# David Wells

## BAN 502

### Module 2 Assignment 2

bike <- read\_csv("~/BAN 502/Module 2/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()  
## )

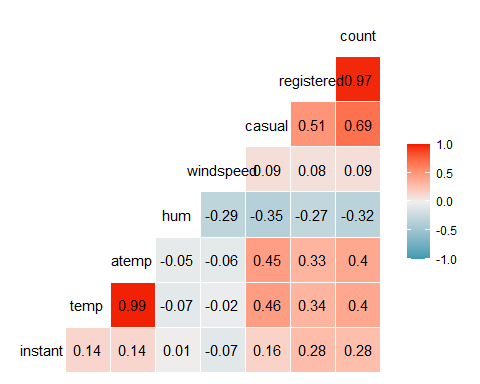
bike = bike %>% mutate(dteday = mdy(dteday))   
  
bike <- bike %>%  
 mutate\_if(sapply(bike, is.character), as.factor)  
  
bike <- bike %>%  
 mutate(hr = as.factor(hr))  
  
#bike <- bike %>%  
 #mutate\_if(sapply(bike, is.factor), as.numeric)  
  
  
str(bike)

## tibble [17,379 x 16] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ instant : num [1:17379] 1 2 3 4 5 6 7 8 9 10 ...  
## $ dteday : Date[1:17379], format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Fall","Spring",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ mnth : Factor w/ 12 levels "Apr","Aug","Dec",..: 5 5 5 5 5 5 5 5 5 5 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ holiday : Factor w/ 2 levels "Holiday","NotHoliday": 2 2 2 2 2 2 2 2 2 2 ...  
## $ weekday : Factor w/ 7 levels "Friday","Monday",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ workingday: Factor w/ 2 levels "NotWorkingDay",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weathersit: Factor w/ 4 levels "HeavyPrecip",..: 4 4 4 4 4 3 4 4 4 4 ...  
## $ temp : num [1:17379] 0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...  
## $ atemp : num [1:17379] 0.288 0.273 0.273 0.288 0.288 ...  
## $ hum : num [1:17379] 0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...  
## $ windspeed : num [1:17379] 0 0 0 0 0 0.0896 0 0 0 0 ...  
## $ casual : num [1:17379] 3 8 5 3 0 0 2 1 1 8 ...  
## $ registered: num [1:17379] 13 32 27 10 1 1 0 2 7 6 ...  
## $ count : num [1:17379] 16 40 32 13 1 1 2 3 8 14 ...  
## - attr(\*, "spec")=  
## .. 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()  
## .. )

# We convert 'hr' to a factor because the number in 'hr' refers to an hour of the day. In this case, the hour is a categorical variable and therefore it makes sense to treat it as a factor rather than a number.

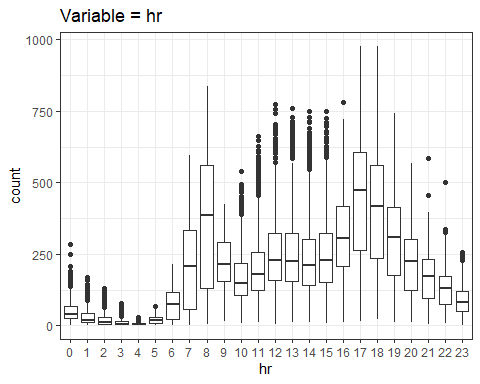
ggcorr(bike, label="TRUE", label\_round = 2)

## Warning in ggcorr(bike, label = "TRUE", label\_round = 2): data in column(s)  
## 'dteday', 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday',  
## 'weathersit' are not numeric and were ignored

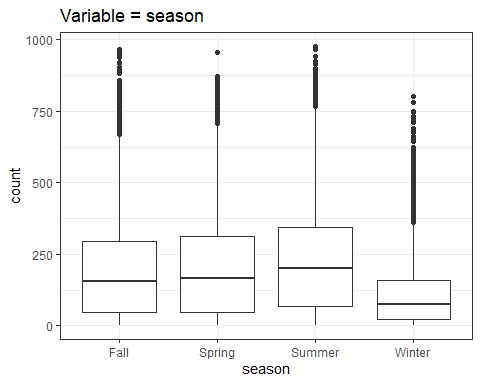


# Both 'temp' and 'atemp' appear to be most correlated with 'count'.

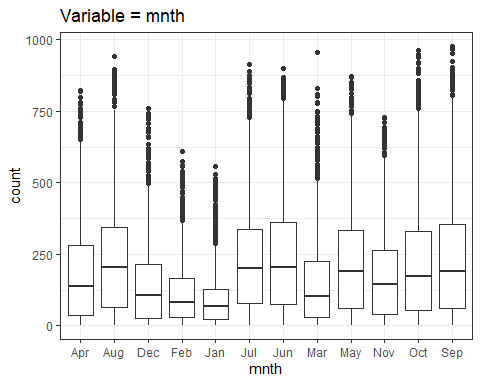
ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw() + ggtitle("Variable = hr")



