# Daniel Rauscher

## Mod 2 Assignment 2

library(tidyverse)

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

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.6 ✓ dplyr 1.0.3  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

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

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 0.1.2 ──

## ✓ broom 0.7.4 ✓ 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(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(leaps)  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(splines)  
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

## Task 1

We converted hr into a factor because the hr column represents a specific time of day which should classify it as a categorical variable, but since we use numbers to describe the time of day it was classified as numerical.

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()  
## )

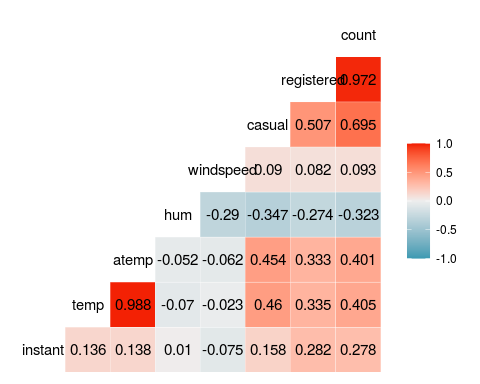
#View(bike)  
bike = bike %>% mutate(dteday = mdy(dteday))   
bike <- bike %>% mutate(season = as\_factor(season)) %>% mutate(holiday = as\_factor(holiday)) %>% mutate(mnth = as\_factor(mnth)) %>% mutate(weekday = as\_factor(weekday)) %>% mutate(workingday = as\_factor(workingday)) %>% mutate(weathersit = as\_factor(weathersit)) %>% mutate(hr = as\_factor(hr))  
# str(bike)

## Task 2

The quantitative variable that appears to be best correlated with count is temp.

ggcorr(bike, label = "TRUE", label\_round =3)

## Warning in ggcorr(bike, label = "TRUE", label\_round = 3): data in column(s)  
## 'dteday', 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday',  
## 'weathersit' are not numeric and were ignored

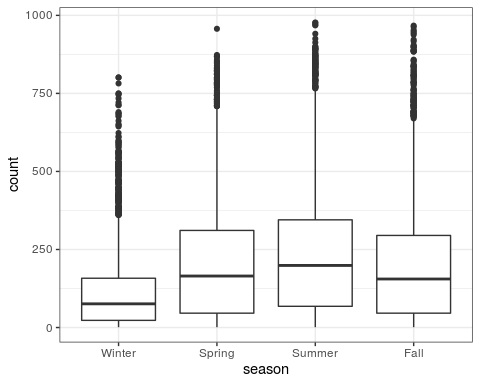


## Task 3

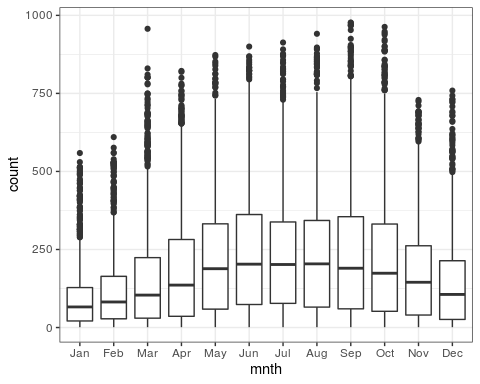
The variables that appear to affect count are weathersit, mnth, and season. Weathersit variable affects count because in worse weather people are less likely to clean their bikes. Mnth and season variables affect count because in the warmer months/seasons it would make sense for people to ride their bikes more, thus needing it cleaned more.

Weekday, holiday and workingday do not appear to affect count because people ride and clean their bikes when the weather is better, which is not dependent on the days or holidays.

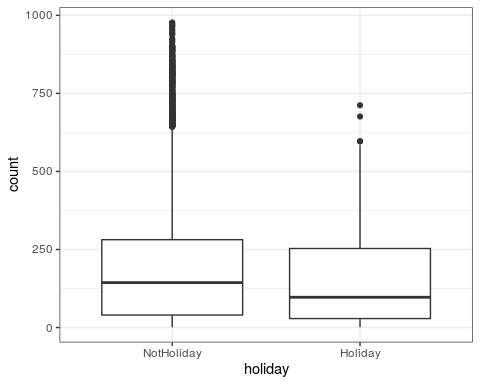
ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()



