8 Processing Text Homework - Exercise 5

Andrew Marshall

7/16/2018

# Data

Data for this exercise.

Year <- c(1936, 1946, 1951, 1963, 1975, 1997, 2006)  
CaloriesPerRecipeMean <- c(2123.8, 2122.3, 2089.9, 2250.0, 2234.2, 2249.6, 3051.9)  
CaloriesPerRecipeSD <- c(1050.0, 1002.3, 1009.6, 1078.6, 1089.2, 1094.8, 1496.2)

# General Instructions

* There are 6 exercises. Choose 4 to be graded.
* One of the exercises must be completed in both SAS and R. Make sure you document this in the output.
* The other 3 exercises are to be complete in either R or SAS. Make sure you document this in the output.

# Exercise 1.

Revisit Exercise 1 from Homework 6. Print a list of messages describing the effect size for each unique pair of means from CaloriesPerRecipe. Your messages should look like

The difference between 1936 and 1945 is ####. This is a ???? difference.  
The difference between 1936 and 1951 is ####. This is a ???? difference.  
...

#### should be replaced with absolute value for the difference between pair of means, and ???? will be small, medium or large. Calculate Cohen for each pair, and use to determine small effects and for large effects.

Print each message on a single line. The result will look better if you use cat in R.

If you use SAS, used scan to iterate over the list macro variables CaloriesPerRecipeMean and CaloriesPerRecipeSD. You can put the results to the log.

# Exercise 2.

Calculate MSW, MSB, and for the data from Wansink Table 1, but start with the strings:

Means <- "268.1 271.1 280.9 294.7 285.6 288.6 384.4"  
StandardDeviations <- "124.8 124.2 116.2 117.7 118.3 122.0 168.3"  
SampleSizes <- "18 18 18 18 18 18 18"

Tokenize the strings, then convert the tokens to a create vectors of numeric values. Use these vectors to compute and print MSW, MSB, and , reusing formula from Homework 4 or 6. Name the vectors appropriately to reuse code.

#This will tokenize the strings into relevant vectors  
  
Means <- as.character(Means)  
StandardDeviations <- as.character(StandardDeviations)  
SampleSizes <- as.character(SampleSizes)  
  
MeansToken <- lapply(strsplit(Means,split =" " ),trimws)  
StandardDeviationsToken <- lapply(strsplit(StandardDeviations,split =" " ),trimws)  
SampleSizesToken <- lapply(strsplit(SampleSizes,split =" " ),trimws)  
  
MeansToken

## [[1]]  
## [1] "268.1" "271.1" "280.9" "294.7" "285.6" "288.6" "384.4"

StandardDeviationsToken

## [[1]]  
## [1] "124.8" "124.2" "116.2" "117.7" "118.3" "122.0" "168.3"

SampleSizesToken

## [[1]]  
## [1] "18" "18" "18" "18" "18" "18" "18"

If you use SAS, do this in a macro. Use local macro variables to accumulate sums, and %put to report the results.

Compare your results from previous homework, or to the resource given in previous homework, to confirm that the text was correctly converted to numeric values.

# Exercise 3.

Reproduce the plot from Homework 1, Getting Started.

Tokenize the text below, then convert each token into a pair of values - mean and standard deviation. Use these values to create the vectors CaloriesPerServingMean and CaloriesPerServingSD. You will need to also tokenize the YearRow to create the Year vector.

This is the table row copied almost verbatim from the Markdown table. Note that ‘|’ is a meta-character in regular expressions (or operator), and R will coerce split into a regular expression when possible. Review the parameters for strsplit for options. You can use regular expressions or use fixed pattern matching at your discretion.

TableRow <- "calories per serving (SD) | 268.1 (124.8) | 271.1 (124.2) | 280.9 (116.2) | 294.7 (117.7) | 285.6 (118.3) | 288.6 (122.0) | 384.4 (168.3)"  
YearRow <- "1936 | 1946 | 1951 | 1963 | 1975 | 1997 | 2006"  
  
#This will extract the Year, Mean and SD values from the provided strings  
  
TableRow <- as.character(TableRow)  
  
#CPS data is split into separate units  
CPSTemp <-lapply(strsplit(TableRow,split =" \\|"),trimws)  
CPSTemp

## [[1]]  
## [1] "calories per serving (SD)" "268.1 (124.8)"   
## [3] "271.1 (124.2)" "280.9 (116.2)"   
## [5] "294.7 (117.7)" "285.6 (118.3)"   
## [7] "288.6 (122.0)" "384.4 (168.3)"

#   
#   
# #Extracts Mean values for evaluation  
# CPSTempMean <- CPSTemp[grepl(" [0-9.] \\(",CPSTemp)]  
# as.numeric(CPSTempMean)  
#   
# CPSTempMean  
  
