### Multiple Regression Assignment 2

## Engel, Alec

bike <- read\_csv("bike\_cleaned.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## instant = col\_double(),  
## dteday = col\_character(),  
## season = col\_character(),  
## mnth = col\_character(),  
## hr = col\_double(),  
## holiday = col\_character(),  
## weekday = col\_character(),  
## workingday = col\_character(),  
## weathersit = col\_character(),  
## temp = col\_double(),  
## atemp = col\_double(),  
## hum = col\_double(),  
## windspeed = col\_double(),  
## casual = col\_double(),  
## registered = col\_double(),  
## count = col\_double()  
## )

bike = bike %>%  
 mutate(dteday = mdy(dteday))

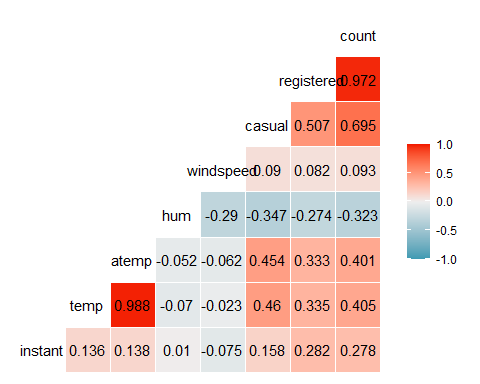
bike = bike %>%  
 mutate\_if(is.character, as.factor)

bike = bike %>%  
 mutate(hr = as\_factor(hr))

Why do we convert the “hr” variable into factor? Why not just leave as numbers?  
**The benefit of converting hours(hr) to factors is that each of the variables (0-23) is only now stored once and we are left with only 24 levels.**

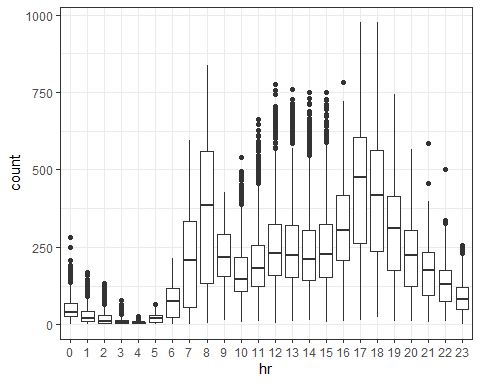
ggcorr(bike, label = TRUE, label\_round = 3)

## Warning in ggcorr(bike, label = TRUE, label\_round = 3): data in column(s)  
## 'dteday', 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday',  
## 'weathersit' are not numeric and were ignored

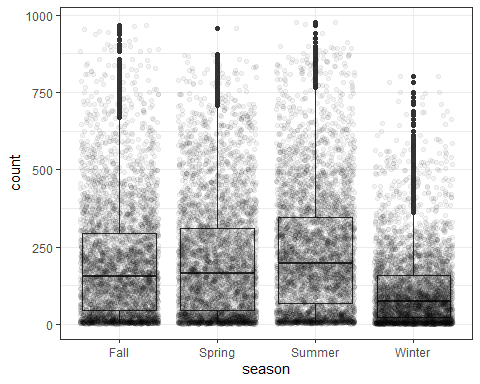


Which of the quantitative variables appears to be best correlated with “count”? **Normalized feeling temperature(atemp) along with normalized temperature(temp) appear to be best correlated with “count”**

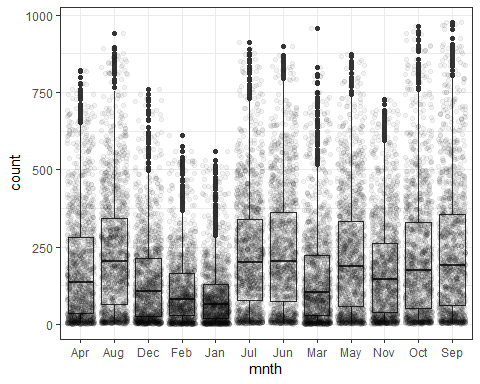
ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw()



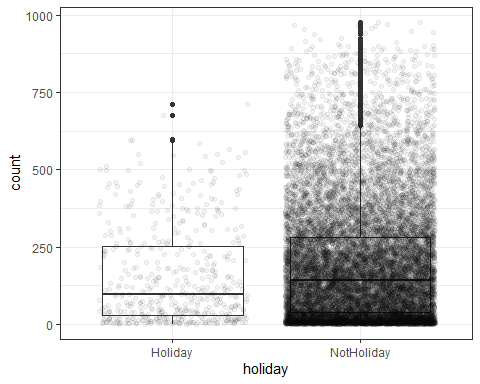
ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + geom\_jitter(alpha = 0.05) + theme\_bw()