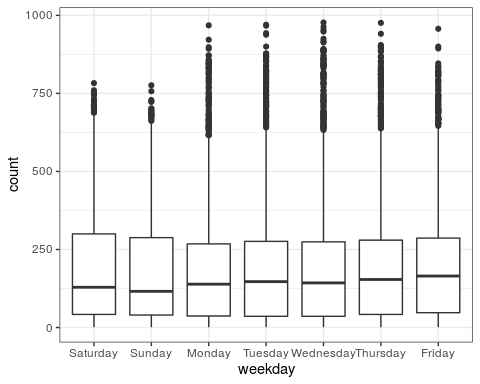
ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()



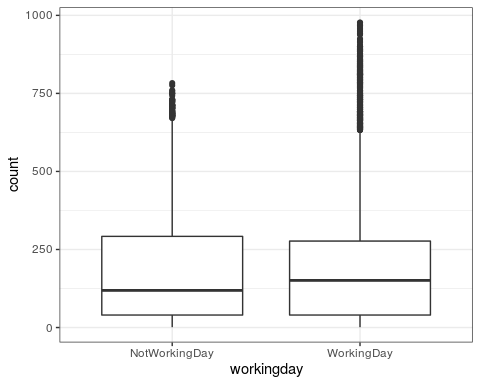
ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()



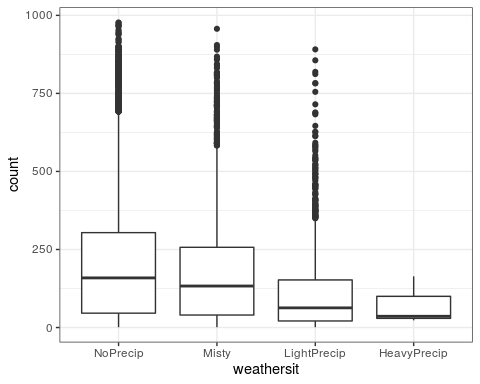
ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()



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



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



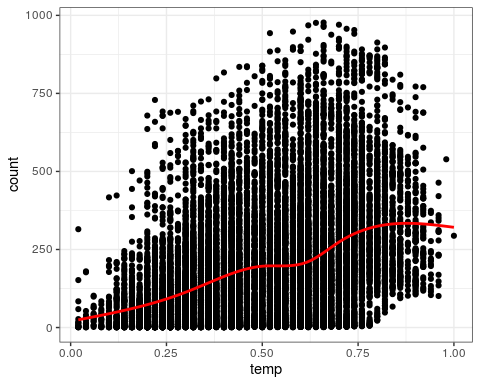
## Task 4

Using the variable temp as the best variable, the model quality is not very good. It is significant but the adjusted R squared value is only 0.1638. When we test the model against our diagnostic assumptions it does not fair well. It could be argued that the model passes assumption 1 and 3 after looking at the graph, but it does not pass assumption 2 or 4.

bike\_recipe <- recipe(count ~ temp, bike)  
  
  
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   
## -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

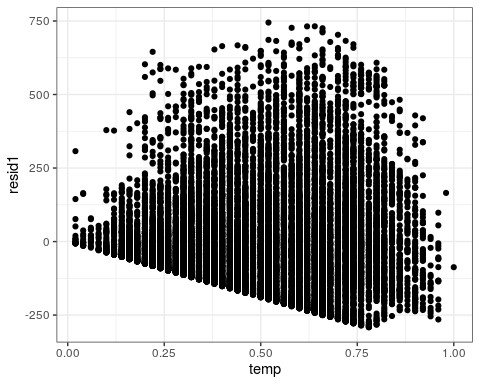
ggplot(bike,aes(x=temp,y=count)) + geom\_point() + theme\_bw() +   
 geom\_smooth(method = lm, formula = y ~ ns(x, df=6), col = "red", se=FALSE)



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

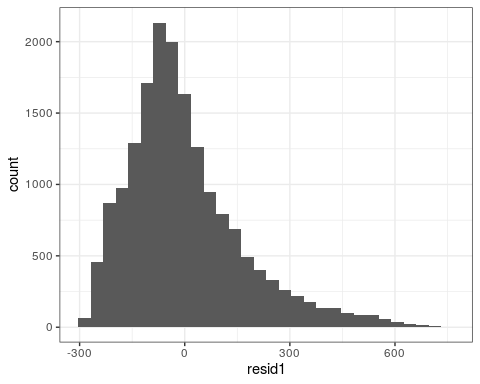
##   
## Durbin-Watson test  
##   
## data: lm\_fit$fit$fit$fit  
## DW = 0.3684, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0

bike\_assumption3 <- bike %>% mutate(resid1 = lm\_fit$fit$fit$fit$residuals)  
ggplot(bike\_assumption3, aes(x=temp, y=resid1)) +  
 geom\_point() +  
 theme\_bw()



ggplot(bike\_assumption3, aes(x=resid1)) + geom\_histogram() + theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Task 5

