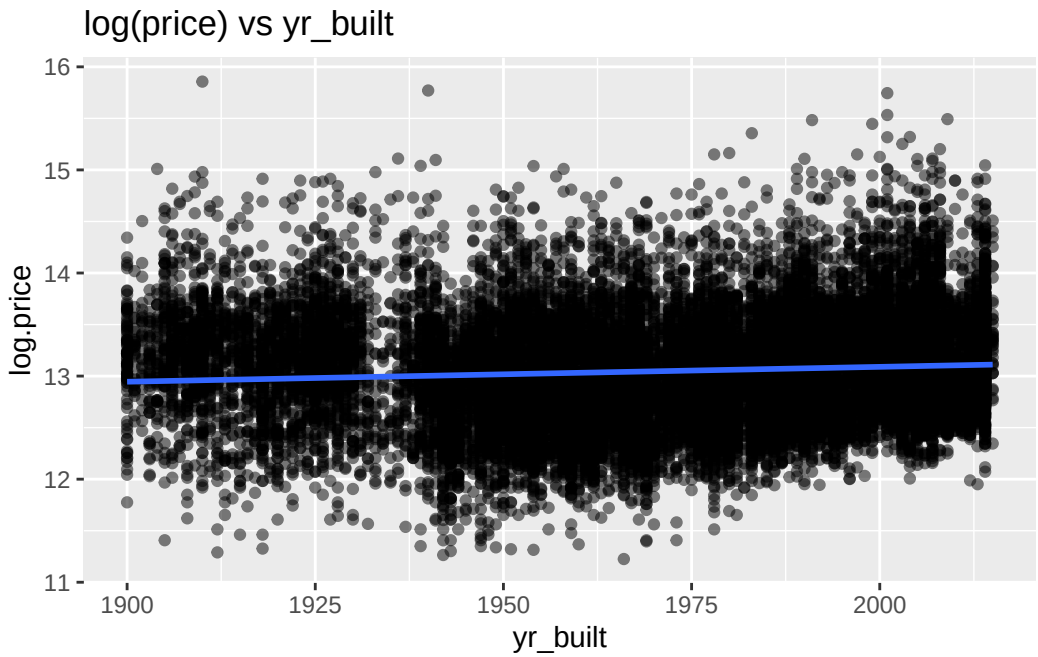

log(price) vs sqft_living
log.price
sqft_living
16
14
12
0
5000
10000


log(price) vs floors
log.price
floors
16
15
14
13
12
11
1.0
1.5
2.0
2.5
3.0
3.5


log(price) vs view
log.price
view
16
15
14
13
12
11
0
1
2
3
4


log(price) vs condition
log.price
16
15
14
13
12
11
1
2
3
4
5
condition


log(price) vs grade
log.price
16
15
14
13
12
11
5
10
grade




```r
# Adding squared terms to data
data$yr_built2 <- data$yr_built^2
data$sqft_living2 <- data$sqft_living^2

# Fit polynomial model
model.log.poly <- lm(log.price ~ bedrooms +
                       bathrooms +
                       sqft_living +
                       sqft_living2 +
                       floors +
                       view +
                       condition +
                       grade +
                       yr_built +
                       yr_built2,
                     data = data)

summary(model.log.poly)
```

```
Call:
lm(formula = log.price ~ bedrooms + bathrooms + sqft_living +
```

```
    sqft_living2 + floors + view + condition + grade + yr_built +
    yr_built2, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2644 -0.2113  0.0138  0.2107  1.4160

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.674e+02  1.082e+01  15.470   <2e-16 ***
bedrooms     -2.978e-02  3.009e-03  -9.895   <2e-16 ***
bathrooms     7.317e-02  4.959e-03  14.754   <2e-16 ***
sqft_living   2.792e-04  8.690e-06  32.132   <2e-16 ***
sqft_living2 -1.782e-08  1.172e-09 -15.212   <2e-16 ***
floors        4.733e-02  5.613e-03   8.432   <2e-16 ***
view          7.222e-02  2.979e-03  24.247   <2e-16 ***
condition     4.640e-02  3.591e-03  12.920   <2e-16 ***
grade         2.176e-01  3.092e-03  70.355   <2e-16 ***
yr_built     -1.545e-01  1.106e-02 -13.978   <2e-16 ***
yr_built2     3.802e-05  2.823e-06  13.468   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.312 on 21602 degrees of freedom
Multiple R-squared:  0.6492,    Adjusted R-squared:  0.649
F-statistic:  3998 on 10 and 21602 DF,  p-value: < 2.2e-16
```

