# Multiple Linear Regression and Special Issues

## Nicole Westrick

### BAN 502: Module 2

library(tidyverse)

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

## v ggplot2 3.3.3 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 0.7.3 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:**

library(readr)  
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()  
## )

View(bike)  
bike = bike %>%   
 mutate(dteday = mdy(dteday))  
bike = bike %>%  
 mutate\_if(sapply(bike, is.character), as.factor)  
bike = bike %>%  
 mutate(hr = as\_factor(hr))

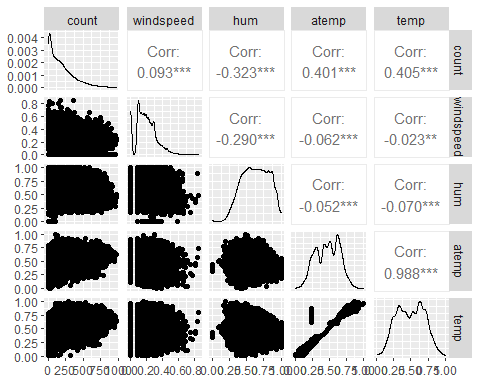
The variable “hr” is converted from number to factor so that the variable can be grouped for analysis.

**Task 2:**

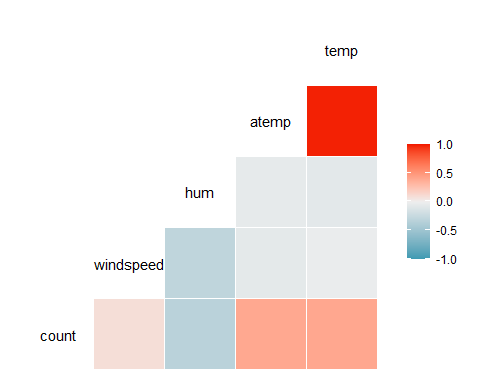
bike2 = bike %>%  
 dplyr::select("count", "windspeed", "hum", "atemp", "temp")  
summary(bike2)

## count windspeed hum atemp   
## Min. : 1.0 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.: 40.0 1st Qu.:0.1045 1st Qu.:0.4800 1st Qu.:0.3333   
## Median :142.0 Median :0.1940 Median :0.6300 Median :0.4848   
## Mean :189.5 Mean :0.1901 Mean :0.6272 Mean :0.4758   
## 3rd Qu.:281.0 3rd Qu.:0.2537 3rd Qu.:0.7800 3rd Qu.:0.6212   
## Max. :977.0 Max. :0.8507 Max. :1.0000 Max. :1.0000   
## temp   
## Min. :0.020   
## 1st Qu.:0.340   
## Median :0.500   
## Mean :0.497   
## 3rd Qu.:0.660   
## Max. :1.000

ggpairs(bike2)



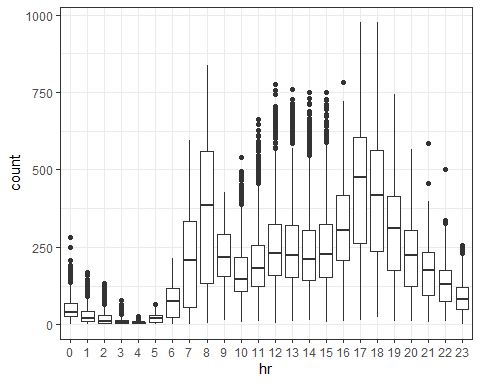
ggcorr(bike2)



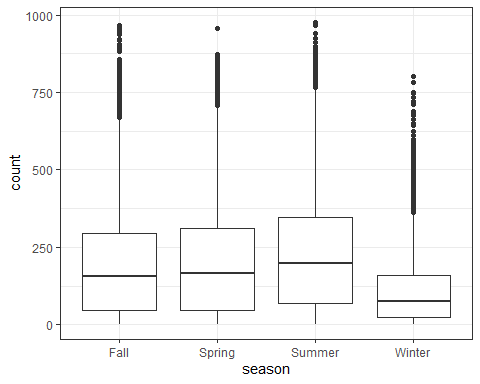
The variable “count” is most strongly correlated with the variables “temp” and “atemp.”

