Stat. Inf. II: Homework 9

Ames SalePrice Model

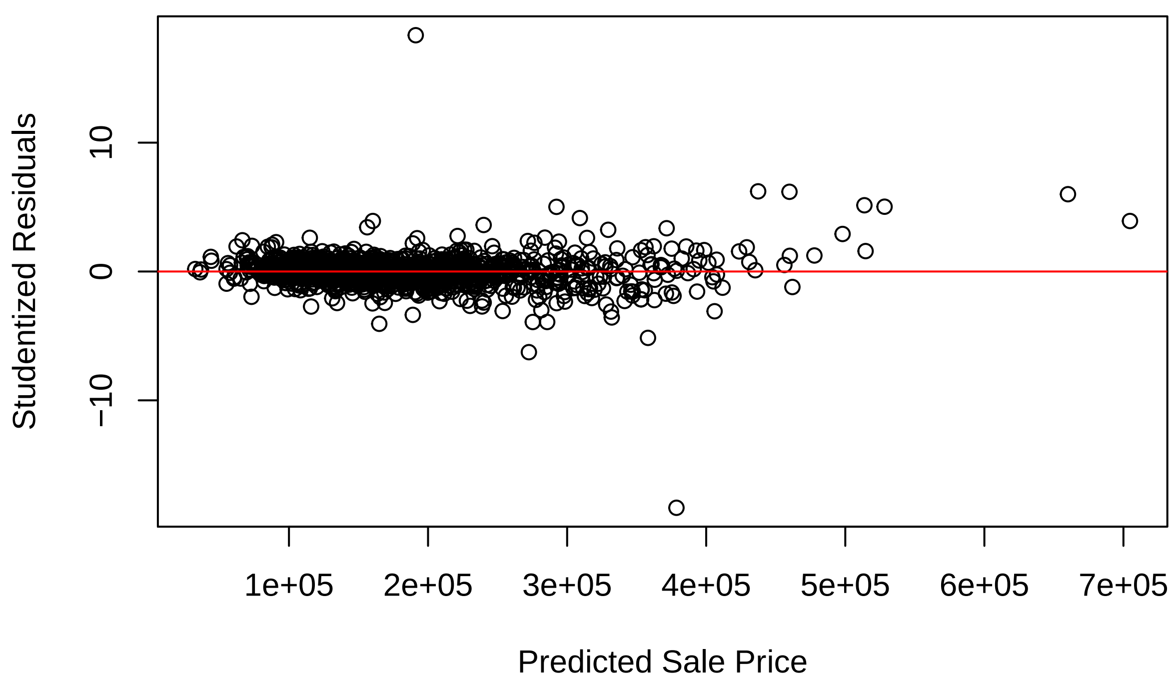
Austin Anderson, Gregory Barber, James Trimarco

Kaggle Competition

The HW you turn in needs to include:

a. At least one residual plot

```
fit <- lm(train$SalePrice ~ ., data = train)
sr.fit <- rstudent(fit)
plot(sr.fit~ fitted(fit), xlab = "Predicted Sale Price", ylab = "Studentized Residuals")
abline(h = 0, col = "red")
```