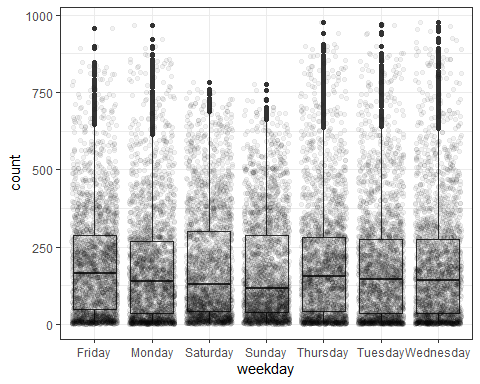
ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + geom\_jitter(alpha = 0.05) + theme\_bw()



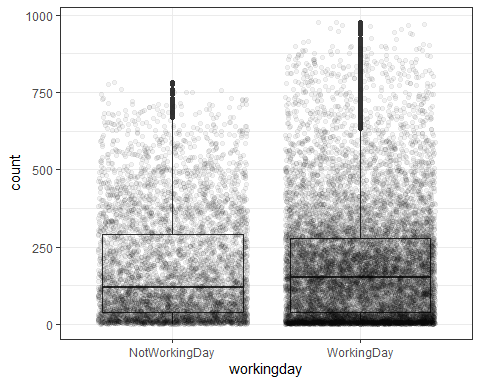
ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + geom\_jitter(alpha = 0.05) + theme\_bw()



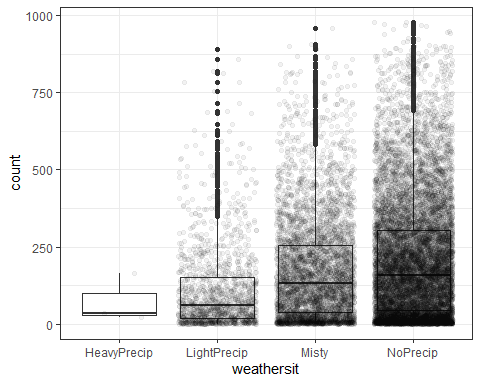
ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + geom\_jitter(alpha = 0.05) + theme\_bw()



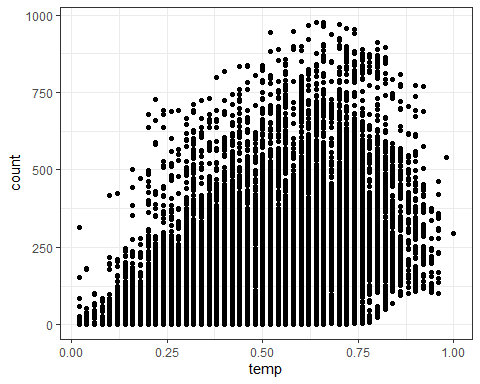
ggplot(bike,aes(x=workingday,y=count)) + geom\_boxplot() + geom\_jitter(alpha = 0.05) + theme\_bw()



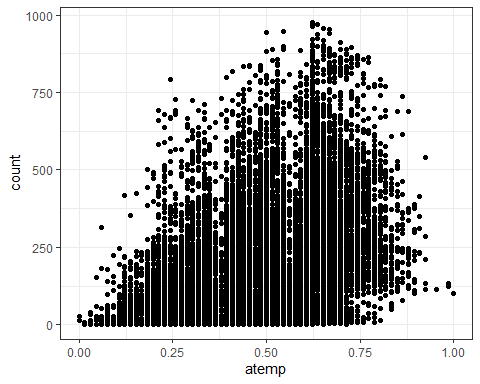
ggplot(bike,aes(x=weathersit,y=count)) + geom\_boxplot() + geom\_jitter(alpha = 0.05) + theme\_bw()



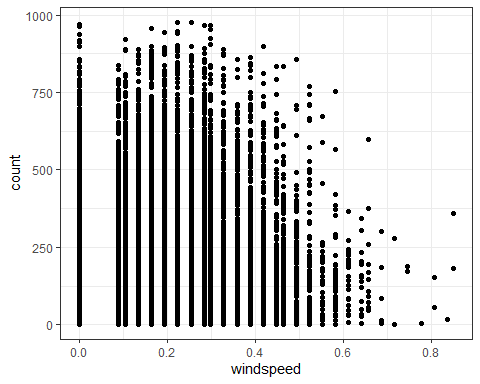
ggplot(bike,aes(x=temp,y=count)) + geom\_point() + theme\_bw()



