MOD 2 ASSIGNMENT 2

Zackery Bradley

9/1/2021

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v 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 --

## v broom 0.7.8 v rsample 0.1.0   
## v dials 0.0.9 v tune 0.1.6   
## v infer 1.0.0 v workflows 0.2.3   
## v modeldata 0.1.1 v workflowsets 0.1.0   
## v parsnip 0.1.5 v yardstick 0.0.8   
## v 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.

#install.packages("glmnet")  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

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

## Loaded glmnet 4.1-2

library(GGally)

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

#install.packages("ggcorrplot")  
library(ggcorrplot)  
library(MASS)

##   
## Attaching package: 'MASS'

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

#install.packages("car")  
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

library(readr)  
#install.packages("devtools")  
library(devtools)

## Loading required package: usethis

##   
## Attaching package: 'devtools'

## The following object is masked from 'package:recipes':  
##   
## check

#install\_version("parsnip", version = "0.1.5", repos = "http://cran.us.r-project.org")  
 library(parsnip)  
bike\_cleaned <- read\_csv("C:/Users/bradl/OneDrive/Desktop/BAN502/Module 2/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()  
## )

library(tidyverse) library(tidymodels) install.packages(“glmnet”) library(glmnet) library(GGally) install.packages(“ggcorrplot”) library(ggcorrplot) library(MASS) install.packages(“car”) library(car) library(lubridate) library(lmtest) library(readr) install.packages(“devtools”) library(devtools) install\_version(“parsnip”, version = “0.1.5”, repos = “<http://cran.us.r-project.org>”) library(parsnip) bike\_cleaned <- read\_csv(“C:/Users/bradl/OneDrive/Desktop/BAN502/Module 2/bike\_cleaned.csv”)

**Task 1**

bike\_cleaned <- read\_csv("C:/Users/bradl/OneDrive/Desktop/BAN502/Module 2/bike\_cleaned.csv")

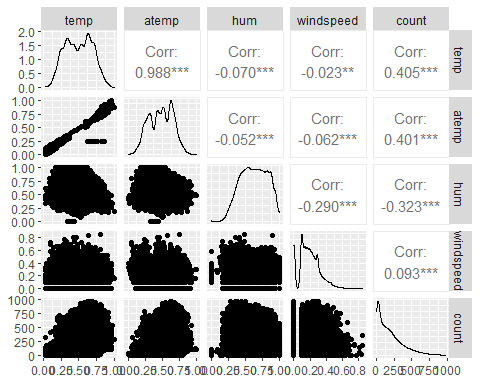
##   
## -- Column specification --------------------------------------------------------  
## cols(  
## instant = col\_double(),  
## dteday = col\_character(),  
## season = col\_character(),  
## mnth = col\_character(),  
## hr = col\_double(),  
## holiday = col\_character(),  
## weekday = col\_character(),  
## 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\_cleaned %>% mutate(dteday =mdy(dteday))   
bike = bike %>% mutate(season = as\_factor(season))  
bike = bike %>% mutate(mnth = as\_factor(mnth))  
bike = bike %>% mutate(holiday = as\_factor(holiday))  
bike = bike %>% mutate(weekday = as\_factor(weekday))  
bike = bike %>% mutate(workingday = as\_factor(workingday))  
bike = bike %>% mutate(weathersit = as\_factor(weathersit))  
  
bike = bike %>% mutate(hr= as\_factor(hr))

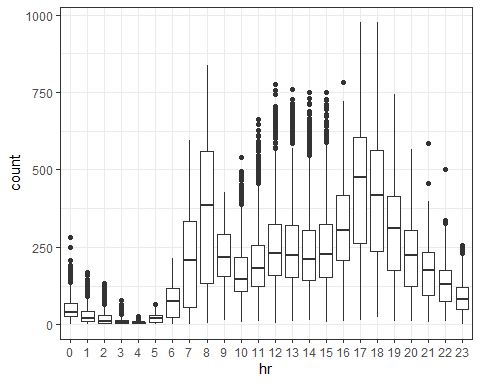
Factor variables can be either numeric or string variables. This makes them useful for statistical modeling.

