# Module 2 Assignment 2

## Matthew Woody

### Package Loads

library("tidyverse")  
library("tidymodels")  
library("glmnet")  
library("GGally")  
library("ggcorrplot")  
library("MASS")  
library("car")  
library("lubridate")  
library("lmtest")

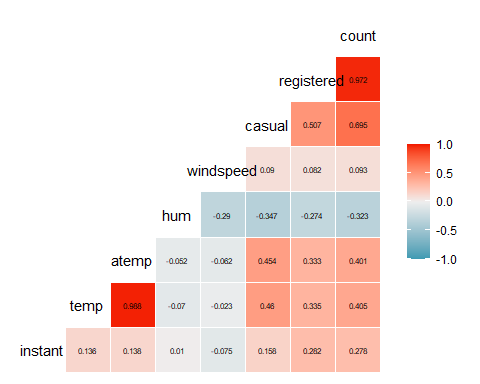
### Task 1

bike <- read\_csv("bike\_cleaned.csv")  
bike <- bike %>% mutate(dteday = mdy(dteday))  
bike <- bike %>% mutate\_if(is.character,as.factor)  
bike$hr = as.factor(bike$hr)  
glimpse(bike)  
summary(bike)

The column hr was converted into a factor since it is actually categorical since it is a “point in time” as opposed to being numeric. Leaving as numbers would produce incorrect results in all of the statistical calculation such as correlation, r-square, p-value, etc.

### Task 2

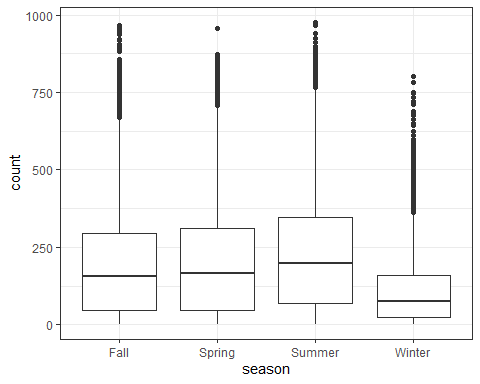
ggcorr(bike, label = TRUE, label\_size = 2, label\_round = 3)



The temp variable seems to have the highest correlation with the count variable at 0.405.

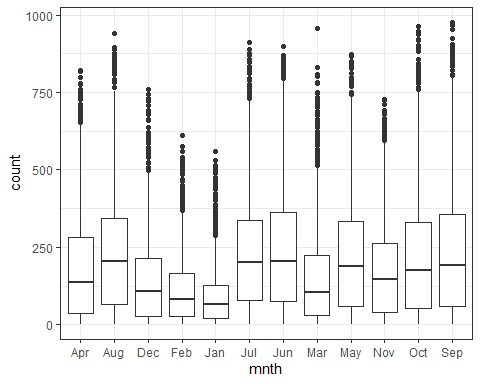
### Task 3

ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()



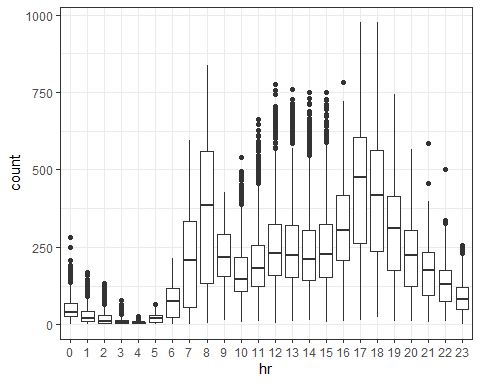
Season does seem to impact count as the number of bike cleanings in winter are significantly lower. There would be multicollinearity with the month. Perhaps temperature would be a better variable.

ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()



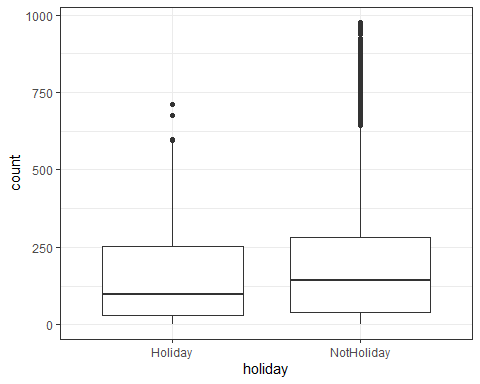
Month does seem to impact count as the number of bike cleanings in winter months are significantly lower. There would be multicollinearity with the season. Perhaps temperature would be a better variable.

ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw()



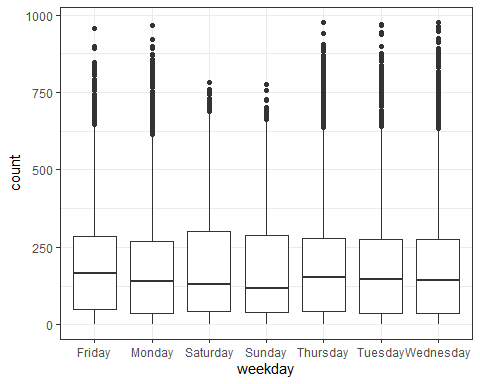
Hour does seem to impact count as the morning and early evening hours are significantly higher in count. Perhaps temperature would be a better variable since bike cleaning would take place during the most comfortable times for temperature during 24 hours.

ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()



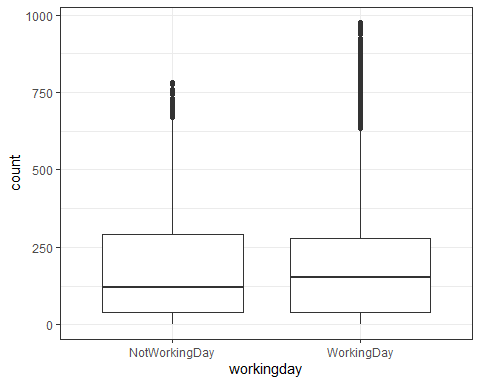
Holiday seems to impact count but would have multicollinarity with the Working Day field.

ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()



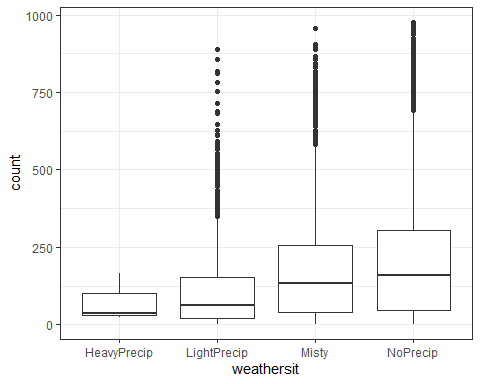
Weekday could possibly have an impact on count, as it seems the counts for Saturday-Sunday are significantly lower. There would be multicollinearity with the working day field.

ggplot(bike,aes(x=workingday,y=count)) + geom\_boxplot() + theme\_bw()



Weekday could possibly have an impact on count, as it seems the counts for Saturday-Sunday are significantly lower. There would be multicollinearity with the Week day field.

ggplot(bike,aes(x=weathersit,y=count)) + geom\_boxplot() + theme\_bw()



Weather Situation seems to have an impact on count, but there could possibly be multicollinarity with the Season and Month fields.

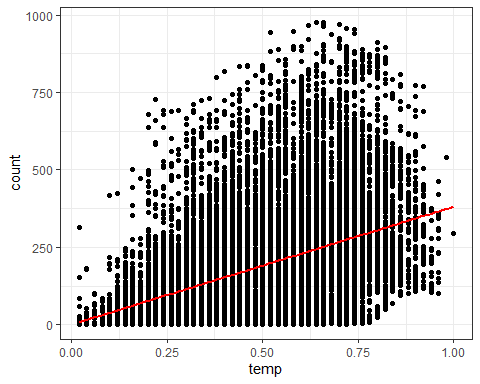
### Task 4

temp\_model = recipe(count ~ temp, bike)  
  