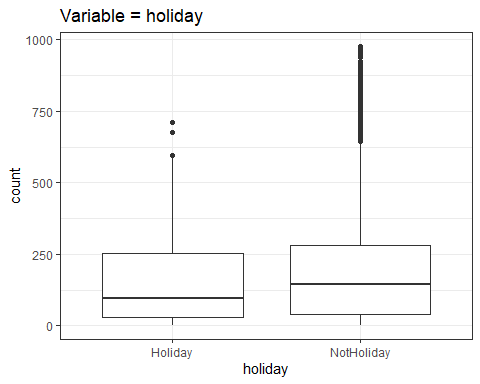
ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw() + ggtitle("Variable = season")



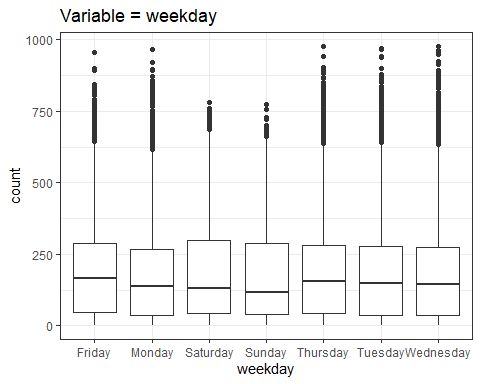
ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw() + ggtitle("Variable = mnth")



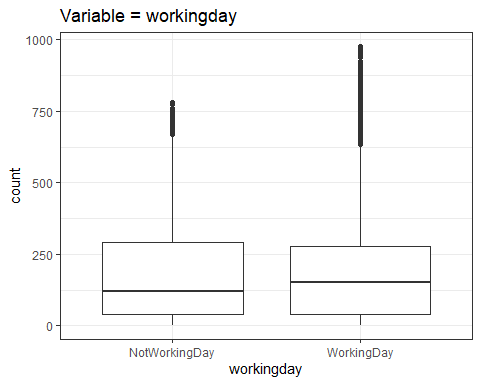
ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw() + ggtitle("Variable = holiday")



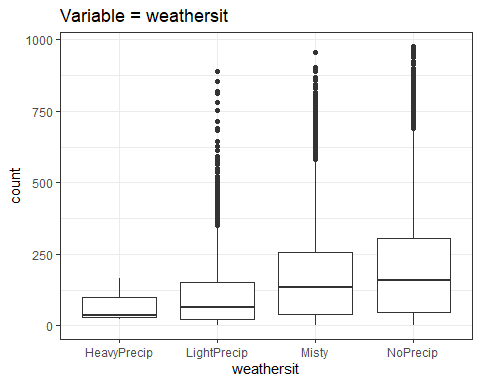
ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw() + ggtitle("Variable = weekday")



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



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



# The variable 'hr' appears to affect 'count' due to the variability of count given the time of day.  
# The variable 'season' appears to not significantly affect 'count', as it seems bike ridership is fairly consistent across seasons.  
# The variable 'mnth' appears to affect 'count'. Ridership varies month to month with the Winter months (Dec, Jan, Feb, and Mar) showing the greatest difference from the remaining months.  
# The variable 'holiday' appears not to have a significant effect as both Holiday and NonHoliday ridership remains constant between them.  
# The variable 'weekday' appears not to have a significant effect as ridership remains constant among the days of the week.  
# The variable 'workingday' appears not to have a significant effect as ridership remains constant between both types of days.  
# The variable 'weathersit' appears to affect 'count'. Ridership on days with HeavyPrecip or LightPrecip have lower values than the other two options.

