Homework 4

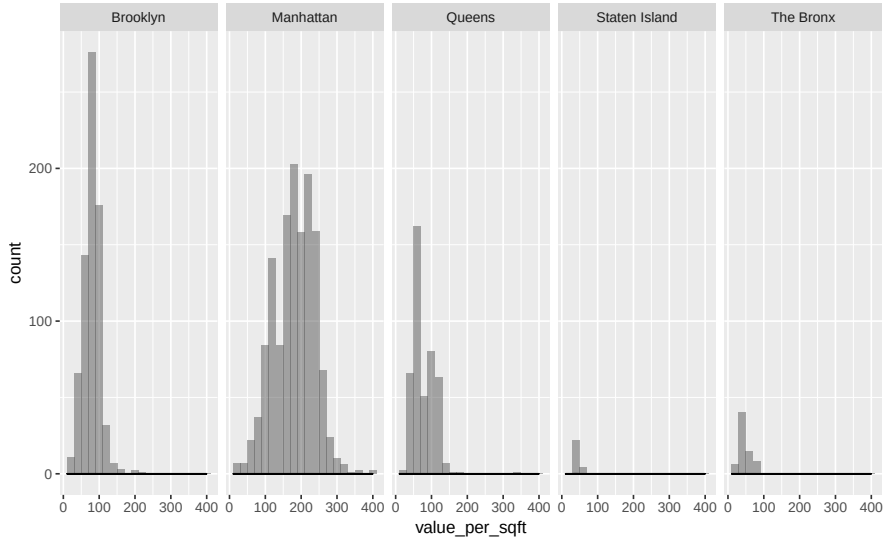
Riley Payung
David Bartling
Quan Tran
Fahd Saleem

2019-11-18

Question 1

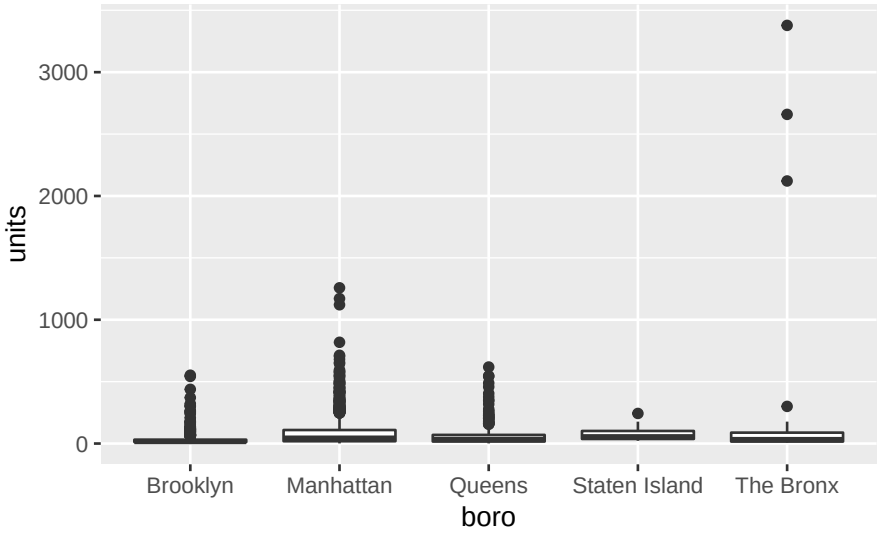
Histogram:

```
ggplot(data = housing_train) +
  geom_histogram(mapping = aes(x = value_per_sqft), position = "identity",binwidth = 20, alpha
  geom_density(mapping = aes(x = value_per_sqft), alpha = 0.5) +
  facet_grid(. ~ boro)
```

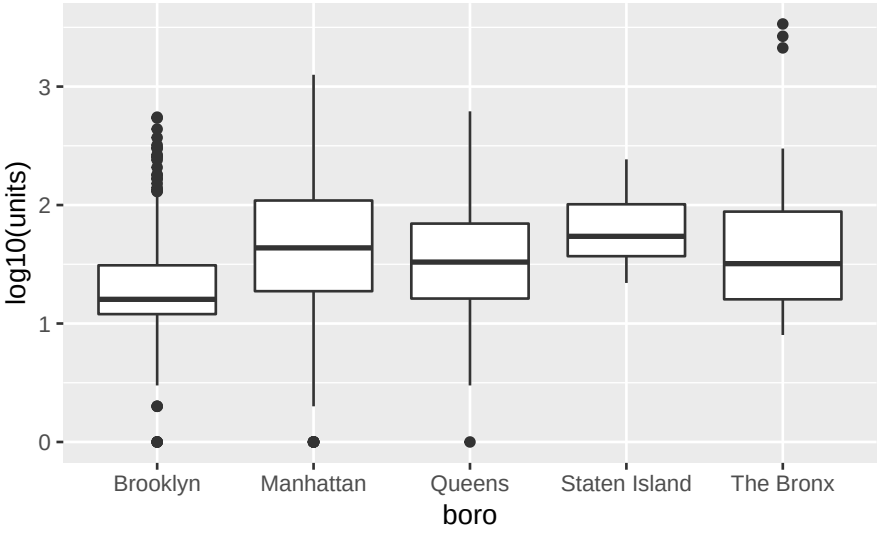


Box Plot of Units:

