STOR 455 Homework #5

40 points - Due Wednesday 3/20 at 5:00pm

Individual Portion - Jake Mareno

Directions: For parts 7 and 10 you should work together, but these parts must be **submitted individually** by each group member. For parts 8 and 9, you must have only **one submission per group**. There will be separate places on Gradescope to submit the individual vs group work.

Situation: Can we predict the selling price of a house in Ames, Iowa based on recorded features of the house? That is your task for this assignment. Each team will get a dataset with information on forty potential predictors and the selling price (in $1,000's) for a sample of homes. The data sets for your group are AmesTrain??.csv and AmesTest??.csv (where ?? corresponds to your group number) A separate file identifies the variables in the Ames Housing data and explains some of the coding.

Part 7. Cross-validation: In some situations, a model might fit the peculiarities of a specific sample of data well, but not reflect structure that is really present in the population. A good test for how your model might work on "real" house prices can be simulated by seeing how well your fitted model does at predicting prices that were NOT in your original sample. This is why we reserved an additional 200 cases as a holdout sample in AmesTest??.csv. Use the group number and AmesTest??.csv corresponding to your group number for homework #3. Import your holdout test data and

```
library(tidyverse)

ames_train <- read_csv("AmesTrain10.csv")
ames_test <- read_csv("AmesTest10.csv")
```

- Compute the predicted Price for each of the cases in the holdout test sample, using your model resulting from the initial fit and residual analysis in parts 1 through 3 of Homework #3.

```
basic_mod <- lm(formula = Price ~ LotFrontage + LotArea + Quality + Condition +
    YearBuilt + YearRemodel + BasementFinSF + BasementSF + GroundSF +
    FullBath + Bedroom + Fireplaces + GarageSF +
    ScreenPorchSF, data = ames_train)

test_price <- predict(basic_mod, newdata = ames_test)
head(test_price)
```

```
##        1        2        3        4        5        6
## 244.9227 141.1920 191.1885 188.7825 181.2795 144.6866
```

- Compute the residuals for the 200 holdout cases.

```
test_resid <- test_price - ames_test$Price
head(test_resid)
```

```
##        1        2        3        4        5        6
## 11.92267 -7.30803 14.70350 16.78255  9.27953 15.68656
```

- Compute the mean and standard deviation of these residuals. Are they close to what you expect from the training model?

```
mean(test_resid)
```

```
## [1] 1.523248
```

```
mean(ames_test$Price)
```

```
## [1] 174.1175
```

```
sd(test_resid)
```

```
## [1] 28.62391
```

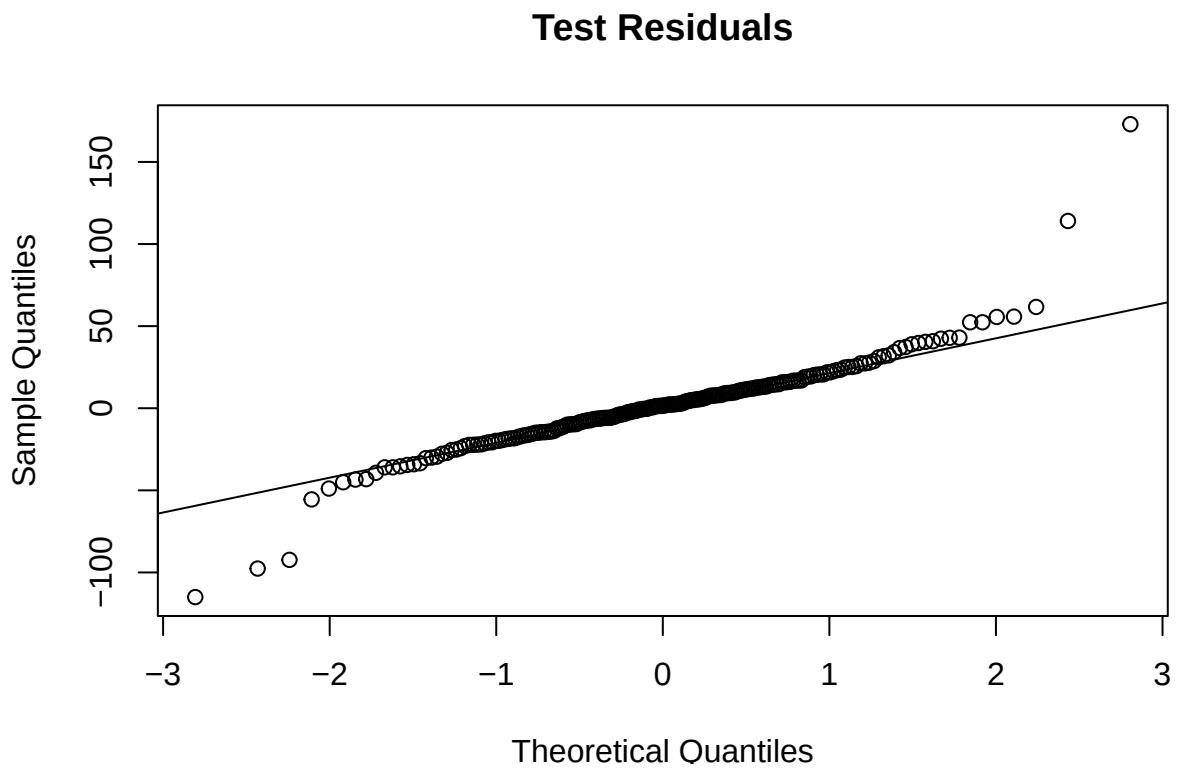
```
sd(basic_mod$residuals)
```

```
## [1] 29.12641
```

