Canadian Rental Price Prediction

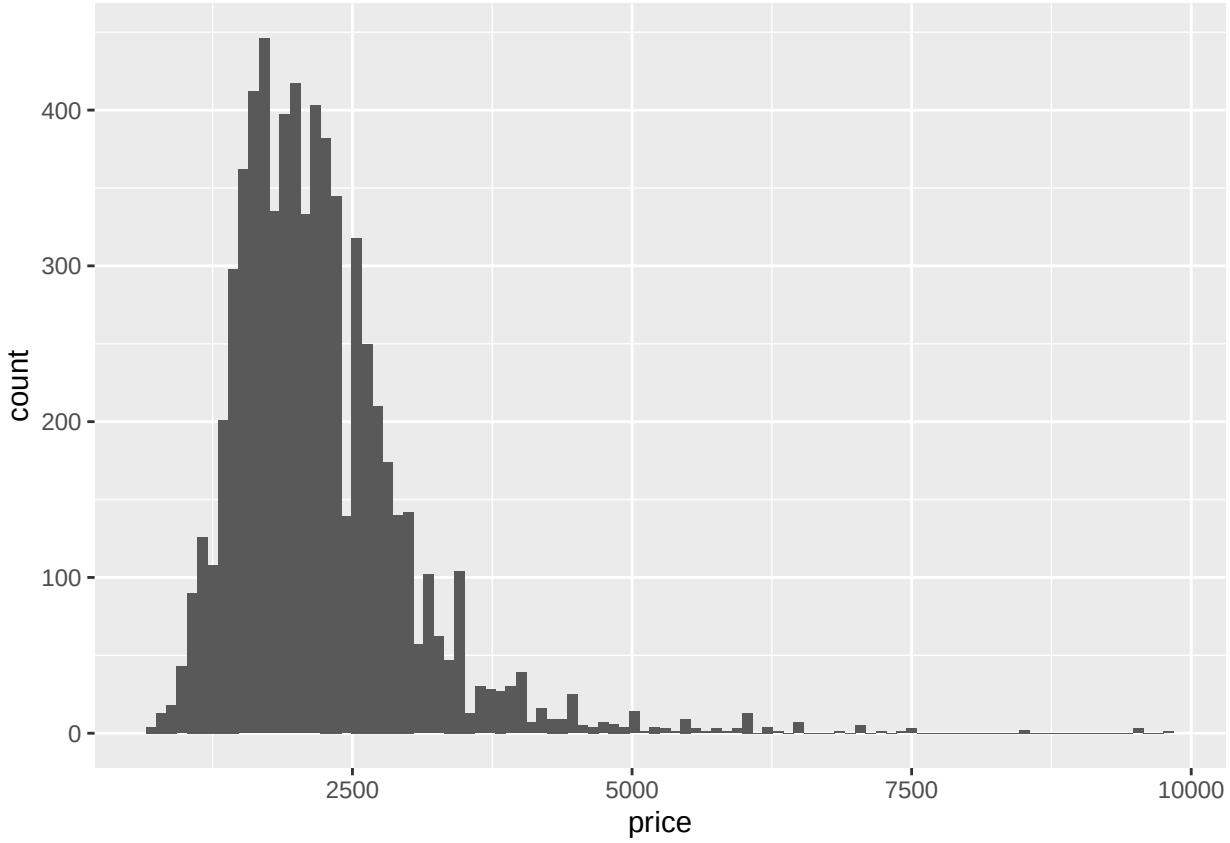
Sadik Gurung

Load and clean

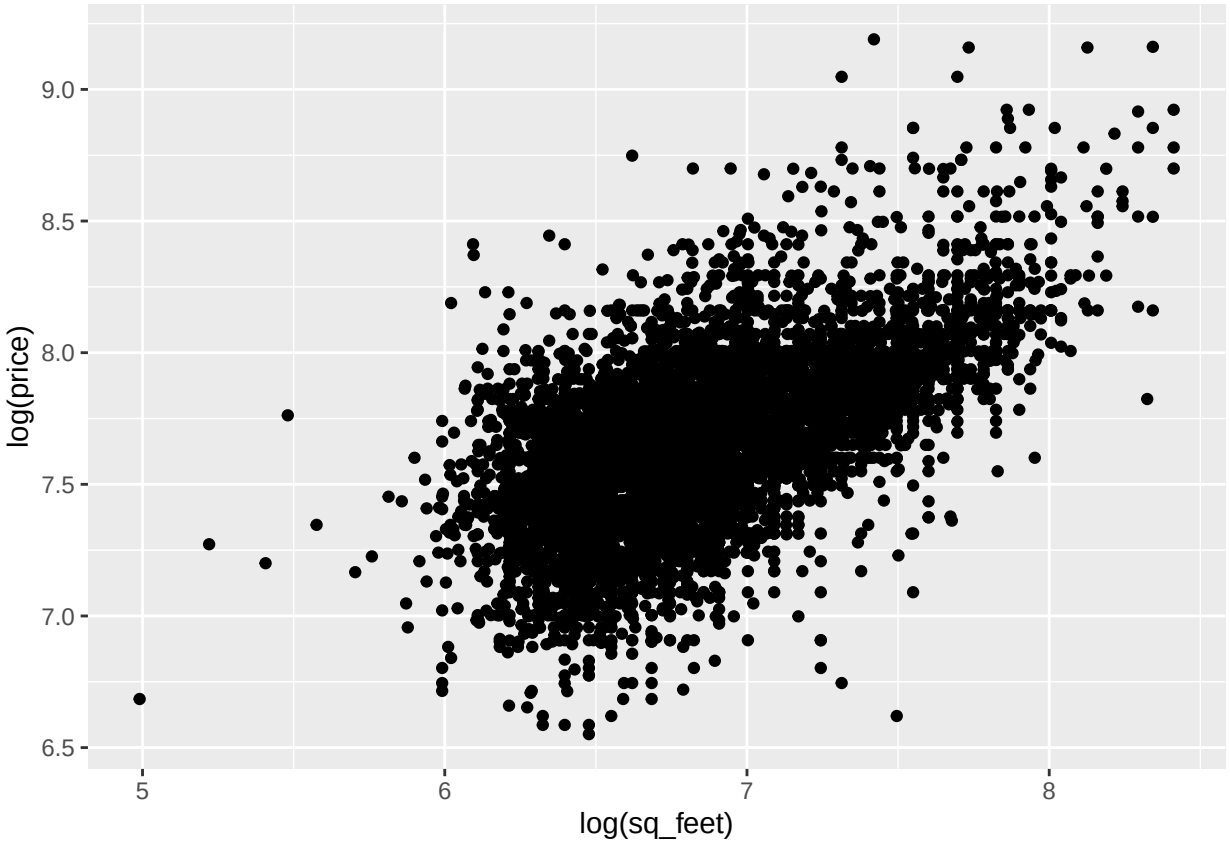
```
rent <- read.csv("rentfaster.csv") %>%
  distinct(rentfaster_id, .keep_all = TRUE) %>%
  mutate(
    bed_clean = beds %>%
      str_replace(regex("studio", ignore_case = TRUE), "0") %>%
      str_replace(regex("none", ignore_case= TRUE), "0"),
    beds = parse_number(bed_clean),

    sq_clean = if_else(str_detect(sq_feet, "\\d"),sq_feet, NA_character_) %>%
      str_replace_all(fixed("+/-"), "") %>%
      str_replace_all(regex("approx.", ignore_case = TRUE), ""),

    sq_feet = parse_number(sq_clean),
    baths_clean = baths %>%
      str_replace(regex("none", ignore_case = TRUE), "0"),

    baths = parse_number(baths_clean),

    dogs = factor(dogs),
    cats = factor(cats)
  ) %>%
  dplyr::select(-rentfaster_id,-address,-link,-availability_date,-lease_term,
                -bed_clean,-sq_clean,-baths_clean,)


rent<- rent %>%
  filter(!type %in% c("Storage", "Parking Spot", "Office Space", "Room For Rent", "Vacation Home"),
         beds >0,
         sq_feet >0,
         price > 0
  ) %>%
  filter(
    between(price,quantile(price,0.001, na.rm = TRUE), quantile(price,0.999, na.rm=TRUE)),
    between(sq_feet,quantile(sq_feet,0.001, na.rm = TRUE), quantile(sq_feet,0.999, na.rm=TRUE))
  )
#reference levels
rent$type <- relevel(factor(rent$type), ref = "House")
rent$province <- relevel(factor(rent$province), ref = "Alberta")
```

Exploratory data analysis

```
#Exploring data
ggplot(data = rent,aes(price))+
  geom_histogram(bins = 100)
```