**Task 3:**

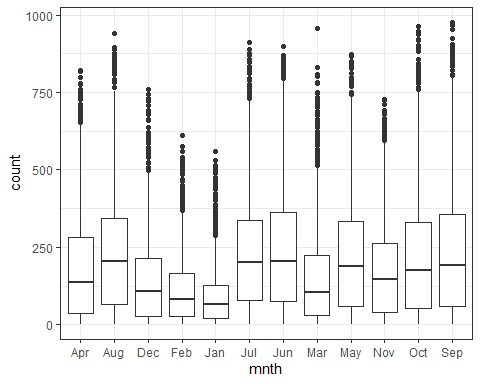
ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw()



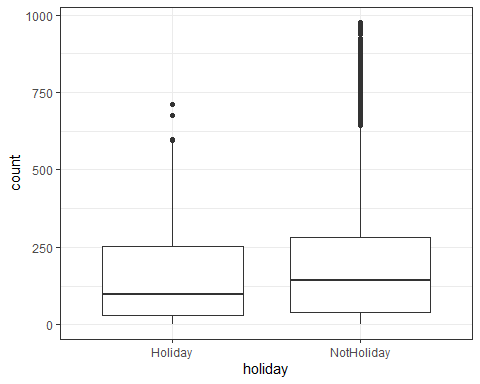
ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()



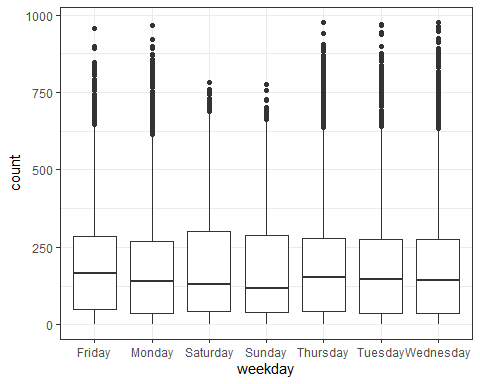
ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()



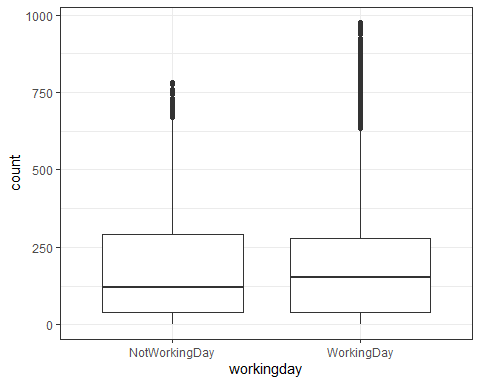
ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()



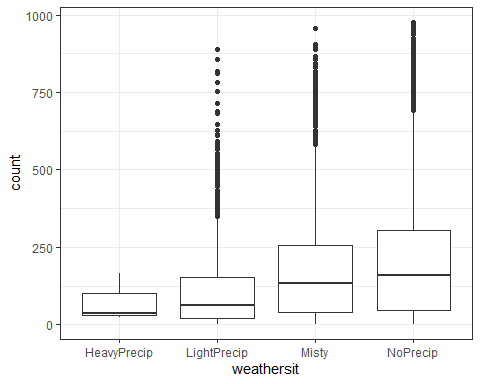
ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()



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



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



The variables “hr,” “mnth,” and “weathersit” appear to affect “count.” It makes sense that these variables would affect the number of rental bikes. More people rent bikes when it is warm outside, so the number of rental bikes in February and January would be lower. It also makes sense that the number of rental bikes is higher during the day when the sun is out, and lower in the very early hours of the day. Lastly, the number of rental bikes is lower during periods of heavy precipitation.

**Task 4:**

bike\_recipe = recipe(count ~ hr, bike) %>%  
 step\_dummy(all\_nominal())  
lm\_model =   
 linear\_reg() %>%   
 set\_engine("lm")   
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   
## -446.45 -60.99 -6.01 50.10 551.49   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 53.898 4.756 11.332 < 2e-16 \*\*\*  
## hr\_X1 -20.522 6.731 -3.049 0.002300 \*\*   
## hr\_X2 -31.028 6.752 -4.595 4.35e-06 \*\*\*  
## hr\_X3 -42.171 6.796 -6.205 5.58e-10 \*\*\*  
## hr\_X4 -47.545 6.796 -6.996 2.73e-12 \*\*\*  
## hr\_X5 -34.008 6.747 -5.040 4.70e-07 \*\*\*  
## hr\_X6 22.146 6.729 3.291 0.000999 \*\*\*  
## hr\_X7 158.167 6.724 23.523 < 2e-16 \*\*\*  
## hr\_X8 305.113 6.724 45.377 < 2e-16 \*\*\*  
## hr\_X9 165.411 6.724 24.600 < 2e-16 \*\*\*  
## hr\_X10 119.770 6.724 17.812 < 2e-16 \*\*\*  
## hr\_X11 154.245 6.724 22.939 < 2e-16 \*\*\*  
## hr\_X12 199.418 6.722 29.668 < 2e-16 \*\*\*  
## hr\_X13 199.763 6.719 29.729 < 2e-16 \*\*\*  
## hr\_X14 187.051 6.719 27.838 < 2e-16 \*\*\*  
## hr\_X15 197.335 6.719 29.368 < 2e-16 \*\*\*  
## hr\_X16 258.085 6.717 38.422 < 2e-16 \*\*\*  
## hr\_X17 407.554 6.717 60.674 < 2e-16 \*\*\*  
## hr\_X18 371.613 6.722 55.286 < 2e-16 \*\*\*  
## hr\_X19 257.625 6.722 38.327 < 2e-16 \*\*\*  
## hr\_X20 172.132 6.722 25.608 < 2e-16 \*\*\*  
## hr\_X21 118.416 6.722 17.617 < 2e-16 \*\*\*  
## hr\_X22 77.437 6.722 11.520 < 2e-16 \*\*\*  
## hr\_X23 33.933 6.722 5.048 4.50e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 128.2 on 17355 degrees of freedom  
## Multiple R-squared: 0.5015, Adjusted R-squared: 0.5008   
## F-statistic: 759.1 on 23 and 17355 DF, p-value: < 2.2e-16

