## Assignment3

## Lydia Suter

## 1/30/2022

## Multiple Linear Regression

In this example, we demonstrate multiple linear regression model-building techniques. The dataset is the bike\_cleaned data. To make things more manageable, this large dataset is reduced to eleven variables (ten predictors and one response).

Begin by loading necessary packages. As usual, you will need to install any of these packages that have not been previously installed.

library(tidyverse) #tidyverse set of packages and functions

## Warning: package 'tidyverse' was built under R version 4.1.2

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.5 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.0.2 v forcats 0.5.1

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

library(tidymodels)

## Warning: package 'tidymodels' was built under R version 4.1.2

## Registered S3 method overwritten by 'tune':  
## method from   
## required\_pkgs.model\_spec parsnip

## -- Attaching packages -------------------------------------- tidymodels 0.1.4 --

## v broom 0.7.9 v rsample 0.1.1   
## v dials 0.0.10 v tune 0.1.6   
## v infer 1.0.0 v workflows 0.2.4   
## v modeldata 0.1.1 v workflowsets 0.1.0   
## v parsnip 0.1.7 v yardstick 0.0.9   
## v recipes 0.1.17

## Warning: package 'dials' was built under R version 4.1.2

## Warning: package 'infer' was built under R version 4.1.2

## Warning: package 'modeldata' was built under R version 4.1.2

## Warning: package 'parsnip' was built under R version 4.1.2

## Warning: package 'recipes' was built under R version 4.1.2

## Warning: package 'rsample' was built under R version 4.1.2

## Warning: package 'tune' was built under R version 4.1.2

## Warning: package 'workflows' was built under R version 4.1.2

## Warning: package 'workflowsets' was built under R version 4.1.2

## Warning: package 'yardstick' was built under R version 4.1.2

## -- 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()  
## \* Use tidymodels\_prefer() to resolve common conflicts.

library(glmnet) #for Lasso, ridge, and elastic net models

## Warning: package 'glmnet' was built under R version 4.1.2

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

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

## Loaded glmnet 4.1-3

library(GGally) #create ggcorr and ggpairs plots

## Warning: package 'GGally' was built under R version 4.1.2

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(gridExtra)

## Warning: package 'gridExtra' was built under R version 4.1.2

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(ggcorrplot) #create an alternative to ggcorr plots

## Warning: package 'ggcorrplot' was built under R version 4.1.2

library(MASS) #access to forward and backward selection algorithms

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(leaps) #best subset selection

## Warning: package 'leaps' was built under R version 4.1.2

library(lmtest) #for the dw test

## Warning: package 'lmtest' was built under R version 4.1.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.2

##   
## Attaching package: 'zoo'

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

library(splines) #for nonlinear fitting  
library(car) #for calculating the variance inflation factor

## Warning: package 'car' was built under R version 4.1.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.1.2

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

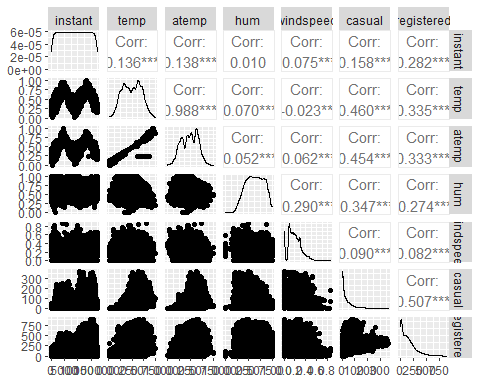
library(readr)  
bike<- read\_csv("bike\_cleaned.csv",  
col\_types = cols(dteday = col\_date(format = "%m/%d/%Y")))

bike$hr<-as.factor(bike$hr)

Why do we convert the “hr” variable into factor? Why not just leave as numbers? Factors represent a very efficient way to store character values, because each unique character value is stored only once, and the data itself is stored as a vector of integers.Perhaps the most important advantage is that they can be used in statistical modeling where they will be implemented correctly, i.e., they will then be assigned the correct number of degrees of freedom. Factor variables are also very useful in many different types of graphics. Furthermore, storing string variables as factor variables is a more efficient use of memory.