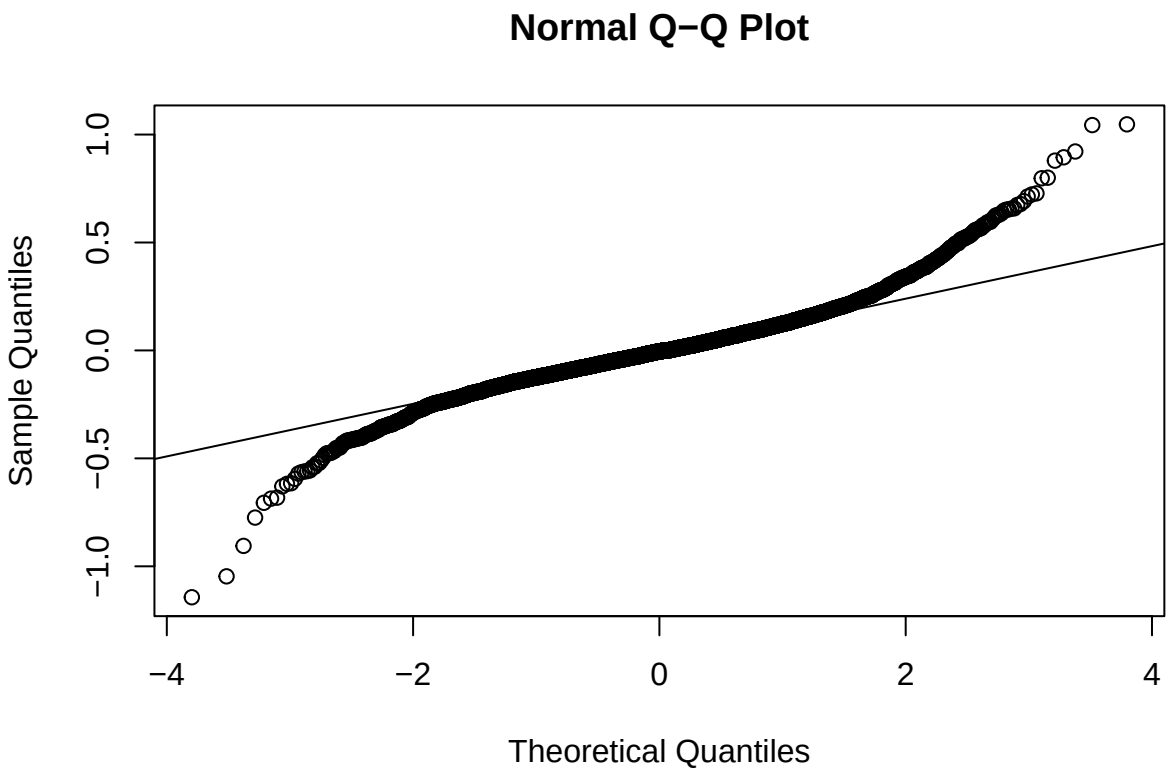



```
ggplot(data = rent, aes(log(sq_feet), log(price)))+
geom_point()
```