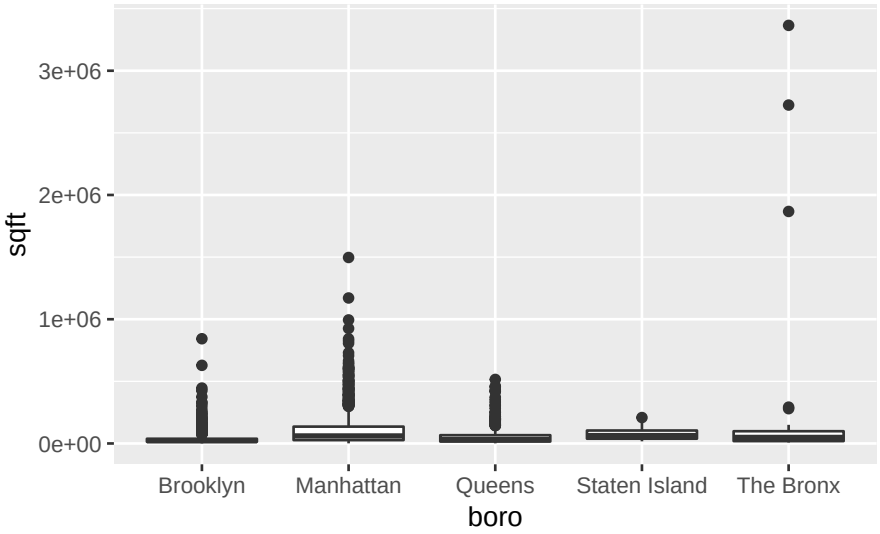
```
ggplot(data = housing_train) +
  geom_boxplot(
    mapping = aes(x = boro, y = units)
  )
```



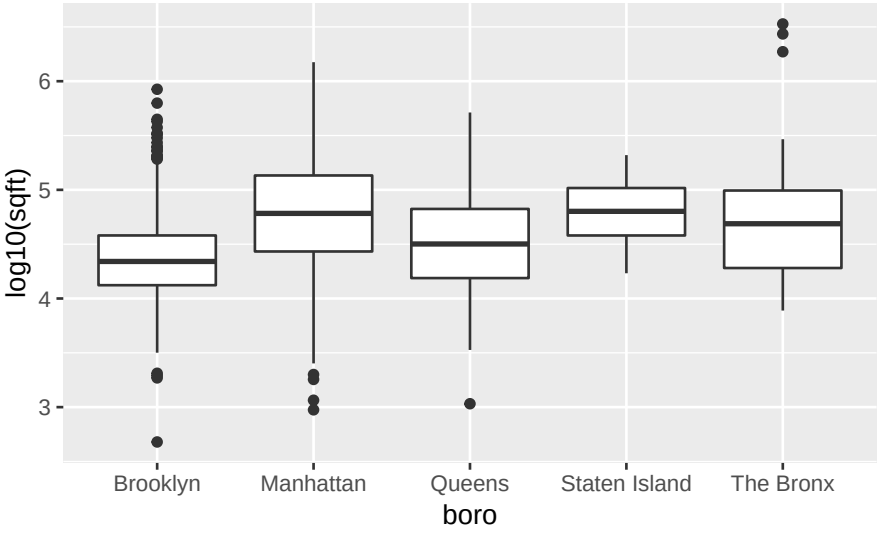

```
ggplot(data = housing_train) +
  geom_boxplot(
    mapping = aes(x = boro, log10(units))
  )
```