After picking a lambda value of 17 the model results in an R squared value of 0.6249. The model agrees with our common sense approach on what was correlated. Variables that had to do with weather were more correlated with count than variables that did not have anything to do with weather.

bike\_ridge<- recipe(count ~ season + mnth + hr + holiday + weekday + workingday + weathersit + temp + atemp + hum + windspeed, bike) %>%   
 step\_ns(temp, deg\_free = 6) %>%  
 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\_ridge)  
  
ridge\_fit = fit(ridge\_wflow, bike)  
  
ridge\_fit

## ══ Workflow [trained] ══════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: linear\_reg()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 4 Recipe Steps  
##   
## ● step\_ns()  
## ● step\_dummy()  
## ● step\_center()  
## ● step\_scale()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
##   
## Call: glmnet::glmnet(x = maybe\_matrix(x), y = y, family = "gaussian", alpha = ~0)   
##   
## Df %Dev Lambda  
## 1 57 0.00 72720  
## 2 57 0.67 66260  
## 3 57 0.74 60370  
## 4 57 0.81 55010  
## 5 57 0.89 50120  
## 6 57 0.97 45670  
## 7 57 1.06 41610  
## 8 57 1.16 37920  
## 9 57 1.28 34550  
## 10 57 1.40 31480  
## 11 57 1.53 28680  
## 12 57 1.67 26130  
## 13 57 1.83 23810  
## 14 57 2.00 21700  
## 15 57 2.19 19770  
## 16 57 2.39 18010  
## 17 57 2.62 16410  
## 18 57 2.86 14960  
## 19 57 3.12 13630  
## 20 57 3.40 12420  
## 21 57 3.71 11310  
## 22 57 4.05 10310  
## 23 57 4.41 9392  
## 24 57 4.80 8558  
## 25 57 5.23 7798  
## 26 57 5.68 7105  
## 27 57 6.18 6474  
## 28 57 6.71 5899  
## 29 57 7.27 5375  
## 30 57 7.88 4897  
## 31 57 8.53 4462  
## 32 57 9.23 4066  
## 33 57 9.97 3705  
## 34 57 10.76 3375  
## 35 57 11.60 3076  
## 36 57 12.48 2802  
## 37 57 13.42 2553  
## 38 57 14.40 2327  
## 39 57 15.44 2120  
## 40 57 16.52 1932  
## 41 57 17.66 1760  
## 42 57 18.84 1604  
## 43 57 20.07 1461  
## 44 57 21.34 1331  
## 45 57 22.66 1213  
## 46 57 24.01 1105  
##   
## ...  
## and 54 more lines.

ridge\_fit %>%   
 pull\_workflow\_fit() %>%  
 pluck("fit")