bike2 = bike %>% dplyr::select("instant","temp", "atemp","hum", "windspeed","casual","registered")

ggpairs(bike2)

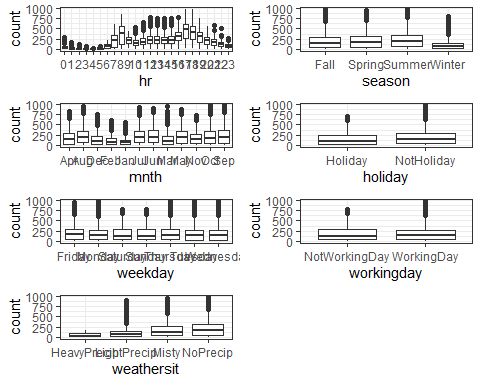


ggcorr(bike2,label = TRUE)



Which of the quantitative variables appears to be best correlated with “count” (ignore the “registered”and “casual” variable as the sum of these two variables equals “count”)? atemp,temp

p0=ggplot(bike,aes(x=hr,y=count)) + geom\_boxplot() + theme\_bw()  
p1=ggplot(bike,aes(x=season,y=count)) + geom\_boxplot() + theme\_bw()  
p2=ggplot(bike,aes(x=mnth,y=count)) + geom\_boxplot() + theme\_bw()  
p3=ggplot(bike,aes(x=holiday,y=count)) + geom\_boxplot() + theme\_bw()  
p4=ggplot(bike,aes(x=weekday,y=count)) + geom\_boxplot() + theme\_bw()  
p5=ggplot(bike,aes(x=workingday,y=count)) + geom\_boxplot() + theme\_bw()  
p6=ggplot(bike,aes(x=weathersit,y=count)) + geom\_boxplot() + theme\_bw()  
grid.arrange(p0,p1,p2,p3,p4,p5,p6, ncol = 2) #arranging ggplot objects in a grid



Repeat this boxplot-based analysis for each of the categorical variables. Which variables appear to affect “count”? Provide a brief explanation as to why you believe that each variable does or does not affect “count”(use your intuition to help you answer this question)hr, mnth,season and whethersit, holiday More overlap in the box plots indicates less association while less overlap in the box plots indicates a stronger association.

**Let’s see how we would do this same model with Tidymodels. We start by building a recipe.**

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

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

**Next we specify the type of model that we are building.**

lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use

**Next we combine the recipe and the model with a workflow.**

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

**Next we fit (execute) the workflow on our dataset.**

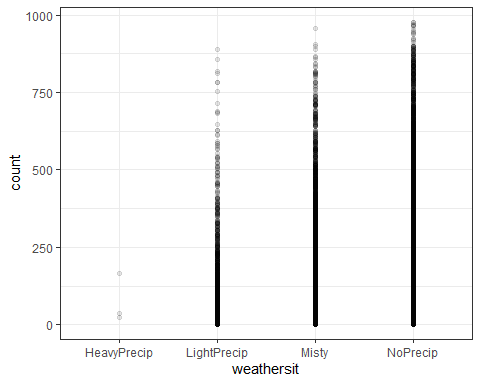
lm\_fit = fit(lm\_wflow, bike)

summary(lm\_fit$fit$fit$fit) #three fits :), the actual fit is embedded deeply in the object

##   
## 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) 74.33 103.60 0.717 0.473  
## weathersitLightPrecip 37.25 103.71 0.359 0.720  
## weathersitMisty 100.83 103.64 0.973 0.331  
## weathersitNoPrecip 130.54 103.62 1.260 0.208  
##   
## 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

ggplot(bike,aes(x=weathersit,y=count)) + geom\_point(alpha=0.1) + geom\_smooth(method = "lm", color = "red") + theme\_bw()