# CPSTempMeanTrim <-lapply(strsplit(CPSTempMean,split ="\\|"),trimws)  
#   
# CPSTempMeanTrim  
  
#Extracts SD values for evaluation  
CPSTempSD <- gsub("(?<=\\()[^()]\*(?=\\))(\*SKIP)(\*F)|.", " ", CPSTemp, perl=T)  
CPSTempSDTrim <-lapply(strsplit(CPSTempSD,split =" \\|"),trimws)  
  
CPSTempSDTrim

## [[1]]  
## [1] "SD 124.8 124.2 116.2 117.7 118.3 122.0 168.3"

# #This converts split string list into data frame for further calculations  
# TableRowMean.df <- data.frame(  
# CPSTempMean  
# )  
# colnames(TableRowMean.df) <- c("CPS.Mean")  
#   
# TableRowMean.df  
  
  
#This converts split string list into data frame for further calculations  
TableRowSD.df <- data.frame(  
 CPSTempSDTrim  
)  
colnames(TableRowSD.df) <- c("CPS.SD")  
  
  
TableRowSD.df

## CPS.SD  
## 1 SD 124.8 124.2 116.2 117.7 118.3 122.0 168.3

#Year data is split into separate units  
Year <- lapply(strsplit(YearRow,split=" \\| "),trimws)  
  
  
#This converts split string list into data frame for further calculations  
Year.df <- data.frame(  
 Year  
)  
colnames(Year.df) <- c("Year")  
Year.df

## Year  
## 1 1936  
## 2 1946  
## 3 1951  
## 4 1963  
## 5 1975  
## 6 1997  
## 7 2006

Run this code (set eval=TRUE) to reproduce the plot.

n <- 18  
alpha <- 0.05  
StandardError <- function(sigma, n) {  
 sigma/sqrt(n)  
}  
ConfidenceInterval <- function(sigma, n) {  
 qt(1-alpha/2, Inf)\*StandardError(sigma,n)  
}  
  
CaloriesPerRecipe <- CaloriesPerServingMean  
Lower <- CaloriesPerRecipe - ConfidenceInterval(CaloriesPerServingSD,n)  
Upper <- CaloriesPerRecipe + ConfidenceInterval(CaloriesPerServingSD,n)  
  
plot(Year, CaloriesPerRecipe,  
 col="blue", pch=19,  
 main="Calories per Recipe",   
 ylab="Calories",   
 ylim=c(min(Lower), max(Upper)))  
lines(Year, CaloriesPerRecipe,  
 lty="dashed", col="blue", lend=2)  
segments(x0=Year,   
 y0=Lower,   
 x1=Year,   
 y1=Upper)

If you choose SAS for this exercise, you will need to create a data table from the text. You can do this in macro language if you wish, but I would recommend doing this in two steps using DATA. I’ve provided a template for the first table.

Add a do loop and scan to ParseTable to tokenize TableRow, outputing one token per table row. Create a second table PlotCookingTooMuch from ParseTable by splitting the tokens into strings, one each for mean and standard deviation, then use the input function to convert each string to a numeric value.

You can do this in one DATA step, if you wish, but I found it easier to debug in two steps.

The code in the comments will plot this table.

# Exercise 4.

Download the two files zero.to.60.csv and quarter.mile.csv. These are records of motorcycle performance for a standing start to 60 mph and for quarter mile time. Each table has a column identifying the make and model for each entry, but this name of the column is different for each table.

## Part a.

There are some duplicates, so compute a mean of Time for each motorcycle, from both tables.

#Assigning path to CSV file to variable PathToZeroTo60  
PathToZeroTo60 = "C:/Users/drewm/Documents/GitHub/code-stat700/zero.to.60.csv"  
  
#Assigning path to CSV file to variable PathToZeroTo60  
PathToquartermile = "C:/Users/drewm/Documents/GitHub/code-stat700/quarter.mile.csv"  
  
#Assigning data from CSV file to data frame  
ZeroTo60.df <- read.delim(PathToZeroTo60,header=TRUE,skip= 0,sep = ",",as.is=TRUE)  
  