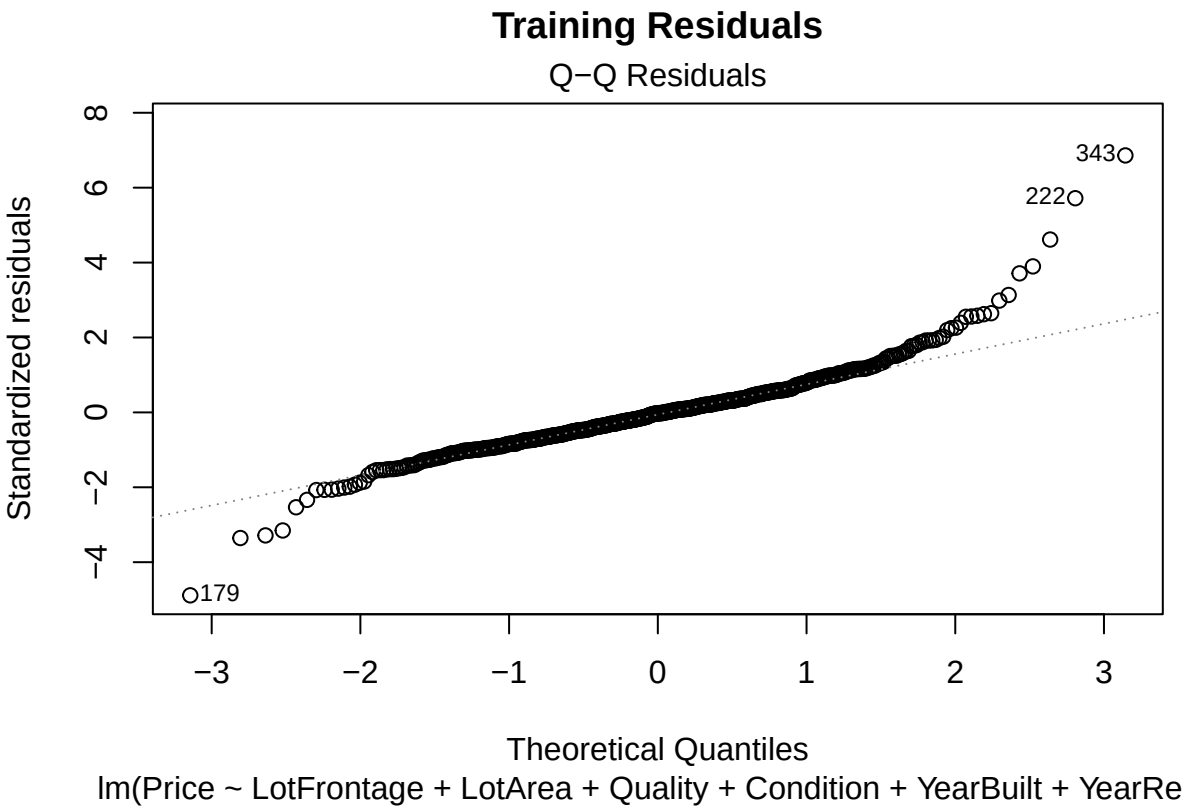
Our mean is only 1.5, which when compared to the mean value of ~174 for `Price` is only a slight deviation from 0, so this indicates the model was a good fit/translated well to the holdout. We also see that the standard deviation of the holdout residuals is only 28.6, which is actually slightly less than the training data's standard deviation of 29.1 and this closeness is again what we want to see.

