STOR 455 Homework #5

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

Group 10

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

```r
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.

```r
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)
```

- Compute the residuals for the 200 holdout cases.

```r
test_resid <- test_price - ames_test$Price
```

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

```r
mean(test_resid)
```

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

```r
sd(test_resid)
```

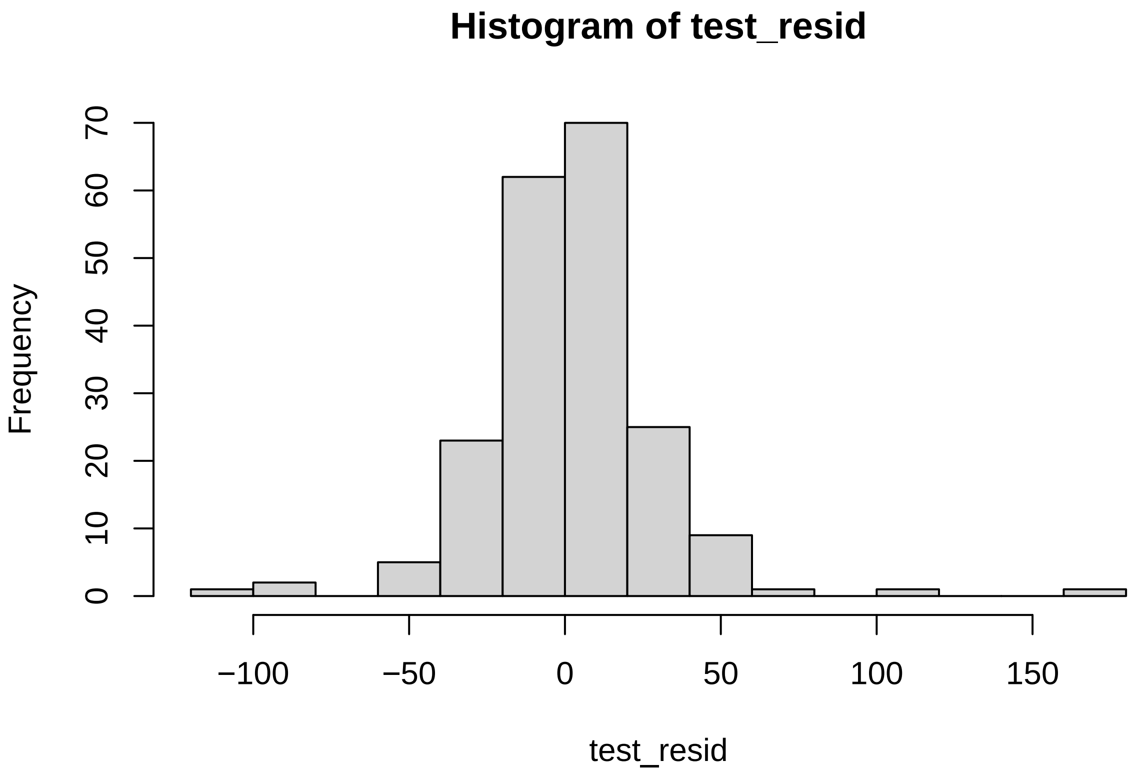
```
## [1] 28.62391
```

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

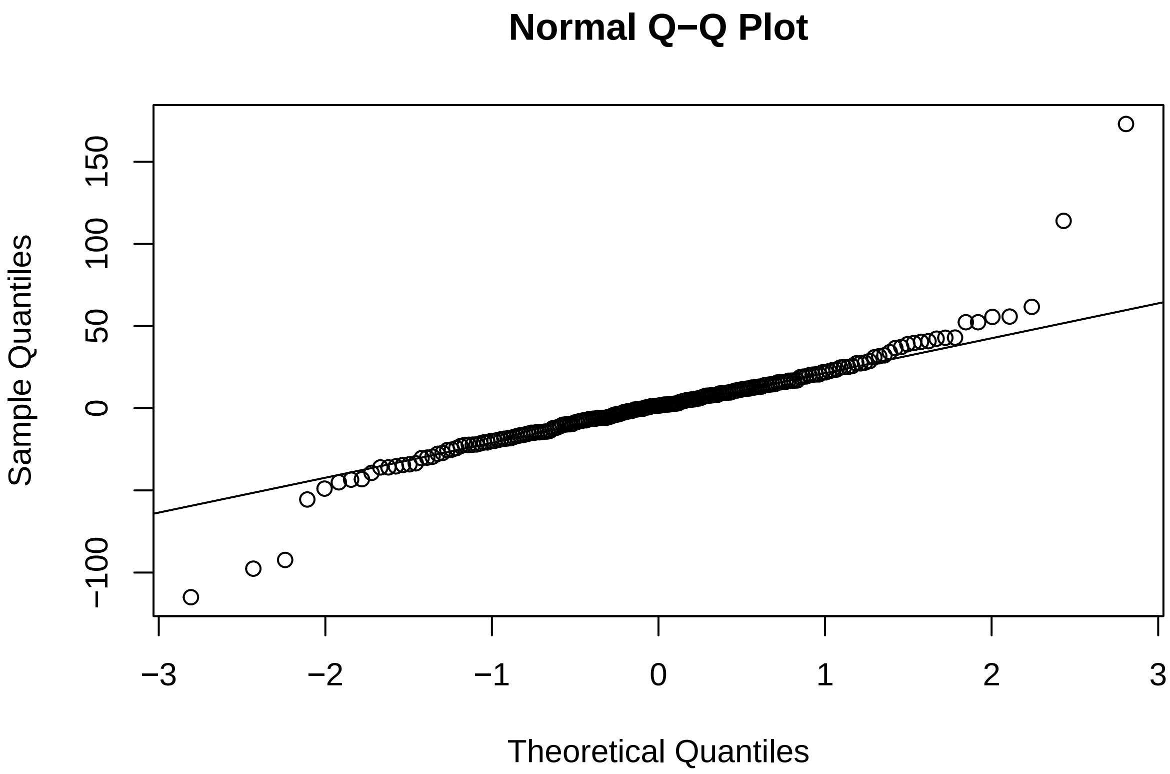
```
## [1] 29.12641
```

- 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?

