title: “Model Basics Exercises” author: “Kim Roth & Johnathan Clementi” date: “4/7/2020” output: word\_document — These are exercises based on the exercises in R for Data Science by Wickham and Grolemund.

The website where the book appears is (and will always be) **free to use**, and is licensed under the [Creative Commons Attribution-NonCommercial-NoDerivs 3.0](http://creativecommons.org/licenses/by-nc-nd/3.0/us/) License.

We will use the data set fueleconomy:vehicles

library(tidyverse)

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

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.2 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(fueleconomy)  
vehicles<-fueleconomy::vehicles

Data for this file.

vehicles

## # A tibble: 33,442 x 12  
## id make model year class trans drive cyl displ fuel hwy cty  
## <int> <chr> <chr> <int> <chr> <chr> <chr> <int> <dbl> <chr> <int> <int>  
## 1 27550 AM Ge~ DJ Po ~ 1984 Specia~ Auto~ 2-Whe~ 4 2.5 Regu~ 17 18  
## 2 28426 AM Ge~ DJ Po ~ 1984 Specia~ Auto~ 2-Whe~ 4 2.5 Regu~ 17 18  
## 3 27549 AM Ge~ FJ8c P~ 1984 Specia~ Auto~ 2-Whe~ 6 4.2 Regu~ 13 13  
## 4 28425 AM Ge~ FJ8c P~ 1984 Specia~ Auto~ 2-Whe~ 6 4.2 Regu~ 13 13  
## 5 1032 AM Ge~ Post O~ 1985 Specia~ Auto~ Rear-~ 4 2.5 Regu~ 17 16  
## 6 1033 AM Ge~ Post O~ 1985 Specia~ Auto~ Rear-~ 6 4.2 Regu~ 13 13  
## 7 3347 ASC I~ GNX 1987 Midsiz~ Auto~ Rear-~ 6 3.8 Prem~ 21 14  
## 8 13309 Acura 2.2CL/~ 1997 Subcom~ Auto~ Front~ 4 2.2 Regu~ 26 20  
## 9 13310 Acura 2.2CL/~ 1997 Subcom~ Manu~ Front~ 4 2.2 Regu~ 28 22  
## 10 13311 Acura 2.2CL/~ 1997 Subcom~ Auto~ Front~ 6 3 Regu~ 26 18  
## # ... with 33,432 more rows

1. Make a linear model of cty as prediced by displ. Give the equation of the line. 1A.

ctyMpgModel = lm(cty~displ, data = vehicles)  
ctyMpgModel

##   
## Call:  
## lm(formula = cty ~ displ, data = vehicles)  
##   
## Coefficients:  
## (Intercept) displ   
## 25.812 -2.518

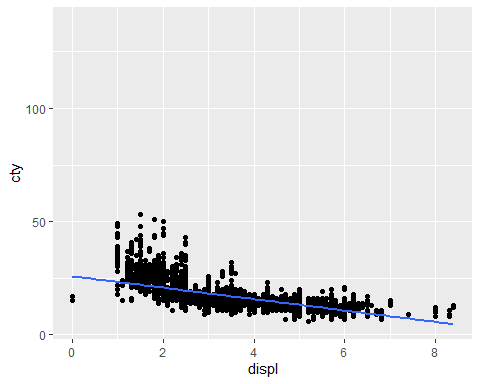
ctyMpgModYInt = 25.812  
ctyMpgModCoef = -2.518

1. Graph the linear model from 1 using ggplot. How does the fit look? 2A. Without further analysis, the line of fit follows the data well. Based on this figure, smaller displacement values are less predictive of city mpg than vehicles with larger displacement.

ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = displ, y = cty)) +  
 geom\_smooth(mapping = aes(x = displ, y = cty), method = "lm", se=0)

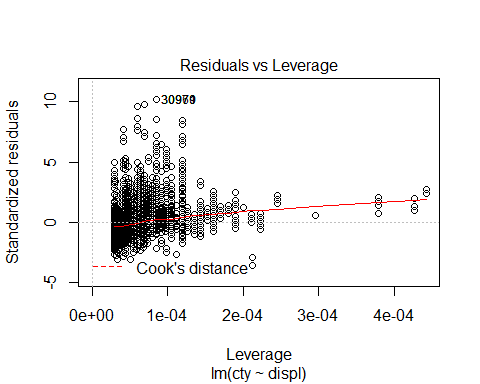
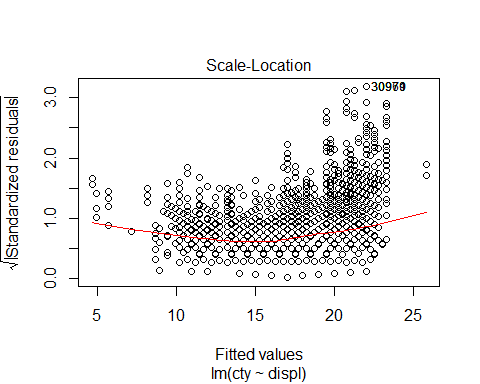
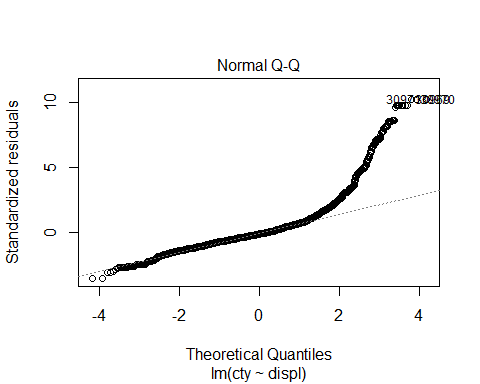
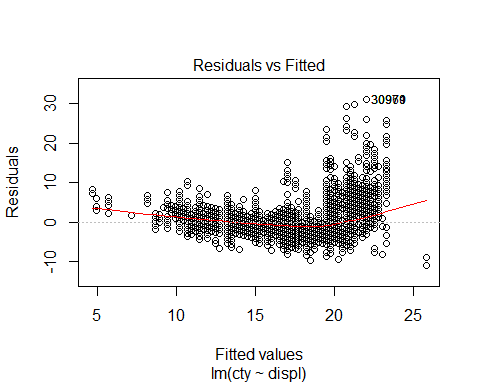
## Warning: Removed 57 rows containing non-finite values (stat\_smooth).

## Warning: Removed 57 rows containing missing values (geom\_point).



1. Look at the first residual graph of the linear model. How does it look? 3A. The dataset has a good constant spread and while the average value closely aligns with a ‘0’ residual in the middle of the data, the upper and lower bounds do not fit as well. Additionally, the spread of residuals at the upper bound means that as displacement increase, the model is less able to predict the city mpg.

plot(ctyMpgModel)



1. Instead of using lm() to fit a straight line, you can use loess() to fit a smooth curve. Repeat the model from 1. using loess() instead of lm(). How does the result compare to geom\_smooth()? What are the advantages and disadvantages to using a curve versus a line here? 4A. The curved line presents a different message to the reader than a straight line does. For instance, looking at the linear model figure, I would identify that there is more variation of city mileage in cars with less displacement. Whereas looking at the locally estimated scatterplot smoothing figure, I would more quickly highlight that cars with smaller displacement have better city mpg ratings.

