# Jon Bailey

library(readr)  
library(lubridate)

##   
## Attaching package: 'lubridate'

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

library(tidyverse)

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

## ✓ ggplot2 3.3.2 ✓ dplyr 1.0.2  
## ✓ tibble 3.0.4 ✓ stringr 1.4.0  
## ✓ tidyr 1.1.2 ✓ forcats 0.5.0  
## ✓ purrr 0.3.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x lubridate::as.difftime() masks base::as.difftime()  
## x lubridate::date() masks base::date()  
## x dplyr::filter() masks stats::filter()  
## x lubridate::intersect() masks base::intersect()  
## x dplyr::lag() masks stats::lag()  
## x lubridate::setdiff() masks base::setdiff()  
## x lubridate::union() masks base::union()

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(GGally)

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

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(splines)  
library(MASS)

##   
## Attaching package: 'MASS'

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

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()  
## )

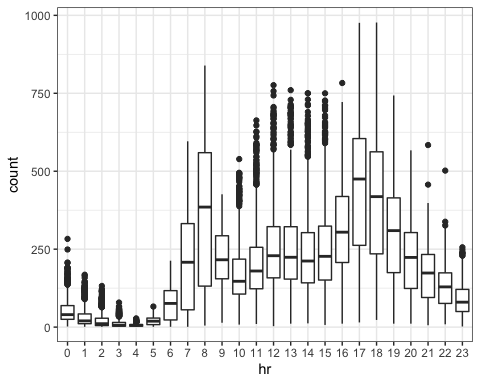
bike = bike %>% mutate(dteday = mdy(dteday))  
  
#mdy is a lubridate package function

bike$season <- as.factor(bike$season)  
bike$mnth <- as.factor(bike$mnth)  
bike$holiday <- as.factor(bike$holiday)  
bike$weekday <- as.factor(bike$weekday)  
bike$workingday <- as.factor(bike$workingday)  
bike$weathersit <- as.factor(bike$weathersit)  
  
bike$hr <- as.factor(bike$hr)

### We must change the hour of the day variables from numeric to factor to show that the numbers do not have the numeric value that they represent (ie hr 24 is not greater than hr 12, they are representative of times of the day).

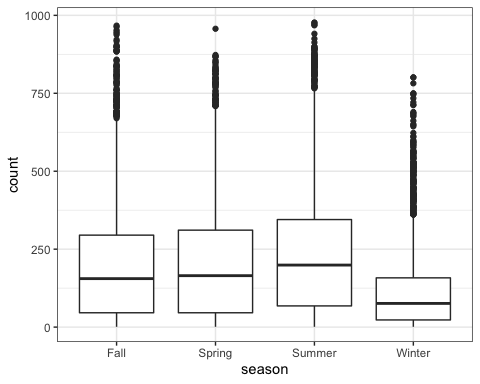
### The variable that appears to be the most correlated with the count variable would be “hr” and you can see that during the night hours the count will dip and be very low, and then during the day hours the count will be much higher.

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



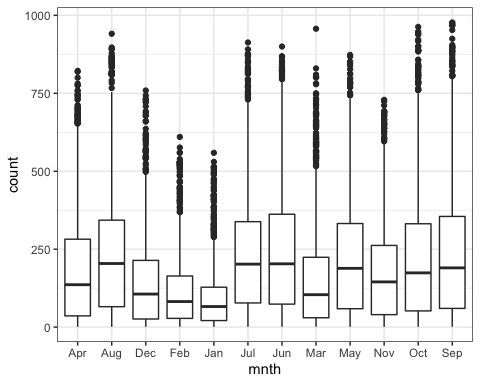
### There is an obvious correlation between hr and count, which can be shown that the hour of the day impacts the count of bikes.

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



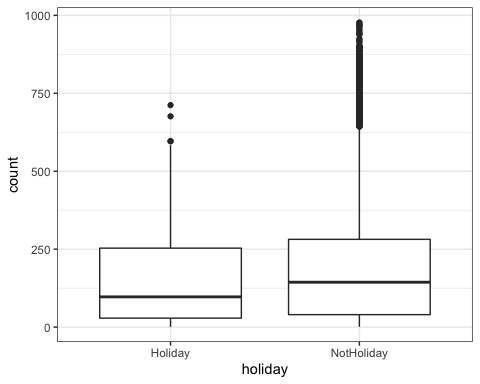
### All seasons are about equal with the count of bikes except for winter, which is shown to have a lower count, most likely due to the cold weather.

