# Module 2 - Linear Regression

## Tate, Levi

### BAN 501

library(tidyverse)

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

## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ 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 ──

## ✓ broom 0.7.2 ✓ 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(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

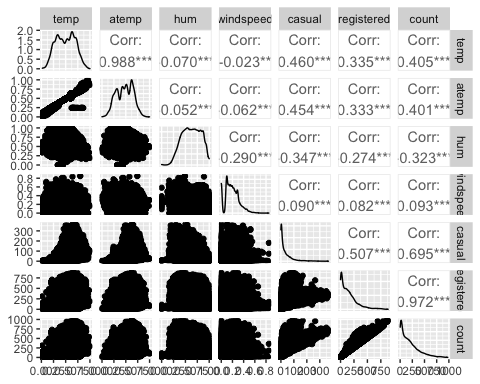
bike <- read\_csv("bike\_cleaned.csv")

##   
## ── 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))  
bike = mutate\_if(bike, is\_character, as\_factor)  
bike = bike %>% mutate(hr = as\_factor(hr))

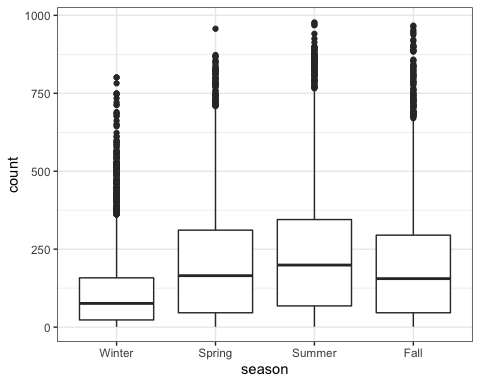
We convert the “hr” variable into factor because it has a limited number of different values(0-23).

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

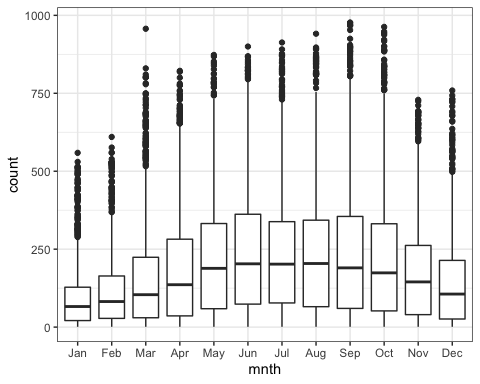


The quantitative variables temp and atemp appear to be the best correlated with count, .405 and .401 respectively.

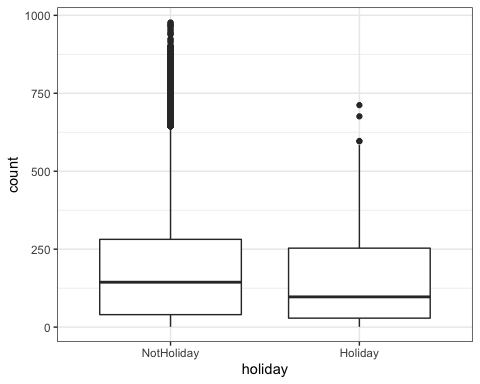
ggplot(bike,aes(x=season,y=count))+ geom\_boxplot()+ theme\_bw()



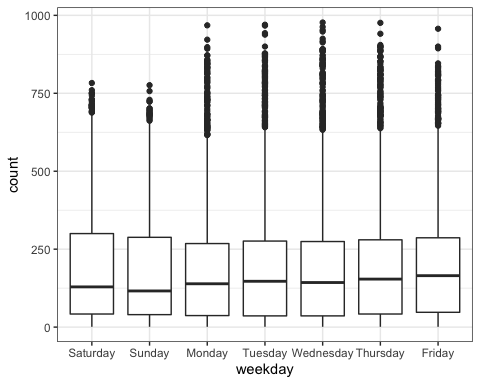
ggplot(bike,aes(x=mnth,y=count))+ geom\_boxplot()+ theme\_bw()