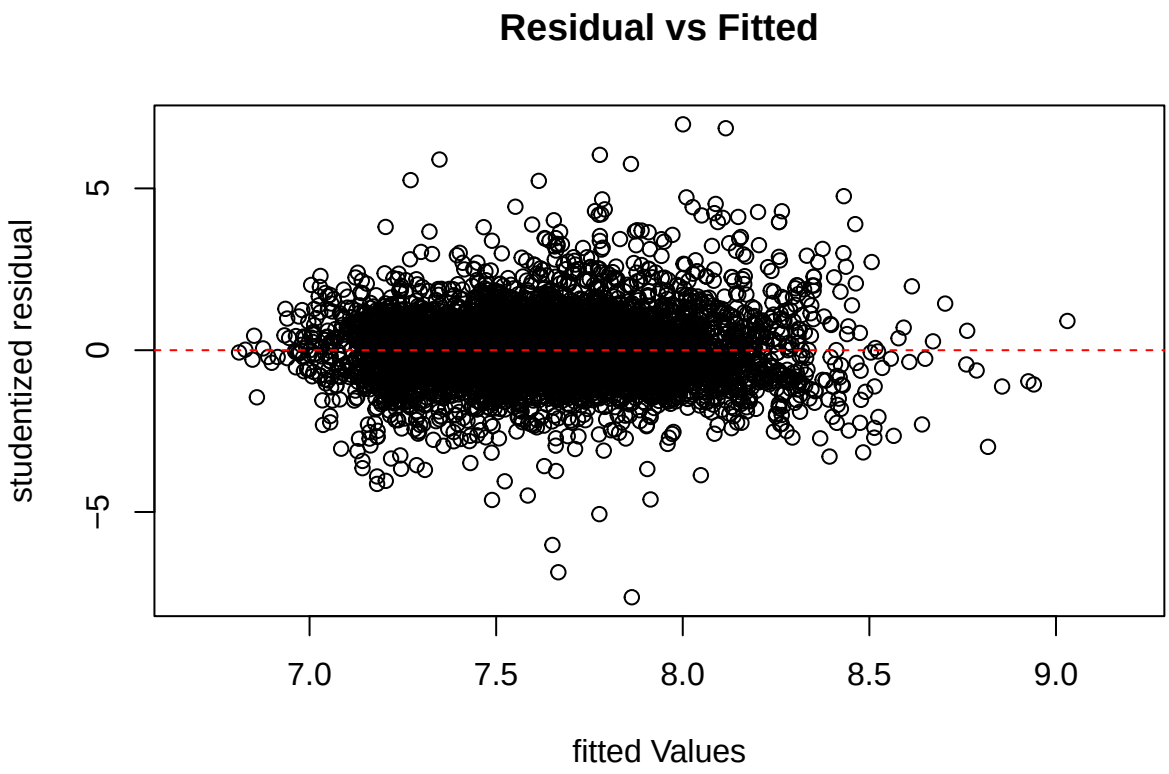


Baseline model and diagnostics

```
#Baseline log linear model
basemodel = lm(log(price) ~log(sq_feet) + beds + baths + type + province +
                 furnishing + smoking + cats + dogs + city + (longitude*latitude),
               data = rent)
#Check if normal
qqnorm(resid(basemodel))
qqline(resid(basemodel))
```




```r
#Check residual vs fitted
res = rstudent(basemodel)
plot(fitted(basemodel), res,
     xlab = "fitted Values",
     ylab = "studentized residual",
     main = "Residual vs Fitted"
)
abline(h=0, lty = 2, col = "red" )
```



Train and Test set

```
#train/test set

set.seed(1)
cities <- unique(rent$city)
test_cities <- sample(cities, size = round(0.2 * length(cities)))

test_set <- filter(rent, city %in% test_cities)
train_set <- filter(rent, !(city %in% test_cities))
```

Models

```
# Model 1: Gamma GLM (log link)
model1<- glm(price ~log(sq_feet) + beds + baths + type + province+
               furnishing + smoking + cats + dogs + latitude:longitude,
             family = Gamma(link="log"),
             data = train_set)

# Model 2: GAM on log(price)
model2 <-mgcv::gam(log(price) ~ s(log(sq_feet), k =30) + beds + baths + type + furnishing +
                     smoking + cats + dogs + province +
```

```
                     s(latitude, longitude, bs="tp", k = 150),method = "REML",
                    data=train_set)
```

Prediction and evaluation metrics

```
# Predictions
pred_model1 <- predict(model1, newdata = test_set, type = "response")
pred_model2 <- exp(predict(model2, newdata = test_set))

# Evaluation Metrics
mae  <- function(y, yhat){
  mean(abs(y - yhat))
  }
mape <- function(y, yhat){
  mean(abs(y - yhat)/y) * 100
}
rmse <- function(y, yhat){
  sqrt(mean((y - yhat)^2))
  }


percent_within <- function(y, yhat, p = 0.10){
  mean(abs(yhat - y) / y <= p) * 100
}

# Accuracy table
tibble(
  percent = c("±5%", "±10%", "±20%", "±30%"),
  GLM = c(percent_within(test_set$price, pred_model1, 0.05),
          percent_within(test_set$price, pred_model1, 0.10),
          percent_within(test_set$price, pred_model1, 0.20),
          percent_within(test_set$price, pred_model1, 0.30)),
  GAM = c(percent_within(test_set$price, pred_model2, 0.05),
          percent_within(test_set$price, pred_model2, 0.10),
          percent_within(test_set$price, pred_model2, 0.20),
          percent_within(test_set$price, pred_model2, 0.30))
)
```

```
## # A tibble: 4 x 3
##   percent   GLM   GAM
##   <chr>   <dbl> <dbl>
## 1 ±5%      14.3  30.1
## 2 ±10%     28.7  53.3
## 3 ±20%     57.9  82.7
## 4 ±30%     77.1  92.8
```