- Construct a plot of the residuals to determine if they are normally distributed. Is this plot what you expect to see considering the training model?

```
qqnorm(test_resid, main = "Test Residuals")
qqline(test_resid)
```




```
plot(basic_mod, 2, main = "Training Residuals")
```



We see that the test/holdout sample and training sample both exhibit similar Q-Q plots that are fairly normal in the middle with a few outliers on either end, which is in line with our expectations.

- Are any holdout cases especially poorly predicted by the training model? If so, identify by the row number(s) in the holdout data. Why might these cases be poorly predicted?

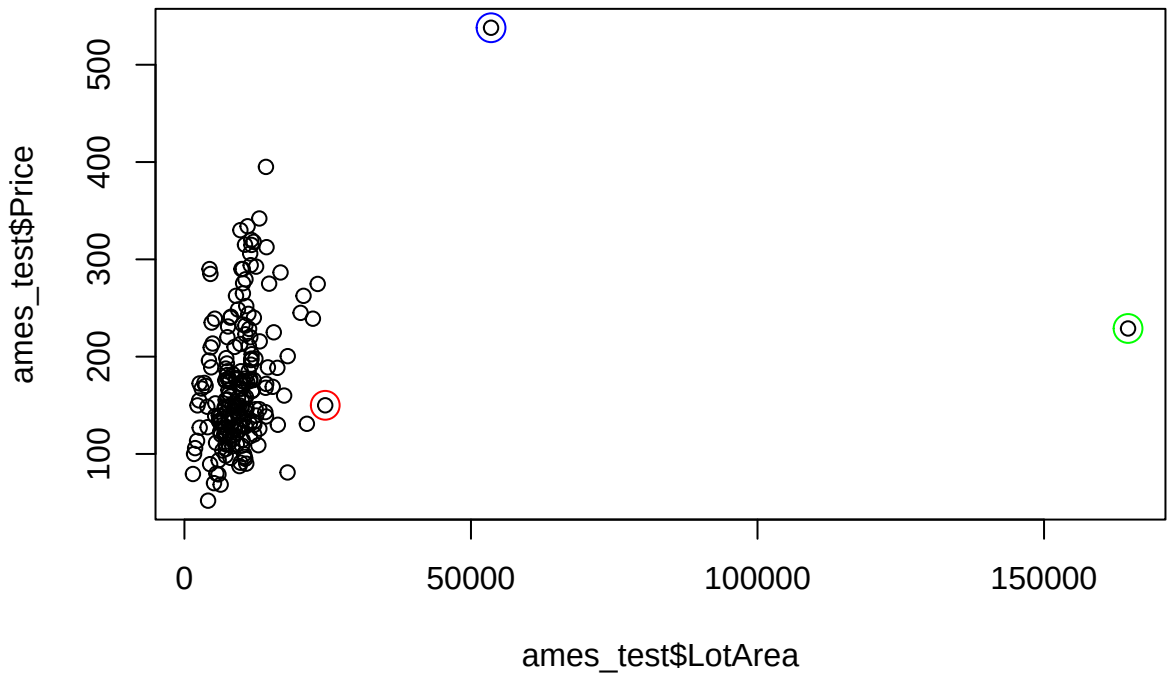
```
head(sort(abs(test_resid)/sd(test_resid), decreasing = TRUE))
```

```
##      118       57       48      175      186       52
## 6.043487 4.017783 3.984220 3.410970 3.225801 2.155098
```

