# Module 2 Assignment 2

## Madison Rauscher

#install.packages("tidyverse")  
#install.packages("tidymodels")  
#install.packages("glmnet")  
#install.packages("GGally")  
#install.packages("ggcorrplot")  
#install.packages("MASS")  
#install.packages("leaps")  
#install.packages("lmtest")  
#install.packages("splines")  
#install.packages("car")  
#install.packages("lubridate")  
  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.6 ✓ dplyr 1.0.3  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 0.1.2 ──

## ✓ broom 0.7.4 ✓ recipes 0.1.15  
## ✓ dials 0.0.9 ✓ rsample 0.0.8   
## ✓ infer 0.5.4 ✓ tune 0.1.2   
## ✓ modeldata 0.1.0 ✓ workflows 0.2.1   
## ✓ parsnip 0.1.5 ✓ yardstick 0.0.7

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.1

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(ggcorrplot)  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(leaps)  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(splines)  
library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

**Task 1**

bike = bike\_cleaned\_2\_ <- read\_csv("bike\_cleaned (2).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()  
## )

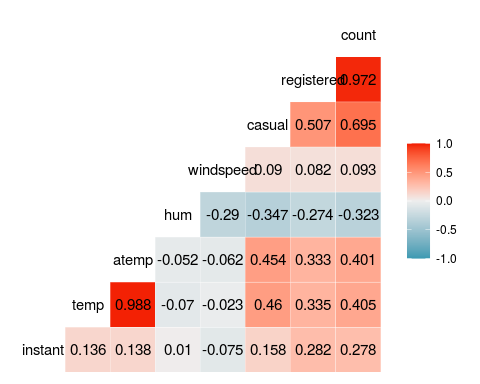
bike = bike %>% mutate(dteday = mdy(dteday))  
bike = bike %>% mutate\_if(is.character, as\_factor)  
bike = bike %>% mutate(hr=factor(hr))

We convert hr into a factor because it is qualitative not quantitative. The hours should be treated as levels to the factor.

**Task 2**

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

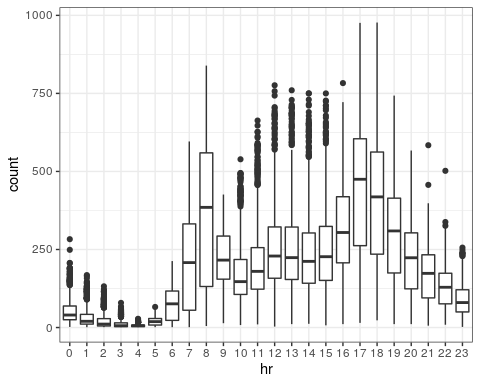
## Warning in ggcorr(bike, label = TRUE, label\_round = 3): data in column(s)  
## 'dteday', 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday',  
## 'weathersit' are not numeric and were ignored



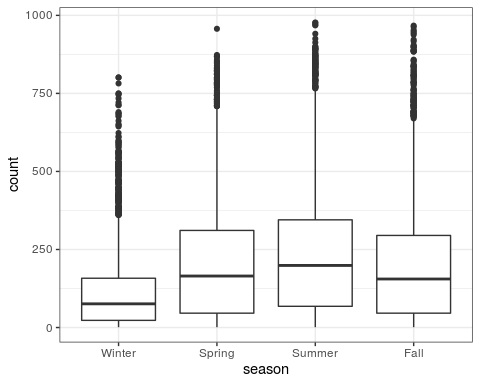
The variable Temp seems to be best correlated with count.

**Task 3**

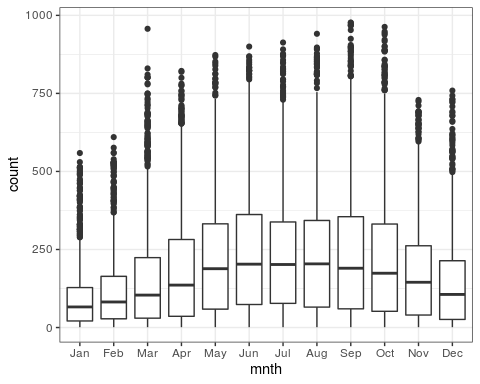
ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw()