**Task 2**

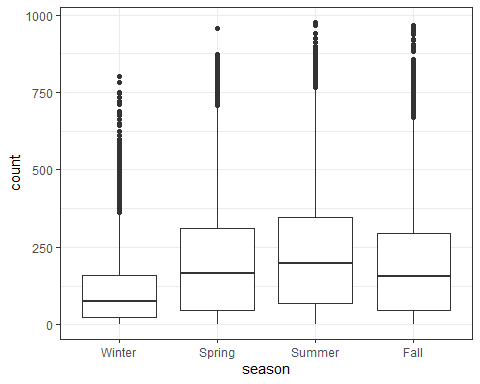
ggpairs(bike, columns = c(10:13, 16))

 from the results of the plot, the variable “temp” seems to be best correlated with count. **Task 3**

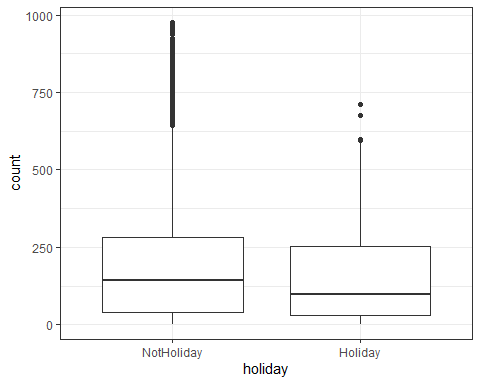
ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw()



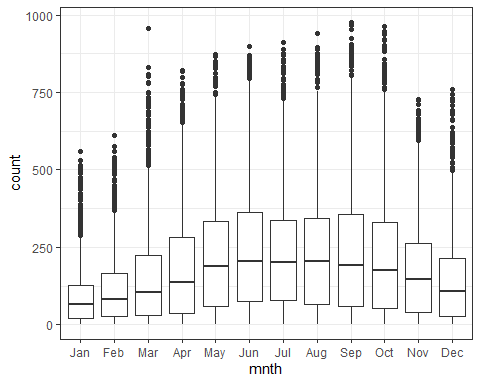
ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()



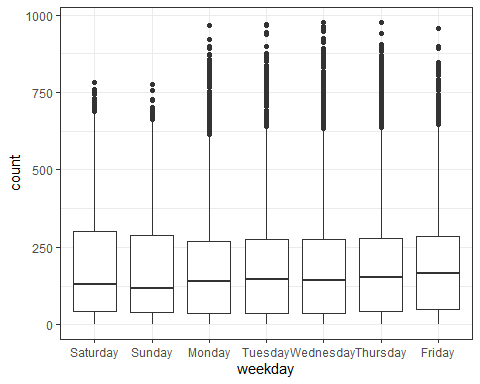
ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()



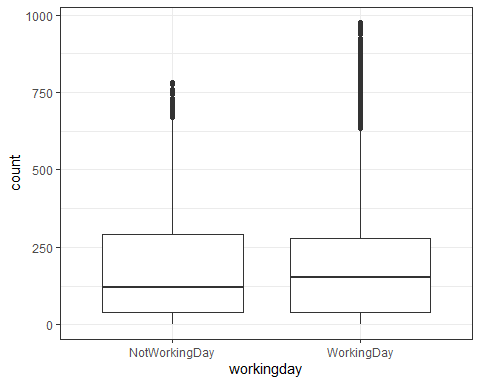
ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()



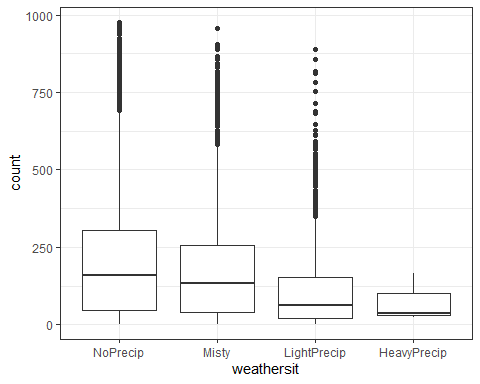
ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()