##   
## Call: glmnet::glmnet(x = maybe\_matrix(x), y = y, family = "gaussian", alpha = ~0)   
##   
## Df %Dev Lambda  
## 1 57 0.00 72720  
## 2 57 0.67 66260  
## 3 57 0.74 60370  
## 4 57 0.81 55010  
## 5 57 0.89 50120  
## 6 57 0.97 45670  
## 7 57 1.06 41610  
## 8 57 1.16 37920  
## 9 57 1.28 34550  
## 10 57 1.40 31480  
## 11 57 1.53 28680  
## 12 57 1.67 26130  
## 13 57 1.83 23810  
## 14 57 2.00 21700  
## 15 57 2.19 19770  
## 16 57 2.39 18010  
## 17 57 2.62 16410  
## 18 57 2.86 14960  
## 19 57 3.12 13630  
## 20 57 3.40 12420  
## 21 57 3.71 11310  
## 22 57 4.05 10310  
## 23 57 4.41 9392  
## 24 57 4.80 8558  
## 25 57 5.23 7798  
## 26 57 5.68 7105  
## 27 57 6.18 6474  
## 28 57 6.71 5899  
## 29 57 7.27 5375  
## 30 57 7.88 4897  
## 31 57 8.53 4462  
## 32 57 9.23 4066  
## 33 57 9.97 3705  
## 34 57 10.76 3375  
## 35 57 11.60 3076  
## 36 57 12.48 2802  
## 37 57 13.42 2553  
## 38 57 14.40 2327  
## 39 57 15.44 2120  
## 40 57 16.52 1932  
## 41 57 17.66 1760  
## 42 57 18.84 1604  
## 43 57 20.07 1461  
## 44 57 21.34 1331  
## 45 57 22.66 1213  
## 46 57 24.01 1105  
## 47 57 25.40 1007  
## 48 57 26.82 918  
## 49 57 28.27 836  
## 50 57 29.74 762  
## 51 57 31.23 694  
## 52 57 32.73 632  
## 53 57 34.24 576  
## 54 57 35.75 525  
## 55 57 37.25 478  
## 56 57 38.74 436  
## 57 57 40.21 397  
## 58 57 41.66 362  
## 59 57 43.08 330  
## 60 57 44.46 300  
## 61 57 45.80 274  
## 62 57 47.09 250  
## 63 57 48.33 227  
## 64 57 49.52 207  
## 65 57 50.64 189  
## 66 57 51.70 172  
## 67 57 52.70 157  
## 68 57 53.64 143  
## 69 57 54.51 130  
## 70 57 55.32 118  
## 71 57 56.07 108  
## 72 57 56.76 98  
## 73 57 57.39 90  
## 74 57 57.97 82  
## 75 57 58.49 74  
## 76 57 58.97 68  
## 77 57 59.40 62  
## 78 57 59.79 56  
## 79 57 60.14 51  
## 80 57 60.46 47  
## 81 57 60.74 43  
## 82 57 61.00 39  
## 83 57 61.23 35  
## 84 57 61.45 32  
## 85 57 61.64 29  
## 86 57 61.81 27  
## 87 57 61.97 24  
## 88 57 62.12 22  
## 89 57 62.25 20  
## 90 57 62.38 18  
## 91 57 62.49 17  
## 92 57 62.60 15  
## 93 57 62.70 14  
## 94 57 62.79 13  
## 95 57 62.88 12  
## 96 57 62.96 11  
## 97 57 63.03 10  
## 98 57 63.10 9  
## 99 57 63.17 8  
## 100 57 63.23 7

ridge\_fit %>%   
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 17)

## 58 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.46308763  
## atemp 24.60575221  
## hum -25.17271911  
## windspeed -2.86236673  
## temp\_ns\_1 -5.66765647  
## temp\_ns\_2 -0.06454337  
## temp\_ns\_3 7.88631738  
## temp\_ns\_4 27.78522075  
## temp\_ns\_5 -2.72487480  
## temp\_ns\_6 1.58092194  
## season\_Spring 9.20584484  
## season\_Summer 1.56136633  
## season\_Fall 19.48906303  
## mnth\_Feb 0.21402512  
## mnth\_Mar 3.62902255  
## mnth\_Apr 2.46089136  
## mnth\_May 3.13471613  
## mnth\_Jun -3.00817788  
## mnth\_Jul -6.08193846  
## mnth\_Aug -1.17989603  
## mnth\_Sep 7.70699272  
## mnth\_Oct 6.79164822  
## mnth\_Nov 3.23924151  
## mnth\_Dec 2.90786911  
## hr\_X1 -18.16860805  
## hr\_X2 -19.39586724  
## hr\_X3 -20.85429295  
## hr\_X4 -21.08948525  
## hr\_X5 -18.43167547  
## hr\_X6 -7.89182714  
## hr\_X7 16.48398810  
## hr\_X8 41.77297172  
## hr\_X9 14.07456476  
## hr\_X10 3.55416197  
## hr\_X11 7.73506149  
## hr\_X12 14.57853828  
## hr\_X13 13.38299669  
## hr\_X14 10.28259406  
## hr\_X15 11.93827573  
## hr\_X16 23.38000829  
## hr\_X17 51.83866279  
## hr\_X18 46.15590097  
## hr\_X19 26.59427259  
## hr\_X20 12.35290080  
## hr\_X21 3.67465578  
## hr\_X22 -2.78187880  
## hr\_X23 -9.64596442  
## holiday\_Holiday -4.31887162  
## weekday\_Sunday -4.35722212  
## weekday\_Monday -2.04112965  
## weekday\_Tuesday -1.06375411  
## weekday\_Wednesday -0.19843703  
## weekday\_Thursday -0.49211550  
## weekday\_Friday 1.25672523  
## workingday\_WorkingDay -0.31774589  
## weathersit\_Misty -1.37040428  
## weathersit\_LightPrecip -13.45771668  
## weathersit\_HeavyPrecip -0.33318369