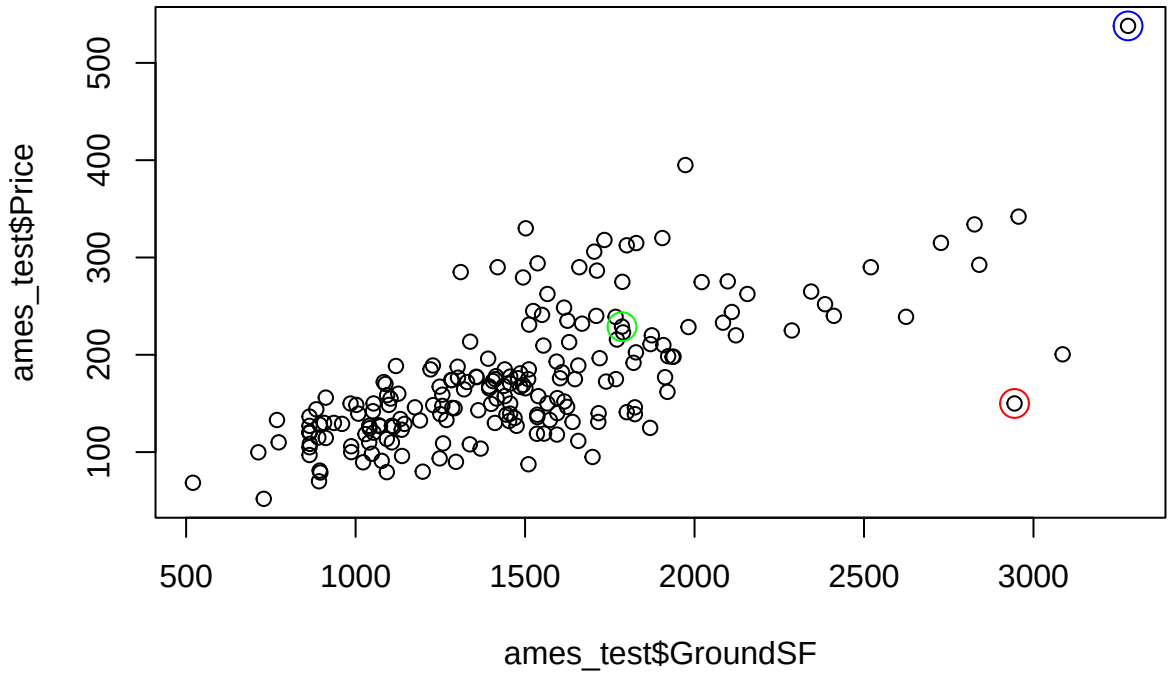
We have 5 observations (row numbers 118, 57, 48, 175, and 186) with a the magnitude of the standardized residual of over 3, which indicates these cases were poorly predicted. I chose to investigate the first 3 more closely as they have such an extreme residual. In fact, the holdout sample has less outliers and is actually closer to normal. We see that observation 118 is a moderate outlier in regards to `LotArea` and `GroundSF`, observation 57 is an extreme outlier for both features, and observation 48 is an extreme outlier for `LotArea` while being normal for `GroundSF`. These deviations from the normal distribution of these features likely accounts for most of the poor accuracy in predicting these observations.

```
# Looked at the numbers for these observations
# ames_test[c(118,57,48),]


plot(ames_test$LotArea, ames_test$Price)
points(ames_test$LotArea[118], ames_test$Price[118], col="red", cex=2)
points(ames_test$LotArea[57], ames_test$Price[57], col="blue", cex=2)
points(ames_test$LotArea[48], ames_test$Price[48], col="green", cex=2)
```




```
plot(ames_test$GroundSF, ames_test$Price)
points(ames_test$GroundSF[118], ames_test$Price[118], col="red", cex=2)
points(ames_test$GroundSF[57], ames_test$Price[57], col="blue", cex=2)
points(ames_test$GroundSF[48], ames_test$Price[48], col="green", cex=2)
```



- Compute the correlation between the predicted values and actual prices for the holdout sample. This is known as the cross-validation correlation. We don't expect the training model to do better at predicting values different from those that were used to build it (as reflected in the original R^2), but an effective model shouldn't do a lot worse at predicting the holdout values. Square the cross-validation correlation to get an R^2 value and subtract it from the original multiple R^2 of the training sample. This is known as the shrinkage. We won't have specific rules about how little the shrinkage should be, but give an opinion on whether the shrinkage looks OK to you or too large in your situation.

```
cross_cor <- cor(ames_test$Price, test_price)

cross_cor^2
```

```
## [1] 0.8319237
```

```
summary(basic_mod)$r.squared
```

```
## [1] 0.862814
```

```
summary(basic_mod)$r.squared - cross_cor^2
```

```
## [1] 0.03089035
```

We get a shrinkage of about 0.031, which is not too much of an adjustment considering our original r-squared of 0.863 and is to be expected when shifting from a model that was designed to exactly match the training set as best as possible when compared to a holdout sample with a bit of random noise.

Part 10. Final Model Again, you may choose to make some additional adjustments to your model after considering the final residual analysis. If you do so, please explain what (and why) you did and provide the summary() for your new final model.