#Displaying data in dataframe  
ZeroTo60.df

## Make.and.model Year Time  
## 1 Suzuki GSX-R1000 2006 2.35  
## 2 Suzuki Hayabusa 2002 2.47  
## 3 Yamaha VMAX 2010 2.50  
## 4 Kawasaki ZX-12R 2002 2.59  
## 5 BMW S1000R 2014 2.60  
## 6 BMW S1000RR 2013 2.60  
## 7 Ducati 1199 Panigale 2014 2.60  
## 8 Ducati Diavel 2015 2.60  
## 9 EBR 1190RX 2014 2.60  
## 10 Honda CBR1000RR SP 2013 2.60  
## 11 Kawasaki Ninja ZX-14R 2012 2.60  
## 12 Yamaha YZF-R1 2015 2.60  
## 13 Kawasaki Ninja H2 2015 2.60  
## 14 Kawasaki ZX-10R 2006 2.60  
## 15 Ducati 1198S 2010 2.70  
## 16 Ducati 899 Panigale 2014 2.70  
## 17 Kawasaki ZX-12R 2000 2.70  
## 18 Yamaha FZ1 2001 2.70  
## 19 Yamaha MT-09 (FZ-09) 2013 2.70  
## 20 BMW S1000RR 2011 2.70  
## 21 Suzuki Hayabusa 2010 2.74  
## 22 BMW K1300S 2009 2.79  
## 23 Ducati 959 Panigale 2016 2.80  
## 24 KTM 1190 Adventure 2014 2.80  
## 25 Suzuki GSX-R600 2003 2.80  
## 26 Yamaha YZF-R6 2003 2.80  
## 27 BMW K1300R 2010 2.81  
## 28 BMW K1200R 2006 2.85  
## 29 Suzuki B-King 2008 2.86  
## 30 Yamaha YZF-R1 2010 2.88  
## 31 Suzuki GSX1200W Inazuma 1998 2.88  
## 32 Ducati 1199 Panigale S 2013 2.89  
## 33 BMW S1000XR 2015 2.80  
## 34 BMW R nineT 2016 2.90  
## 35 BMW R1200GS 2014 2.90  
## 36 Ducati Streetfighter 848 2012 2.90  
## 37 Suzuki GSX-R750 2013 2.90  
## 38 Yamaha FZR1000 1990 2.90  
## 39 Yamaha MT-09 (FZ-09) 2015 2.90  
## 40 Yamaha MT-10/FZ-10 2017 2.90  
## 41 Yamaha YZF-R1 2016 2.90  
## 42 Ducati 999 2003 2.90  
## 43 Ducati 749S 2003 2.90  
## 44 Kawasaki ZX-6R 2003 2.90  
## 45 Honda CBR900RR 1994 2.90  
## 46 Suzuki RF900 1994 2.90  
## 47 Kawasaki 1400GTR/Concours 14 2011 2.90  
## 48 Honda CBR1100XX 1999 2.91  
## 49 Ducati Desmosedici RR 2008 2.96  
## 50 Aprilia Tuono RSV-R 2003 3.00  
## 51 Kawasaki GPZ900R 1984 3.00  
## 52 Kawasaki Ninja ZX-9R 1994 3.00  
## 53 Suzuki Bandit 1250S 2007 3.00  
## 54 Suzuki GSX-R1100 1986 3.00  
## 55 BMW R1200RT 2014 3.00  
## 56 Kawasaki Ninja ZX-6R 2006 3.06  
## 57 Suzuki SV1000s 2003 3.08  
## 58 Ducati Multistrada 1200S 2010 3.09  
## 59 Honda CBR600F4i 2003 3.10  
## 60 Buell 1125R 2008 3.15  
## 61 Suzuki TL1000R 1998 3.22  
## 62 Triumph Speed Triple 2006 3.25  
## 63 Honda CBR600 1990 3.30  
## 64 Honda V65 Sabre 1984 3.30  
## 65 Kawasaki ZZ-R600 1990 3.30  
## 66 Triumph Rocket III Roadster 2010 3.30  
## 67 Zero SR 2014 3.30  
## 68 Honda CBR600RR 2003 3.30  
## 69 Yamaha VMAX 1997 3.33  
## 70 Ducati 1299 Panigale S 2015 3.33  
## 71 Suzuki GSX-R1000 2006 3.35  
## 72 Triumph Street Triple R 2009 3.36  
## 73 Harley-Davidson VRSCR 2006 3.40  
## 74 Honda CBR600F 1987 3.40  
## 75 Kawasaki Zephyr / ZR1100 1993 3.40  
## 76 Suzuki TL1000S 1997 3.40  
## 77 Ducati Sport1000 2000 3.44  
## 78 Kawasaki Ninja ZX-7R 1996 3.44  
## 79 Yamaha FZ6 2004 3.44  
## 80 Triumph Daytona 900 1994 3.40  
## 81 Suzuki GSX600F 1990 3.50

#Assigning data from CSV file to data frame  
Quartermile.df <- read.delim(PathToquartermile,header=TRUE,skip= 0,sep = ",",as.is=TRUE)  
  