## `geom\_smooth()` using formula 'y ~ x'



**Multiple Linear Regression**

bike3 = bike %>% dplyr::select("season","mnth","hr", "holiday","weekday","workingday", "weathersit", "temp", "atemp","hum", "windspeed", "count" )

summary(bike3) #statistical summary

## season mnth hr holiday   
## Length:17379 Length:17379 16 : 730 Length:17379   
## Class :character Class :character 17 : 730 Class :character   
## Mode :character Mode :character 13 : 729 Mode :character   
## 14 : 729   
## 15 : 729   
## 12 : 728   
## (Other):13004   
## weekday workingday weathersit temp   
## Length:17379 Length:17379 Length:17379 Min. :0.020   
## Class :character Class :character Class :character 1st Qu.:0.340   
## Mode :character Mode :character Mode :character Median :0.500   
## Mean :0.497   
## 3rd Qu.:0.660   
## Max. :1.000   
##   
## atemp hum windspeed count   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. : 1.0   
## 1st Qu.:0.3333 1st Qu.:0.4800 1st Qu.:0.1045 1st Qu.: 40.0   
## Median :0.4848 Median :0.6300 Median :0.1940 Median :142.0   
## Mean :0.4758 Mean :0.6272 Mean :0.1901 Mean :189.5   
## 3rd Qu.:0.6212 3rd Qu.:0.7800 3rd Qu.:0.2537 3rd Qu.:281.0   
## Max. :1.0000 Max. :1.0000 Max. :0.8507 Max. :977.0   
##

glimpse(bike3) #use of glimpse to hide the read\_csv attributes

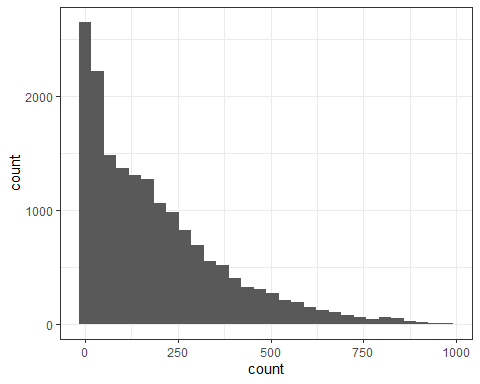
## Rows: 17,379  
## Columns: 12  
## $ season <chr> "Winter", "Winter", "Winter", "Winter", "Winter", "Winter",~  
## $ mnth <chr> "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Ja~  
## $ hr <fct> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~  
## $ holiday <chr> "NotHoliday", "NotHoliday", "NotHoliday", "NotHoliday", "No~  
## $ weekday <chr> "Saturday", "Saturday", "Saturday", "Saturday", "Saturday",~  
## $ workingday <chr> "NotWorkingDay", "NotWorkingDay", "NotWorkingDay", "NotWork~  
## $ weathersit <chr> "NoPrecip", "NoPrecip", "NoPrecip", "NoPrecip", "NoPrecip",~  
## $ temp <dbl> 0.24, 0.22, 0.22, 0.24, 0.24, 0.24, 0.22, 0.20, 0.24, 0.32,~  
## $ atemp <dbl> 0.2879, 0.2727, 0.2727, 0.2879, 0.2879, 0.2576, 0.2727, 0.2~  
## $ hum <dbl> 0.81, 0.80, 0.80, 0.75, 0.75, 0.75, 0.80, 0.86, 0.75, 0.76,~  
## $ windspeed <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0896, 0.0000, 0.0~  
## $ count <dbl> 16, 40, 32, 13, 1, 1, 2, 3, 8, 14, 36, 56, 84, 94, 106, 110~

### Data Exploration

Begin exploring the data by looking at a plot of our response variable only (choose a histogram for a single quantitative variable)

ggplot(bike3, aes(x=count)) + geom\_histogram() + theme\_bw()

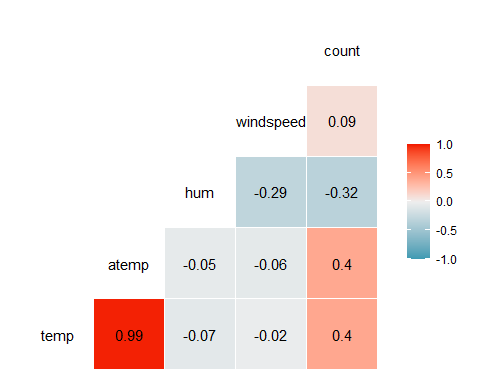
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 The data is somewhat skewed to the right.