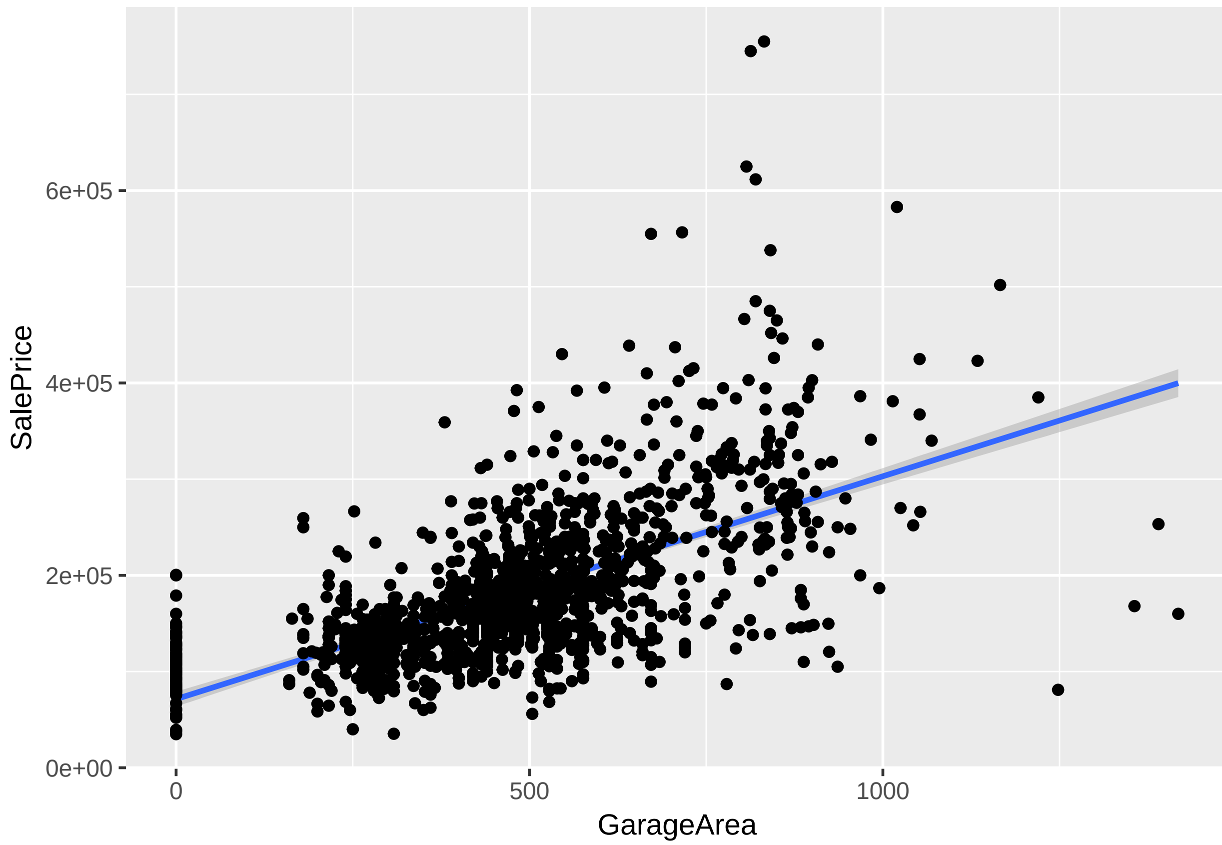


The studentized residuals seem to have fairly constant variance, suggesting a linear model is appropriate. There are a couple outliers present, which we took note of. There may also be a slight pattern emerging as the residuals become more positive as the predicted price increases. In our modeling code, we removed outliers in key quantitative predictors.

b. At least one interpretation of a multiple regression coefficient using a 95% confidence interval for that coefficient,

First we'll inspect the variable to see if it is normally distributed around the mean of the response. The plot suggest a very rough normal distribution, but definitely not perfect.

```
ggplot(train, aes(x = GarageArea, y = SalePrice)) +
  geom_smooth(method = "lm") +
  geom_point()
```



Now we get a confidence interval for the slope. Since we know the assumptions of this method aren't entirely met, we have to look at this confidence interval with some skepticism.

```
cis <- confint(fit,parm = 'GarageArea')
cis
```

```
##               2.5 %   97.5 %
## GarageArea 2.277881 33.63239
```

For every one square foot increase in Garage Area, we predict sale price to increase by at least $2.28 and at most $33.63, with 95% confidence while keeping all other predictors fixed.

c. The final model that you submitted with a paragraph describing how you came up with that model. Supress (via `echo`) R output from intermediate steps, only show me the important steps.