#Displaying data in dataframe  
Quartermile.df

## Model.Year Motorcycle Time  
## 1 2012 Kawasaki Ninja ZX-14R 9.47  
## 2 2008 Ducati Desmosedici RR 9.49  
## 3 2015 Kawasaki Ninja H2 9.62  
## 4 2008 Suzuki Hayabusa 9.70  
## 5 2010 Kawasaki ZX-10R 9.72  
## 6 2008 Ducati 1098R 9.75  
## 7 2013 BMW HP4 9.76  
## 8 2006 Kawasaki ZX-10R 9.76  
## 9 2015 Yamaha YZF-R1 9.83  
## 10 2014 Ducati 1199 Panigale 9.84  
## 11 2002 Kawasaki ZX-12R 9.87  
## 12 2004 Yamaha YZF-R1S 9.90  
## 13 2011 Ducati 1199 Panigale S 9.91  
## 14 2011 BMW S1000RR 9.93  
## 15 2014 BMW S1000R 9.94  
## 16 1998 Kawasaki ZX-9R 9.99  
## 17 2008 Suzuki B-King 9.99  
## 18 2010 Ducati 1198S 10.00  
## 19 2012 Honda CBR1000RR 10.00  
## 20 2012 Suzuki GSX-R1000 10.03  
## 21 2010 Yamaha YZF-R1 10.05  
## 22 2016 Yamaha YZF-R1 10.11  
## 23 2010 Yamaha VMAX 10.11  
## 24 2014 EBR 1190RX 10.12  
## 25 2002 Kawasaki ZZ-R1200 10.12  
## 26 2012 Kawasaki ZX-10R ABS 10.14  
## 27 2010 Aprilia RSV4 Factory 10.16  
## 28 2015 Ducati 1299 Panigale S 10.18  
## 29 1997 Honda CBR1100XX 10.20  
## 30 2009 BMW K1300S 10.22  
## 31 2012 Yamaha YZF-R1 10.24  
## 32 2013 KTM 1190 RC8 R 10.26  
## 33 2006 BMW K1200R 10.30  
## 34 2008 Ducati 1098 10.31  
## 35 2013 Kawasaki ZX-10R ABS 10.32  
## 36 2003 Ducati 999 10.36  
## 37 1997 Yamaha YZF1000R 10.38  
## 38 2013 MV Agusta F4RR 10.40  
## 39 2013 Suzuki GSX-R750 10.41  
## 40 1997 Kawasaki ZX-11 10.42  
## 41 2005 Honda RC51 10.44  
## 42 2003 Aprilia Tuono RSV-R 10.49  
## 43 2010 MV Agusta Brutale 1090RR 10.50  
## 44 1994 Suzuki RF900 10.50  
## 45 2008 Buell 1125R 10.51  
## 46 2014 Triumph Speed Triple R 10.55  
## 47 1991 Suzuki GSX-R1100M 10.55  
## 48 2011 Kawasaki 1400GTR/Concours 14 10.56  
## 49 2006 Suzuki GSX-R1000 10.58  
## 50 2003 Suzuki SV1000s 10.59  
## 51 1994 Honda CBR900RR 10.60  
## 52 2001 Yamaha FZ1 10.62  
## 53 2010 Ducati Multistrada 1200S 10.62  
## 54 2011 Ducati Diavel 10.63  
## 55 2013 Yamaha MT-09 (FZ-09) 10.66  
## 56 2003 Honda CBR600RR 10.68  
## 57 1994 Kawasaki Ninja ZX-9R 10.70  
## 58 2003 Kawasaki ZX-6R 10.73  
## 59 1997 Ducati 916 10.77  
## 60 1991 Kawasaki ZZR-1100 10.78  
## 61 2006 Kawasaki Ninja ZX-6R 10.78  
## 62 1998 Suzuki TL1000R 10.79  
## 63 2003 Suzuki GSX-R600 10.79  
## 64 2006 Triumph Speed Triple 10.82  
## 65 1983 Honda V65 Magna 10.84  
## 66 2003 Yamaha YZF-R6 10.84  
## 67 2002 MV Agusta F4 Senna 10.85  
## 68 1997 Suzuki TL1000S 10.85  
## 69 1991 Yamaha FZR1000RU 10.87  
## 70 1984 Suzuki GS1150ES 10.94  
## 71 2012 Ducati Streetfighter 848 10.94  
## 72 1986 Honda VFR750F 10.95  
## 73 2006 Triumph Daytona 675 10.96  
## 74 2015 Yamaha MT-09 (FZ-09) 10.97  
## 75 2013 Yamaha FJR1300A 10.98  
## 76 2001 Aprilia RST1000 Futura 11.04  
## 77 1986 Suzuki GSX-R1100 11.04  
## 78 1983 Honda V65 Magna 11.07  
## 79 1997 