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



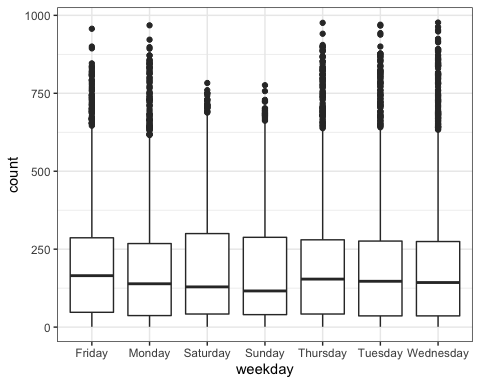
### January is the lowest count due to the cold weather. July and June are the highest months due to warmer weather, with May, August and September close behind.

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



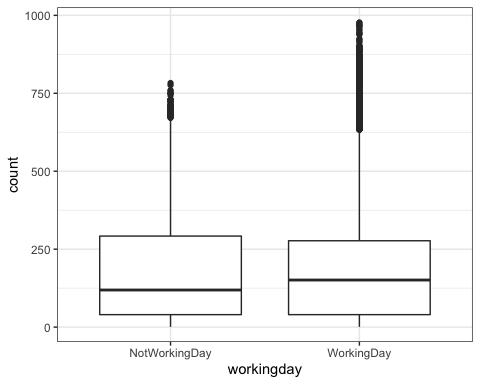
### Surprisingly, there are more bikers on NotHoliday than on Holiday, with a greater variability in the higher count values of the NotHoliday.

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



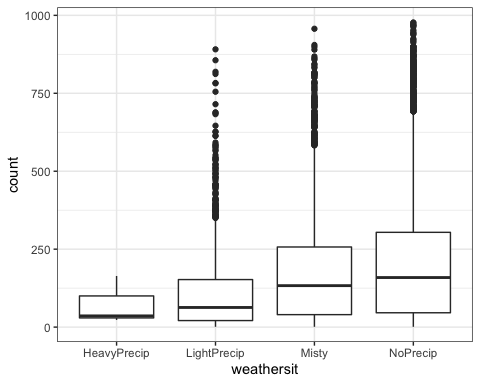
### Weekdays are fairly consistent with very little correlation, but Friday and Saturday are slightly above the others.

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



### There is very little difference between the count on NotWorkingDay and WorkingDay.

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



### There is a large difference in the count for the above 4 variables, and it is shown that the count is much higher on NoPrecip days than on days with different precipitation levels.

### The above model shows the hr variable (most correlated with the count variable) and shows each specific hour of the day with the estimated count of bikes at that specific hour. The temperature and precipitation values are also shown to have an impact on the number of bikers.

# Ridge Regression

bike2 = bike %>% dplyr::select("season", "mnth", "hr", "holiday", "weekday", "workingday", "weathersit", "temp", "atemp", "hum", "windspeed", "count")  
  
bike\_recipe = recipe(count ~ season + mnth + hr + holiday + weekday + workingday + weathersit + temp + atemp + hum + windspeed, data = bike2) %>%  
 step\_dummy(all\_nominal()) %>%   
 step\_center(all\_predictors()) %>%  
 step\_scale(all\_predictors())  
  
ridge\_model = linear\_reg(mixture = 0) %>% #mixture is 0 for ridge, 1 for lasso  
 set\_engine("glmnet")  
  
ridge\_wflow = workflow() %>%  
 add\_model(ridge\_model) %>%  
 add\_recipe(bike\_recipe)  
  
ridge\_fit = fit(ridge\_wflow, bike2)

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

### Based on the above Lambda values suggested, I will choose the value of 30.

ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 30)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 24.7608664  
## atemp 24.3401839  
## hum -24.4757351  
## windspeed -1.8375628  
## season\_Spring -1.9696496  
## season\_Summer -5.9637409  
## season\_Winter -14.8899184  
## mnth\_Aug -0.3121922  
## mnth\_Dec 0.9058286  
## mnth\_Feb -2.3854903  
## mnth\_Jan -2.6820662  
## mnth\_Jul -6.2059740  
## mnth\_Jun -1.8125541  
## mnth\_Mar 0.2075416  
## mnth\_May 2.8178202  
## mnth\_Nov 2.2777050  
## mnth\_Oct 8.0569833  
## mnth\_Sep 7.9205987  
## hr\_X1 -19.4424332  
## hr\_X2 -20.6748056  
## hr\_X3 -22.0851546  
## hr\_X4 -22.4146131  
## hr\_X5 -19.9327607  
## hr\_X6 -9.9921297  
## hr\_X7 13.1158327  
## hr\_X8 37.0862793  
## hr\_X9 11.1162855  
## hr\_X10 1.2248065  
## hr\_X11 5.1875749  
## hr\_X12 11.5928052  
## hr\_X13 10.4957075  
## hr\_X14 7.5819635  
## hr\_X15 9.1676935  
## hr\_X16 19.9787338  
## hr\_X17 46.7759457  
## hr\_X18 41.4520209  
## hr\_X19 23.0862085  
## hr\_X20 9.6319765  
## hr\_X21 1.4118615  
## hr\_X22 -4.7497397  
## hr\_X23 -11.2911010  
## holiday\_NotHoliday 3.3545925  
## weekday\_Monday -1.6789317  
## weekday\_Saturday 1.5208355  
## weekday\_Sunday -2.7367975  
## weekday\_Thursday -0.7025073  
## weekday\_Tuesday -1.1073966  
## weekday\_Wednesday -0.2854113  
## workingday\_WorkingDay 2.1271880  
## weathersit\_LightPrecip -11.1467697  
## weathersit\_Misty 2.1617532  
## weathersit\_NoPrecip 4.4239482