lm\_model =   
 linear\_reg() %>%   
 set\_engine("lm")   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(temp\_model)  
  
lm\_fit = fit(lm\_wflow, bike)  
  
summary(lm\_fit$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -291.37 -110.23 -32.86 76.77 744.76   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.0356 3.4827 -0.01 0.992   
## temp 381.2949 6.5344 58.35 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 165.9 on 17377 degrees of freedom  
## Multiple R-squared: 0.1638, Adjusted R-squared: 0.1638   
## F-statistic: 3405 on 1 and 17377 DF, p-value: < 2.2e-16

ggplot(bike, aes(x=temp,y=count)) + geom\_point() +   
 geom\_smooth(method="lm",se=FALSE, color="red") + theme\_bw()

## `geom\_smooth()` using formula 'y ~ x'

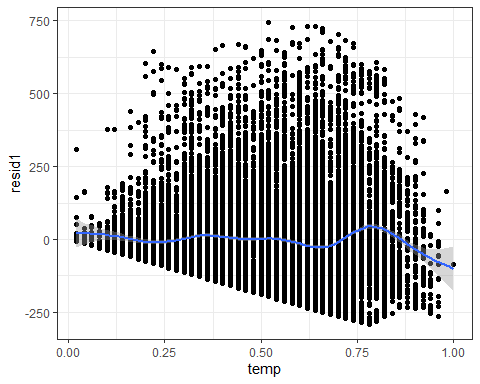


dwtest(lm\_fit$fit$fit$fit)

##   
## Durbin-Watson test  
##   
## data: lm\_fit$fit$fit$fit  
## DW = 0.3684, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0

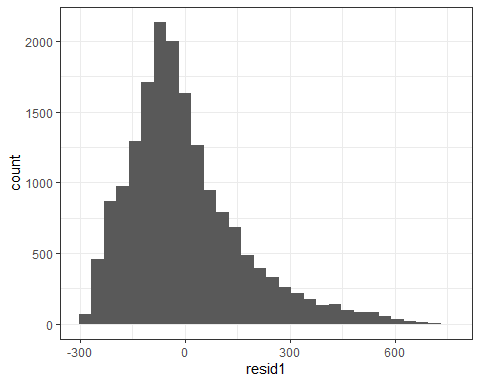
bike = bike %>% mutate(resid1 = lm\_fit$fit$fit$fit$residuals)   
  
ggplot(bike,aes(x=temp,y=resid1)) + geom\_point()+ geom\_smooth() + theme\_bw()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



ggplot(bike,aes(x=resid1)) + geom\_histogram() + theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



There seems to be a “mixed-bag” over whether temperature alone is a good predictor for count. The R-Squared is low at .1638 while the p-value seems to indicate that temperature is a significant variable. A scatter plot and best fit line of the data points seems to show something of a linear relationship to a certain temperature, although the relationship between count and temperature seems to diminish somewhat at higher temperatures. Running the diagnostics on the model, the Durbin-Watson test shows that the residuals are independent. Plotting the residuals in a scatter plot shows that a non-linear effect is present and a histogram plot shows that there is a normal distribution. Adding additional variables would increase the r-squared and improve the model.

### Task 5

bike2<-bike %>% dplyr::select(-instant,-dteday,-registered,-casual,-resid1)  
  
bike\_ridge<-recipe(count~.,bike2) %>%  
 step\_dummy(all\_nominal())%>%  
 step\_center(all\_predictors()) %>%   
 step\_scale(all\_predictors())   
   
ridge\_model =   
 linear\_reg(mixture = 0) %>%   
 set\_engine("glmnet")   
  
ridge\_wflow =   
 workflow() %>%   
 add\_model(ridge\_model) %>%   
 add\_recipe(bike\_ridge)  
  
ridge\_fit = fit(ridge\_wflow, bike2)

ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit")