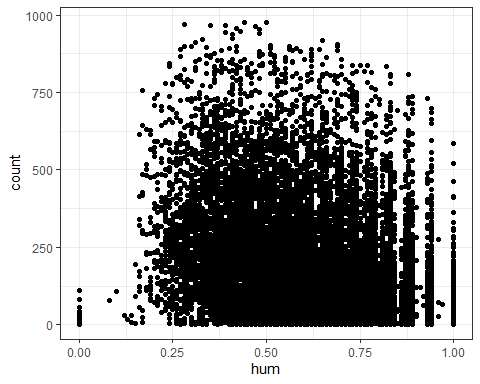
ggplot(bike,aes(x=atemp,y=count)) + geom\_point() + theme\_bw()



ggplot(bike,aes(x=windspeed,y=count)) + geom\_point() + theme\_bw()



ggplot(bike,aes(x=hum,y=count)) + geom\_point() + theme\_bw()



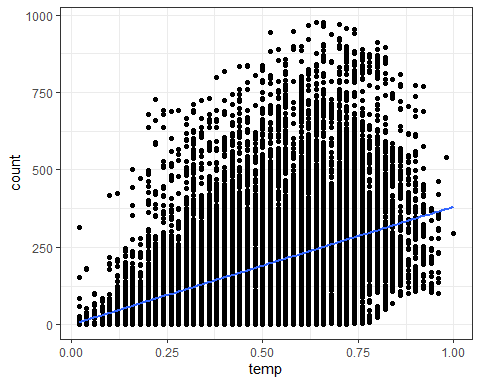
Which variables appear to affect “count”? Provide a brief explanation as to why you believe that each variable does or does not affect “count”.  
**Based on these boxplots; I see that season, month, weather, temperature, humidity and wind speed affect the count of total bike rentals while holiday, weekday, and working days do not. The time of the year affecting bike rentals makes sense due to nicer outdoor weather conditions and longer days. This correlation of nicer weather conditions during seasonal times aligning with increased bike rental users justifies our season, month, weather, and temperature variables. We also see sort of a cap on temp and humidity as people’s interest in renting a bike would obviously reduce with temperatures in the hundreds or on high humidity days. It also makes sense that we would see less bike rentals on days in which wind speeds were increased as this would make it more difficult to ride a bike. Surprisingly to me, holidays don’t look to have affected the count but that may be due to that fact that there are so few holidays each year and our observations would be less for holidays rather than non-holidays. I see weekdays and working days along the same lines as I would assume more bike rentals on the weekends when people have free time for leisure rides. However, this makes sense that weekends and workdays would be similar since many people could be renting bikes for transportation to and from work.**

temp\_recipe = recipe(count ~ temp, bike)  
  
lm\_model =   
 linear\_reg() %>%  
 set\_engine("lm")  
  
lm\_wflow =   
 workflow() %>%  
 add\_model(lm\_model) %>%  
 add\_recipe(temp\_recipe)  
  
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) + theme\_bw()

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



**In this model, we see a decent R squared value in comparison to other variables and we see a slope coefficient significance of less than .05 showing that temp is a significant predictor of count. The model also shows normalized temperature in celcius correlating to count of bike rentals with a slope of 381.2949 and proving that as temperature rises, the count of bike rentals also rise.**

recipe2 = recipe(count ~season+mnth+hr+holiday+weekday+workingday+weathersit+temp+atemp+hum+windspeed, bike)%>%  
 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(recipe2)  
  
ridge\_fit = fit(ridge\_wflow, bike)  
  