### The largest range of coefficient values is the hour of the day, with 5:00pm as the highest positive value for time. This makes sense, as maany bikers would want to go at this time of the day. The lowest time value is 4:00am, which makes sense as many bikers would not be out at this time.

### Another variable that has a high coefficient value is the temperature, which makes sense as to being a large factor in determining whether a biker will decide to go for a ride. Humidity is a somewhat important variable, as this could affect decisions of going for a ride.

### The weather variable shows that most bikers prefer no precipitation much more than light precipation, which shows that bikers do not like to be in the rain.

# Lasso Regression

bike\_recipe = recipe(count ~ season + mnth + hr + holiday + weekday + workingday + weathersit + temp + atemp + hum + windspeed, data = bike2) %>%  
 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(ridge\_model) %>%  
 add\_recipe(bike\_recipe)  
  
lasso\_fit = fit(ridge\_wflow,bike2)

lasso\_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

### From the above values, the lambda value of 30 looks to be a good judge for this dataset.

lasso\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 30)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## temp 24.7608664  
## atemp 24.3401839  
## hum -24.4757351  
## windspeed -1.8375628  
## season\_Spring -1.9696496  
## season\_Summer -5.9637409  
## season\_Winter -14.8899184  
## mnth\_Aug -0.3121922  
## mnth\_Dec 0.9058286  
## mnth\_Feb -2.3854903  
## mnth\_Jan -2.6820662  
## mnth\_Jul -6.2059740  
## mnth\_Jun -1.8125541  
## mnth\_Mar 0.2075416  
## mnth\_May 2.8178202  
## mnth\_Nov 2.2777050  
## mnth\_Oct 8.0569833  
## mnth\_Sep 7.9205987  
## hr\_X1 -19.4424332  
## hr\_X2 -20.6748056  
## hr\_X3 -22.0851546  
## hr\_X4 -22.4146131  
## hr\_X5 -19.9327607  
## hr\_X6 -9.9921297  
## hr\_X7 13.1158327  
## hr\_X8 37.0862793  
## hr\_X9 11.1162855  
## hr\_X10 1.2248065  
## hr\_X11 5.1875749  
## hr\_X12 11.5928052  
## hr\_X13 10.4957075  
## hr\_X14 7.5819635  
## hr\_X15 9.1676935  
## hr\_X16 19.9787338  
## hr\_X17 46.7759457  
## hr\_X18 41.4520209  
## hr\_X19 23.0862085  
## hr\_X20 9.6319765  
## hr\_X21 1.4118615  
## hr\_X22 -4.7497397  
## hr\_X23 -11.2911010  
## holiday\_NotHoliday 3.3545925  
## weekday\_Monday -1.6789317  
## weekday\_Saturday 1.5208355  
## weekday\_Sunday -2.7367975  
## weekday\_Thursday -0.7025073  
## weekday\_Tuesday -1.1073966  
## weekday\_Wednesday -0.2854113  
## workingday\_WorkingDay 2.1271880  
## weathersit\_LightPrecip -11.1467697  
## weathersit\_Misty 2.1617532  
## weathersit\_NoPrecip 4.4239482

### From the above dataset, two high values in terms of time are 8:00am and 5:00pm, which definitely makes sense, as these two values are separate from the workday and are good times of the day. A value that is not significant is holiday versus not holiday, which is somewhat surprising, as I thought that more bikers would be out on holidays when they have time.

### A value that is significant is the temperature value, as the temperature can greatly affect the decision of whether or not to go on a bike ride.

### A variable that does not show much significance is a specific day of the week to show higher numbers of bikers. This is surprising that the numbers remain fairly consistent, as I thought that the numbers would be higher on Saturdays and Sundays since people are not working.