bike\_simple = recipe(count ~ hr, bike)  
bike\_simple

## Data Recipe  
##   
## Inputs:  
##   
## role #variables  
## outcome 1  
## predictor 1

lm\_model =   
 linear\_reg()   
 set\_engine

## function (object, engine, ...)   
## {  
## if (!inherits(object, "model\_spec")) {  
## rlang::abort("`object` should have class 'model\_spec'.")  
## }  
## if (!is.character(engine) | length(engine) != 1)   
## rlang::abort("`engine` should be a single character value.")  
## object$engine <- engine  
## object <- check\_engine(object)  
## new\_model\_spec(cls = class(object)[1], args = object$args,   
## eng\_args = enquos(...), mode = object$mode, method = NULL,   
## engine = object$engine)  
## }  
## <bytecode: 0x0000000025736ce0>  
## <environment: namespace:parsnip>

lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(bike\_simple)  
   
lm\_fit = fit(lm\_wflow, bike)

## Warning: Engine set to `lm`.

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 \*\*\*  
## hr1 -20.522 6.731 -3.049 0.002300 \*\*   
## hr2 -31.028 6.752 -4.595 4.35e-06 \*\*\*  
## hr3 -42.171 6.796 -6.205 5.58e-10 \*\*\*  
## hr4 -47.545 6.796 -6.996 2.73e-12 \*\*\*  
## hr5 -34.008 6.747 -5.040 4.70e-07 \*\*\*  
## hr6 22.146 6.729 3.291 0.000999 \*\*\*  
## hr7 158.167 6.724 23.523 < 2e-16 \*\*\*  
## hr8 305.113 6.724 45.377 < 2e-16 \*\*\*  
## hr9 165.411 6.724 24.600 < 2e-16 \*\*\*  
## hr10 119.770 6.724 17.812 < 2e-16 \*\*\*  
## hr11 154.245 6.724 22.939 < 2e-16 \*\*\*  
## hr12 199.418 6.722 29.668 < 2e-16 \*\*\*  
## hr13 199.763 6.719 29.729 < 2e-16 \*\*\*  
## hr14 187.051 6.719 27.838 < 2e-16 \*\*\*  
## hr15 197.335 6.719 29.368 < 2e-16 \*\*\*  
## hr16 258.085 6.717 38.422 < 2e-16 \*\*\*  
## hr17 407.554 6.717 60.674 < 2e-16 \*\*\*  
## hr18 371.613 6.722 55.286 < 2e-16 \*\*\*  
## hr19 257.625 6.722 38.327 < 2e-16 \*\*\*  
## hr20 172.132 6.722 25.608 < 2e-16 \*\*\*  
## hr21 118.416 6.722 17.617 < 2e-16 \*\*\*  
## hr22 77.437 6.722 11.520 < 2e-16 \*\*\*  
## hr23 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

# 'hr' seems to be a good predictor of 'count', as each hour is significant to at least p<0.01. However, the Adjusted R-square value is 0.5008, which leads me to believe it's not the only predictor or even the best single predictor.

bike\_recipe2 = recipe(count ~., bike) %>%   
 step\_rm(instant,dteday,registered,casual) %>%  
 step\_dummy(all\_nominal()) %>%  
 step\_center(temp,atemp,hum,windspeed) %>%   
 step\_scale(temp,atemp,hum,windspeed)  
   
ridge\_model =   
 linear\_reg(mixture = 0) %>%   
 set\_engine("glmnet")   
  
ridge\_wflow =   
 workflow() %>%   
 add\_model(ridge\_model) %>%   
 add\_recipe(bike\_recipe2)  
  
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

## == 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\_fit2 = ridge\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%   
 coef(s = 82)  
ridge\_fit2

