### Module 2 Assignment 2

### Tiffany Morris

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

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.1 ✓ dplyr 1.0.6  
## ✓ tidyr 1.1.3 ✓ 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 0.1.3 ──

## ✓ broom 0.7.6 ✓ rsample 0.1.0   
## ✓ dials 0.0.9 ✓ tune 0.1.5   
## ✓ infer 0.5.4 ✓ workflows 0.2.2   
## ✓ modeldata 0.1.0 ✓ workflowsets 0.0.2   
## ✓ parsnip 0.1.5 ✓ yardstick 0.0.8   
## ✓ recipes 0.1.16

## ── 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()  
## • Use tidymodels\_prefer() to resolve common conflicts.

library(GGally)

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

library(ggcorrplot)   
library(gridExtra)

##   
## Attaching package: 'gridExtra'

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

library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

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

## Loaded glmnet 4.1-1

library(MASS)

##   
## Attaching package: 'MASS'

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

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

library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

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

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()  
## )

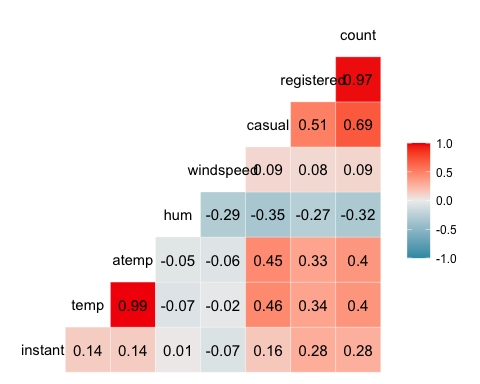
bike = bike%>%   
 mutate(dteday =mdy(dteday)) %>%  
 mutate\_if(is.character, as.factor) %>%  
 mutate(hr = as\_factor(hr))

**Why do we convert the “hr” variable into factor? Why not just leave as numbers?**

In order to make hr a categorical variable.

ggcorr(bike, label = "TRUE", label\_round = 2)

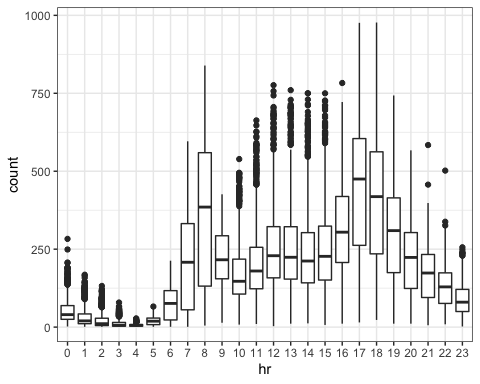
## Warning in ggcorr(bike, label = "TRUE", label\_round = 2): data in column(s)  
## 'dteday', 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday',  
## 'weathersit' are not numeric and were ignored



**Which of the quantitative variables appears to be best correlated with “count” (ignore the “registered”and “casual” variable as the sum of these two variables equals “count”)?**

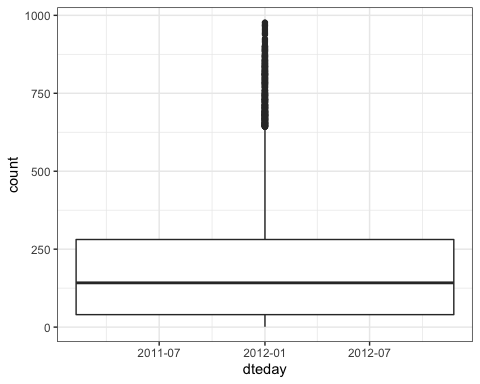
Atemp and temp are best correlated with count.

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

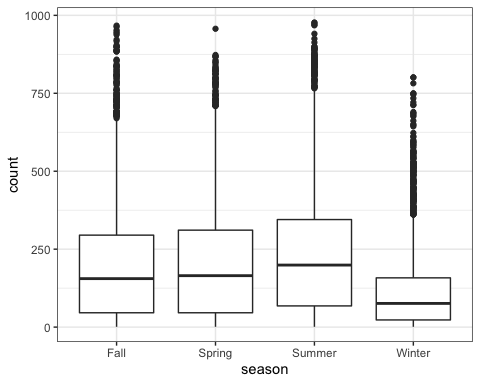


ggplot(bike,aes(x=dteday,y=count))+ geom\_boxplot()+ theme\_bw()