Next we look at correlation. This is a logical step since almost all of our variables are quantitative.

ggcorr(bike3, label = "TRUE", label\_round = 2)

## Warning in ggcorr(bike3, label = "TRUE", label\_round = 2): data in column(s)  
## 'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday', 'weathersit' are not  
## numeric and were ignored



bike3\_recipe = recipe(count ~ weathersit + season +holiday+mnth+temp+atemp+hum+windspeed+weekday+workingday, bike3)  
  
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(bike3\_recipe)  
  
lm\_fit2 = fit(lm\_wflow, bike3)

summary(lm\_fit2$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -385.77 -101.88 -29.47 62.51 674.23   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 166.256 89.208 1.864 0.062381 .   
## weathersitLightPrecip -102.255 87.791 -1.165 0.244137   
## weathersitMisty -84.107 87.739 -0.959 0.337772   
## weathersitNoPrecip -97.158 87.746 -1.107 0.268192   
## seasonSpring -15.074 8.539 -1.765 0.077528 .   
## seasonSummer -46.396 7.693 -6.031 1.66e-09 \*\*\*  
## seasonWinter -62.789 7.280 -8.625 < 2e-16 \*\*\*  
## holidayNotHoliday 34.476 7.277 4.738 2.18e-06 \*\*\*  
## mnthAug -33.522 9.660 -3.470 0.000521 \*\*\*  
## mnthDec 66.326 9.457 7.014 2.41e-12 \*\*\*  
## mnthFeb 63.208 9.481 6.667 2.70e-11 \*\*\*  
## mnthJan 84.085 9.626 8.735 < 2e-16 \*\*\*  
## mnthJul -77.033 9.753 -7.898 2.99e-15 \*\*\*  
## mnthJun -57.517 6.712 -8.569 < 2e-16 \*\*\*  
## mnthMar 42.433 7.377 5.752 8.98e-09 \*\*\*  
## mnthMay -2.943 5.946 -0.495 0.620688   
## mnthNov 44.253 10.384 4.261 2.04e-05 \*\*\*  
## mnthOct 40.914 10.254 3.990 6.63e-05 \*\*\*  
## mnthSep 29.918 9.113 3.283 0.001030 \*\*   
## temp 447.253 43.525 10.276 < 2e-16 \*\*\*  
## atemp 124.218 45.644 2.721 0.006506 \*\*   
## hum -291.532 7.382 -39.492 < 2e-16 \*\*\*  
## windspeed 24.143 10.466 2.307 0.021078 \*   
## weekdayMonday -3.768 4.415 -0.853 0.393454   
## weekdaySaturday 1.046 4.302 0.243 0.807853   
## weekdaySunday -9.796 4.311 -2.272 0.023077 \*   
## weekdayThursday -8.376 4.319 -1.939 0.052467 .   
## weekdayTuesday -6.075 4.333 -1.402 0.160982   
## weekdayWednesday -2.689 4.322 -0.622 0.533815   
## workingdayWorkingDay NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 151.7 on 17350 degrees of freedom  
## Multiple R-squared: 0.3014, Adjusted R-squared: 0.3003   
## F-statistic: 267.4 on 28 and 17350 DF, p-value: < 2.2e-16

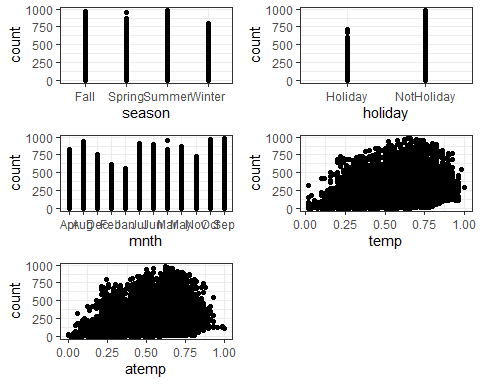
Provide a brief commentary on the resulting model.there is no significance in above results the R-squared is 0.282.removed some variables which have a p value more than 0.05.