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



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

 hr hr does have an affect on count as we can see that the counts are typically higher during midday season season does have an affect on count, as we can see that count is typically higher in warmer seasons holiday while we can see than “non-holidays” tend to have higher sales numbers, both relationships have around the same quartiles. holiday does not have a significant affect on count. mnth mnth has an affect on count. we can see that count is usually higher in the warmer months. weekday while counts are usually higher mid week, the quartiles seem to remain steady throughout the week. weekday does not have an affect on count. workingday the quartiles show that there is no real affect on workingday and count as there is not significant change throughout. weathersit weathersit does have an affect on count. count is typically higher on non-rainy days. **Task 4**

weathersit\_simple = recipe(count ~ weathersit, bike)   
#1st df, org df  
  
  
weathersitlm\_model=   
 #2nd df  
linear\_reg() %>%  
set\_engine("lm")  
  
weathersitlm\_wflow=   
 #3rd df  
workflow() %>%  
add\_model(weathersitlm\_model) %>%   
 #2nd df  
add\_recipe(weathersit\_simple)   
#1st df  
  
weathersitlm\_fit =   
#4th df  
fit(weathersitlm\_wflow,bike)   
#3rd df, org df  
  
summary(weathersitlm\_fit$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -203.87 -141.87 -45.17 89.13 781.83   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 204.869 1.680 121.969 <2e-16 \*\*\*  
## weathersitMisty -29.704 3.148 -9.437 <2e-16 \*\*\*  
## weathersitLightPrecip -93.290 5.051 -18.469 <2e-16 \*\*\*  
## weathersitHeavyPrecip -130.536 103.616 -1.260 0.208   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 179.4 on 17375 degrees of freedom  
## Multiple R-squared: 0.02149, Adjusted R-squared: 0.02132   
## F-statistic: 127.2 on 3 and 17375 DF, p-value: < 2.2e-16

#4th df

with the adjusted r-squared being .02, the model seems to be of good quality. **Task 5**

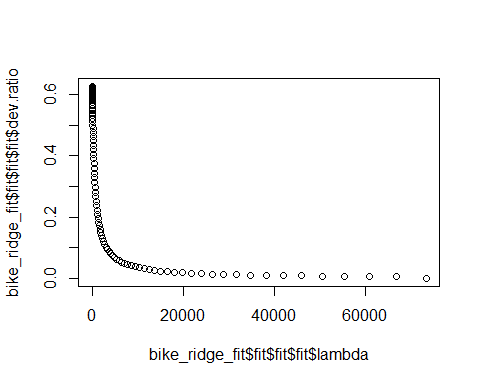
str(bike)

## spec\_tbl\_df [17,379 x 16] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ instant : num [1:17379] 1 2 3 4 5 6 7 8 9 10 ...  
## $ dteday : Date[1:17379], format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Winter","Spring",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ mnth : Factor w/ 12 levels "Jan","Feb","Mar",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ holiday : Factor w/ 2 levels "NotHoliday","Holiday": 1 1 1 1 1 1 1 1 1 1 ...  
## $ weekday : Factor w/ 7 levels "Saturday","Sunday",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ workingday: Factor w/ 2 levels "NotWorkingDay",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weathersit: Factor w/ 4 levels "NoPrecip","Misty",..: 1 1 1 1 1 2 1 1 1 1 ...  
## $ temp : num [1:17379] 0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...  
## $ atemp : num [1:17379] 0.288 0.273 0.273 0.288 0.288 ...  
## $ hum : num [1:17379] 0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...  
## $ windspeed : num [1:17379] 0 0 0 0 0 0.0896 0 0 0 0 ...  
## $ casual : num [1:17379] 3 8 5 3 0 0 2 1 1 8 ...  
## $ registered: num [1:17379] 13 32 27 10 1 1 0 2 7 6 ...  
## $ count : num [1:17379] 16 40 32 13 1 1 2 3 8 14 ...  
## - attr(\*, "spec")=  
## .. 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\_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())  
  