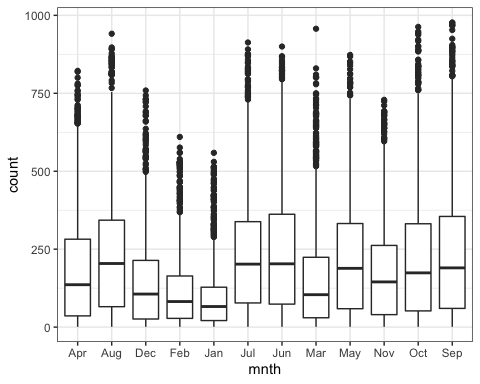
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?



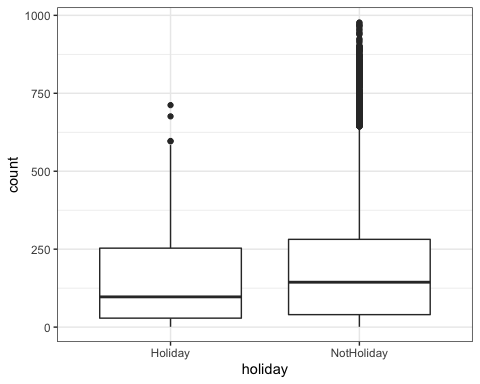
ggplot(bike,aes(x=season,y=count))+ geom\_boxplot()+ theme\_bw()



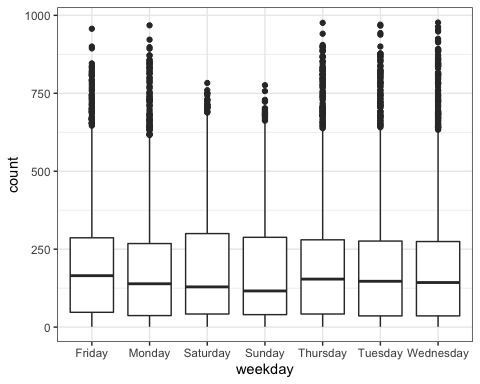
ggplot(bike,aes(x=mnth,y=count))+ geom\_boxplot()+ theme\_bw()



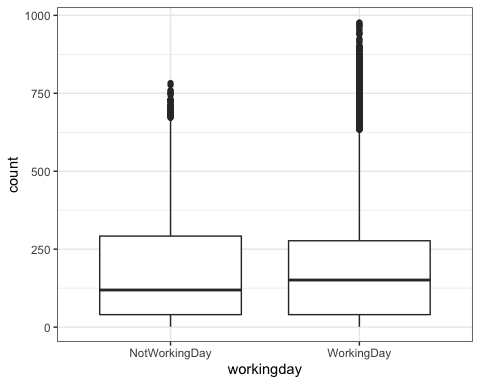
ggplot(bike,aes(x=holiday,y=count))+ geom\_boxplot()+ theme\_bw()



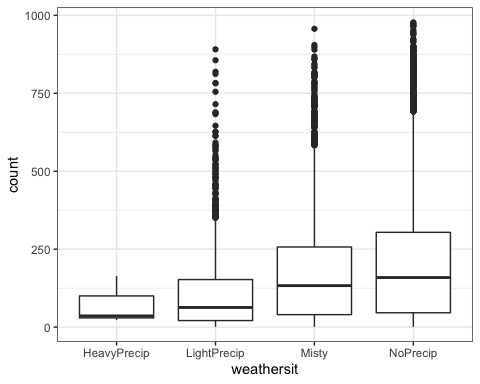
ggplot(bike,aes(x=weekday,y=count))+ geom\_boxplot()+ theme\_bw()



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



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



**Which variables appear to affect“count”? Provide a brief explanation as to why you believe that each variable does or does not affect “count”.**

Weathersit affects count, when there is no to little precipitation count is higher - likely no one wants to ride a bike in wet weather. Season affects count, in the Winter, count decreases - likely no one wants to ride a bike in cold weather. Holiday affects count, count is higher when there is not a holiday - likely no commuting on holidays so less bike usage. Weekday affects count, lower count on weekend days (saturday and sunday) - likely no commute on weekends. Workingday affects count, higher count on workingday - likely more commuting on a work day versus non work day. Mnth affects count, lower counts in cooler, colder months - likely due to weather. Dteday has no effect on count, likely because the combined view of counts is similar over time.