Yamaha VMAX 11.09  
## 80 2003 Honda CBR600F4i 11.11  
## 81 1983 Honda CB1100F 11.13  
## 82 2003 Ducati 749S 11.15  
## 83 2008 Triumph Street Triple 11.17  
## 84 1996 Kawasaki Ninja ZX-7R 11.17  
## 85 1981 Kawasaki GPZ1100 11.18  
## 86 1984 Kawasaki GPZ900R 11.18  
## 87 1993 Honda CBR1000F 11.19  
## 88 2014 BMW R1200RT 11.22  
## 89 1981 Suzuki Katana 11.32  
## 90 2004 Yamaha FZ6 11.32  
## 91 2012 Triumph Tiger Explorer 1200 11.32  
## 92 1979 Honda CBX 11.36  
## 93 2001 Harley-Davidson VRSC V-Rod 11.38  
## 94 1980 Suzuki GS1100E 11.39  
## 95 2011 BMW F800R 11.39  
## 96 1994 Triumph Daytona 900 11.40  
## 97 1984 Kawasaki GPZ750 Turbo 11.40  
## 98 2004 Triumph Rocket III 11.44  
## 99 2010 Triumph Rocket III Roadster 11.48  
## 100 2008 Moto Guzzi Griso 1200 8V 11.51  
## 101 1990 Kawasaki ZZ-R600 11.55  
## 102 2003 Ducati Multistrada 11.60  
## 103 2006 Suzuki Boulevard M109R 11.62  
## 104 2003 Suzuki SV650s 11.63  
## 105 2009 Ducati Sport1000 11.64  
## 106 2002 Triumph Speed Four 11.65  
## 107 1985 Yamaha XJ750X Maxim-X 11.66  
## 108 1991 Triumph Trophy 1200 (120ps) 11.70  
## 109 1983 Honda CX650 Turbo 11.75  
## 110 1978 Yamaha XS1100 11.78  
## 111 1999 BMW R1100S 11.78  
## 112 1978 Suzuki GS1000 11.83  
## 113 2005 Ducati Multistrada 1000DS 11.88  
## 114 2011 Yamaha XT1200Z Super T??n??r?? 11.89  
## 115 1971 Norton Dunstall 810 11.90  
## 116 2007 Victory 8-Ball 11.91  
## 117 2015 Indian Scout 11.93  
## 118 1979 Kawasaki KZ1300 11.93  
## 119 1982 Kawasaki GPZ750 11.93  
## 120 1980 Honda CBX 11.93  
## 121 1987 Honda CBR600F 11.94  
## 122 1993 Kawasaki Zephyr / ZR1100 11.95  
## 123 1993 Triumph Trophy 1200 (110ps) 12.00  
## Final.speed  
## 1 152.83 mph (245.96 km/h)  
## 2 152.80 mph (245.91 km/h)  
## 3 152.01 mph (244.64 km/h)  
## 4 148.48 mph (238.96 km/h)  
## 5 150.0 mph (241.4 km/h)  
## 6 148.6 mph (239.1 km/h)  
## 7 152.4 mph (245.3 km/h)  
## 8 149.08 mph (239.92 km/h)  
## 9 149.91 mph (241.26 km/h)  
## 10 145.68 mph (234.45 km/h)  
## 11 146.29 mph (235.43 km/h)  
## 12 144.98 mph (233.32 km/h)  
## 13 145.95 mph (234.88 km/h)  
## 14 149.8 mph (241.1 km/h)  
## 15 141.74 mph (228.11 km/h)  
## 16 136.80 mph (220.16 km/h)  
## 17 138.42 mph (222.77 km/h)  
## 18 144.9 mph (233.2 km/h)  
## 19 141.7 mph (228.0 km/h)  
## 20 143.8 mph (231.4 km/h)  
## 21 138.51 mph (222.91 km/h)  
## 22 146.62 mph (235.96 km/h)  
## 23 137 mph (220 km/h)  
## 24 140.29 mph (225.77 km/h)  
## 25 136.9 mph (220.3 km/h)  
## 26 144.4 mph (232.4 km/h)  
## 27 142.75 mph (229.73 km/h)  
## 28 151.56 mph (243.91 km/h)  
## 29 136.1 mph (219.0 km/h)  
## 30 135.04 mph (217.33 km/h)  
## 31 140.0 mph (225.3 km/h)  
## 32 137.90 mph (221.93 km/h)  
## 33 132.76 mph (213.66 km/h)  
## 34 142.56 mph (229.43 km/h)  
## 35 140.87 mph (226.71 km/h)  
## 36 132.89 mph (213.87 