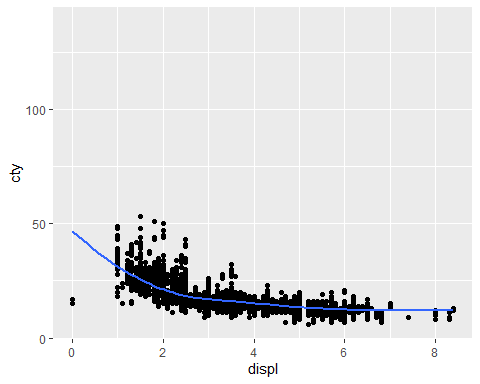
ctyMpgCurveModel = loess(cty~displ, data = vehicles)  
ctyMpgCurveModel

## Call:  
## loess(formula = cty ~ displ, data = vehicles)  
##   
## Number of Observations: 33385   
## Equivalent Number of Parameters: 4.98   
## Residual Standard Error: 2.673

ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = displ, y = cty)) +  
 geom\_smooth(mapping = aes(x = displ, y = cty), method = "loess", se=0)

## Warning: Removed 57 rows containing non-finite values (stat\_smooth).

## Warning: Removed 57 rows containing missing values (geom\_point).



In case you want to know more LOESS stands for, locally estimated scatterplot smoothing,it essentially puts together a bunch of linear for parts of the data, see <https://simplystatistics.org/2014/02/13/loess-explained-in-a-gif/>

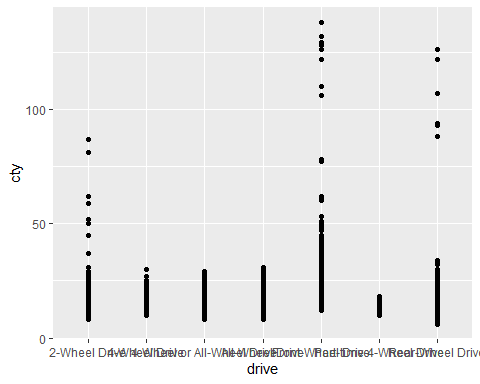
1. Predict the cty for displ of 1.9. 5A.

predictedDispl = (ctyMpgModCoef \* 1.9) + ctyMpgModYInt  
predictedDispl

## [1] 21.0278

1. Graph a scatterplot of cty vs. drive. Do you think a model is a good idea? 6A.

ggplot(data = vehicles) +  
 geom\_point(mapping = aes(x = drive, y = cty))



1. Make a linear model for cty as predicted by drive.

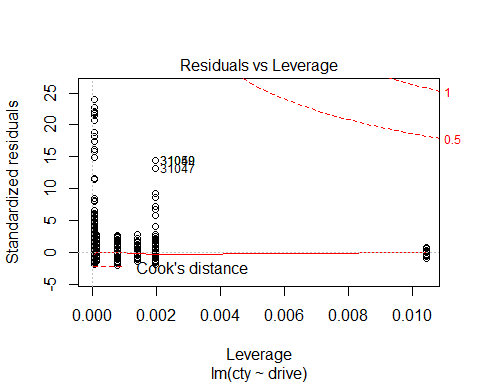
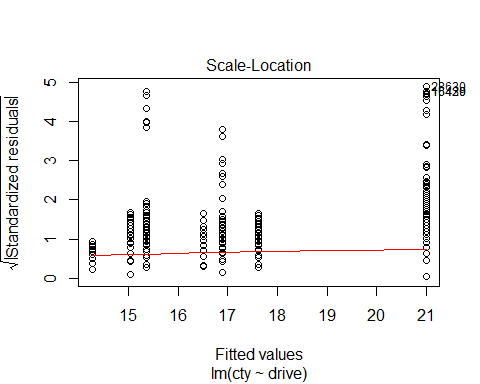
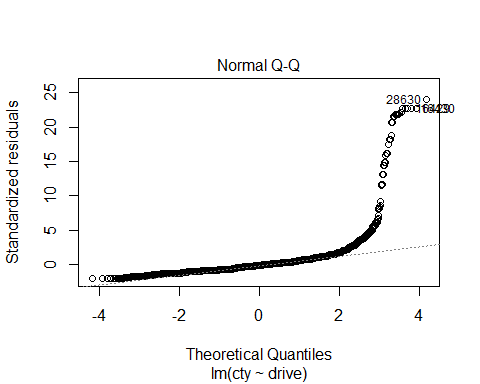
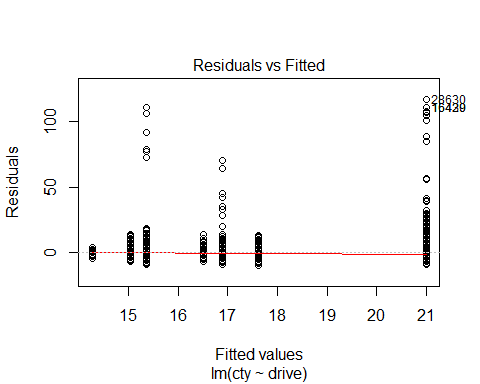
ctyDriveModel = lm(cty~drive, data = vehicles)  
ctyDriveModel