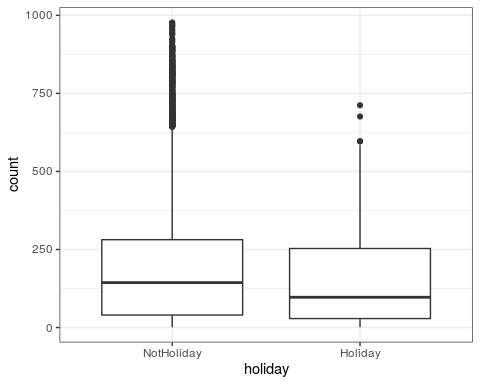
ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()



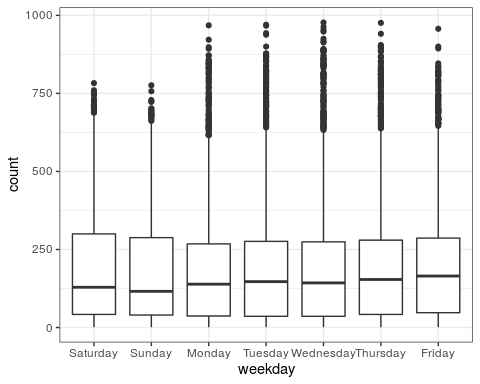
ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()



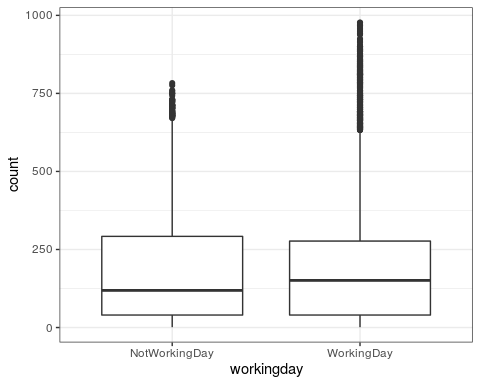
ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()



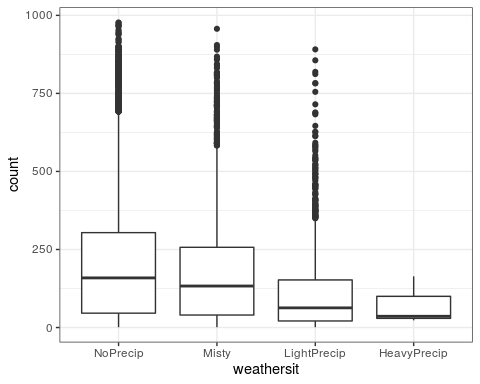
ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()



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



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



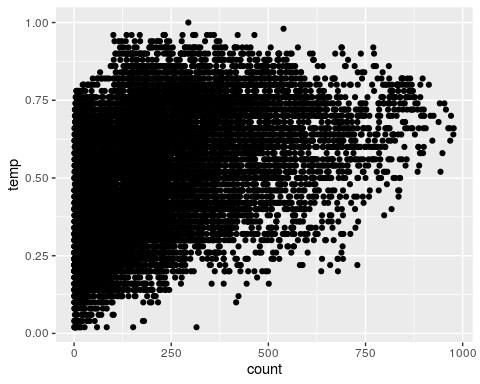
Hour does affect count because during normal business hours, interquartile range (IQR) is higher than overnight likely because this is when people are able to get their bikes cleaned. Season affects count because the IQR is higher in the summer and spring than the cooler seasons which makes sense as you would use your bike more when it is warmer. Month affects count because the warmer the month, the higher the IQR because people will want to ride more when it’s warmer. Weather affects count because the nicer the weather, the higher the IQR which makes sense because more people will use their bikes when the weather is nice and therefore need it cleaned. Holiday, weekday and working day doe not seem to affect count as the boxplots are pretty uniform within each variable. This is likely because people will ride their bike and therefore need it cleaned regardless what day it is; it is more dependent on weather.