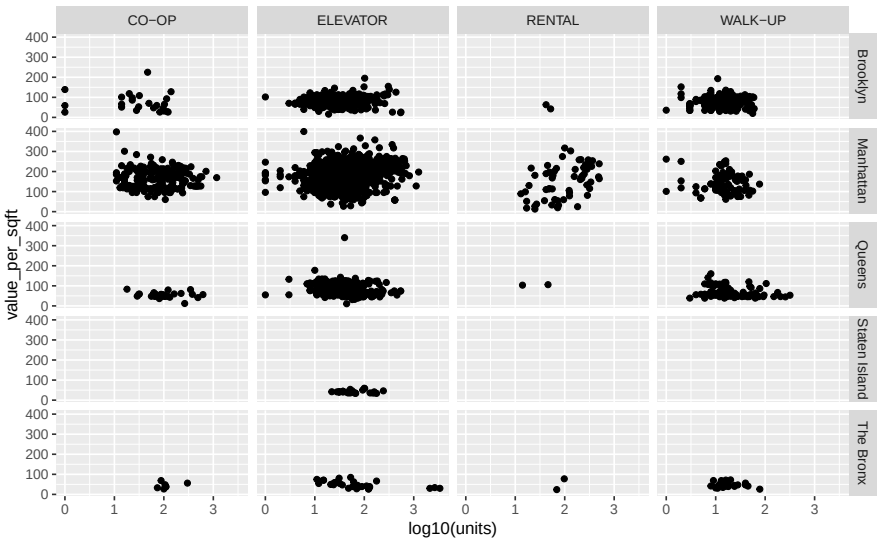

```
ggplot(data = housing_train) +
  geom_boxplot(
    mapping = aes(x = boro, y = sqft)
  )
```




```
ggplot(data = housing_train) +
  geom_boxplot(
    mapping = aes(x = boro, log10(sqft))
  )
```




```
ggplot(data = housing_train) +
  geom_point(mapping = aes(log10(units), y = value_per_sqft)) +
   facet_grid(boro ~ class)
```



Question 2

```
housing_train_filtered <- housing_train %>%
  filter(units < 2000)
```

Question 3

```
boro_model <- rep_kfold_cv(
   data = housing_train_filtered,
   k = 10,
   model = value_per_sqft ~ boro,
   cv_reps = 3)
```

```
## Warning: Unquoting language objects with `!!!` is deprecated as of rlang 0.4.0.
## Please use `!!` instead.
##
##   # Bad:
##   dplyr::select(data, !!!enquo(x))
##
##   # Good:
##   dplyr::select(data, !!enquo(x))    # Unquote single quosure
##   dplyr::select(data, !!!enquos(x))  # Splice list of quosures
##
## This warning is displayed once per session.
```

```
class_model <- rep_kfold_cv(
   data = housing_train_filtered,
   k = 10,
   model = value_per_sqft ~ class,
   cv_reps = 3)
units_model <- rep_kfold_cv(
   data = housing_train_filtered,
```

```r
   k = 10,
   model = value_per_sqft ~ units,
   cv_reps = 3)
sqft_model <- rep_kfold_cv(
   data = housing_train_filtered,
   k = 10,
   model = value_per_sqft ~ sqft,
   cv_reps = 3)
year_model <- rep_kfold_cv(
   data = housing_train_filtered,
   k = 10,
   model = value_per_sqft ~ year_built,
   cv_reps = 3)
```

The model that performed the best so far was the boro model, which had the highest r^2 value of 0.5803 and the lowest MSE of 1963.717.

Question 4

Build and cross-validate at least 3 multivariate models that predict `value_per_sqft`, using the k-fold cross-validation parameters `k = 10` and `cv_reps = 3`.

Which of your models performs the best? Is it markedly better than your best model in the last question?

```r
model1<- rep_kfold_cv(
  data = housing_train_filtered,
  k = 10,
  model = value_per_sqft ~ boro + units,
  cv_reps = 3)
  head(units_model)
```

r_squared	mse	adjusted_mse
0.0514973	4431.205	4430.739

```r
model2 <- rep_kfold_cv(
  data = housing_train_filtered,
  k = 10,
  model = value_per_sqft ~ boro + class + sqft,
  cv_reps = 3)
  head(sqft_model)
```

```

<div class="kable-table">

 r_squared        mse   adjusted_mse
----------  ---------  -------------
 0.0910938   4246.178       4245.509

</div>


```r
model3 <- rep_kfold_cv(
  data = housing_train_filtered,
  k = 10,
  model = value_per_sqft ~ boro + units + sqft + class + year_built,
  cv_reps = 3)
head(year_model)
```

<div class="kable-table">

 r_squared   mse   adjusted_mse
---------- ---- -------------
        NA   NA   NA

</div>

model 3 is the model that performs the best since the r_squared value is closer to one, meaning that the model is doing a good job at getting the variability of the response. However, when compared to model 2 the r_squared value is barley better but regardless is still slightly higher.

Question 5

Now that you've selected your model, train it on the full dataset from question 2:

```r
final_model <- lm(value_per_sqft ~ boro + units, data = housing_train_filtered)

final_model2 <- lm(value_per_sqft ~ boro + class + sqft, data = housing_train_filtered)

final_model3 <- lm(value_per_sqft ~ boro + units + sqft + class + year_built, data = housing_tr
```

where `model_formula` is the best performing model from the previous question. Use `final_model` to calculate the mean-square error for predictions on the *testing* dataset:

```r
final_model %>%
  mse(housing_test)
```

```
## [1] 2256.422
```
```

```
final_model2 %>%
  mse(housing_test)
```

```
## [1] 2159.989
```

```
final_model3 %>%
  mse(housing_test)
```

```
## [1] 1711.361
```

This score is useful because it is absolute and allows you to compare how well your model perfo
Can you do better than a MSE score of 2000?