##   
## Call:  
## lm(formula = cty ~ drive, data = vehicles)  
##   
## Coefficients:  
## (Intercept) drive4-Wheel Drive   
## 16.8915 -0.3779   
## drive4-Wheel or All-Wheel Drive driveAll-Wheel Drive   
## -1.8465 0.7241   
## driveFront-Wheel Drive drivePart-time 4-Wheel Drive   
## 4.1005 -2.6207   
## driveRear-Wheel Drive   
## -1.5210

ctyDriveFWDCoef = 4.1005  
ctyDriveYInt = 16.8915

1. Look at the first residual graph of the linear model from 7. How does it look? 8A. Judging by the average line vs the 0 residual line, the drive variable is a good predictor of city mpg.

plot(ctyDriveModel)



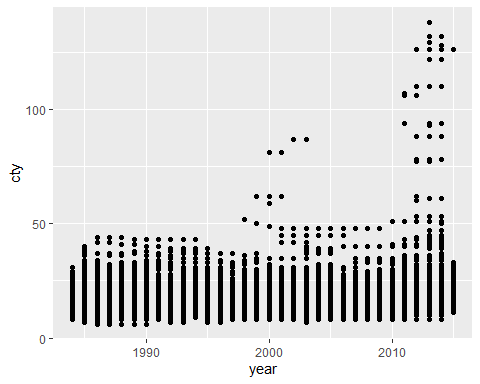
1. Predict the cty for the drive of 2-Wheel Drive. 9A.

predictedDrive = (ctyDriveFWDCoef \* 1) + ctyDriveYInt  
predictedDrive

## [1] 20.992

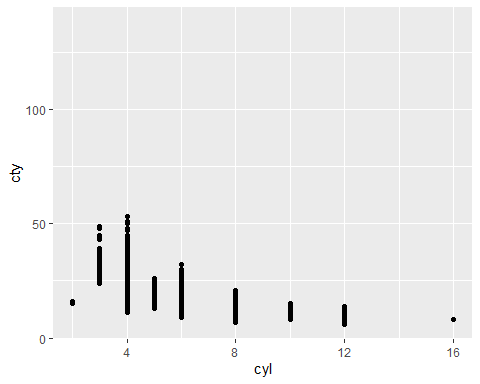
1. Make scatterplots for the quantitative variables versus cty. Which looks like it is going to make the best linear model. Why? 10A. City mpg and hwy mpg plotted against eachother seem like the best for a linear model. The spread of data indicate that generally as cty increases, so does hwy.

ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = year, y = cty))



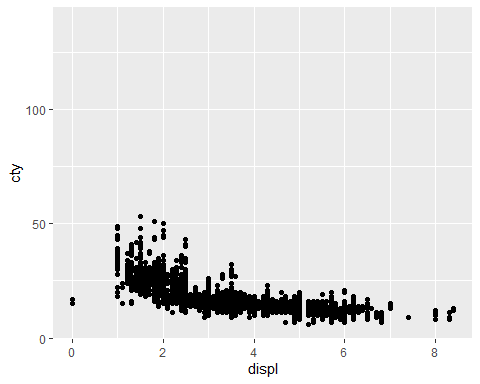
ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = cyl, y = cty))

## Warning: Removed 58 rows containing missing values (geom\_point).

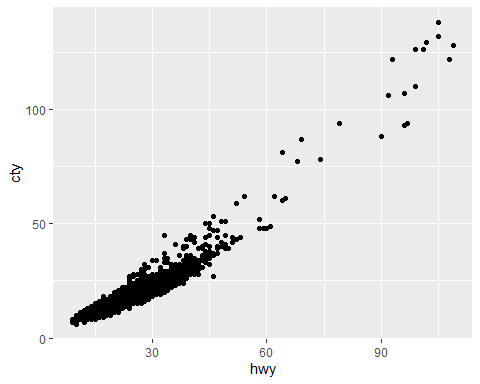


ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = displ, y = cty))

## Warning: Removed 57 rows containing missing values (geom\_point).

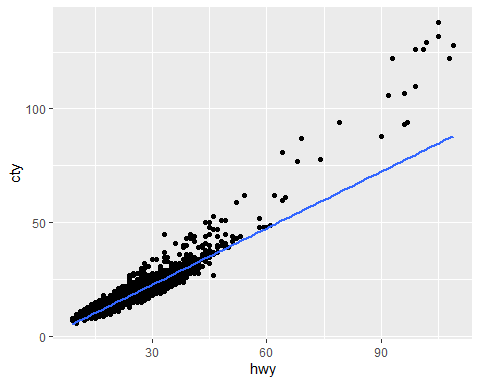


ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = hwy, y = cty))



1. Make a linear model based on either your best pick in 10 or the second best pick if displ was best pick and graph it. 11A.

ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = hwy, y = cty)) +  
 geom\_smooth(mapping = aes(x = hwy, y = cty), method = "lm", se = 0)



12.Which of the models predicting hwy would you prefer to use and why? 12A. I would use the locally estimated scatterplot smoothing method rather than the linear model method as it more closely follows the spread of data.

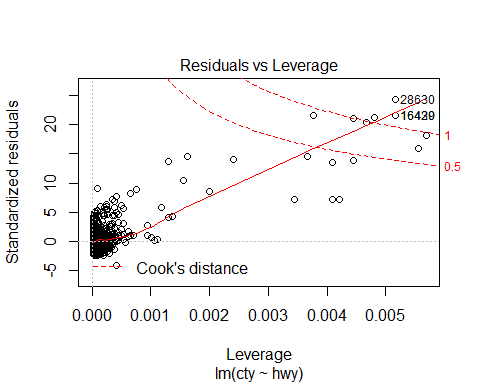
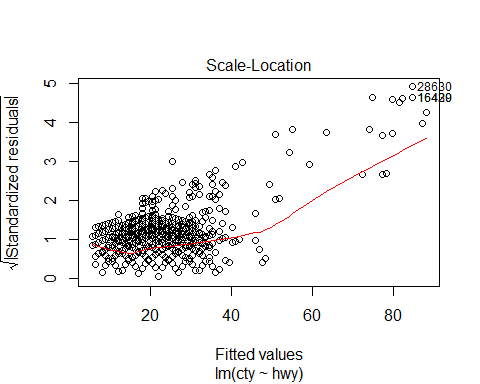
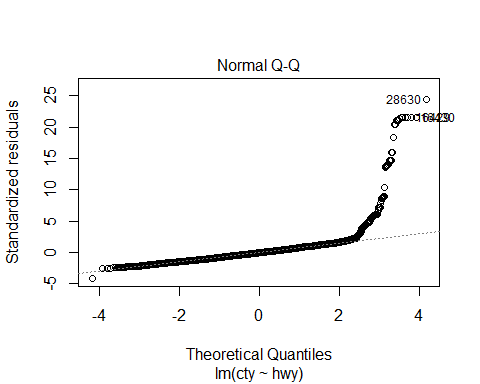
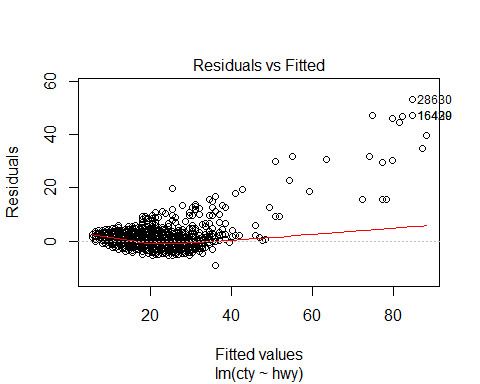
hwyCtyLmModel = lm(cty~hwy, data = vehicles)  
hwyCtyLmModel