dwtest(lm\_fit$fit$fit$fit)

##   
## Durbin-Watson test  
##   
## data: lm\_fit$fit$fit$fit  
## DW = 0.30859, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0

bike2\_recipe = recipe(count ~ temp, bike)  
lm\_model =   
 linear\_reg() %>%   
 set\_engine("lm")   
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(bike2\_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 built two models; one using the “best” variable from my visualization analysis (“hr”), and one using the “best” variable from my correlation analysis (“temp”). Of the two, hour appears to be the best predictor for the count of rental bikes. The quality of the model for hour as a predictor of count appears good. The linear regression analysis yielded significance with a significant p-value and an adjusted R-squared of 0.5008. Diagnostics also look good.

**Task 5:**

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(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 0.61 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)

## 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

The ridge regression model includes all variables except “instant,” “dteday,” “registered,” and “casual.” I chose a Lambda of 15, with a R-squared of 0.6205, because the R-squared values leveled off around 0.62. The variables with slopes farthest away from 0 have the largest affect on the predicted variable (“count”). We can easily see which variables these are by examining the slopes for all variables included in the model.

**Task 6:**

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(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 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 = 0.440)

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.463087634  
## temp 31.802161623  
## atemp 21.278651960  
## hum -22.892060298  
## windspeed -4.385708207  
## season\_Spring -7.109080624  
## season\_Summer -13.575316276  
## season\_Winter -22.560924582  
## mnth\_Aug .   
## mnth\_Dec .   
## mnth\_Feb .   
## mnth\_Jan .   
## mnth\_Jul -6.927594534  
## mnth\_Jun -2.313165169  
## mnth\_Mar 1.709357360  
## mnth\_May 2.057792250  
## mnth\_Nov .   
## mnth\_Oct 5.435385102  
## mnth\_Sep 7.903123654  
## hr\_X1 -8.694976832  
## hr\_X2 -10.209487618  
## hr\_X3 -12.022795179  
## hr\_X4 -12.377773963  
## hr\_X5 -9.322701641  
## hr\_X6 1.468825611  
## hr\_X7 28.307752558  
## hr\_X8 56.088821633  
## hr\_X9 26.032150985  
## hr\_X10 14.583811735  
## hr\_X11 19.165083877  
## hr\_X12 26.628003663  
## hr\_X13 25.372664129  
## hr\_X14 22.026341180  
## hr\_X15 23.873802384  
## hr\_X16 36.422652517  
## hr\_X17 67.479381993  
## hr\_X18 61.269117522  
## hr\_X19 39.931523379  
## hr\_X20 24.322954227  
## hr\_X21 14.729532502  
## hr\_X22 7.570873621  
## hr\_X23 0.001506087  
## holiday\_NotHoliday 4.118310588  
## weekday\_Monday -1.214881067  
## weekday\_Saturday 0.366257645  
## weekday\_Sunday -3.966435296  
## weekday\_Thursday -0.046273120  
## weekday\_Tuesday -0.616987200  
## weekday\_Wednesday .   
## workingday\_WorkingDay .   
## weathersit\_LightPrecip -14.446608530  
## weathersit\_Misty .   
## weathersit\_NoPrecip 2.262150638

The lasso regression model includes all variables except “instant,” “dteday,” “registered,” and “casual.” I chose a Lambda of 0.440, with a R-squared of 0.6305, because the R-squared values leveled off around 0.63. This model clearly shows the irrelevant variables, because their slopes are no longer visible (zeroed out). Looking at the slopes that were not zeroed out, we can see which variables have the biggest affect on our predicted variable.

The ridge and lasso regression models both aid the modeling process by helping to narrow down which variables have the greatest affect on the variable you hope to predict. In this case, we wanted to build a model to predict the number of rental bikes. This variable was named “count.” Data was available from several different variables including temperature, time of day, time of week, month, and weather condition. A ridge and lasso regression model was built to determine which variables would best fit the model to predict “count.” The lasso regression model proved most helpful in eliminating unnecessary variables from the model. In the end, it appears that time of day, or “hr,” is the single best predictor of number of rental bikes, or “count.”