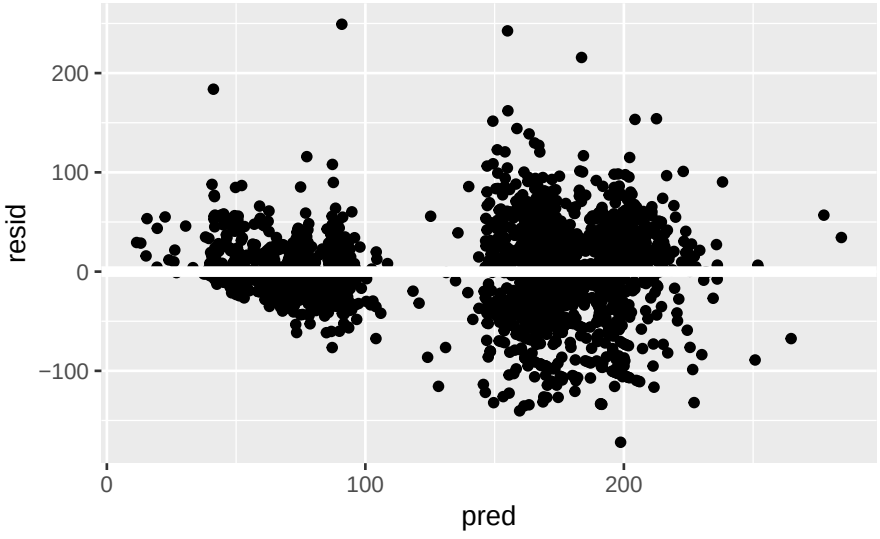
- Yes, you can do better than a mean square error of 2000, as shown through using multiple vari

Question 6

```
final_model3_df <- housing_train_filtered %>%
  add_predictions(final_model3) %>%
  add_residuals(final_model3)
```

```
# Residual vs predicted plot
ggplot(data = final_model3_df) +
geom_point(mapping = aes(x = pred, y = resid)) +
geom_ref_line(h = 0)
```

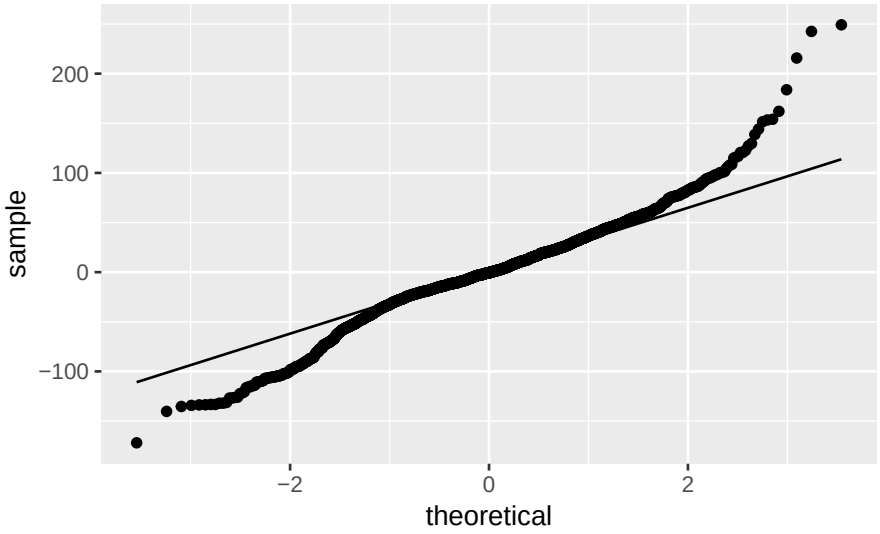
```
## Warning: Removed 96 rows containing missing values (geom_point).
```




```
# qq plot
ggplot(data = final_model3_df) +
geom_qq(mapping = aes(sample = resid)) +
  geom_qq_line(mapping = aes(sample = resid))
```

```
## Warning: Removed 96 rows containing non-finite values (stat_qq).
```

```
## Warning: Removed 96 rows containing non-finite values (stat_qq_line).
```



The spread of the residuals is quite large with most of the points surrounding 200 and 50. The shape is relatively normal because our residuals follow a linear sample with some outliers at the beginning and the end of the qq-line.