bike\_simple = recipe(count ~ hr, bike)  
  
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(bike\_simple)  
  
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   
## -446.45 -60.99 -6.01 50.10 551.49   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 53.898 4.756 11.332 < 2e-16 \*\*\*  
## hr1 -20.522 6.731 -3.049 0.002300 \*\*   
## hr2 -31.028 6.752 -4.595 4.35e-06 \*\*\*  
## hr3 -42.171 6.796 -6.205 5.58e-10 \*\*\*  
## hr4 -47.545 6.796 -6.996 2.73e-12 \*\*\*  
## hr5 -34.008 6.747 -5.040 4.70e-07 \*\*\*  
## hr6 22.146 6.729 3.291 0.000999 \*\*\*  
## hr7 158.167 6.724 23.523 < 2e-16 \*\*\*  
## hr8 305.113 6.724 45.377 < 2e-16 \*\*\*  
## hr9 165.411 6.724 24.600 < 2e-16 \*\*\*  
## hr10 119.770 6.724 17.812 < 2e-16 \*\*\*  
## hr11 154.245 6.724 22.939 < 2e-16 \*\*\*  
## hr12 199.418 6.722 29.668 < 2e-16 \*\*\*  
## hr13 199.763 6.719 29.729 < 2e-16 \*\*\*  
## hr14 187.051 6.719 27.838 < 2e-16 \*\*\*  
## hr15 197.335 6.719 29.368 < 2e-16 \*\*\*  
## hr16 258.085 6.717 38.422 < 2e-16 \*\*\*  
## hr17 407.554 6.717 60.674 < 2e-16 \*\*\*  
## hr18 371.613 6.722 55.286 < 2e-16 \*\*\*  
## hr19 257.625 6.722 38.327 < 2e-16 \*\*\*  
## hr20 172.132 6.722 25.608 < 2e-16 \*\*\*  
## hr21 118.416 6.722 17.617 < 2e-16 \*\*\*  
## hr22 77.437 6.722 11.520 < 2e-16 \*\*\*  
## hr23 33.933 6.722 5.048 4.50e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 128.2 on 17355 degrees of freedom  
## Multiple R-squared: 0.5015, Adjusted R-squared: 0.5008   
## F-statistic: 759.1 on 23 and 17355 DF, p-value: < 2.2e-16

**Comment on the quality of the model.**

When using hr as a predictor, the model appears to be decent quality with only knowing the R-squared value of 0.5015.

bike\_recipe = recipe(count ~. , bike) %>%  
 step\_rm(instant,dteday, registered, casual) %>%  
 step\_dummy(all\_nominal()) %>%   
 step\_center(all\_predictors()) %>% #centers the predictors  
 step\_scale(all\_predictors()) #scales the predictors  
  
ridge\_model = #give the model type a name   
 linear\_reg(mixture = 0) %>% #mixture = 1 sets up Lasso  
 set\_engine("glmnet") #specify the specify type of linear tool we want to use   
  
ridge\_wflow =   
 workflow() %>%   
 add\_model(ridge\_model) %>%   
 add\_recipe(bike\_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 0.61 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 = 12) #show the coefficients for our selected lambda value