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 164.2269950  
## temp 21.1053680  
## atemp 21.1697360  
## hum -22.8100840  
## windspeed 0.7863886  
## season\_Spring -0.2989202  
## season\_Summer -4.8013631  
## season\_Winter -27.0667868  
## mnth\_Aug 1.1122345  
## mnth\_Dec -0.7936524  
## mnth\_Feb -14.6922420  
## mnth\_Jan -16.8438506  
## mnth\_Jul -14.5054539  
## mnth\_Jun -1.5034073  
## mnth\_Mar -3.8826738  
## mnth\_May 9.9909816  
## mnth\_Nov 6.3626044  
## mnth\_Oct 26.0139620  
## mnth\_Sep 24.2587845  
## hr\_X1 -89.4793751  
## hr\_X2 -95.1226510  
## hr\_X3 -102.0310904  
## hr\_X4 -103.7283796  
## hr\_X5 -92.4833079  
## hr\_X6 -52.1059480  
## hr\_X7 41.2871849  
## hr\_X8 138.5386716  
## hr\_X9 34.2727026  
## hr\_X10 -4.9905023  
## hr\_X11 11.7802198  
## hr\_X12 38.0671396  
## hr\_X13 34.0018126  
## hr\_X14 22.3948797  
## hr\_X15 28.8083639  
## hr\_X16 72.1528870  
## hr\_X17 179.7824781  
## hr\_X18 158.2804005  
## hr\_X19 83.8461178  
## hr\_X20 29.1666571  
## hr\_X21 -4.1953814  
## hr\_X22 -29.3491949  
## hr\_X23 -56.0310757  
## holiday\_NotHoliday 16.4552140  
## weekday\_Monday -3.4412746  
## weekday\_Saturday 3.3849222  
## weekday\_Sunday -6.2902901  
## weekday\_Thursday -0.4708931  
## weekday\_Tuesday -1.5598888  
## weekday\_Wednesday 0.1354380  
## workingday\_WorkingDay 3.7681469  
## weathersit\_LightPrecip -34.3986538  
## weathersit\_Misty 3.3865217  
## weathersit\_NoPrecip 8.5455350

# I chose a coefficient 's=82' as once Lambda approached 82, I felt there was little gain to be had. The difference between this value and subsequent smaller Lambdas was less than 0.30 and approaching zero.

bike\_recipe3 = recipe(count ~., bike) %>%   
 step\_rm(instant,dteday,registered,casual) %>%  
 step\_dummy(all\_nominal()) %>%  
 step\_center(temp,atemp,hum,windspeed) %>%   
 step\_scale(temp,atemp,hum,windspeed)  
   
lasso\_model =   
 linear\_reg(mixture = 1) %>%   
 set\_engine("glmnet")   
  
lasso\_wflow =   
 workflow() %>%   
 add\_model(lasso\_model) %>%   
 add\_recipe(bike\_recipe3)  
  
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

## == 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\_fit2 = lasso\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%   
 coef(s = 0.639)  
lasso\_fit2

## 53 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 105.0573477  
## temp 30.3378993  
## atemp 22.3038242  
## hum -23.1096346  
## windspeed -3.9787303  
## season\_Spring -13.9871279  
## season\_Summer -28.6480998  
## season\_Winter -51.0535316  
## mnth\_Aug .   
## mnth\_Dec .   
## mnth\_Feb .   
## mnth\_Jan .   
## mnth\_Jul -24.3361674  
## mnth\_Jun -8.1444122  
## mnth\_Mar 4.7200071  
## mnth\_May 5.9964342  
## mnth\_Nov .   
## mnth\_Oct 20.3878384  
## mnth\_Sep 28.1400483  
## hr\_X1 -48.9851335  
## hr\_X2 -56.8642070  
## hr\_X3 -66.7022097  
## hr\_X4 -68.5422544  
## hr\_X5 -52.3429074  
## hr\_X6 .   
## hr\_X7 133.9116183  
## hr\_X8 272.6352839  
## hr\_X9 122.4635796  
## hr\_X10 65.2737306  
## hr\_X11 88.1469387  
## hr\_X12 125.2990504  
## hr\_X13 118.9531363  
## hr\_X14 102.2426086  
## hr\_X15 111.4619513  
## hr\_X16 173.9519447  
## hr\_X17 328.7679505  
## hr\_X18 298.2248962  
## hr\_X19 191.7734418  
## hr\_X20 113.9035318  
## hr\_X21 66.0645392  
## hr\_X22 30.3524742  
## hr\_X23 -1.3927568  
## holiday\_NotHoliday 23.5925904  
## weekday\_Monday -2.8013844  
## weekday\_Saturday 0.8494848  
## weekday\_Sunday -10.4336273  
## weekday\_Thursday .   
## weekday\_Tuesday -0.8984223  
## weekday\_Wednesday .   
## workingday\_WorkingDay .   
## weathersit\_LightPrecip -52.2288118  
## weathersit\_Misty .   
## weathersit\_NoPrecip 4.1397814

# Using the Lasso model with the Lambda I chose [coef(s = 0.639)], I was able to exclude certain variables from the model, whereas with the Ridge model, I was not able to do so.