km/h)  
## 37 132.72 mph (213.59 km/h)  
## 38 140.98 mph (226.89 km/h)  
## 39 135.89 mph (218.69 km/h)  
## 40 133.02 mph (214.07 km/h)  
## 41 133.5 mph (214.8 km/h)  
## 42 132.15 mph (212.67 km/h)  
## 43 130.98 mph (210.79 km/h)  
## 44 133.1 mph (214.2 km/h)  
## 45 134.32 mph (216.17 km/h)  
## 46 129.32 mph (208.12 km/h)  
## 47 133.7 mph (215.2 km/h)  
## 48 127.68 mph (205.48 km/h)  
## 49 144.25 mph (232.15 km/h)  
## 50 130.81 mph (210.52 km/h)  
## 51 131.1 mph (211.0 km/h)  
## 52 130.02 mph (209.25 km/h)  
## 53 127.62 mph (205.38 km/h)  
## 54 128 mph (206 km/h)  
## 55 125.24 mph (201.55 km/h)  
## 56 130.07 mph (209.33 km/h)  
## 57 129.7 mph (208.7 km/h)  
## 58 127.95 mph (205.92 km/h)  
## 59 126.3 mph (203.3 km/h)  
## 60 132 mph (212 km/h)  
## 61 127.12 mph (204.58 km/h)  
## 62 129.54 mph (208.47 km/h)  
## 63 127.89 mph (205.82 km/h)  
## 64 125.01 mph (201.18 km/h)  
## 65 124.82 mph (200.88 km/h)  
## 66 127.81 mph (205.69 km/h)  
## 67 131.85 mph (212.19 km/h)  
## 68 127.70 mph (205.51 km/h)  
## 69 130 mph (210 km/h)  
## 70 125 mph (201 km/h)  
## 71 125.58 mph (202.10 km/h)  
## 72 113.95 mph (183.38 km/h)  
## 73 127.32 mph (204.90 km/h)  
## 74 122.58 mph (197.27 km/h)  
## 75 122.29 mph (196.81 km/h)  
## 76 121.12 mph (194.92 km/h)  
## 77 123.13 mph (198.16 km/h)  
## 78 123.62 mph (198.95 km/h)  
## 79 121.49 mph (195.52 km/h)  
## 80 123.74 mph (199.14 km/h)  
## 81 120.48 mph (193.89 km/h)  
## 82 123.04 mph (198.01 km/h)  
## 83 120.06 mph (193.22 km/h)  
## 84 122.27 mph (196.77 km/h)  
## 85 119.10 mph (191.67 km/h)  
## 86 121.65 mph (195.78 km/h)  
## 87 121.24 mph (195.12 km/h)  
## 88 119.24 mph (191.90 km/h)  
## 89 120.00 mph (193.12 km/h)  
## 90 118.01 mph (189.92 km/h)  
## 91 118.07 mph (190.02 km/h)  
## 92 118.11 mph (190.08 km/h)  
## 93 119.07 mph (191.62 km/h)  
## 94 118.42 mph (190.58 km/h)  
## 95 116.83 mph (188.02 km/h)  
## 96 119.1 mph (191.7 km/h)  
## 97 118.42 mph (190.58 km/h)  
## 98 118.33 mph (190.43 km/h)  
## 99 115 mph (185 km/h)  
## 100 116.00 mph (186.68 km/h)  
## 101 119.00 mph (191.51 km/h)  
## 102 113.77 mph (183.10 km/h)  
## 103 114.91 mph (184.93 km/h)  
## 104 114.41 mph (184.13 km/h)  
## 105 136.50 mph (219.68 km/h)  
## 106 114.9 mph (184.9 km/h)  
## 107 116.1 mph (186.8 km/h)  
## 108 120 mph (190 km/h)  
## 109 112.21 mph (180.58 km/h)  
## 110 114.21 mph (183.80 km/h)  
## 111 114.73 mph (184.64 km/h)  
## 112 107.88 mph (173.62 km/h)  
## 113 113.10 mph (182.02 km/h)  
## 114 109.57 mph (176.34 km/h)  
## 115 107.88 mph (173.62 km/h)  
## 116 113.26 mph (182.27 km/h)  
## 117 110.05 mph (177.11 km/h)  
## 118 114.79 mph (184.74 km/h)  
## 119 109.62 mph (176.42 km/h)  
## 120 114.06 mph (183.56 km/h)  
## 121 111.42 mph (179.31 km/h)  
## 122 114.5 mph (184.3 km/h)  
## 123 117 mph (188 km/h)