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 27.3259525  
## atemp 25.5119756  
## hum -24.3005321  
## windspeed -3.4171757  
## season\_Spring -4.2241260  
## season\_Summer -9.3550826  
## season\_Winter -18.9064808  
## mnth\_Aug -0.6001469  
## mnth\_Dec 1.5439404  
## mnth\_Feb -0.2911153  
## mnth\_Jan -0.2730016  
## mnth\_Jul -7.2646798  
## mnth\_Jun -2.5075615  
## mnth\_Mar 1.7910975  
## mnth\_May 2.7801130  
## mnth\_Nov 1.9730844  
## mnth\_Oct 7.6253186  
## mnth\_Sep 8.2784035  
## hr\_X1 -16.9490129  
## hr\_X2 -18.3050471  
## hr\_X3 -19.8899283  
## hr\_X4 -20.2105744  
## hr\_X5 -17.4414885  
## hr\_X6 -6.5556585  
## hr\_X7 18.6335743  
## hr\_X8 44.7123577  
## hr\_X9 16.3822339  
## hr\_X10 5.5540068  
## hr\_X11 9.8056259  
## hr\_X12 16.7570302  
## hr\_X13 15.5314138  
## hr\_X14 12.3516080  
## hr\_X15 14.0791812  
## hr\_X16 25.8793192  
## hr\_X17 55.1048625  
## hr\_X18 49.3161731  
## hr\_X19 29.3161538  
## hr\_X20 14.6814303  
## hr\_X21 5.7233279  
## hr\_X22 -0.9693679  
## hr\_X23 -8.0744601  
## holiday\_NotHoliday 3.5397599  
## weekday\_Monday -2.1027275  
## weekday\_Saturday 1.6315274  
## weekday\_Sunday -3.0813484  
## weekday\_Thursday -1.1416716  
## weekday\_Tuesday -1.5990943  
## weekday\_Wednesday -0.6579325  
## workingday\_WorkingDay 2.3812919  
## weathersit\_LightPrecip -12.0867151  
## weathersit\_Misty 2.3732542  
## weathersit\_NoPrecip 4.7840387

**I chose a lamba value of 12, r-squared of .6232. This model appears to be a good model, particularly because I believe we are dealing with multicollinearity. The ridge model does a good job driving predictors to close to zero.**

bike\_recipe = recipe(count ~. , bike) %>%  
 step\_rm(instant,dteday, registered, casual) %>%  
 step\_dummy(all\_nominal()) %>%   
 step\_center(all\_predictors()) %>% #centers the predictors  
 step\_scale(all\_predictors()) #scales the predictors  
  
lasso\_model = #give the model type a name   
 linear\_reg(mixture = 1) %>% #mixture = 1 sets up Lasso  
 set\_engine("glmnet") #specify the specify type of linear tool we want to use   
  
lasso\_wflow =   
 workflow() %>%   
 add\_model(lasso\_model) %>%   
 add\_recipe(bike\_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 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 = 1.777) #show the coefficients for our selected lambda value

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.46308763  
## temp 22.27903015  
## atemp 28.04153354  
## hum -24.60137239  
## windspeed -1.73197338  
## season\_Spring -0.09891348  
## season\_Summer -6.24319562  
## season\_Winter -17.95028935  
## mnth\_Aug .   
## mnth\_Dec .   
## mnth\_Feb .   
## mnth\_Jan .   
## mnth\_Jul -6.05521550  
## mnth\_Jun -1.66808644  
## mnth\_Mar .   
## mnth\_May .   
## mnth\_Nov 0.78179817  
## mnth\_Oct 7.23934584  
## mnth\_Sep 7.07180282  
## hr\_X1 -15.39931065  
## hr\_X2 -16.85694487  
## hr\_X3 -18.55151550  
## hr\_X4 -18.93134812  
## hr\_X5 -15.97800156  
## hr\_X6 -4.44781830  
## hr\_X7 18.90735201  
## hr\_X8 46.63445080  
## hr\_X9 16.50246922  
## hr\_X10 5.02794837  
## hr\_X11 9.58292649  
## hr\_X12 16.97958369  
## hr\_X13 15.71370722  
## hr\_X14 12.33737616  
## hr\_X15 14.18734221  
## hr\_X16 26.74899180  
## hr\_X17 57.81187050  
## hr\_X18 51.64099059  
## hr\_X19 30.38856821  
## hr\_X20 14.84050804  
## hr\_X21 5.31947876  
## hr\_X22 .   
## hr\_X23 -5.97042336  
## holiday\_NotHoliday 2.92783717  
## weekday\_Monday .   
## weekday\_Saturday .   
## weekday\_Sunday -2.21660437  
## weekday\_Thursday .   
## weekday\_Tuesday .   
## weekday\_Wednesday .   
## workingday\_WorkingDay .   
## weathersit\_LightPrecip -13.31504663  
## weathersit\_Misty .   
## weathersit\_NoPrecip 0.25372579

**I chose a lamba value of 1.777, r-squared of .6213. This model appears to have driven 14 variables to zero and eliminating them as predictors. Those include workingday,weekday\_wednesday, weathersit\_misty, and mnth\_december,feb,jan, and november, and others.**