We ran into a lot of errors having to do with the levels of the dummy variables not matching in the train and test sets. Our solution involves briefly joining the two datasets into one, which ensures that factors have the same levels.

We then separated the data and removed outliers.

```
outlier_vars <- c("LotArea", "BsmtFinSF1", "TotalBsmtSF", "X1stFlrSF", "GrLivArea", "GarageArea", "OpenI

replace_outliers <- function(dataframe){
   dataframe %>%
           map_at(outlier_vars, ~ replace(.x, .x %in% boxplot.stats(.x, coef = 3)$out, NA)) %>%
           bind_cols
}

train <- replace_outliers(train)

train <- train %>% drop_na(outlier_vars)
```

Define contrasts function

Required for creating model matrix

Create factors

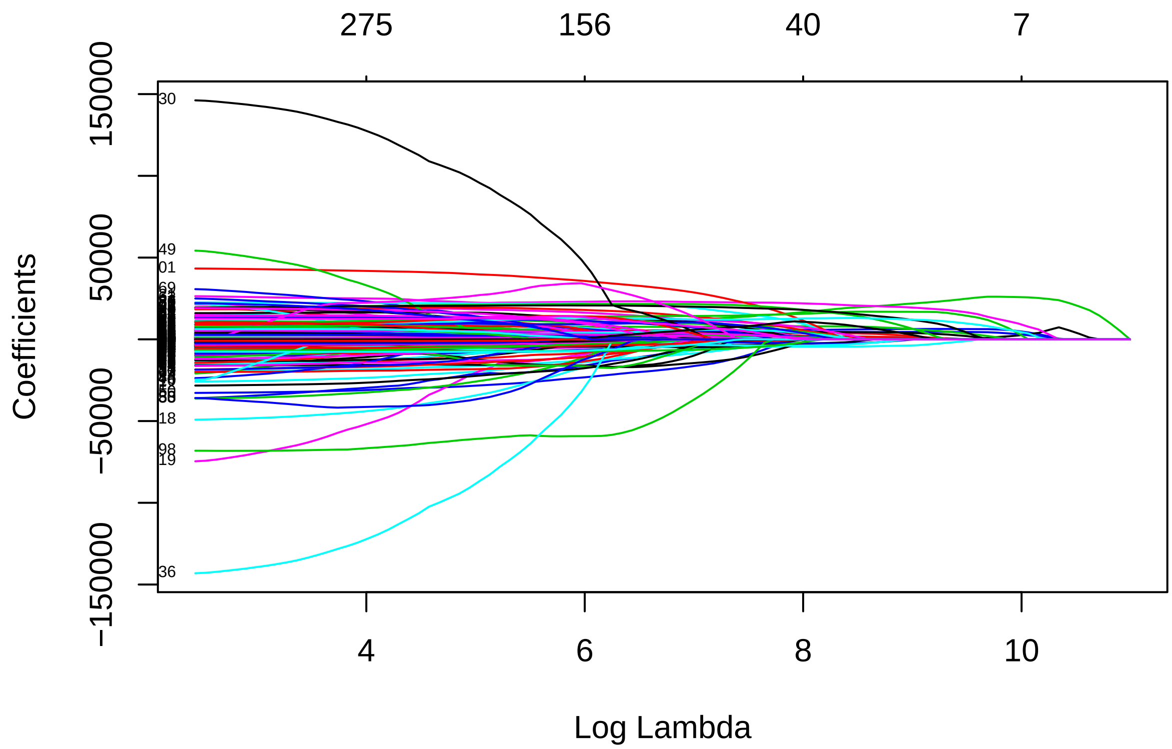
This matrix contains only the factors. The numerical data is accessed separately.