## Task 6

Using the lasso model we can more clearly see what variables were not correlated to the variable count because they have fallen out of the model. If we wanted to take out more variables we could increase lambda.

One of the implications of the model results from the ridge and lasso methods is there are clearly some variables that are not correlated with count and should be removed from the model. The adjusted R squared value of around .62 in both models is OK, but not great. Since temp has a non linear relationship with count, adding a spline helped improve the adjusted R squared value.

bike\_lasso <- recipe(count ~ season + mnth + hr + holiday + weekday + workingday + weathersit + temp + atemp + hum + windspeed, bike) %>%   
 step\_ns(temp, deg\_free = 6) %>%  
 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\_ridge)  
  
lasso\_fit = fit(lasso\_wflow, bike)  
  
lasso\_fit

## ══ Workflow [trained] ══════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: linear\_reg()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 4 Recipe Steps  
##   
## ● step\_ns()  
## ● 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 72.720  
## 2 1 2.73 66.260  
## 3 1 4.99 60.370  
## 4 3 7.91 55.010  
## 5 3 11.96 50.120  
## 6 4 15.82 45.670  
## 7 4 19.53 41.610  
## 8 6 22.98 37.920  
## 9 6 26.59 34.550  
## 10 6 29.58 31.480  
## 11 8 32.37 28.680  
## 12 11 35.33 26.130  
## 13 12 38.43 23.810  
## 14 12 41.13 21.700  
## 15 13 43.45 19.770  
## 16 14 45.55 18.010  
## 17 15 47.52 16.410  
## 18 15 49.25 14.960  
## 19 16 50.78 13.630  
## 20 17 52.09 12.420  
## 21 18 53.28 11.310  
## 22 19 54.34 10.310  
## 23 22 55.26 9.392  
## 24 24 56.18 8.558  
## 25 26 56.98 7.798  
## 26 28 57.70 7.105  
## 27 29 58.36 6.474  
## 28 29 58.95 5.899  
## 29 30 59.44 5.375  
## 30 31 59.91 4.897  
## 31 33 60.30 4.462  
## 32 34 60.65 4.066  
## 33 35 60.98 3.705  
## 34 35 61.26 3.375  
## 35 36 61.49 3.076  
## 36 36 61.69 2.802  
## 37 38 61.88 2.553  
## 38 38 62.15 2.327  
## 39 41 62.38 2.120  
## 40 42 62.55 1.932  
## 41 42 62.69 1.760  
## 42 42 62.81 1.604  
## 43 42 62.91 1.461  
## 44 43 63.00 1.331  
## 45 44 63.12 1.213  
## 46 44 63.22 1.105  
##   
## ...  
## and 35 more lines.