```
hist(test_resid, breaks=20)
```



```
qqnorm(test_resid)
qqline(test_resid)
```



- 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), decreasing = TRUE))
```

```
##       118       57        48        175       186        52
## 172.98821 115.00464 114.04394  97.63528  92.33502  61.68733
```

- 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)

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

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

Part 8. Find a "fancy model": Again using AmesTrain??.csv, where ?? corresponds to your new group number in homework #5, to build a regression model to predict Price. In addition to the quantitative predictors, you may now consider models with

- Categorical variables - Just put these in the model and let R take care of making the indicator predictors (and picking one category to leave out). Use factor( ) to treat a numeric variable as categorical. You'll see the coefficients for each indicator when you look at the summary( ) and they will be grouped together in the ANOVA. Be careful, since adding a single categorical variable with a lot of categories might actually be adding a lot of new indicator terms.

- Transformations of predictors - You can include functions of quantitative predictors. Probably best to use the I( ) notation so you don't need to create new columns when you run the predictions for the test data.
- Transformations of the response - You might address curvature or skewness in residual plots by transforming the response prices with a function like log(Price), sqrt(Price), Priceˆ2, etc.. These should generally not need the I( ) notation to make these adjustments. IMPORTANT: If you transform Price, be sure to reverse the transformation when making final predictions!
- Combinations of variables - This might include interactions or other combinations. You do not need the I( ) notation when making an interaction using a categorical predictor (e.g. GroundSF*CentralAir).

Keep general track of the approaches you try and explain what guides your decisions as you select a new set of predictors (but again you don't need to give full details of every model you consider). Along the way you should consider some residual analysis.

