# BAN 502 - Module 2 Assignment 2

## Kristian Marlowe

### Multiple Linear Regression

First, we will load packages necessary for the assignment.

#install.packages("tidyverse","gridExtra","car", "glmnetn")  
library(tidyverse)

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

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.2 ✓ 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(gridExtra)

##   
## Attaching package: 'gridExtra'

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

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(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(lubridate)

##   
## Attaching package: 'lubridate'

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

#Task 1 - Read in data and tidy data

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))  
summary(bike)

## instant dteday season mnth   
## Min. : 1 Min. :2011-01-01 Length:17379 Length:17379   
## 1st Qu.: 4346 1st Qu.:2011-07-04 Class :character Class :character   
## Median : 8690 Median :2012-01-02 Mode :character Mode :character   
## Mean : 8690 Mean :2012-01-02   
## 3rd Qu.:13034 3rd Qu.:2012-07-02   
## Max. :17379 Max. :2012-12-31   
## hr holiday weekday workingday   
## Min. : 0.00 Length:17379 Length:17379 Length:17379   
## 1st Qu.: 6.00 Class :character Class :character Class :character   
## Median :12.00 Mode :character Mode :character Mode :character   
## Mean :11.55   
## 3rd Qu.:18.00   
## Max. :23.00   
## weathersit temp atemp hum   
## Length:17379 Min. :0.020 Min. :0.0000 Min. :0.0000   
## Class :character 1st Qu.:0.340 1st Qu.:0.3333 1st Qu.:0.4800   
## Mode :character Median :0.500 Median :0.4848 Median :0.6300   
## Mean :0.497 Mean :0.4758 Mean :0.6272   
## 3rd Qu.:0.660 3rd Qu.:0.6212 3rd Qu.:0.7800   
## Max. :1.000 Max. :1.0000 Max. :1.0000   
## windspeed casual registered count   
## Min. :0.0000 Min. : 0.00 Min. : 0.0 Min. : 1.0   
## 1st Qu.:0.1045 1st Qu.: 4.00 1st Qu.: 34.0 1st Qu.: 40.0   
## Median :0.1940 Median : 17.00 Median :115.0 Median :142.0   
## Mean :0.1901 Mean : 35.68 Mean :153.8 Mean :189.5   
## 3rd Qu.:0.2537 3rd Qu.: 48.00 3rd Qu.:220.0 3rd Qu.:281.0   
## Max. :0.8507 Max. :367.00 Max. :886.0 Max. :977.0

bike <- bike %>%  
 mutate\_if(sapply(bike, is.character), as.factor)  
  
bike = bike %>% mutate(hr = as.factor(hr))  
summary(bike)

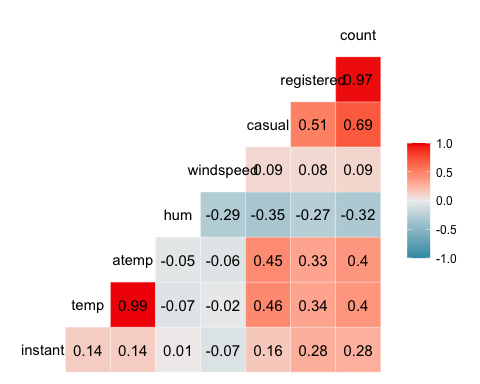
## instant dteday season mnth   
## Min. : 1 Min. :2011-01-01 Fall :4232 Jul :1488   
## 1st Qu.: 4346 1st Qu.:2011-07-04 Spring:4409 May :1488   
## Median : 8690 Median :2012-01-02 Summer:4496 Dec :1483   
## Mean : 8690 Mean :2012-01-02 Winter:4242 Aug :1475   
## 3rd Qu.:13034 3rd Qu.:2012-07-02 Mar :1473   
## Max. :17379 Max. :2012-12-31 Oct :1451   
## (Other):8521   
## hr holiday weekday workingday   
## 16 : 730 Holiday : 500 Friday :2487 NotWorkingDay: 5514   
## 17 : 730 NotHoliday:16879 Monday :2479 WorkingDay :11865   
## 13 : 729 Saturday :2512   
## 14 : 729 Sunday :2502   
## 15 : 729 Thursday :2471   
## 12 : 728 Tuesday :2453   
## (Other):13004 Wednesday:2475   
## weathersit temp atemp hum   
## HeavyPrecip: 3 Min. :0.020 Min. :0.0000 Min. :0.0000   
## LightPrecip: 1419 1st Qu.:0.340 1st Qu.:0.3333 1st Qu.:0.4800   
## Misty : 4544 Median :0.500 Median :0.4848 Median :0.6300   
## NoPrecip :11413 Mean :0.497 Mean :0.4758 Mean :0.6272   
## 3rd Qu.:0.660 3rd Qu.:0.6212 3rd Qu.:0.7800   
## Max. :1.000 Max. :1.0000 Max. :1.0000   
##   
## windspeed casual registered count   
## Min. :0.0000 Min. : 0.00 Min. : 0.0 Min. : 1.0   
## 1st Qu.:0.1045 1st Qu.: 4.00 1st Qu.: 34.0 1st Qu.: 40.0   
## Median :0.1940 Median : 17.00 Median :115.0 Median :142.0   
## Mean :0.1901 Mean : 35.68 Mean :153.8 Mean :189.5   
## 3rd Qu.:0.2537 3rd Qu.: 48.00 3rd Qu.:220.0 3rd Qu.:281.0   
## Max. :0.8507 Max. :367.00 Max. :886.0 Max. :977.0   
##