ZeroTo60Mean <- mean(ZeroTo60.df$Time)  
ZeroTo60Mean

## [1] 2.953333

QuartermileMean <- mean(Quartermile.df$Time)  
QuartermileMean

## [1] 10.80789

## Part b.

Create a new table with these means, but use only those motorcycles that are in both tables. You will need to merge these by names.

# TableMatch <- merge.data.frame(ZeroTo60.df,Quartermile.df,by.x = 'Make.and.model',by.y = 'Motorcycle')  
TableMatch <- merge(ZeroTo60.df,Quartermile.df,by.x = c('Make.and.model','Year'), by.y = c('Motorcycle','Model.Year'))  
  
#Renaming Columns  
colnames(TableMatch) <- c("Make and Model","Year","0-to-60 Time","Quarter Mile Time", "Final Speed")  
  
#This will sort the merged table by year  
TableMatch[order(TableMatch$Year),]

## Make and Model Year 0-to-60 Time Quarter Mile Time  
## 22 Kawasaki GPZ900R 1984 3.00 11.18  
## 36 Suzuki GSX-R1100 1986 3.00 11.04  
## 17 Honda CBR600F 1987 3.40 11.94  
## 32 Kawasaki ZZ-R600 1990 3.30 11.55  
## 28 Kawasaki Zephyr / ZR1100 1993 3.40 11.95  
## 20 Honda CBR900RR 1994 2.90 10.60  
## 27 Kawasaki Ninja ZX-9R 1994 3.00 10.70  
## 39 Suzuki RF900 1994 2.90 10.50  
## 43 Triumph Daytona 900 1994 3.40 11.40  
## 26 Kawasaki Ninja ZX-7R 1996 3.44 11.17  
## 42 Suzuki TL1000S 1997 3.40 10.85  
## 50 Yamaha VMAX 1997 3.33 11.09  
## 41 Suzuki TL1000R 1998 3.22 10.79  
## 46 Yamaha FZ1 2001 2.70 10.62  
## 30 Kawasaki ZX-12R 2002 2.59 9.87  
## 1 Aprilia Tuono RSV-R 2003 3.00 10.49  
## 11 Ducati 749S 2003 2.90 11.15  
## 12 Ducati 999 2003 2.90 10.36  
## 18 Honda CBR600F4i 2003 3.10 11.11  
## 19 Honda CBR600RR 2003 3.30 10.68  
## 31 Kawasaki ZX-6R 2003 2.90 10.73  
## 37 Suzuki GSX-R600 2003 2.80 10.79  
## 40 Suzuki SV1000s 2003 3.08 10.59  
## 55 Yamaha YZF-R6 2003 2.80 10.84  
## 47 Yamaha FZ6 2004 3.44 11.32  
## 2 BMW K1200R 2006 2.85 10.30  
## 25 Kawasaki Ninja ZX-6R 2006 3.06 10.78  
## 29 Kawasaki ZX-10R 2006 2.60 9.76  
## 34 Suzuki GSX-R1000 2006 3.35 10.58  
## 35 Suzuki GSX-R1000 2006 2.35 10.58  
## 45 Triumph Speed Triple 2006 3.25 10.82  
## 7 Buell 1125R 2008 3.15 10.51  
## 13 Ducati Desmosedici RR 2008 2.96 9.49  
## 33 Suzuki B-King 2008 2.86 9.99  
## 3 BMW K1300S 2009 2.79 10.22  
## 8 Ducati 1198S 2010 2.70 10.00  
## 14 Ducati Multistrada 1200S 2010 3.09 10.62  
## 44 Triumph Rocket III Roadster 2010 3.30 11.48  
## 51 Yamaha VMAX 2010 2.50 10.11  
## 52 Yamaha YZF-R1 2010 2.88 10.05  
## 6 BMW S1000RR 2011 2.70 9.93  
## 21 Kawasaki 1400GTR/Concours 14 2011 2.90 10.56  
## 15 Ducati Streetfighter 848 2012 2.90 10.94  
## 24 Kawasaki Ninja ZX-14R 2012 2.60 9.47  
## 38 Suzuki GSX-R750 2013 2.90 10.41  
## 48 Yamaha MT-09 (FZ-09) 2013 2.70 10.66  
## 4 BMW R1200RT 2014 3.00 11.22  
## 5 BMW S1000R 2014 2.60 9.94  
## 9 Ducati 1199 Panigale 2014 2.60 9.84  
## 16 EBR 1190RX 2014 2.60 10.12  
## 10 Ducati 1299 Panigale S 2015 3.33 10.18  
## 23 Kawasaki Ninja H2 2015 2.60 9.62  
## 49 Yamaha MT-09 (FZ-09) 2015 2.90 10.97  
## 53 Yamaha YZF-R1 2015 2.60 9.83  
## 54 Yamaha YZF-R1 2016 2.90 10.11  
## Final Speed  
## 22 121.65 mph (195.78 km/h)  
## 36 123.13 mph (198.16 km/h)  
## 17 111.42 mph (179.31 km/h)  
## 32 119.00 mph (191.51 km/h)  
## 28 114.5 mph (184.3 km/h)  
## 20 131.1 mph (211.0 km/h)  
## 27 129.7 mph (208.7 km/h)  
## 39 133.1 mph (214.2 km/h)  
## 43 119.1 mph (191.7 km/h)  
## 26 122.27 mph (196.77 km/h)  
## 42 127.70 mph (205.51 km/h)  
## 50 121.49 mph (195.52 km/h)  
## 41 129.54 mph (208.47 km/h)  
## 46 130.02 mph (209.25 km/h)  
## 30 146.29 mph (235.43 km/h)  
## 1 132.15 mph (212.67 km/h)  
## 11 123.04 mph (198.01 km/h)  
## 12 132.89 mph (213.87 km/h)  
## 18 123.74 mph (199.14 km/h)  
## 19 130.07 mph (209.33 km/h)  
## 31 127.95 mph (205.92 km/h)  
## 37 127.89 mph (205.82 km/h)  
## 40 130.81 mph (210.52 km/h)  
## 55 127.81 mph (205.69 km/h)  
## 47 118.01 mph (189.92 km/h)  
## 2 132.76 mph (213.66 km/h)  
## 25 127.12 mph (204.58 km/h)  
## 29 149.08 mph (239.92 km/h)  
## 34 144.25 mph (232.15 km/h)  
## 35 144.25 mph (232.15 km/h)  
## 45 125.01 mph (201.18 km/h)  
## 7 134.32 mph (216.17 km/h)  
## 13 152.80 mph (245.91 km/h)  
## 33 138.42 mph (222.77 km/h)  
## 3 135.04 mph (217.33 km/h)  
## 8 144.9 mph (233.2 km/h)  
## 14 127.62 mph (205.38 km/h)  
## 44 115 mph (185 km/h)  
## 51 137 mph (220 km/h)  
## 52 138.51 mph (222.91 km/h)  
## 6 149.8 mph (241.1 km/h)  
## 21 127.68 mph (205.48 km/h)  
## 15 125.58 mph (202.10 km/h)  
## 24 152.83 mph (245.96 km/h)  
## 38 135.89 mph (218.69 km/h)  
## 48 125.24 mph (201.55 km/h)  
## 4 119.24 mph (191.90 km/h)  
## 5 141.74 mph (228.11 km/h)  
## 9 145.68 mph (234.45 km/h)  
## 16 140.29 mph (225.77 km/h)  
## 10 151.56 mph (243.91 km/h)  
## 23 152.01 mph (244.64 km/h)  
## 49 122.58 mph (197.27 km/h)  
## 53 149.91 mph (241.26 km/h)  
## 54 146.62 mph (235.96 km/h)