##   
## Call: glmnet::glmnet(x = maybe\_matrix(x), y = y, family = "gaussian", alpha = ~0)   
##   
## Df %Dev Lambda  
## 1 52 0.00 73420  
## 2 52 0.61 66900  
## 3 52 0.67 60950  
## 4 52 0.74 55540  
## 5 52 0.81 50600  
## 6 52 0.88 46110  
## 7 52 0.97 42010  
## 8 52 1.06 38280  
## 9 52 1.16 34880  
## 10 52 1.27 31780  
## 11 52 1.39 28960  
## 12 52 1.53 26390  
## 13 52 1.67 24040  
## 14 52 1.83 21910  
## 15 52 2.00 19960  
## 16 52 2.19 18190  
## 17 52 2.40 16570  
## 18 52 2.62 15100  
## 19 52 2.86 13760  
## 20 52 3.13 12540  
## 21 52 3.41 11420  
## 22 52 3.72 10410  
## 23 52 4.06 9482  
## 24 52 4.43 8640  
## 25 52 4.83 7872  
## 26 52 5.26 7173  
## 27 52 5.72 6536  
## 28 52 6.22 5955  
## 29 52 6.76 5426  
## 30 52 7.34 4944  
## 31 52 7.96 4505  
## 32 52 8.62 4105  
## 33 52 9.33 3740  
## 34 52 10.09 3408  
## 35 52 10.90 3105  
## 36 52 11.76 2829  
## 37 52 12.67 2578  
## 38 52 13.63 2349  
## 39 52 14.65 2140  
## 40 52 15.72 1950  
## 41 52 16.83 1777  
## 42 52 18.01 1619  
## 43 52 19.23 1475  
## 44 52 20.49 1344  
## 45 52 21.81 1225  
## 46 52 23.16 1116  
## 47 52 24.56 1017  
## 48 52 25.98 926  
## 49 52 27.44 844  
## 50 52 28.93 769  
## 51 52 30.43 701  
## 52 52 31.95 639  
## 53 52 33.48 582  
## 54 52 35.01 530  
## 55 52 36.53 483  
## 56 52 38.04 440  
## 57 52 39.54 401  
## 58 52 41.01 365  
## 59 52 42.44 333  
## 60 52 43.84 303  
## 61 52 45.20 276  
## 62 52 46.51 252  
## 63 52 47.77 230  
## 64 52 48.96 209  
## 65 52 50.10 190  
## 66 52 51.18 174  
## 67 52 52.19 158  
## 68 52 53.14 144  
## 69 52 54.02 131  
## 70 52 54.83 120  
## 71 52 55.59 109  
## 72 52 56.28 99  
## 73 52 56.91 91  
## 74 52 57.49 82  
## 75 52 58.01 75  
## 76 52 58.48 68  
## 77 52 58.91 62  
## 78 52 59.30 57  
## 79 52 59.64 52  
## 80 52 59.96 47  
## 81 52 60.24 43  
## 82 52 60.49 39  
## 83 52 60.72 36  
## 84 52 60.93 33  
## 85 52 61.11 30  
## 86 52 61.28 27  
## 87 52 61.44 25  
## 88 52 61.58 22  
## 89 52 61.71 20  
## 90 52 61.83 19  
## 91 52 61.95 17  
## 92 52 62.05 15  
## 93 52 62.14 14  
## 94 52 62.23 13  
## 95 52 62.32 12  
## 96 52 62.40 11  
## 97 52 62.47 10  
## 98 52 62.54 9  
## 99 52 62.60 8  
## 100 52 62.66 7

ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%   
 coef(s = 43)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 23.5783786  
## atemp 23.4074844  
## hum -24.1899670  
## windspeed -0.9701015  
## season\_Spring -1.2298905  
## season\_Summer -4.5618556  
## season\_Winter -13.5609299  
## mnth\_Aug -0.1273960  
## mnth\_Dec 0.5273145  
## mnth\_Feb -3.0879704  
## mnth\_Jan -3.5349931  
## mnth\_Jul -5.5560858  
## mnth\_Jun -1.3725328  
## mnth\_Mar -0.3449820  
## mnth\_May 2.8405731  
## mnth\_Nov 2.1844274  
## mnth\_Oct 7.9347356  
## mnth\_Sep 7.5906064  
## hr\_X1 -19.4378757  
## hr\_X2 -20.6036498  
## hr\_X3 -21.9250887  
## hr\_X4 -22.2581593  
## hr\_X5 -19.9348546  
## hr\_X6 -10.5735561  
## hr\_X7 11.2396911  
## hr\_X8 33.8957503  
## hr\_X9 9.4129227  
## hr\_X10 0.1149916  
## hr\_X11 3.9028472  
## hr\_X12 9.9749541  
## hr\_X13 8.9623156  
## hr\_X14 6.2189247  
## hr\_X15 7.7168016  
## hr\_X16 17.9094935  
## hr\_X17 43.1850556  
## hr\_X18 38.1457010  
## hr\_X19 20.7921586  
## hr\_X20 8.0698926  
## hr\_X21 0.3022574  
## hr\_X22 -5.5310114  
## hr\_X23 -11.7224965  
## holiday\_NotHoliday 3.1907439  
## weekday\_Monday -1.5025324  
## weekday\_Saturday 1.4316015  
## weekday\_Sunday -2.5647987  
## weekday\_Thursday -0.5058227  
## weekday\_Tuesday -0.8964398  
## weekday\_Wednesday -0.1443014  
## workingday\_WorkingDay 2.0030564  
## weathersit\_LightPrecip -10.6265662  
## weathersit\_Misty 1.9838868  
## weathersit\_NoPrecip 4.2840229

Pulling in all the variables, I selected the lambda value of 43, which would give me an R-Squared value of 60%. Many of the variables no doubt have multicollinarity, so the Ridge model would be beneficial to use for the purposes of modeling all the predictors. The model shows considerable variability in the slope coefficients for hour, season and month.

### Task 6

bike\_lasso<-recipe(count~.,bike2) %>%  
step\_dummy(all\_nominal()) %>%   
 step\_center(all\_predictors()) %>%   
 step\_scale(all\_predictors())  
  
lasso\_model =  
 linear\_reg(mixture = 1) %>%   
 set\_engine("glmnet")  
  
lasso\_wflow =   
 workflow() %>%   
 add\_model(lasso\_model) %>%   
 add\_recipe(bike\_lasso)  
  
lasso\_fit = fit(lasso\_wflow, bike2)

lasso\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit")

