# Multiple Linear Regression Assignment

## Drew Rabil

### Module 2 - Assignment 2

library(tidyverse)

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

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.0 ✓ dplyr 1.0.5  
## ✓ 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(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(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(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

## Task 1

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)) #mdy is a lubridate package function  
bike <- bike %>% mutate\_if(is.character,as.factor) #convert remaining character variables to factors  
bike <- bike %>% mutate(hr = as\_factor(hr)) #convert the"hr" variable into a factor  
  
summary(bike) #statistical summary

## 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   
##

glimpse(bike) #neater format that hides the read\_csv attributes

## Rows: 17,379  
## Columns: 16  
## $ instant <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, …  
## $ dteday <date> 2011-01-01, 2011-01-01, 2011-01-01, 2011-01-01, 2011-01-01…  
## $ season <fct> Winter, Winter, Winter, Winter, Winter, Winter, Winter, Win…  
## $ mnth <fct> Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan,…  
## $ hr <fct> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1…  
## $ holiday <fct> NotHoliday, NotHoliday, NotHoliday, NotHoliday, NotHoliday,…  
## $ weekday <fct> Saturday, Saturday, Saturday, Saturday, Saturday, Saturday,…  
## $ workingday <fct> NotWorkingDay, NotWorkingDay, NotWorkingDay, NotWorkingDay,…  
## $ weathersit <fct> NoPrecip, NoPrecip, NoPrecip, NoPrecip, NoPrecip, Misty, No…  
## $ temp <dbl> 0.24, 0.22, 0.22, 0.24, 0.24, 0.24, 0.22, 0.20, 0.24, 0.32,…  
## $ atemp <dbl> 0.2879, 0.2727, 0.2727, 0.2879, 0.2879, 0.2576, 0.2727, 0.2…  
## $ hum <dbl> 0.81, 0.80, 0.80, 0.75, 0.75, 0.75, 0.80, 0.86, 0.75, 0.76,…  
## $ windspeed <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0896, 0.0000, 0.0…  
## $ casual <dbl> 3, 8, 5, 3, 0, 0, 2, 1, 1, 8, 12, 26, 29, 47, 35, 40, 41, 1…  
## $ registered <dbl> 13, 32, 27, 10, 1, 1, 0, 2, 7, 6, 24, 30, 55, 47, 71, 70, 5…  
## $ count <dbl> 16, 40, 32, 13, 1, 1, 2, 3, 8, 14, 36, 56, 84, 94, 106, 110…

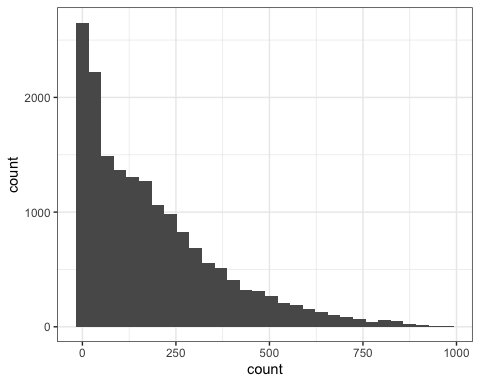
**Why do we convert the “hr” variable into factor? Why not just leave as numbers?**

Even though the “hr” variable is represented by numbers, these numbers represented the hour of the day from 0-24. The numbers are no value to them because they stand for a certain hour of the day (you cannot take the average of the hour of day, it does not make sense). Therefore, these numbers are actually, in reality, categorical values that need to be changed for my analysis.

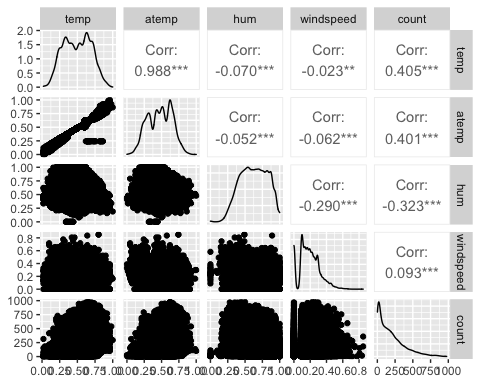
## Task 2

ggplot(bike, aes(x=count)) + geom\_histogram() + theme\_bw()

