### Module 2: Assignment 1 – Simple Linear Regression

## Lea Shipley

1: There are 6 variables and 153 observations in this dataset on daily air quality measurements in New York from May to September of 1973. All variables are integers, with the exception of “Wind,” which is numeric. There is missing data (37 missing from the variable “Ozone” and 7 missing from the variable “Solar.R”). The response (Y) variable is “Ozone”.

2: The new dataframe, “air2,” still contains 5 variables, but now has 111 rows after deleting the rows with missing data.

3: “Temp” is most strongly correlated with the “Ozone” variable. “Day” is least strongly correlated with the “Ozone” variable.

4: “Temp” and “Ozone” have a positive correlation.

5:

1. The R-squared value is decent. The predictor variable, temperature, is statistically significant with a p-value below 0.05.
2. The slope coefficient falls within the range of the 95% confidence interval 1.96 to 2.91.

8:

Assumption 1: The predictor and response variable do have a linear relationship.

Assumption 2: The residuals are likely independent, as the Durbin-Watson test resulted in a p-value greater than 0.05.

Assumption 3: Variance of the residuals is roughly symmetric overall, however there appears to be some non-constant variance with some outliers.

Assumption 4: The model residuals are approximately normally-distributed with some positive skew due to outliers.

9: The linear regression model constructed could be used to make predictions of ozone levels based on certain temperatures to issue warnings of ozone pollution levels. I would caution when using the model that it is certainly not perfect, and to perhaps use it as more of a general guide. There are some outliers in the dataset. Other factors such wind speeds, humidity levels, and traffic levels can also have an effect on ozone levels. Also, when temperature is in the 80-85° level there is more variance in the ozone level.

library(tidyverse)

## -- Attaching packages ------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

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

library(GGally)

## Warning: package 'GGally' was built under R version 3.6.2

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

##   
## Attaching package: 'GGally'

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

library(car)

## Warning: package 'car' was built under R version 3.6.2

## 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(lmtest)

## Warning: package 'lmtest' was built under R version 3.6.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.6.2

##   
## Attaching package: 'zoo'

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

**Task1**

air = airquality  
  
str(air)

## 'data.frame': 153 obs. of 6 variables:  
## $ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...  
## $ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...  
## $ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...  
## $ Temp : int 67 72 74 62 56 66 65 59 61 69 ...  
## $ Month : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ Day : int 1 2 3 4 5 6 7 8 9 10 ...

summary(air)

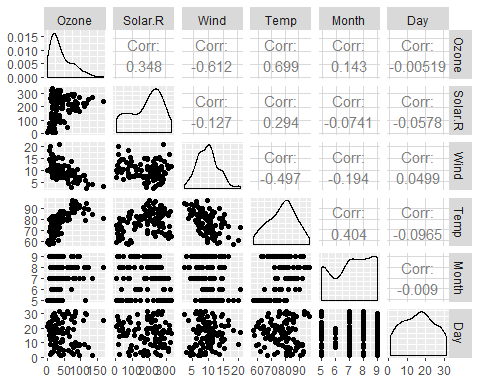
## Ozone Solar.R Wind Temp   
## Min. : 1.00 Min. : 7.0 Min. : 1.700 Min. :56.00   
## 1st Qu.: 18.00 1st Qu.:115.8 1st Qu.: 7.400 1st Qu.:72.00   
## Median : 31.50 Median :205.0 Median : 9.700 Median :79.00   
## Mean : 42.13 Mean :185.9 Mean : 9.958 Mean :77.88   
## 3rd Qu.: 63.25 3rd Qu.:258.8 3rd Qu.:11.500 3rd Qu.:85.00   
## Max. :168.00 Max. :334.0 Max. :20.700 Max. :97.00   
## NA's :37 NA's :7   
## Month Day   
## Min. :5.000 Min. : 1.0   
## 1st Qu.:6.000 1st Qu.: 8.0   
## Median :7.000 Median :16.0   
## Mean :6.993 Mean :15.8   
## 3rd Qu.:8.000 3rd Qu.:23.0   
## Max. :9.000 Max. :31.0   
##

**Task 2**

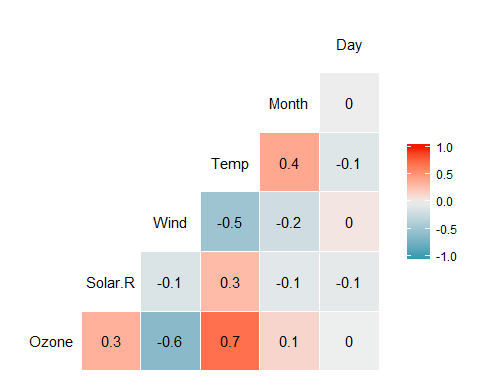
air2 = air %>% drop\_na() #delete any row with an NA value

**Task 3**

ggpairs(air2)

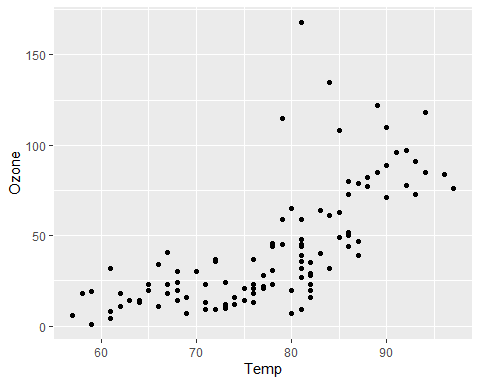


ggcorr(air2, label = TRUE)