And fit on the training data.

```
quants_train <- scale(train[, quant_idx])

x_train  <- data.matrix(data.frame(quants_train, factors_train))
y_train <- data.matrix(train$SalePrice)

glmmod <- glmnet(x_train, y_train, alpha=1, family="gaussian")

coef(glmmod)[, 15][coef(glmmod)[, 20] > 0]
```

```
##     (Intercept)         LotArea     OverallQual       YearBuilt
##    1.758922e+05    0.000000e+00    2.608691e+04    0.000000e+00
##    YearRemodAdd     TotalBsmtSF       X1stFlrSF      GarageArea
##    0.000000e+00    6.225732e+03    1.123839e+00    4.336620e+03
##    YearBuilt_SQ YearRemodAdd_SQ   BsmtFinSF1_SQ    GrLivArea_SQ
##    2.335046e+02    0.000000e+00    2.051912e+03    1.378872e+04
##     ExterQualEx       BsmtQualEx    KitchenQualEx     GarageCars3
##    0.000000e+00    1.184360e+04    0.000000e+00    8.089928e+03
```

```
plot(glmmod, xvar = "lambda", label = TRUE)
```



One could rewrite the list of coefficients printed above in linear regression notation as:

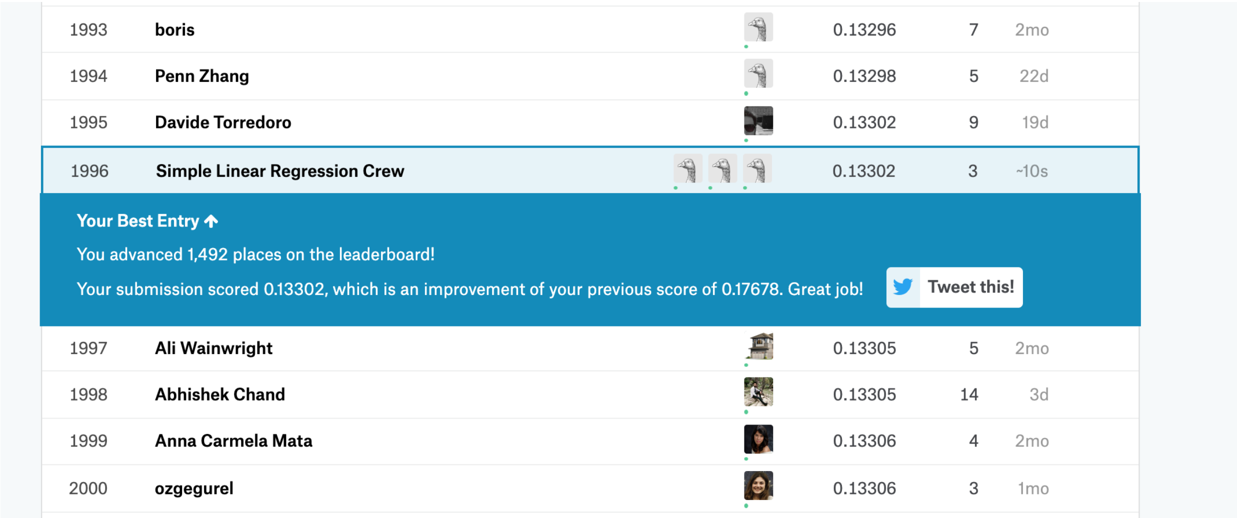
$$\mu_{saleprice} = 173,568 + 25,500 OveralQual + 6,013 TotalBsmtSF + 4025 GarageArea + 13,874 GrLivArea^2 \ldots$$

We used cross validation to check the right value for lambda. #### Cross validation

```
cv.model <- cv.glmnet(x_train, y_train,
                      alpha=1, type.measure = "mse", nfolds = 10)

(best.lambda <- cv.model$lambda.min)
```

```
## [1] 391.0568
```

```
plot(cv.model)
```