Bike\_Ridge\_Model=  
linear\_reg(mixture = 0)%>%  
set\_engine("glmnet")  
  
ridge\_wflow =  
workflow() %>%  
add\_model(Bike\_Ridge\_Model) %>%  
add\_recipe(bike\_recipe)   
  
bike\_ridge\_fit = fit(ridge\_wflow,bike)  
bike\_ridge\_fit

## == Workflow [trained] ==========================================================  
## Preprocessor: Recipe  
## Model: linear\_reg()  
##   
## -- Preprocessor ----------------------------------------------------------------  
## 3 Recipe Steps  
##   
## \* step\_dummy()  
## \* step\_center()  
## \* step\_scale()  
##   
## -- Model -----------------------------------------------------------------------  
##   
## 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  
##   
## ...  
## and 54 more lines.

plot(bike\_ridge\_fit$fit$fit$fit$lambda,bike\_ridge\_fit$fit$fit$fit$dev.ratio)



bike\_ridge\_fit %>%  
pull\_workflow\_fit() %>%  
pluck("fit")

## Warning: `pull\_workflow\_fit()` was deprecated in workflows 0.2.3.  
## Please use `extract\_fit\_parsnip()` instead.

##   
## 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

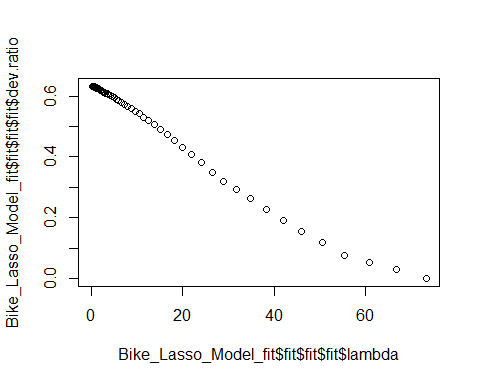
An appropriate Lambda would be 17 on an R squared value of .6196. This model, or alpha, we can see form the plot that this represents a good regularlization strength.

**Task 6**

bike\_recipe\_lasso=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())   
  
Bike\_Lasso\_Model=  
linear\_reg(mixture = 1) %>%  
set\_engine("glmnet")  
Bike\_Lasso\_Wflow =  
workflow() %>%  
add\_model(Bike\_Lasso\_Model) %>%  
add\_recipe(bike\_recipe\_lasso)  
  
Bike\_Lasso\_Model\_fit= fit(Bike\_Lasso\_Wflow,bike)  
Bike\_Lasso\_Model\_fit

## == Workflow [trained] ==========================================================  
## Preprocessor: Recipe  
## Model: linear\_reg()  
##   
## -- Preprocessor ----------------------------------------------------------------  
## 3 Recipe Steps  
##   
## \* step\_dummy()  
## \* step\_center()  
## \* step\_scale()  
##   
## -- Model -----------------------------------------------------------------------  
##   
## 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  
##   
## ...  
## and 31 more lines.

plot(Bike\_Lasso\_Model\_fit$fit$fit$fit$lambda,Bike\_Lasso\_Model\_fit$fit$fit$fit$dev.ratio)



Bike\_Lasso\_Model\_fit %>%  
pull\_workflow\_fit() %>%  
pluck("fit")

## Warning: `pull\_workflow\_fit()` was deprecated in workflows 0.2.3.  
## Please use `extract\_fit\_parsnip()` instead.

##   
## 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

bike\_ridge\_fit %>%  
pull\_workflow\_fit() %>%  
pluck("fit")

## Warning: `pull\_workflow\_fit()` was deprecated in workflows 0.2.3.  
## Please use `extract\_fit\_parsnip()` instead.

##   
## 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

An appropriate Lambda would be 19 on an R squared value of .6185. This model, or lamda, we can see form the plot that this represents a predictor of the model.