##   
## Call:  
## lm(formula = cty ~ hwy, data = vehicles)  
##   
## Coefficients:  
## (Intercept) hwy   
## -1.9775 0.8266

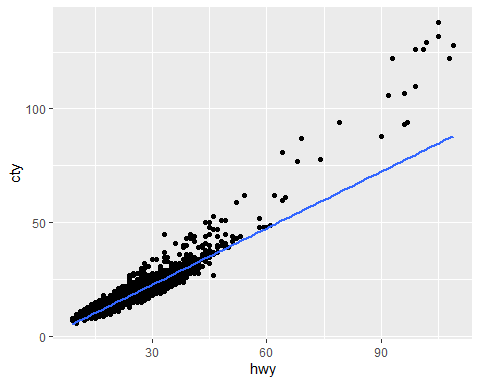
hwyCtyLoessModel = loess(cty~hwy, data = vehicles)  
hwyCtyLoessModel

## Call:  
## loess(formula = cty ~ hwy, data = vehicles)  
##   
## Number of Observations: 33442   
## Equivalent Number of Parameters: 5.55   
## Residual Standard Error: 1.712

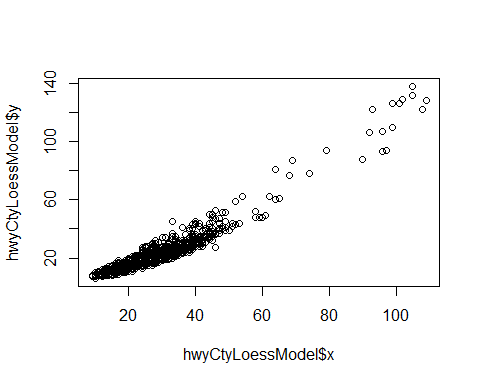
plot(hwyCtyLmModel)



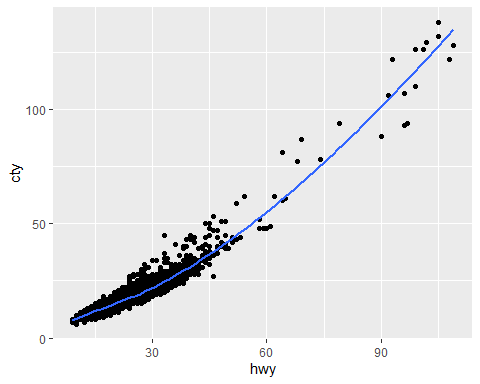
ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = hwy, y = cty)) +  
 geom\_smooth(mapping = aes(x = hwy, y = cty), method = "lm", se = 0)



plot(hwyCtyLoessModel)



ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = hwy, y = cty)) +  
 geom\_smooth(mapping = aes(x = hwy, y = cty), method = "loess", se = 0)