We convert the “hr” variable as a factor so that we can use it in statistical analysis. This will allow us to compare trends within the hours together to make assumptions about trends.

## Task 2 - Finding Correlation with “Count”

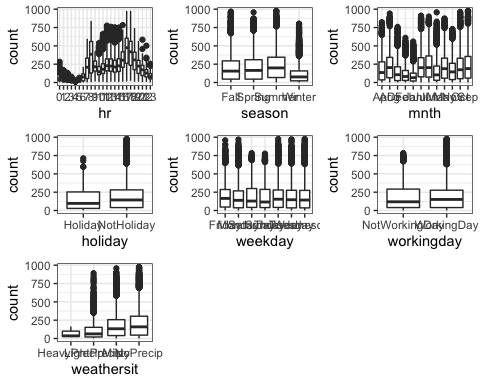
ggcorr(bike,label = TRUE,label\_round = 2)

## 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

 The variables most correlated with ‘Count’ seem to be atemp, temp, and hum.

## Task 3 - Visualization of Correlation

p1 = ggplot(bike, aes(x=hr, y=count))+geom\_boxplot() +theme\_bw()  
p2 = ggplot(bike, aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()  
p3 = ggplot(bike, aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()  
p4 = ggplot(bike, aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()  
p5 = ggplot(bike, aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()  
p6 = ggplot(bike, aes(x=workingday,y=count)) + geom\_boxplot() + theme\_bw()  
p7 = ggplot(bike, aes(x=weathersit,y=count)) + geom\_boxplot() + theme\_bw()  
grid.arrange(p1,p2,p3,p4,p5,p6,p7, ncol = 3)

 The variables that seem to have the most effect are ‘hr’, ‘mnth’, and possibly ‘weathersit’.The ‘hr’ variable I could see having an effect because you wouldn’t expect bikes being rented during the middle of the night. The month and weather could also cause effects on the count because the better the weather, the more people might want to rent bikes and vice versa. Depending on the weather, could affect the month and the renting of bicycles. I would typically expect the winter months to have less renters as well. The variables like ‘holiday’ ‘season’ ‘weekday’ and ‘workday’ do not show much of a change in the medians or the spread of data, suggesting not much of an effect on the count variable. This could be due to different vacation times that people may want to rent bikes, or different reasons that people may want to bike to work or just for fun that don’t necessarily depend on the day or if it’s a holiday or workday or not.

## Task 4 - Baselike Analysis

recipe1 = recipe(count ~ hr, bike)  
  
lm\_model =   
 linear\_reg() %>%   
 set\_engine("lm") #  
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(recipe1)  
  
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

This model shows ‘hr’ is a fairly good predictor of count. The r-squared value is 0.5008, which is fairly good, all the different levels are significant and the p-value is less than an alpha of 0.05, suggesting a relationship between ‘hr’ and ‘count’.

## Task 5 - Ridge Models

bike\_recipe = recipe(count ~., bike) %>%   
 step\_rm("instant", "dteday", "registered", "casual")%>%  
 step\_dummy(all\_nominal())%>%  
 step\_center(all\_predictors()) %>%   
 step\_scale(all\_predictors())  
   
bike\_model =   
 linear\_reg(mixture = 0) %>%   
 set\_engine("glmnet")   
  
ridge\_wflow =   
 workflow() %>%   
 add\_model(bike\_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

I would say here we could use ~ 14 as our lambda for a r-squared value of 0.6214.

ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%   
 coef(s = 14)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 26.9322080  
## atemp 25.4374336  
## hum -24.3872098  
## windspeed -3.2118848  
## season\_Spring -3.8063116  
## season\_Summer -8.8103923  
## season\_Winter -18.1844127  
## mnth\_Aug -0.5646545  
## mnth\_Dec 1.4802609  
## mnth\_Feb -0.6440979  
## mnth\_Jan -0.6685513  
## mnth\_Jul -7.1363370  
## mnth\_Jun -2.4317034  
## mnth\_Mar 1.5259885  
## mnth\_May 2.7779790  
## mnth\_Nov 2.0814212  
## mnth\_Oct 7.7618270  
## mnth\_Sep 8.2612097  
## hr\_X1 -17.5986546  
## hr\_X2 -18.9371231  
## hr\_X3 -20.4962452  
## hr\_X4 -20.8177236  
## hr\_X5 -18.0875988  
## hr\_X6 -7.3172954  
## hr\_X7 17.6208006  
## hr\_X8 43.4454231  
## hr\_X9 15.3962179  
## hr\_X10 4.6785524  
## hr\_X11 8.8938068  
## hr\_X12 15.7777024  
## hr\_X13 14.5664533  
## hr\_X14 11.4178830  
## hr\_X15 13.1284064  
## hr\_X16 24.8101206  
## hr\_X17 53.7451944  
## hr\_X18 48.0139259  
## hr\_X19 28.2123198  
## hr\_X20 13.7209215  
## hr\_X21 4.8533726  
## hr\_X22 -1.7743971  
## hr\_X23 -8.8109533  
## holiday\_NotHoliday 3.5275585  
## weekday\_Monday -2.0369288  
## weekday\_Saturday 1.6220092  
## weekday\_Sunday -3.0321669  
## weekday\_Thursday -1.0770015  
## weekday\_Tuesday -1.5247793  
## weekday\_Wednesday -0.5988896  
## workingday\_WorkingDay 2.3455297  
## weathersit\_LightPrecip -11.9614909  
## weathersit\_Misty 2.3532329  
## weathersit\_NoPrecip 4.7228610

Overall, this model is a fairly good fit. We can notice the coefficients ‘make sense’ based on the type of weather, month of the year, and hours. We can see there are a variety of factors included such as temp, atemp, weathersit, weekday, hr, holiday, month, season, windspeed and humidity.

## Task 6 Lasso Prediction

bike\_recipe = recipe(count ~., bike) %>%   
 step\_rm("instant", "dteday", "registered", "casual")%>%  
 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\_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 =0.530 )

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 31.1582266  
## atemp 21.7151111  
## hum -22.9820579  
## windspeed -4.2012120  
## season\_Spring -6.6522176  
## season\_Summer -13.1132449  
## season\_Winter -22.2812337  
## mnth\_Aug .   
## mnth\_Dec .   
## mnth\_Feb .   
## mnth\_Jan .   
## mnth\_Jul -6.8696529  
## mnth\_Jun -2.2781092  
## mnth\_Mar 1.5318945  
## mnth\_May 1.8887890  
## mnth\_Nov .   
## mnth\_Oct 5.5263066  
## mnth\_Sep 7.8346026  
## hr\_X1 -9.1285997  
## hr\_X2 -10.6399912  
## hr\_X3 -12.4450348  
## hr\_X4 -12.8029510  
## hr\_X5 -9.7545983  
## hr\_X6 0.8574517  
## hr\_X7 27.6918368  
## hr\_X8 55.4701412  
## hr\_X9 25.4097841  
## hr\_X10 13.9612392  
## hr\_X11 18.5422758  
## hr\_X12 26.0020579  
## hr\_X13 24.7469422  
## hr\_X14 21.3991618  
## hr\_X15 23.2470540  
## hr\_X16 35.7966142  
## hr\_X17 66.8532906  
## hr\_X18 60.6448465  
## hr\_X19 39.3117530  
## hr\_X20 23.7064923  
## hr\_X21 14.1169226  
## hr\_X22 6.9598348  
## hr\_X23 .   
## holiday\_NotHoliday 4.0393743  
## weekday\_Monday -1.1020924  
## weekday\_Saturday 0.3422609  
## weekday\_Sunday -3.8224118  
## weekday\_Thursday .   
## weekday\_Tuesday -0.4719994  
## weekday\_Wednesday .   
## workingday\_WorkingDay .   
## weathersit\_LightPrecip -14.3841027  
## weathersit\_Misty .   
## weathersit\_NoPrecip 2.1298570

In the lasso model, I chose a Lambda value of 0.530 for an r-squared value of 0.6304 which suggests that the Lasso model is removing variables. This does improve the r-squared slightly from the Ridge model as well. We can further see this in our output of our slope values. We have variables not included such as hr\_X23, weekday\_Wednesday, weathersit\_Misty, and more. We can see the coefficients are changing as well. These two models are similar, however the Lasso model allows for less variables to be included for a little more simplified model in this instance. Both models do allow us to predict a count of bicycles rented based on certain conditions.