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



ggpairs(bike, columns = c("temp","atemp","hum","windspeed","count"))

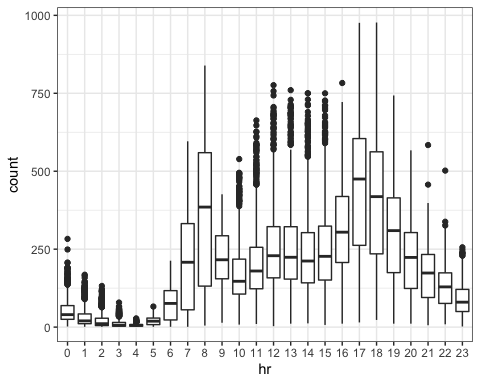


**Which of the quantitative variables appears to be best correlated with “count” (ignore the “registered” and “casual” variable as the sum of these two variables equals “count”)?**

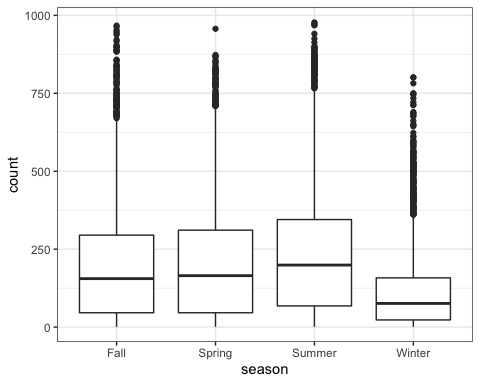
The temperature(“temp”) variable appears to be best correlated with “count” with a correlation of 0.405.

## Task 3

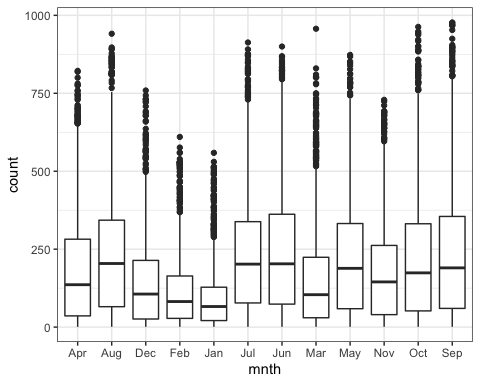
ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw()



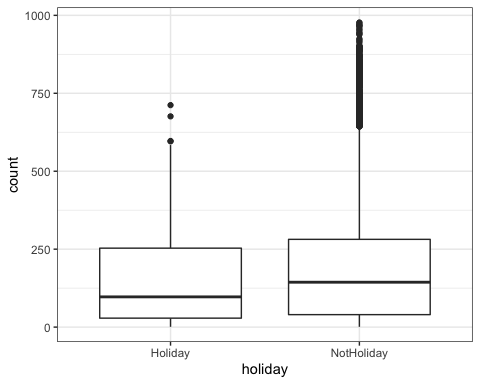
ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()



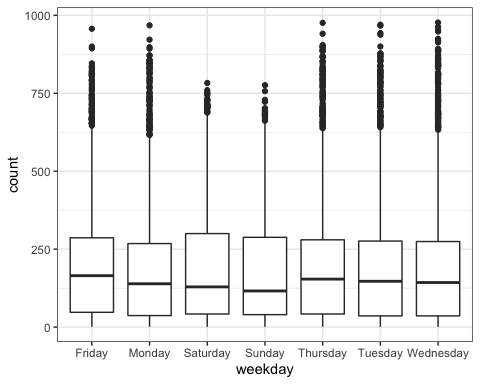
ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()



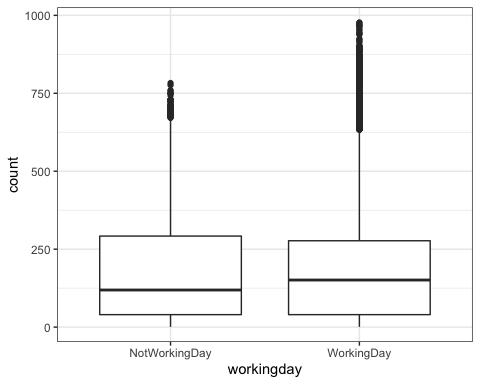
ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()