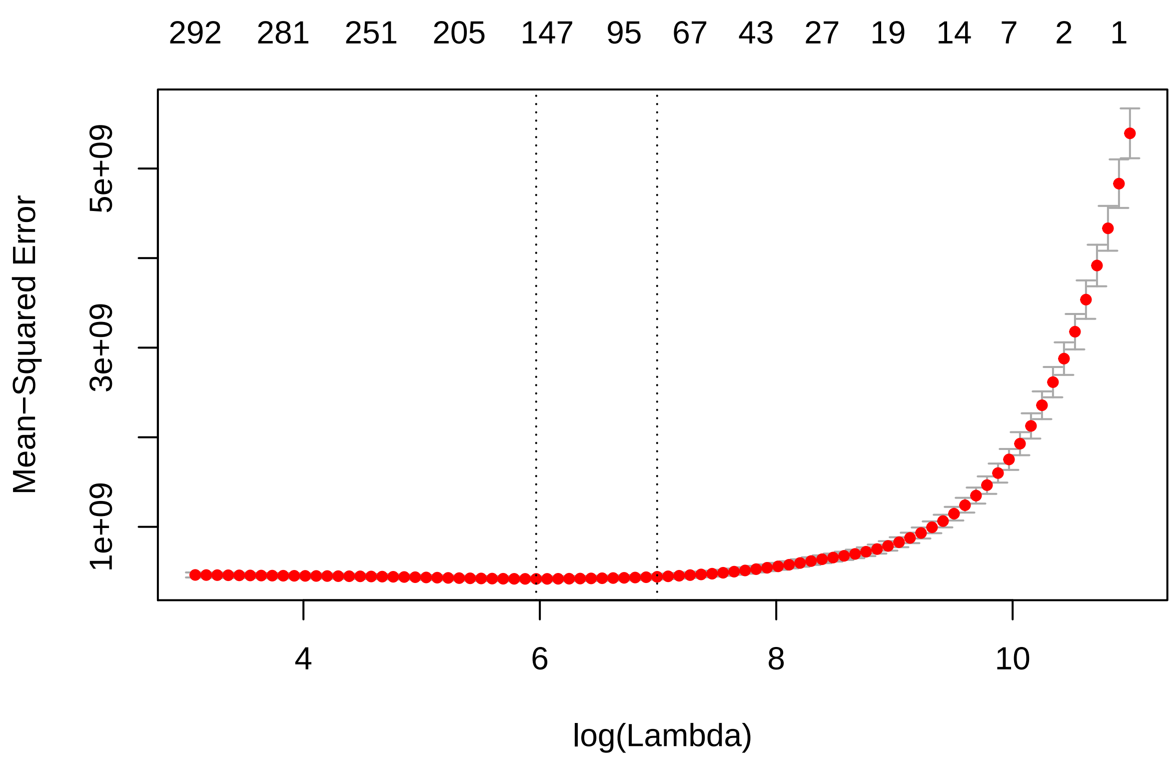


1993	boris	0.13296	7	2mo
1994	Penn Zhang	0.13298	5	22d
1995	Davide Torredoro	0.13302	9	19d
1996	Simple Linear Regression Crew	0.13302	3	~10s

Your Best Entry ↑
You advanced 1,492 places on the leaderboard!
Your submission scored 0.13302, which is an improvement of your previous score of 0.17678. Great job!
Tweet this!

1997	Ali Wainwright	0.13305	5	2mo
1998	Abhishek Chand	0.13305	14	3d
1999	Anna Carmela Mata	0.13306	4	2mo
2000	ozgegurel	0.13306	3	1mo

Figure 1: A caption




```
#coef(cv.model)
```

This model got us to position 1996 on Kaggle!

One oddness of this experience is that we tried something called elasticnet, which mixes in part of the output from ridge regression and part from lasso. This model was giving us great estimated RMSE values – like $21,500 or so. But that model did not do well on the test data – we got a worse score than our first at 0.18

or so. We're not sure why cross validation suggested this model was the best, and we'd like to understand the experience better.