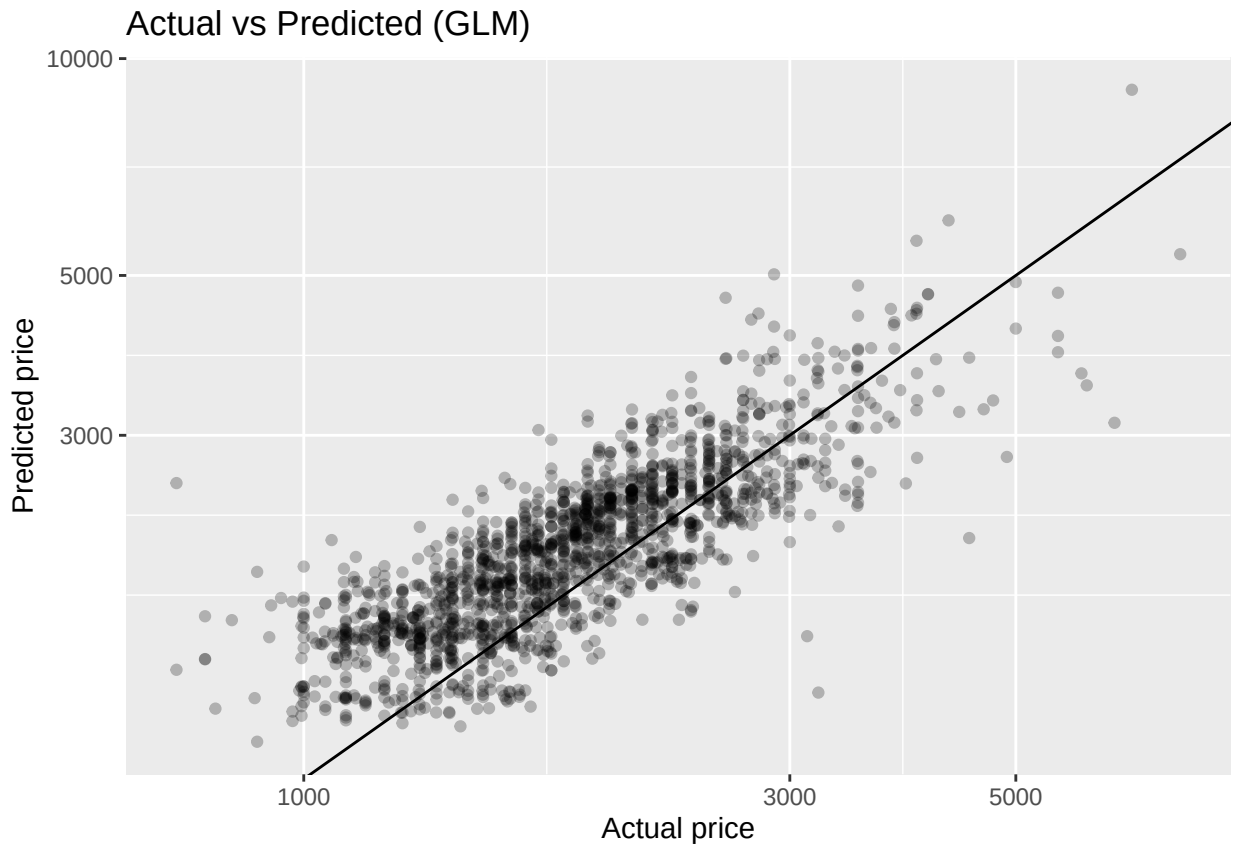
```
# Metrics table
tibble(
  model = c("Gamma GLM", "GAM"),
  MAE = c(mae(test_set$price, pred_model1), mae(test_set$price, pred_model2)),
  RMSE  = c(rmse(test_set$price, pred_model1), rmse(test_set$price, pred_model2)),
```

```
  MAPE = c(mape(test_set$price, pred_model1), mape(test_set$price, pred_model2))
)
```

```
## # A tibble: 2 x 4
##   model       MAE  RMSE  MAPE
##   <chr>     <dbl> <dbl> <dbl>
## 1 Gamma GLM  368.  473.  20.8
## 2 GAM        254.  422.  12.4
```

Actual vs Predicted plot

```
plot_test_set <- test_set %>%
  mutate(pred_model1 = pred_model1, pred_model2 = pred_model2)

#Actual vs Predicted (GLM)
plot_test_set %>%
  ggplot(aes(x = price, y = pred_model1)) +
  geom_point(alpha = 0.25) +
  scale_x_log10() + scale_y_log10() +
  geom_abline(slope = 1, intercept = 0) +
  labs(title = "Actual vs Predicted (GLM)", x = "Actual price", y = "Predicted price")
```



```r
#Actual vs Predicted (GAM)
plot_test_set %>%
  ggplot(aes(x = price, y = pred_model2)) +
  geom_point(alpha = 0.25) +
  scale_x_log10() + scale_y_log10() +
  geom_abline(slope = 1, intercept = 0) +
  labs(title = "Actual vs Predicted (GAM)", x = "Actual price", y = "Predicted price")
```