## Assignment 2

### Underwood, Katie

#install.packages("glmnet")  
#install.packages("ggcorrplot")  
#install.packages("MASS")  
#install.packages("car")  
#install.packages("lubridate")  
#install.packages("lmtest")  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

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

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidymodels)

## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --

## v broom 0.7.2 v recipes 0.1.15  
## v dials 0.0.9 v rsample 0.0.8   
## v infer 0.5.4 v tune 0.1.2   
## v modeldata 0.1.0 v workflows 0.2.1   
## v parsnip 0.1.5 v 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(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

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

## Loaded glmnet 4.1

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

In this data set, hr = a time of day. This makes this value categorical rather than numerical, just like days of the week are categorical.

bike\_cleaned <- 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()  
## )

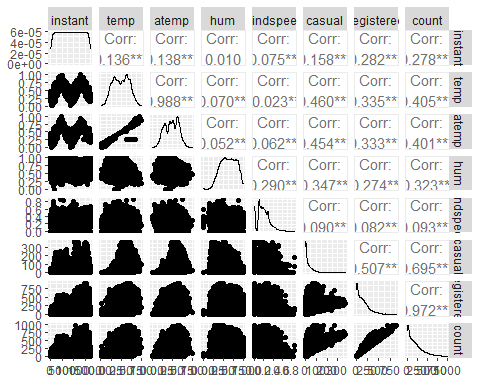
bike=bike\_cleaned  
#str(bike)  
#summary(bike)

bike = bike%>% 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))  
#str(bike)

## Task2

Temp is best correlated to Count

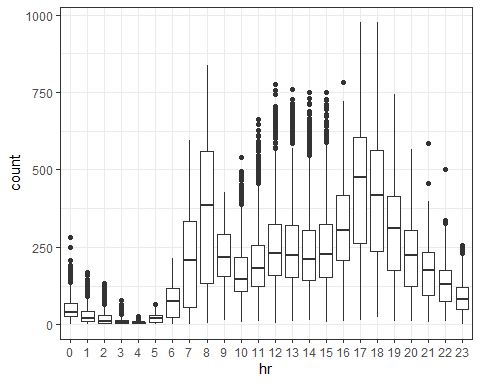
ggpairs(bike, columns = c(1,10:16))



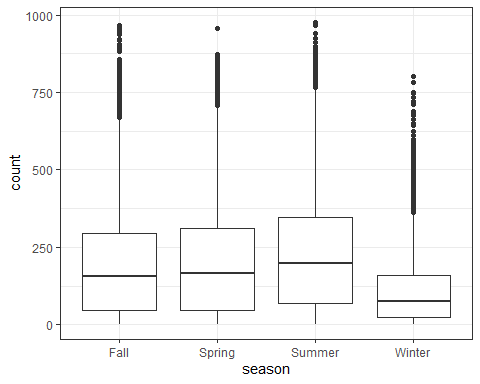
## Task3

hr does seem to impact count, in addition to month, and weathersit. This does make intuitive sense to me. Most people bike when it is warm and/or sunny which is what these three variables account for.

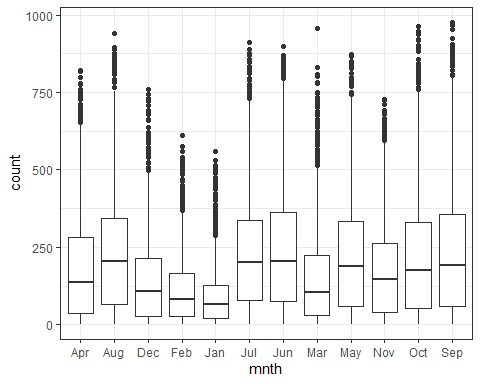
ggplot(bike,aes(x=hr,y=count))+ geom\_boxplot()+ theme\_bw()



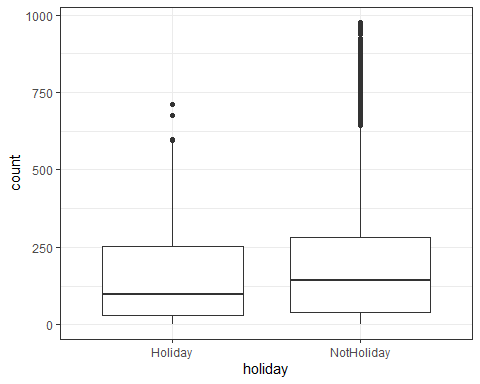
ggplot(bike,aes(x=season,y=count))+ geom\_boxplot()+ theme\_bw()



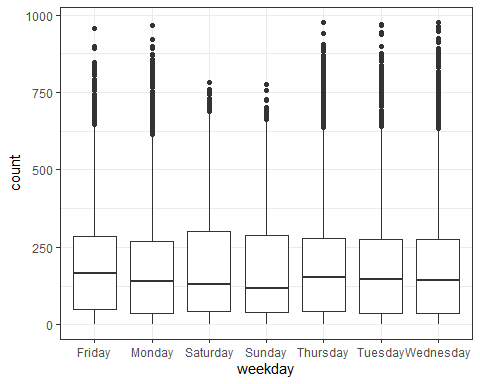
ggplot(bike,aes(x=mnth,y=count))+ geom\_boxplot()+ theme\_bw()