##   
## Call: glmnet::glmnet(x = maybe\_matrix(x), y = y, family = "gaussian", alpha = ~1)   
##   
## Df %Dev Lambda  
## 1 0 0.00 73.420  
## 2 1 2.78 66.900  
## 3 1 5.09 60.950  
## 4 3 7.60 55.540  
## 5 3 11.69 50.600  
## 6 4 15.44 46.110  
## 7 4 19.18 42.010  
## 8 6 22.56 38.280  
## 9 6 26.23 34.880  
## 10 6 29.28 31.780  
## 11 8 32.06 28.960  
## 12 11 34.97 26.390  
## 13 12 38.11 24.040  
## 14 12 40.86 21.910  
## 15 14 43.28 19.960  
## 16 14 45.50 18.190  
## 17 15 47.37 16.570  
## 18 15 49.03 15.100  
## 19 16 50.55 13.760  
## 20 16 51.81 12.540  
## 21 18 52.98 11.420  
## 22 19 54.01 10.410  
## 23 21 54.90 9.482  
## 24 24 55.78 8.640  
## 25 25 56.58 7.872  
## 26 26 57.29 7.173  
## 27 27 57.91 6.536  
## 28 27 58.47 5.955  
## 29 28 58.95 5.426  
## 30 28 59.38 4.944  
## 31 29 59.74 4.505  
## 32 31 60.09 4.105  
## 33 32 60.41 3.740  
## 34 32 60.69 3.408  
## 35 32 60.92 3.105  
## 36 33 61.11 2.829  
## 37 36 61.30 2.578  
## 38 37 61.60 2.349  
## 39 36 61.82 2.140  
## 40 36 61.98 1.950  
## 41 38 62.13 1.777  
## 42 39 62.25 1.619  
## 43 40 62.36 1.475  
## 44 41 62.46 1.344  
## 45 42 62.58 1.225  
## 46 42 62.69 1.116  
## 47 42 62.77 1.017  
## 48 41 62.84 0.926  
## 49 42 62.89 0.844  
## 50 42 62.92 0.769  
## 51 42 62.96 0.701  
## 52 42 62.98 0.639  
## 53 42 63.01 0.582  
## 54 42 63.04 0.530  
## 55 42 63.05 0.483  
## 56 43 63.07 0.440  
## 57 44 63.09 0.401  
## 58 45 63.11 0.365  
## 59 45 63.13 0.333  
## 60 45 63.14 0.303  
## 61 46 63.15 0.276  
## 62 49 63.16 0.252  
## 63 49 63.17 0.230  
## 64 49 63.18 0.209  
## 65 49 63.19 0.190  
## 66 49 63.19 0.174  
## 67 49 63.20 0.158  
## 68 49 63.20 0.144  
## 69 49 63.21 0.131  
## 70 48 63.21 0.120  
## 71 48 63.21 0.109  
## 72 48 63.21 0.099  
## 73 48 63.22 0.091  
## 74 49 63.22 0.082  
## 75 49 63.22 0.075  
## 76 49 63.22 0.068  
## 77 49 63.22 0.062  
## 78 49 63.22 0.057  
## 79 50 63.22 0.052  
## 80 50 63.22 0.047  
## 81 50 63.22 0.043

lasso\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%   
 coef(s = .582)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.463087634  
## temp 30.797724272  
## atemp 21.964059648  
## hum -23.037824043  
## windspeed -4.092837857  
## season\_Spring -6.401578787  
## season\_Summer -12.862600717  
## season\_Winter -22.125902659  
## mnth\_Aug .   
## mnth\_Dec .   
## mnth\_Feb .   
## mnth\_Jan .   
## mnth\_Jul -6.834778851  
## mnth\_Jun -2.257008878  
## mnth\_Mar 1.430222892  
## mnth\_May 1.793051529  
## mnth\_Nov .   
## mnth\_Oct 5.573551344  
## mnth\_Sep 7.796042536  
## hr\_X1 -9.394812737  
## hr\_X2 -10.904086982  
## hr\_X3 -12.704007326  
## hr\_X4 -13.063354270  
## hr\_X5 -10.018932096  
## hr\_X6 0.489447070  
## hr\_X7 27.321080245  
## hr\_X8 55.097653993  
## hr\_X9 25.034847883  
## hr\_X10 13.585808618  
## hr\_X11 18.166388556  
## hr\_X12 25.624118681  
## hr\_X13 24.368974784  
## hr\_X14 21.020250109  
## hr\_X15 22.868417466  
## hr\_X16 35.418573294  
## hr\_X17 66.475568877  
## hr\_X18 60.268630645  
## hr\_X19 38.938650105  
## hr\_X20 23.335719184  
## hr\_X21 13.748760493  
## hr\_X22 6.592922663  
## hr\_X23 -0.000833544  
## holiday\_NotHoliday 3.994300881  
## weekday\_Monday -1.043576634  
## weekday\_Saturday 0.321401515  
## weekday\_Sunday -3.746680381  
## weekday\_Thursday .   
## weekday\_Tuesday -0.396343791  
## weekday\_Wednesday .   
## workingday\_WorkingDay .   
## weathersit\_LightPrecip -14.347570257  
## weathersit\_Misty .   
## weathersit\_NoPrecip 2.052181456

Using the Lasso model method, I chose the Lambda coefficient value of .582 which would give me an r-squared value of 63%. The Lasso method zeroed out many of the variables.

Given that there is quite definitely multicollinarity in our predictor variables, perhaps the ridge method is a better choice for determining predictability by variable.