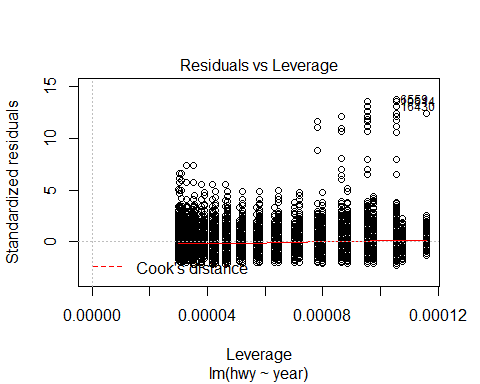
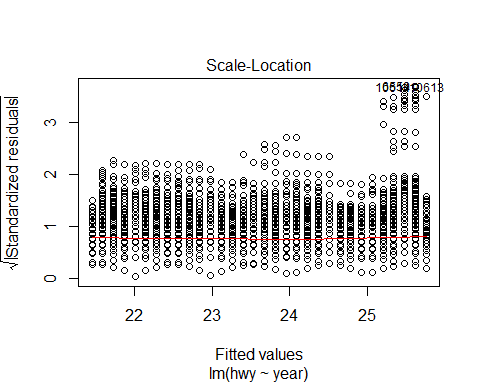
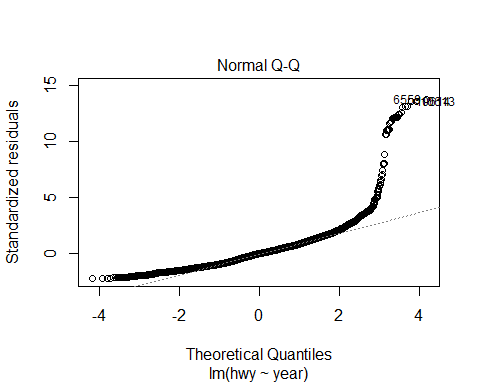
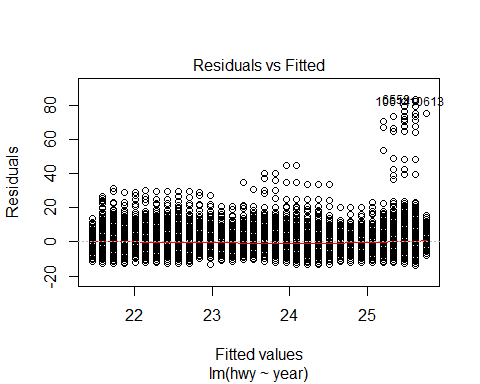
13.Choose a new pair of variables to make a linear model. The response variable must be quanitiative. Graph your model and give an equation. 13A.y = (0.1391 \* x) - 254.4613

hwyYearModel = lm(hwy~year, data = vehicles)  
hwyYearModel

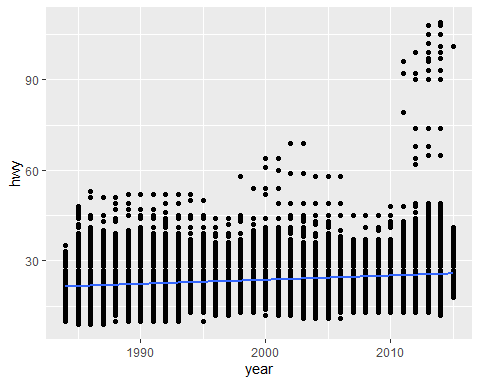
##   
## Call:  
## lm(formula = hwy ~ year, data = vehicles)  
##   
## Coefficients:  
## (Intercept) year   
## -254.4613 0.1391

hwyYearFunc = function(x){  
 (0.1391 \* x) - 254.4613  
}

plot(hwyYearModel)



ggplot(data = vehicles) +   
 geom\_point(mapping = aes(x = year, y = hwy)) +  
 geom\_smooth(mapping = aes(x = year, y = hwy), method = "lm", se = 0)



1. How is the fit? 14A. Considering the data presented, the fit is good - as we progress in time, generally cars are becoming more fuel efficient.
2. Make a prediction with your model. 15A.

hwyYearFunc(2030)

## [1] 27.9117