Notes/Tips:

- WARNING: When using a categorical predictor with multiple categories in regsubsets( ), R will create indicators and treat them as separate predictors when deciding which to put into a model. So you might get a model with quantitative predictors like LotArea and GroundSF along with specific indicators like GarageQTA and HouseStyle1Story. This may not be very useful, since we should generally use all indicators for a categorical predictor if we include one in the model. On the other hand, when using the step( ) function, R will generally keep the multiple indicators for different categories of the same variable together as a unit.
- In some cases the indicators created for different categorical variables will have identical values. For example, if you include both GarageC and GarageQ in a model, R will produce values for each of the indicators. The indicators for GarageQNone and GarageCNone (equal to one only for houses that don't have a garage) will be identical. This may be handled differently in R depending on the procedure. regsubsets( ) may give a "warning" about variables being linearly dependent. You can still use the results, just be aware that some variables are completely dependent. lm( ) might give output with coefficients (and tests) of some predictors listed as NA. This is not a problem, R is just automatically deleting one of the redundant variables. If you are predicting for a house with no garage you might have a coefficient to use for GarageQNone but then you don't need to worry about having one for GarageCNone.
- If your residual analysis from homework #3 or an early model here suggest you might want to do a transformation for the response variable (Price), do so *before* fitting a lot more models. No sense fine tuning a set of predictors for Price, then deciding you should be predicting log(Price) or Priceˆ2. So make that decision fairly early, but don't get too picky and expect to get perfect plot of residuals versus fits or an exact normal quantile plot.
- Similarly, if you decide that some data cases should be dropped from the training set, don't wait until late in the process to do so. For example, if you spot a *very* large residual you should look at the characteristics for that house to see if it should be deleted. Don't forget about the value of simple plots (like a scatterplot of Price vs. LotArea) for helping to see what is going on and recognize extreme cases. Be sure to document any adjustments you make in the final report.
- Comparing C_p from different predictor pools - While Mallow's C_p is a useful tool for comparing models from the same pool of predictors. You should not use it to compare models based on different predictor pools. For example, if you add a bunch of categorical variables to all the quantitative predictors from homework #3 to make a new "full" model, then find C_p from a model that you fit in homework #3, it will be worse than it was before. If you look at the formula for calculating C_p, you will see that all that has changed is MSE for the full model after adding the new batch of predictors.
- I should be able to follow the steps you use when selecting a model. I certainly don't need to see every bit of output, but it might help to include more of the R commands you use. For example, saying you used backward elimination is not very helpful when I don't know what you start with for the full model or pool of predictors (e.g. did you include Condition and Quality as numeric predictors? or did you decide to

eliminate one of GroundSF, FirstSF, or SecondSF due to redundancy?). The easiest way to convey this in many cases is to show the R command you used. It is fine to abbreviate the output (for example, delete many steps in a stepwise procedure using trace=FALSE), but it would be helpful if you identified the parts you do include. For example, a sentence like “After 12 steps of the stepwise procedure, we have the output below for the fitted model.” Similarly, I don’t need to see 600 residuals, using head and sort can show the important ones.

- Once you have settled on a response, made adjustments to the data (if needed), and chosen a set of predictors, be sure to include the summary( )for your “fancy” model at this stage.

Part 9: Cross-validation for your “fancy” model Redo the cross-validation analysis with your test data for your new fancy model. Use AmesTest??.csv, where ?? corresponds to your new group number for homework #5. Discuss how the various measures (mean of residuals, std. dev of residuals, shape of the distributions of residuals, cross-validation correlation, and shrinkage) compare to the results you had for your basic model. Don’t worry about looking for poorly predicted cases this time. If you transformed the response variable, consider how to take this into account for your residual analysis. In order to compare residuals they should have the same units!

Note on missing categories:

> When creating the predictions using predict(yourmodel,AmesTest) you may see an error like:
>
> Error in model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels) : factor HouseStyle has new levels 1.5Unf, 2.5Fin, 2.5Unf
>
> This occurs because the holdout sample has a value for the categorical variable that was not present in your training sample, so there is no indicator in your model to handle that case. To get a prediction for that house, you’ll need to switch the category to one that is in your training data. In the example above you might choose to replace the “2.5Fin” house style with “2Story”. If you are not sure what category to use, try whatever R uses as the “left out” reference category. Be sure to record any changes like this that you make.

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.