Suppose that you are interested in a house in Ames that has the characteristics listed below. Construct a 95% confidence interval for the mean price of such houses.

A 2 story 11 room home, built in 1987 and remodeled in 1999 on a 21540 sq. ft. lot with 328 feet of road frontage. Overall quality is good (7) and condition is average (5). The quality and condition of the exterior are both good (Gd) and it has a poured concrete foundation. There is an 757 sq. foot basement that has excellent height, but is completely unfinished and has no bath facilities. Heating comes from a gas air furnace that is in excellent condition and there is central air conditioning. The house has 2432 sq. ft. of living space above ground, 1485 on the first floor and 947 on the second, with 4 bedrooms, 2 full and one half baths, and 1 fireplace. The 2 car, built-in garage has 588 sq. ft. of space and is average (TA) for both quality and construction. The only porches or decks is a 205 sq. ft. open porch in the front.

```
ames_train_adjusted <- ames_train |>
  mutate(Quality = factor(Quality),
         Condition = factor(Condition),
         age_built = 2010 - YearBuilt)

fancy_mod <- lm(log(Price) ~ Quality + GroundSF + Condition + YearBuilt +
    BasementFinSF + ExteriorQ + HouseStyle + GarageSF + Foundation +
    LotArea + LotConfig + GarageType + Bedroom + TotalRooms +
    ExteriorC + BasementHBath + I(GarageSF^2) + GarageC + HeatingQC +
    BasementSF + BasementC + BasementHt + KitchenQ + GarageQ +
    ScreenPorchSF + Fireplaces + WoodDeckSF + FullBath + HalfBath +
    SecondSF + FirstSF + BasementFin + BasementUnFinSF + BasementFBath +
    YearRemodel + Heating + EnclosedPorchSF + LotFrontage + I(age_built^2) +
    GarageCars + I(GarageCars^2) + CentralAir + OpenPorchSF +
    Bedroom:TotalRooms + LotArea:LotFrontage + GarageSF:GarageCars,
    data = ames_train_adjusted)

count(ames_test, LotConfig)
```

```
## # A tibble: 5 x 2
##   LotConfig     n
##   <chr>     <int>
## 1 Corner       42
## 2 CulDSac      17
## 3 FR2           6
```

```
## 4 FR3           1
## 5 Inside      134
```

```
summary(basic_mod)$r.squared
```

```
## [1] 0.862814
```

```
summary(fancy_mod)$r.squared
```

```
## [1] 0.9326211
```

```
house_ex <- data.frame(
  Order = 0,
  Price = 0,
  LotFrontage = 328,
  LotArea = 21540,
  LotConfig = "Inside",
  HouseStyle = "2Story",
  Quality = 7,
  Condition = 5,
  YearBuilt = 1987,
  YearRemodel = 1999,
  ExteriorQ = "Gd",
  ExteriorC = "Gd",
  Foundation = "PConc",
  BasementHt = "Ex",
  BasementC = "None",
  BasementFin = "Unf",
  BasementFinSF = 0,
  BasementUnFinSF = 757,
  BasementSF = 757,
  Heating = "GasA",
  HeatingQC = "Ex",
  CentralAir = "Y",
  FirstSF = 1485,
  SecondSF = 947,
  GroundSF = 2432,
  BasementFBath = 0,
  BasementHBath = 0,
  FullBath = 2,
  HalfBath = 1,
  Bedroom = 4,
  KitchenQ = "TA",
  TotalRooms = 11,
  Fireplaces = 1,
  GarageType = "BuiltIn",
  GarageCars = 2,
  GarageSF = 588,
  GarageQ = "TA",
  GarageC = "TA",
  WoodDeckSF = 0,
  OpenPorchSF = 205,
  EnclosedPorchSF = 0,
  ScreenPorchSF = 0
) |>
  mutate(Quality = factor(Quality),
         Condition = factor(Condition),
```

```
         age_built = 2010 - YearBuilt)

exp(predict.lm(fancy_mod, house_ex, interval="confidence", level=0.95))
```

```
##        fit      lwr      upr
## 1 314.1368 249.1536 396.0686
```

NOTE: The description did not provide info for LotConfig or KitchenQ, so I went with "Inside" config (most common) and "TA" kitchen quality (average).

Our fancy model predicts a 95% confidence interval for the mean price of this house as [249.15, 396.07] with a mean value of 314.14, where these figures represents thousands of dollars.