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 72.720  
## 2 1 2.73 66.260  
## 3 1 4.99 60.370  
## 4 3 7.91 55.010  
## 5 3 11.96 50.120  
## 6 4 15.82 45.670  
## 7 4 19.53 41.610  
## 8 6 22.98 37.920  
## 9 6 26.59 34.550  
## 10 6 29.58 31.480  
## 11 8 32.37 28.680  
## 12 11 35.33 26.130  
## 13 12 38.43 23.810  
## 14 12 41.13 21.700  
## 15 13 43.45 19.770  
## 16 14 45.55 18.010  
## 17 15 47.52 16.410  
## 18 15 49.25 14.960  
## 19 16 50.78 13.630  
## 20 17 52.09 12.420  
## 21 18 53.28 11.310  
## 22 19 54.34 10.310  
## 23 22 55.26 9.392  
## 24 24 56.18 8.558  
## 25 26 56.98 7.798  
## 26 28 57.70 7.105  
## 27 29 58.36 6.474  
## 28 29 58.95 5.899  
## 29 30 59.44 5.375  
## 30 31 59.91 4.897  
## 31 33 60.30 4.462  
## 32 34 60.65 4.066  
## 33 35 60.98 3.705  
## 34 35 61.26 3.375  
## 35 36 61.49 3.076  
## 36 36 61.69 2.802  
## 37 38 61.88 2.553  
## 38 38 62.15 2.327  
## 39 41 62.38 2.120  
## 40 42 62.55 1.932  
## 41 42 62.69 1.760  
## 42 42 62.81 1.604  
## 43 42 62.91 1.461  
## 44 43 63.00 1.331  
## 45 44 63.12 1.213  
## 46 44 63.22 1.105  
## 47 44 63.30 1.007  
## 48 43 63.36 0.918  
## 49 45 63.40 0.836  
## 50 45 63.44 0.762  
## 51 46 63.47 0.694  
## 52 48 63.51 0.632  
## 53 49 63.55 0.576  
## 54 49 63.58 0.525  
## 55 49 63.60 0.478  
## 56 52 63.62 0.436  
## 57 53 63.65 0.397  
## 58 53 63.67 0.362  
## 59 53 63.70 0.330  
## 60 53 63.71 0.300  
## 61 53 63.73 0.274  
## 62 55 63.74 0.250  
## 63 55 63.75 0.227  
## 64 55 63.76 0.207  
## 65 55 63.76 0.189  
## 66 55 63.77 0.172  
## 67 55 63.78 0.157  
## 68 56 63.78 0.143  
## 69 55 63.78 0.130  
## 70 55 63.79 0.118  
## 71 55 63.79 0.108  
## 72 55 63.79 0.098  
## 73 55 63.79 0.090  
## 74 55 63.80 0.082  
## 75 55 63.80 0.074  
## 76 55 63.80 0.068  
## 77 55 63.80 0.062  
## 78 55 63.80 0.056  
## 79 55 63.80 0.051  
## 80 54 63.80 0.047  
## 81 55 63.80 0.043

lasso\_fit %>%   
 pull\_workflow\_fit() %>%  
 pluck("fit") %>%  
 coef(s = 1.007)

## 58 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 189.4630876  
## atemp 26.4859187  
## hum -25.0031327  
## windspeed -2.8837044  
## temp\_ns\_1 -1.5570535  
## temp\_ns\_2 1.9531199  
## temp\_ns\_3 9.9232802  
## temp\_ns\_4 29.6005214  
## temp\_ns\_5 .   
## temp\_ns\_6 .   
## season\_Spring 11.1513923  
## season\_Summer .   
## season\_Fall 23.6039837  
## mnth\_Feb .   
## mnth\_Mar 1.8289586  
## mnth\_Apr .   
## mnth\_May 0.7704410  
## mnth\_Jun -3.1975738  
## mnth\_Jul -5.1537924  
## mnth\_Aug .   
## mnth\_Sep 7.0412044  
## mnth\_Oct 3.1719037  
## mnth\_Nov .   
## mnth\_Dec .   
## hr\_X1 -11.3946580  
## hr\_X2 -12.7682607  
## hr\_X3 -14.4494154  
## hr\_X4 -14.7123796  
## hr\_X5 -11.6949445  
## hr\_X6 -0.1636066  
## hr\_X7 24.4815213  
## hr\_X8 52.0136535  
## hr\_X9 21.7899442  
## hr\_X10 10.3216109  
## hr\_X11 14.8686563  
## hr\_X12 22.3514303  
## hr\_X13 21.0796337  
## hr\_X14 17.7245100  
## hr\_X15 19.5431743  
## hr\_X16 32.0162623  
## hr\_X17 63.0237517  
## hr\_X18 56.7885799  
## hr\_X19 35.4570317  
## hr\_X20 19.9413150  
## hr\_X21 10.4744983  
## hr\_X22 3.4407952  
## hr\_X23 -2.0973847  
## holiday\_Holiday -3.6267399  
## weekday\_Sunday -3.1399314  
## weekday\_Monday -1.0278938  
## weekday\_Tuesday .   
## weekday\_Wednesday .   
## weekday\_Thursday .   
## weekday\_Friday 0.9223513  
## workingday\_WorkingDay .   
## weathersit\_Misty -0.4451954  
## weathersit\_LightPrecip -13.6728366  
## weathersit\_HeavyPrecip .