- Bedrooms: For each additional bedroom, the expected log of the house price decreases by approximately 0.0237. The plot vs `log(price)` shows a somewhat linear relationship, but there is a high concentration of data points at certain bedroom counts, particularly between 2 and 5 bedrooms. The linearity assumption holds somewhat well, but there are outliers.
- Bathrooms: For each additional bathroom per bedroom, the expected log of the house price increases by approximately 0.0850, so more bathrooms contribute positively to the price. The relationship with `log(price)` appears to be more linear than with bedrooms, but there is some spread. Still, the linearity assumption seems plausible here.
- Sqft Living: For each additional square foot of living space, the expected log of the house price increases by approximately 0.0001664. The relationship with `log(price)` looks quite linear, especially for lower square footages. However, as the square footage increases beyond 5,000 sqft, there is some curvature, so a quadratic term might improve the fit.

- Floors: Each additional floor increases the expected log of the house price by approximately 0.0857. There is a slight linear trend with `log(price)`, but the data is heavily concentrated around 1 to 2 floors. Because it is a categorical variable, the linear relationship is less clear.
- View: If the house has been viewed, the expected log of the house price increases by approximately 0.0674. The plot vs `log(price)` shows a somewhat linear relationship, but like floors, this variable is also categorical, so the linear relationship less straightforward.
- Condition: Each one-unit increase in the condition rating increases the expected log of the house price by approximately 0.0423. The relationship with `log(price)` here is almost flat, indicating that `condition` has a very weak effect on `log(price)` in a linear sense, which also does not support the linearity assumption well.
- Grade: Each one-unit increase in the grade increases the expected log of the house price by approximately 0.2218. This is a strong positive effect compared to the other covariates. The plot vs `log(price)` shows a strong linear relationship. The linearity assumption is very plausible.
- Year Built: For each additional year, the expected log of the house price decreases by approximately 0.005526. The plot vs `log(price)` shows a slight downward trend, but the relationship seems weak and noisy. The linearity assumption is not strongly supported.
- Adding the squared terms has indeed improved the model fit. The model now explains slightly more of the variance in `log(price)` and better captures the non-linear relationships present in the data. The adjusted R-squared has increased from 0.6425 to 0.6490. The squared terms are significant so their inclusion is justified.

Exercise (d)

```
set.seed(1122)

# Set training and test set
train.indices <- sample(1:nrow(data), 10806)
train.set <- data[train.indices, ]
test.set <- data[-train.indices, ]

# Model from (b) on training set
model.b <- lm(log.price ~ bedrooms +
                bathrooms +
                sqft_living +
                floors +
                view +
```

```
                condition +
                grade +
                yr_built,
              data = train.set)

# Model from (c) on training set
model.c <- lm(log.price ~ bedrooms +
                bathrooms +
                sqft_living +
                sqft_living2 +
                floors +
                view +
                condition +
                grade +
                yr_built +
                yr_built2,
              data = train.set)

# Predictions
pred.b <- predict(model.b, newdata = test.set)
pred.c <- predict(model.c, newdata = test.set)

# MSE
mse.b <- mean((pred.b - test.set$log.price)^2)
mse.c <- mean((pred.c - test.set$log.price)^2)

paste("MSE model (b):", mse.b)
```

```
[1] "MSE model (b): 0.0977362287598035"
```

```
paste("MSE model (c):", mse.c)
```

```
[1] "MSE model (c): 0.0965438867986728"
```

```
# New model with an interaction term
# between grade and sqft_living
# & polynomial for bathroom
model.extended <- lm(log.price ~ bedrooms +
                       bathrooms +
                       I(bathrooms^2) +
```

```
                          sqft_living +
                          sqft_living2 +
                          floors +
                          view +
                          condition +
                          grade +
                          yr_built +
                          yr_built2 +
                          grade:sqft_living,
                        data = train.set)

pred.extended <- predict(model.extended, newdata = test.set)
mse.extended <- mean((pred.extended - test.set$log.price)^2)
paste("MSE extended model:", mse.extended)
```

```
[1] "MSE extended model: 0.0962785211652264"
```

Model (c) clearly performs better than model (b). The extended model provides the best fit among the three models, but could not beat the prediction error of 0.09557445.