**Task 4**

ggplot(air2, aes(x = Temp, y = Ozone)) + geom\_point()



**Task 5**

model1 = lm(Ozone ~ Temp, air2)   
summary(model1)

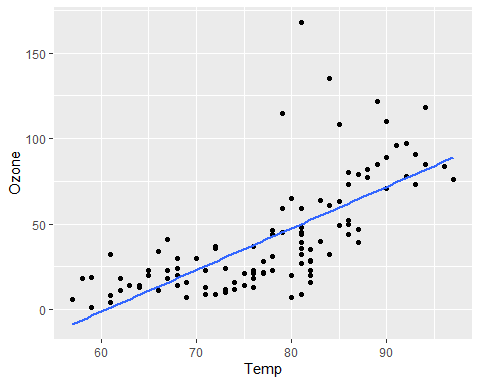
##   
## Call:  
## lm(formula = Ozone ~ Temp, data = air2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.922 -17.459 -0.874 10.444 118.078   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -147.6461 18.7553 -7.872 2.76e-12 \*\*\*  
## Temp 2.4391 0.2393 10.192 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 23.92 on 109 degrees of freedom  
## Multiple R-squared: 0.488, Adjusted R-squared: 0.4833   
## F-statistic: 103.9 on 1 and 109 DF, p-value: < 2.2e-16

confint(model1)

## 2.5 % 97.5 %  
## (Intercept) -184.818372 -110.473773  
## Temp 1.964787 2.913433

**Task 6**

ggplot(air2, aes(x = Temp, y = Ozone)) + geom\_point() + geom\_smooth(method = "lm", se = FALSE)



**Task 7**

testdata = data.frame(Temp = 80)  
predict(model1, newdata = testdata, interval = "predict")

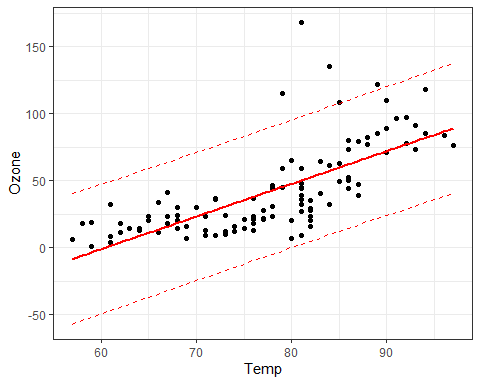
## fit lwr upr  
## 1 47.48272 -0.1510188 95.11646

**Task 8**

temp\_var = predict(model1, interval = "prediction")

## Warning in predict.lm(model1, interval = "prediction"): predictions on current data refer to \_future\_ responses

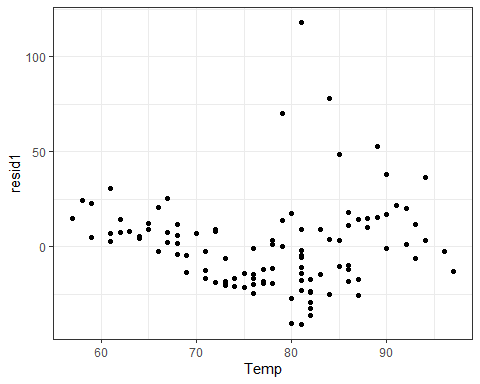
new\_df = cbind(air2, temp\_var)  
  
ggplot(new\_df, aes(x = Temp, y = Ozone)) +   
 geom\_point() +   
 geom\_smooth(method = "lm", se = FALSE, color = "red") +  
 geom\_line(aes(y=lwr), color = "red", linetype = "dashed") +  
 geom\_line(aes(y=upr), color = "red", linetype = "dashed") +  
 theme\_bw()



dwtest(model1)

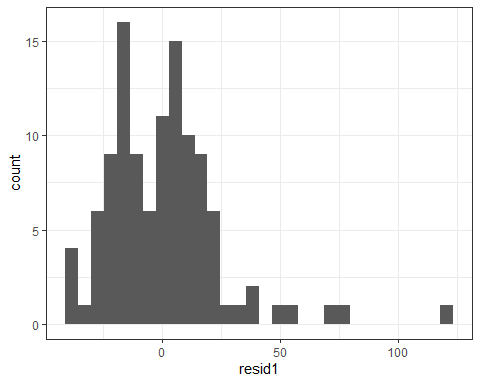
##   
## Durbin-Watson test  
##   
## data: model1  
## DW = 1.8644, p-value = 0.2123  
## alternative hypothesis: true autocorrelation is greater than 0

air2 = air2 %>% mutate(resid1 = model1$residuals) #add the model residuals to our data frame  
ggplot(air2,aes(x=Temp,y=resid1)) + geom\_point() + theme\_bw()

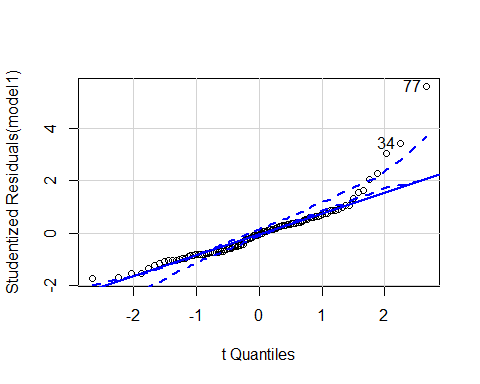


ggplot(air2,aes(x=resid1)) + geom\_histogram() + theme\_bw()

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



qqPlot(model1)



## [1] 34 77