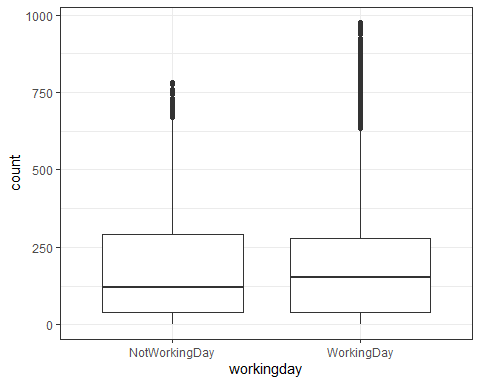
ggplot(bike,aes(x=holiday,y=count))+ geom\_boxplot()+ theme\_bw()



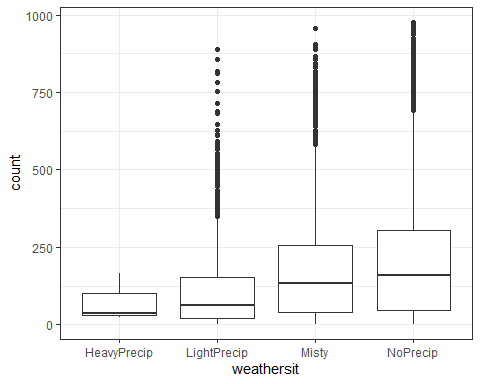
ggplot(bike,aes(x=weekday,y=count))+ geom\_boxplot()+ theme\_bw()



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



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



##Task4 I chose the hr variable to build the model. The R squared value around .5 is an “ok” value for this, not good not bad. Looking at the slopes for each hr value, makes sense to me. The highest values are 4-7pm, with the count being almost 0 during the middle of the night.

hr\_recipe = recipe(count ~ hr, bike) %>%  
 step\_dummy(hr)  
  
lm\_model = linear\_reg() %>%   
 set\_engine("lm")  
  
lm\_workflow = workflow() %>%   
 add\_model(lm\_model) %>%  
 add\_recipe(hr\_recipe)  
  
lm\_fit = fit(lm\_workflow, bike)  
  
#summary(lm\_fit$fit$fit$fit)

## Task 5

Lambda value of 131

count\_recipe = recipe(count ~ temp + atemp + hum + windspeed, bike) %>%  
 step\_dummy(all\_nominal()) %>%  
 step\_center(all\_predictors()) %>%  
 step\_scale(all\_predictors())  
  
ridge\_model =  
 linear\_reg(mixture = 0) %>%  
 set\_engine("glmnet")  
  
ridge\_wflow =  
 workflow() %>%  
 add\_model(ridge\_model) %>%  
 add\_recipe(count\_recipe)  
  
ridge\_fit = fit(ridge\_wflow, 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 4 0.00 73420  
## 2 4 0.24 66900  
## 3 4 0.26 60950  
## 4 4 0.28 55540  
## 5 4 0.31 50600  
## 6 4 0.34 46110  
## 7 4 0.37 42010  
## 8 4 0.41 38280  
## 9 4 0.45 34880  
## 10 4 0.49 31780  
## 11 4 0.54 28960  
## 12 4 0.59 26390  
## 13 4 0.65 24040  
## 14 4 0.71 21910  
## 15 4 0.78 19960  
## 16 4 0.85 18190  
## 17 4 0.93 16570  
## 18 4 1.02 15100  
## 19 4 1.11 13760  
## 20 4 1.22 12540  
## 21 4 1.33 11420  
## 22 4 1.46 10410  
## 23 4 1.59 9482  
## 24 4 1.74 8640  
## 25 4 1.90 7872  
## 26 4 2.07 7173  
## 27 4 2.26 6536  
## 28 4 2.46 5955  
## 29 4 2.68 5426  
## 30 4 2.91 4944  
## 31 4 3.17 4505  
## 32 4 3.45 4105  
## 33 4 3.74 3740  
## 34 4 4.06 3408  
## 35 4 4.40 3105  
## 36 4 4.76 2829  
## 37 4 5.15 2578  
## 38 4 5.56 2349  
## 39 4 6.00 2140  
## 40 4 6.46 1950  
## 41 4 6.95 1777  
## 42 4 7.46 1619  
## 43 4 8.00 1475  
## 44 4 8.56 1344  
## 45 4 9.14 1225  
## 46 4 9.74 1116  
##   
## ...  
## and 54 more lines.

ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 131)

## 5 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.463088  
## temp 26.298654  
## atemp 26.418792  
## hum -31.156434  
## windspeed 5.880701

## Task 6

Lambda chosen as 15.100 Using Lasso allowed me to realize that windspeed doesn’t have a significant enough effect to keep it in the model.

count\_recipe = recipe(count ~ temp + atemp + hum + windspeed, bike) %>%  
 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(count\_recipe)  
  
lasso\_fit = fit(lasso\_wflow, bike)  
  
lasso\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 15.100)

## 5 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.46309  
## temp 29.25552  
## atemp 26.64924  
## hum -40.04359  
## windspeed .