**Task 4**

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

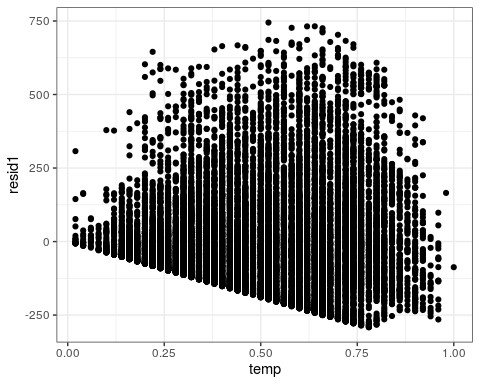
#Assumption 1  
ggplot(bike, aes(count, temp)) +  
 geom\_point()



#Assumption 2  
dwtest(lm\_fit$fit$fit$fit)

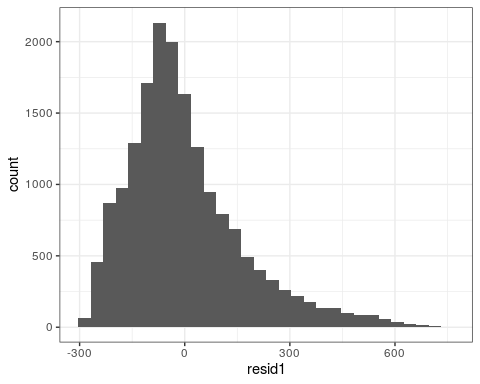
##   
## Durbin-Watson test  
##   
## data: lm\_fit$fit$fit$fit  
## DW = 0.3684, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0

#Assumption 3  
bike2 = bike %>% mutate(resid1=lm\_fit$fit$fit$fit$residuals)  
ggplot(bike2, aes(temp, resid1))+  
 geom\_point() +  
 theme\_bw()



#Assumption 4  
ggplot(bike2, aes(x=resid1)) +  
 geom\_histogram() +  
 theme\_bw()

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



When looking at the quality of the model when using the 4 assumptions, we find that it fails assumption 1 because it is not linear, it fails assumption 2 because the p-value is very small meaning the model errors are not independent, it may pass assumption 3 because the plot is slightly random, but it fails assumption 4 because the histogram is skewed to the right. Overall I would say this is not a valid model.

**Task 5**

bike5\_recipe = 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(bike5\_recipe)  
   
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 0.56 66900  
## 3 52 0.61 60950  
## 4 52 0.67 55540  
## 5 52 0.74 50600  
## 6 52 0.81 46110  
## 7 52 0.89 42010  
## 8 52 0.97 38280  
## 9 52 1.07 34880  
## 10 52 1.17 31780  
## 11 52 1.28 28960  
## 12 52 1.40 26390  
## 13 52 1.54 24040  
## 14 52 1.68 21910  
## 15 52 1.84 19960  
## 16 52 2.01 18190  
## 17 52 2.20 16570  
## 18 52 2.41 15100  
## 19 52 2.64 13760  
## 20 52 2.88 12540  
## 21 52 3.15 11420  
## 22 52 3.44 10410  
## 23 52 3.75 9482  
## 24 52 4.10 8640  
## 25 52 4.47 7872  
## 26 52 4.87 7173  
## 27 52 5.31 6536  
## 28 52 5.78 5955  
## 29 52 6.29 5426  
## 30 52 6.83 4944  
## 31 52 7.42 4505  
## 32 52 8.06 4105  
## 33 52 8.73 3740  
## 34 52 9.46 3408  
## 35 52 10.24 3105  
## 36 52 11.07 2829  
## 37 52 11.95 2578  
## 38 52 12.88 2349  
## 39 52 13.88 2140  
## 40 52 14.92 1950  
## 41 52 16.02 1777  
## 42 52 17.18 1619  
## 43 52 18.39 1475  
## 44 52 19.65 1344  
## 45 52 20.96 1225  
## 46 52 22.32 1116  
## 47 52 23.73 1017  
## 48 52 25.17 926  
## 49 52 26.65 844  
## 50 52 28.16 769  
## 51 52 29.70 701  
## 52 52 31.25 639  
## 53 52 32.82 582  
## 54 52 34.39 530  
## 55 52 35.96 483  
## 56 52 37.51 440  
## 57 52 39.06 401  
## 58 52 40.57 365  
## 59 52 42.06 333  
## 60 52 43.50 303  
## 61 52 44.90 276  
## 62 52 46.25 252  
## 63 52 47.55 230  
## 64 52 48.78 209  
## 65 52 49.95 190  
## 66 52 51.06 174  
## 67 52 52.10 158  
## 68 52 53.07 144  
## 69 52 53.97 131  
## 70 52 54.80 120  
## 71 52 55.57 109  
## 72 52 56.28 99  
## 73 52 56.92 91  
## 74 52 57.50 82  
## 75 52 58.03 75  
## 76 52 58.51 68  
## 77 52 58.94 62  
## 78 52 59.33 57  
## 79 52 59.68 52  
## 80 52 60.00 47  
## 81 52 60.28 43  
## 82 52 60.53 39  
## 83 52 60.76 36  
## 84 52 60.96 33  
## 85 52 61.15 30  
## 86 52 61.31 27  
## 87 52 61.47 25  
## 88 52 61.61 22  
## 89 52 61.73 20  
## 90 52 61.85 19  
## 91 52 61.96 17  
## 92 52 62.06 15  
## 93 52 62.16 14  
## 94 52 62.24 13  
## 95 52 62.33 12  
## 96 52 62.41 11  
## 97 52 62.48 10  
## 98 52 62.54 9  
## 99 52 62.61 8  
## 100 52 62.67 7

ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 39)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 26.4014627  
## atemp 26.0200068  
## hum -24.4309327  
## windspeed -1.1544426  
## season\_Spring 7.2367518  
## season\_Summer 1.7911095  
## season\_Fall 14.9709749  
## mnth\_Feb -2.7594548  
## mnth\_Mar 0.4497667  
## mnth\_Apr 0.7722786  
## mnth\_May 4.1952360  
## mnth\_Jun -0.2663428  
## mnth\_Jul -4.1975823  
## mnth\_Aug 1.6506364  
## mnth\_Sep 8.7524823  
## mnth\_Oct 6.4660137  
## mnth\_Nov 1.4020101  
## mnth\_Dec 0.2466391  
## hr\_X1 -19.3602530  
## hr\_X2 -20.5147950  
## hr\_X3 -21.8196890  
## hr\_X4 -22.1402992  
## hr\_X5 -19.7497078  
## hr\_X6 -10.2197671  
## hr\_X7 11.9351895  
## hr\_X8 34.9101445  
## hr\_X9 9.8898753  
## hr\_X10 0.3214413  
## hr\_X11 4.0673680  
## hr\_X12 10.1526779  
## hr\_X13 9.0588472  
## hr\_X14 6.2241894  
## hr\_X15 7.7373617  
## hr\_X16 18.1384805  
## hr\_X17 43.9034689  
## hr\_X18 38.8411432  
## hr\_X19 21.2639929  
## hr\_X20 8.3934054  
## hr\_X21 0.5501699  
## hr\_X22 -5.3322837  
## hr\_X23 -11.5943644  
## holiday\_Holiday -3.8049016  
## weekday\_Sunday -3.4257063  
## weekday\_Monday -1.3476228  
## weekday\_Tuesday -0.7900084  
## weekday\_Wednesday 0.0629562  
## weekday\_Thursday -0.2846677  
## weekday\_Friday 0.9724405  
## workingday\_WorkingDay 0.3456444  
## weathersit\_Misty -1.1987278  
## weathersit\_LightPrecip -12.4762421  
## weathersit\_HeavyPrecip -0.2771623

At the lambda that I chose, the R squared value is .6053. This lambda drives the beta values towards 0. The coefficients seem to be accurate as they show temp and a temp being highly correlated and some of the early morning hours to be negatively correlated.