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 = 33)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.46308763  
## temp 24.45441286  
## atemp 24.12124084  
## hum -24.42775453  
## windspeed -1.62051521  
## season\_Spring -1.76697581  
## season\_Summer -5.59348270  
## season\_Winter -14.52597694  
## mnth\_Aug -0.26814429  
## mnth\_Dec 0.81066338  
## mnth\_Feb -2.58231387  
## mnth\_Jan -2.91817359  
## mnth\_Jul -6.04851305  
## mnth\_Jun -1.70476046  
## mnth\_Mar 0.05494344  
## mnth\_May 2.82545575  
## mnth\_Nov 2.26336322  
## mnth\_Oct 8.04251861  
## mnth\_Sep 7.84444467  
## hr\_X1 -19.49419963  
## hr\_X2 -20.71029244  
## hr\_X3 -22.09864455  
## hr\_X4 -22.42926318  
## hr\_X5 -19.98644132  
## hr\_X6 -10.18605592  
## hr\_X7 12.60948704  
## hr\_X8 36.26296911  
## hr\_X9 10.65092319  
## hr\_X10 0.90221331  
## hr\_X11 4.82267637  
## hr\_X12 11.14727693  
## hr\_X13 10.07038001  
## hr\_X14 7.19745344  
## hr\_X15 8.76189449  
## hr\_X16 19.42376443  
## hr\_X17 45.85421148  
## hr\_X18 40.59899044  
## hr\_X19 22.47746327  
## hr\_X20 9.19979320  
## hr\_X21 1.08905582  
## hr\_X22 -4.99338082  
## hr\_X23 -11.45041235  
## holiday\_NotHoliday 3.31675381  
## weekday\_Monday -1.63243394  
## weekday\_Saturday 1.49997767  
## weekday\_Sunday -2.69352756  
## weekday\_Thursday -0.65127025  
## weekday\_Tuesday -1.05182134  
## weekday\_Wednesday -0.24683507  
## workingday\_WorkingDay 2.09547343  
## weathersit\_LightPrecip -11.01783610  
## weathersit\_Misty 2.12229134  
## weathersit\_NoPrecip 4.38733640

**For ridge, I chose a lambda value here of 33. This gave us an R squared value of 0.6093. A decent amount of these variables appear to be not significant as they have slope coefficients close to zero. Some of the weekdays, hours and months fall into this category.**

recipe3 = recipe(count ~season+mnth+hr+holiday+weekday+workingday+weathersit+temp+atemp+hum+windspeed, bike)%>%  
 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(recipe3)  
  
lasso\_fit = fit(lasso\_wflow, bike)  
  
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 = .333)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 33.0112540  
## atemp 20.4054377  
## hum -22.7586783  
## windspeed -4.6153624  
## season\_Spring -7.7182163  
## season\_Summer -14.1181181  
## season\_Winter -22.9252792  
## mnth\_Aug -0.1234883  
## mnth\_Dec .   
## mnth\_Feb .   
## mnth\_Jan .   
## mnth\_Jul -7.0843432  
## mnth\_Jun -2.4018677  
## mnth\_Mar 1.9181302  
## mnth\_May 2.2399794  
## mnth\_Nov .   
## mnth\_Oct 5.2827883  
## mnth\_Sep 7.9146330  
## hr\_X1 -7.5087277  
## hr\_X2 -9.0305733  
## hr\_X3 -10.8615820  
## hr\_X4 -11.2121433  
## hr\_X5 -8.1404442  
## hr\_X6 2.8681672  
## hr\_X7 29.7136483  
## hr\_X8 57.4974481  
## hr\_X9 27.4440293  
## hr\_X10 15.9942998  
## hr\_X11 20.5748472  
## hr\_X12 28.0413682  
## hr\_X13 26.7854967  
## hr\_X14 23.4403253  
## hr\_X15 25.2868697  
## hr\_X16 37.8352741  
## hr\_X17 68.8925855  
## hr\_X18 62.6802106  
## hr\_X19 41.3381467  
## hr\_X20 25.7268294  
## hr\_X21 16.1292539  
## hr\_X22 8.9688133  
## hr\_X23 1.3819731  
## holiday\_NotHoliday 4.2166729  
## weekday\_Monday -1.4112067  
## weekday\_Saturday 0.3315789  
## weekday\_Sunday -4.1990631  
## weekday\_Thursday -0.2793349  
## weekday\_Tuesday -0.8528197  
## weekday\_Wednesday .   
## workingday\_WorkingDay .   
## weathersit\_LightPrecip -14.5409497  
## weathersit\_Misty .   
## weathersit\_NoPrecip 2.4246071

**For lasso, I chose a lambda value here of .333. This gave us an R squared value of 0.6313. Now we see multiple variables that are not significant zeroed out with slope coefficients of zero. I also can clearly see temperature, hour, and season having the strongest slope coefficients.**

What are the implications of the model results from the ridge and lasso methods?  
**I am leaning towards the use of the lasso model over the ridge model in this scenario. With the lasso model we were able to see December, February, January, November, Wednesdays, working days, and misty precipitation removed from the model and therefor cleaning up our variables so that we can focus on the significant ones. This looks to be very beneficial in locating the strongest predictor variables towards count of bikes rented to be temperature, seasons, and hour during the day.**