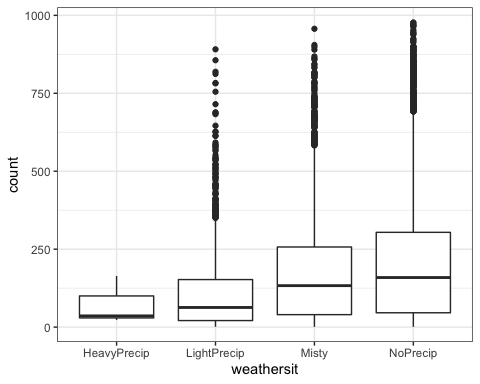
ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()



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



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



Based on the seven boxplots above for the categorical variables, it appears the variables that most affect “count” are hr, mnth, season, and weathersit. The workingday, weekday, and holiday variables appear to not affect “count”.

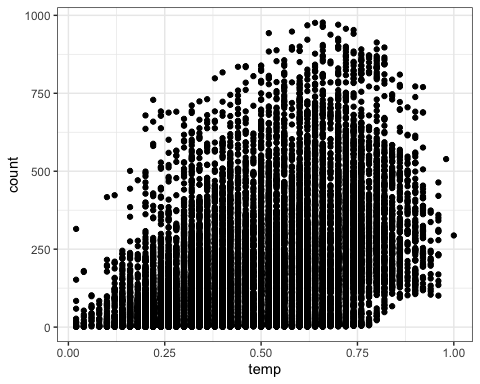
## Task 4

bike\_recipe = recipe(count ~ temp, bike)  
  
lm\_model = #name of the model type   
 linear\_reg() %>% #specify that I am doing a linear regression  
 set\_engine("lm") #specify the specify type of linear tool I want to use   
  
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

I chose the temperature variable (“temp”) as the best variable to build a model as the single predictor of “count”. I thought it was between the temp and the hr variables and decided to go with the temp variable. The summary of the model shows that the temp variable is of significance in terms of predicitng count because of its very small p-value. However the R-squared value was only 0.1638 causing me to reconsider which variable was actually the best predcitor for count.

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



## Task 5

bike\_recipe2 = recipe(count ~., bike) %>%   
 step\_rm(instant,dteday,registered,casual) %>% #excludes unnecessary variables from analysis  
 step\_dummy(all\_nominal()) %>% #makes hr categorical  
 step\_center(all\_predictors()) %>% #centers the predictors  
 step\_scale(all\_predictors()) #scales the predictors  
   
ridge\_model = #name of the model type   
 linear\_reg(mixture = 0) %>% #mixture = 0 for Ridge Regression  
 set\_engine("glmnet") #specify the specify type of linear tool I want to use   
  
ridge\_wflow =   
 workflow() %>%   
 add\_model(ridge\_model) %>%   
 add\_recipe(bike\_recipe2)  
  
ridge\_fit = fit(ridge\_wflow, bike)  
  
ridge\_fit

## ══ Workflow [trained] ══════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: linear\_reg()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 4 Recipe Steps  
##   
## • step\_rm()  
## • 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 0.61 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  
##   
## ...  
## and 54 more lines.

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 = 15) #show the coefficients for our selected lambda value