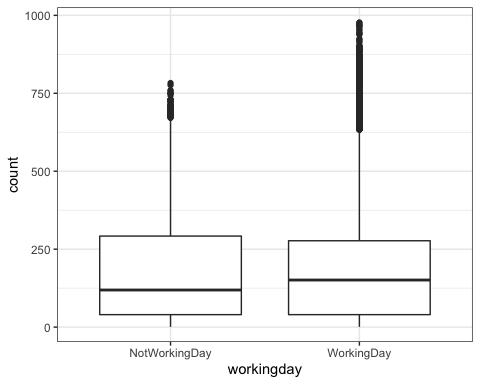
ggplot(bike,aes(x=holiday,y=count))+ geom\_boxplot()+ theme\_bw()



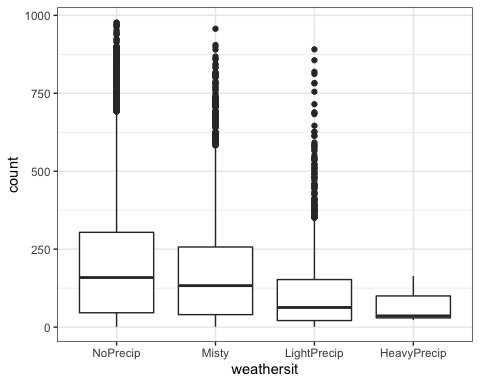
ggplot(bike,aes(x=weekday,y=count))+ geom\_boxplot()+ theme\_bw()



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



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



Season: Higher counts in the summer and spring for bike rentals.  
Mnth: In the months of summer and spring (April to late summer, early fall), bike rentals are at their best.  
Holiday: Non-Holidays have higher bike rentals compared to holidays.  
Weekday: Weekdays(Monday-Friday) have higher bike rental counts.  
Workingday: Working days are sightly higher in bike rentals when compared to Not working days.  
Weathersit: Bike rentals are affected by the weather. Higher counts with noPrecip and counts slowoy decrease as the weather worsens(Misty to HeavyPrecip).

WeatherAndBikes = recipe(count ~ weathersit, bike) %>%  
 step\_dummy(weathersit)  
  
lm\_model =   
 linear\_reg() %>%  
 set\_engine("lm")  
  
lm\_wflow =  
 workflow() %>%  
add\_model(lm\_model) %>%  
 add\_recipe(WeatherAndBikes)  
  
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   
## -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 \*\*\*  
## weathersit\_Misty -29.704 3.148 -9.437 <2e-16 \*\*\*  
## weathersit\_LightPrecip -93.290 5.051 -18.469 <2e-16 \*\*\*  
## weathersit\_HeavyPrecip -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

This model for predicting count based upon the weathersit variable has a very low R-squared value of .02149. The intercept, Misty, and light Precip are all significant, however Heavy Precip is not considered significant due to a higher p value than .05.

ridge\_bike\_recipe = recipe(count ~ ., bike) %>%  
 step\_rm(instant, dteday, registered, casual) %>%  
 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(ridge\_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 0.56 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

ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit")%>%  
 coef(s = 27)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.463087634  
## temp 27.248592633  
## atemp 26.550982055  
## hum -24.646475544  
## windspeed -1.981088347  
## season\_Spring 8.177779798  
## season\_Summer 2.139789567  
## season\_Fall 16.545291804  
## mnth\_Feb -2.150645949  
## mnth\_Mar 0.933499424  
## mnth\_Apr 0.835014271  
## mnth\_May 4.172096335  
## mnth\_Jun -0.639333574  
## mnth\_Jul -4.709769072  
## mnth\_Aug 1.561037105  
## mnth\_Sep 9.063841721  
## mnth\_Oct 6.370048221  
## mnth\_Nov 1.226748674  
## mnth\_Dec 0.406416266  
## hr\_X1 -19.211480242  
## hr\_X2 -20.433035317  
## hr\_X3 -21.830941001  
## hr\_X4 -22.147010073  
## hr\_X5 -19.603793820  
## hr\_X6 -9.520042091  
## hr\_X7 13.871918021  
## hr\_X8 38.103890428  
## hr\_X9 11.677055789  
## hr\_X10 1.551111353  
## hr\_X11 5.471272320  
## hr\_X12 11.881615590  
## hr\_X13 10.711431982  
## hr\_X14 7.717685812  
## hr\_X15 9.314937911  
## hr\_X16 20.302152998  
## hr\_X17 47.512045716  
## hr\_X18 42.173782712  
## hr\_X19 23.625752206  
## hr\_X20 10.052229841  
## hr\_X21 1.773313250  
## hr\_X22 -4.425598862  
## hr\_X23 -11.022725379  
## holiday\_Holiday -4.052734680  
## weekday\_Sunday -3.792841284  
## weekday\_Monday -1.438380893  
## weekday\_Tuesday -0.911112222  
## weekday\_Wednesday 0.003547205  
## weekday\_Thursday -0.400958006  
## weekday\_Friday 0.955642722  
## workingday\_WorkingDay 0.134925521  
## weathersit\_Misty -1.336793220  
## weathersit\_LightPrecip -13.202365519  
## weathersit\_HeavyPrecip -0.277643455

After using all variables to predict the count of bike rentals(excluding instant, dteday, registered, and casual) this ridge regression model has an R-squared value of .6131 when using a lambda value of 27.

lasso\_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(lasso\_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 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

lasso\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 0.365)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 1.894631e+02  
## temp 3.679226e+01  
## atemp 1.933714e+01  
## hum -2.282619e+01  
## windspeed -4.543234e+00  
## season\_Spring 1.268978e+01  
## season\_Summer 5.162601e+00  
## season\_Fall 2.283594e+01  
## mnth\_Feb .   
## mnth\_Mar 1.740166e+00  
## mnth\_Apr .   
## mnth\_May 2.446445e+00  
## mnth\_Jun -2.230590e+00  
## mnth\_Jul -6.643269e+00  
## mnth\_Aug .   
## mnth\_Sep 8.137791e+00  
## mnth\_Oct 4.173446e+00  
## mnth\_Nov -4.842442e-01  
## mnth\_Dec .   
## hr\_X1 -7.850716e+00  
## hr\_X2 -9.346168e+00  
## hr\_X3 -1.115147e+01  
## hr\_X4 -1.148130e+01  
## hr\_X5 -8.405388e+00  
## hr\_X6 2.539946e+00  
## hr\_X7 2.936241e+01  
## hr\_X8 5.709892e+01  
## hr\_X9 2.697726e+01  
## hr\_X10 1.545558e+01  
## hr\_X11 1.997065e+01  
## hr\_X12 2.738065e+01  
## hr\_X13 2.608317e+01  
## hr\_X14 2.270712e+01  
## hr\_X15 2.454325e+01  
## hr\_X16 3.710156e+01  
## hr\_X17 6.818983e+01  
## hr\_X18 6.201876e+01  
## hr\_X19 4.073406e+01  
## hr\_X20 2.517343e+01  
## hr\_X21 1.561842e+01  
## hr\_X22 8.489782e+00  
## hr\_X23 9.290610e-01  
## holiday\_Holiday -4.287721e+00  
## weekday\_Sunday -3.946562e+00  
## weekday\_Monday -1.157532e+00  
## weekday\_Tuesday -6.628867e-01  
## weekday\_Wednesday .   
## weekday\_Thursday -5.885086e-02  
## weekday\_Friday 8.549135e-01  
## workingday\_WorkingDay -9.966968e-04  
## weathersit\_Misty -2.023847e+00  
## weathersit\_LightPrecip -1.573573e+01  
## weathersit\_HeavyPrecip -2.251595e-02

As lambda gets smaller, it allows for more variables to appear in the model and explains more variance. When using a lambda value of 0.365 with the same variables from the ridge model, the R-squared value is .6311.

The implications of the model results from the ridge and lasso methods can be explained by the use of the penalty term for large slope coefficients. With the use of regularization methods, predicting the count of bike rentals will less likely be overfitted and these models could be applied to new data.