**Task 6**

bike5\_recipe = 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(bike5\_recipe)  
   
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 13 43.19 19.960  
## 16 14 45.32 18.190  
## 17 15 47.30 16.570  
## 18 15 49.05 15.100  
## 19 16 50.59 13.760  
## 20 17 51.90 12.540  
## 21 18 53.13 11.420  
## 22 18 54.16 10.410  
## 23 19 55.02 9.482  
## 24 22 55.90 8.640  
## 25 23 56.68 7.872  
## 26 25 57.37 7.173  
## 27 26 58.00 6.536  
## 28 27 58.56 5.955  
## 29 27 59.04 5.426  
## 30 30 59.47 4.944  
## 31 31 59.86 4.505  
## 32 32 60.19 4.105  
## 33 32 60.51 3.740  
## 34 33 60.79 3.408  
## 35 33 61.02 3.105  
## 36 33 61.20 2.829  
## 37 34 61.37 2.578  
## 38 37 61.65 2.349  
## 39 37 61.86 2.140  
## 40 37 62.03 1.950  
## 41 38 62.16 1.777  
## 42 38 62.27 1.619  
## 43 38 62.37 1.475  
## 44 41 62.46 1.344  
## 45 41 62.58 1.225  
## 46 42 62.67 1.116  
## 47 42 62.76 1.017  
## 48 41 62.81 0.926  
## 49 42 62.86 0.844  
## 50 43 62.90 0.769  
## 51 43 62.94 0.701  
## 52 44 62.97 0.639  
## 53 43 63.01 0.582  
## 54 44 63.03 0.530  
## 55 44 63.05 0.483  
## 56 43 63.07 0.440  
## 57 44 63.09 0.401  
## 58 46 63.11 0.365  
## 59 47 63.13 0.333  
## 60 48 63.14 0.303  
## 61 48 63.15 0.276  
## 62 48 63.16 0.252  
## 63 48 63.17 0.230  
## 64 48 63.18 0.209  
## 65 48 63.19 0.190  
## 66 48 63.19 0.174  
## 67 49 63.20 0.158  
## 68 49 63.20 0.144  
## 69 49 63.20 0.131  
## 70 51 63.21 0.120  
## 71 51 63.21 0.109  
## 72 51 63.21 0.099  
## 73 51 63.21 0.091  
## 74 51 63.22 0.082  
## 75 51 63.22 0.075  
## 76 51 63.22 0.068  
## 77 51 63.22 0.062

lasso\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 3.105)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 31.9126047  
## atemp 27.6457675  
## hum -25.7584504  
## windspeed .   
## season\_Spring 5.4375122  
## season\_Summer .   
## season\_Fall 18.3002497  
## mnth\_Feb .   
## mnth\_Mar .   
## mnth\_Apr .   
## mnth\_May 1.2377517  
## mnth\_Jun .   
## mnth\_Jul -4.7840106  
## mnth\_Aug .   
## mnth\_Sep 5.7074472  
## mnth\_Oct 1.7540258  
## mnth\_Nov .   
## mnth\_Dec .   
## hr\_X1 -18.7857255  
## hr\_X2 -20.1114697  
## hr\_X3 -21.6550771  
## hr\_X4 -21.9817754  
## hr\_X5 -19.0066785  
## hr\_X6 -7.5296420  
## hr\_X7 13.1391741  
## hr\_X8 40.6713080  
## hr\_X9 10.2469100  
## hr\_X10 .   
## hr\_X11 2.7500663  
## hr\_X12 9.8840911  
## hr\_X13 8.4265699  
## hr\_X14 4.8969401  
## hr\_X15 6.6892403  
## hr\_X16 19.2729968  
## hr\_X17 50.4519538  
## hr\_X18 44.4853033  
## hr\_X19 23.5066361  
## hr\_X20 8.1672868  
## hr\_X21 .   
## hr\_X22 -2.1484687  
## hr\_X23 -9.6046253  
## holiday\_Holiday -1.6497092  
## weekday\_Sunday -0.6338193  
## weekday\_Monday .   
## weekday\_Tuesday .   
## weekday\_Wednesday .   
## weekday\_Thursday .   
## weekday\_Friday .   
## workingday\_WorkingDay .   
## weathersit\_Misty .   
## weathersit\_LightPrecip -11.0363483  
## weathersit\_HeavyPrecip .

At the lambda I chose the R squared is .6102 which is slightly higher than the ridge model. This model shows that some coefficients are now 0. It removed the coefficients that were so close to 0 that they should not be in the model and only the most correlated remain.

The implications of the ridge and lasso methods are that they reduce the likelihood of overfitting and multicollinearity. To do this, the models ad a penalty term which reduces the model complexity.Lasso goes so far as to drive out any variables that are extremely uncorrelated.