smaller\_set = recipe(count ~ season +holiday+temp+atemp+hum+windspeed, bike3)  
  
  
lm\_model\_s = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm")  
  
lm\_small =   
 workflow() %>%   
 add\_model(lm\_model\_s) %>%   
 add\_recipe(smaller\_set)  
  
lm\_fit3 = fit(lm\_small, bike3)  
  
summary(lm\_fit3$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -366.33 -102.23 -30.15 64.60 726.69   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 153.766 10.168 15.122 < 2e-16 \*\*\*  
## seasonSpring -58.170 3.525 -16.504 < 2e-16 \*\*\*  
## seasonSummer -99.896 4.361 -22.906 < 2e-16 \*\*\*  
## seasonWinter -55.880 3.655 -15.290 < 2e-16 \*\*\*  
## holidayNotHoliday 27.931 6.986 3.998 6.42e-05 \*\*\*  
## temp 323.825 41.997 7.711 1.32e-14 \*\*\*  
## atemp 147.413 45.469 3.242 0.001189 \*\*   
## hum -280.058 6.477 -43.237 < 2e-16 \*\*\*  
## windspeed 38.921 10.352 3.760 0.000171 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 153.7 on 17370 degrees of freedom  
## Multiple R-squared: 0.282, Adjusted R-squared: 0.2817   
## F-statistic: 852.8 on 8 and 17370 DF, p-value: < 2.2e-16

Provide a brief commentary on the resulting model.there is no significance in above results the R-squared is 0.282. but the p-value is much significant.

p1 = ggplot(bike3, aes(x=season,y=count)) + geom\_point() + theme\_bw()  
p2 = ggplot(bike3, aes(x=holiday,y=count)) + geom\_point() + theme\_bw()  
p3 = ggplot(bike3, aes(x=mnth,y=count)) + geom\_point() + theme\_bw()  
p4 = ggplot(bike3, aes(x=temp,y=count)) + geom\_point() + theme\_bw()  
p5 = ggplot(bike3, aes(x=atemp,y=count)) + geom\_point() + theme\_bw()  
  
grid.arrange(p1,p2,p3,p4,p5, ncol = 2) #arranging ggplot objects in a grid



smaller\_set = recipe(count ~ season +holiday+temp+atemp+hum+windspeed, bike3)  
  
  
lm\_model\_s = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm")  
  
lm\_small =   
 workflow() %>%   
 add\_model(lm\_model\_s) %>%   
 add\_recipe(smaller\_set)  
  
lm\_fit3 = fit(lm\_small, bike3)  
  
summary(lm\_fit3$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -366.33 -102.23 -30.15 64.60 726.69   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 153.766 10.168 15.122 < 2e-16 \*\*\*  
## seasonSpring -58.170 3.525 -16.504 < 2e-16 \*\*\*  
## seasonSummer -99.896 4.361 -22.906 < 2e-16 \*\*\*  
## seasonWinter -55.880 3.655 -15.290 < 2e-16 \*\*\*  
## holidayNotHoliday 27.931 6.986 3.998 6.42e-05 \*\*\*  
## temp 323.825 41.997 7.711 1.32e-14 \*\*\*  
## atemp 147.413 45.469 3.242 0.001189 \*\*   
## hum -280.058 6.477 -43.237 < 2e-16 \*\*\*  
## windspeed 38.921 10.352 3.760 0.000171 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 153.7 on 17370 degrees of freedom  
## Multiple R-squared: 0.282, Adjusted R-squared: 0.2817   
## F-statistic: 852.8 on 8 and 17370 DF, p-value: < 2.2e-16

Provide a brief commentary on the resulting model (including multicollinearity).season,holiday,temp,atemp,hum,windspeed best predictors to predict count based on p value.