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 26.7532434  
## atemp 25.3879860  
## hum -24.4200853  
## windspeed -3.1130775  
## season\_Spring -3.6303396  
## season\_Summer -8.5673762  
## season\_Winter -17.8739281  
## mnth\_Aug -0.5483913  
## mnth\_Dec 1.4444342  
## mnth\_Feb -0.8017114  
## mnth\_Jan -0.8466135  
## mnth\_Jul -7.0744473  
## mnth\_Jun -2.3930242  
## mnth\_Mar 1.4075247  
## mnth\_May 2.7785951  
## mnth\_Nov 2.1194400  
## mnth\_Oct 7.8121707  
## mnth\_Sep 8.2468829  
## hr\_X1 -17.8480612  
## hr\_X2 -19.1785207  
## hr\_X3 -20.7260196  
## hr\_X4 -21.0480586  
## hr\_X5 -18.3362739  
## hr\_X6 -7.6224647  
## hr\_X7 17.1923218  
## hr\_X8 42.8921178  
## hr\_X9 14.9813035  
## hr\_X10 4.3182180  
## hr\_X11 8.5158162  
## hr\_X12 15.3667805  
## hr\_X13 14.1627022  
## hr\_X14 11.0295137  
## hr\_X15 12.7315731  
## hr\_X16 24.3549371  
## hr\_X17 53.1469698  
## hr\_X18 47.4435570  
## hr\_X19 27.7390318  
## hr\_X20 13.3177313  
## hr\_X21 4.4943000  
## hr\_X22 -2.1018385  
## hr\_X23 -9.1049346  
## holiday\_NotHoliday 3.5197764  
## weekday\_Monday -2.0069789  
## weekday\_Saturday 1.6165455  
## weekday\_Sunday -3.0094160  
## weekday\_Thursday -1.0470594  
## weekday\_Tuesday -1.4906040  
## weekday\_Wednesday -0.5720151  
## workingday\_WorkingDay 2.3285301  
## weathersit\_LightPrecip -11.9012596  
## weathersit\_Misty 2.3435435  
## weathersit\_NoPrecip 4.6966964

## Task 6

bike\_recipe2 = recipe(count ~., bike) %>%   
 step\_rm(instant,dteday,registered,casual) %>% #excludes unnecessary variables from analysis  
 step\_dummy(all\_nominal()) %>% #makes hr categorical   
 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 for a Lasso Regression  
 set\_engine("glmnet") #specify the specify type of linear tool I want to use   
  
lasso\_wflow =   
 workflow() %>%   
 add\_model(lasso\_model) %>%   
 add\_recipe(bike\_recipe2)  
  
lasso\_fit = fit(lasso\_wflow, bike)  
  
lasso\_fit

## ══ Workflow [trained] ══════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: linear\_reg()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 4 Recipe Steps  
##   
## • step\_rm()  
## • 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 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  
##   
## ...  
## and 35 more lines.

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 = 0.639) #show the coefficients for our selected lambda value

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 30.3378993  
## atemp 22.3038242  
## hum -23.1096346  
## windspeed -3.9787303  
## season\_Spring -6.0863412  
## season\_Summer -12.5459972  
## season\_Winter -21.9304192  
## mnth\_Aug .   
## mnth\_Dec .   
## mnth\_Feb .   
## mnth\_Jan .   
## mnth\_Jul -6.8095275  
## mnth\_Jun -2.2452217  
## mnth\_Mar 1.3146563  
## mnth\_May 1.6778683  
## mnth\_Nov .   
## mnth\_Oct 5.6399236  
## mnth\_Sep 7.7501898  
## hr\_X1 -9.7879921  
## hr\_X2 -11.2945608  
## hr\_X3 -13.0878522  
## hr\_X4 -13.4488932  
## hr\_X5 -10.4104307  
## hr\_X6 .   
## hr\_X7 26.8105904  
## hr\_X8 54.5846059  
## hr\_X9 24.5185661  
## hr\_X10 13.0685244  
## hr\_X11 17.6479942  
## hr\_X12 25.1027524  
## hr\_X13 23.8470407  
## hr\_X14 20.4970102  
## hr\_X15 22.3452510  
## hr\_X16 34.8957476  
## hr\_X17 65.9527172  
## hr\_X18 59.7471863  
## hr\_X19 38.4204126  
## hr\_X20 22.8197432  
## hr\_X21 13.2355494  
## hr\_X22 6.0808972  
## hr\_X23 -0.2790287  
## holiday\_NotHoliday 3.9438619  
## weekday\_Monday -0.9796971  
## weekday\_Saturday 0.2987204  
## weekday\_Sunday -3.6628941  
## weekday\_Thursday .   
## weekday\_Tuesday -0.3128158  
## weekday\_Wednesday .   
## workingday\_WorkingDay .   
## weathersit\_LightPrecip -14.3022873  
## weathersit\_Misty .   
## weathersit\_NoPrecip 1.9656521

**What are the implications of the model results from the ridge and lasso methods?**

The selected lambda for the lasso regression of 0.639 shows that some coefficients do in fact go to 0 and fallen out from the model. The ridge regression did bring the coefficients closer to 0, but none of them reached 0. These regressions allow us to have a simpler model to analyze.