#This will provide the mean for the newly merged columns.  
mean(TableMatch$'0-to-60 Time')

## [1] 2.958545

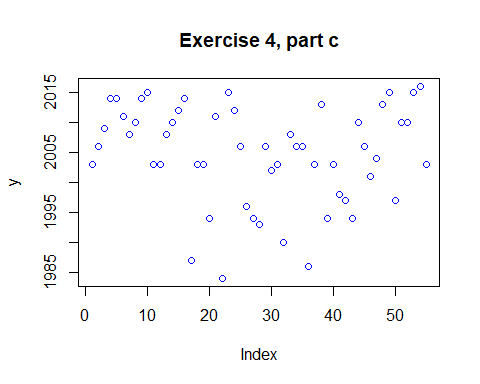
mean(TableMatch$'Quarter Mile Time')

## [1] 10.58909

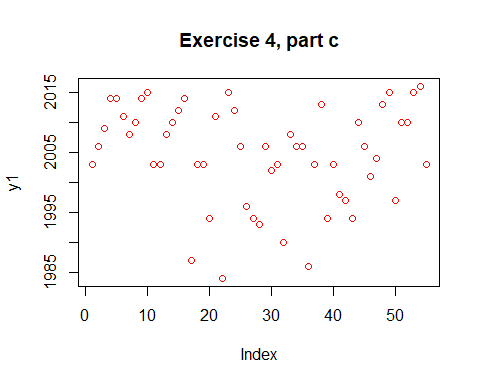
## Part c.

Plot the relationship between 0-to-60 time and quarter mile time.

#Assignments for time y plots  
x <- c(TableMatch$Year)  
y <- c(TableMatch$Time.x)  
  
#Assignments for time X plots  
x1 <- c(TableMatch$Year)  
y1 <- c(TableMatch$Time.y)  
  
  
# Relationship between 0-to-60 time and quarter-mile time  
plot(x,y,main = "Exercise 4, part c",col="blue",type="p")



plot(x,y1,main = "Exercise 4, part c",col="red",type="p")



# Exercise 5.

Read either file, zero.to.60.csv or quarter.mile.csv, from Exercise 4 into a table. Use partial matching to show the following sets of entries. You can assume make is the first word in the motorcycle name, and the model are the remaining words.

## Part a

What entries in this list were made by BMW?

#Assigning path to CSV file to variable PathToZeroTo60  
PathToZeroTo60 = "C:/Users/drewm/Documents/GitHub/code-stat700/zero.to.60.csv"  
  
#Assigning data from CSV file to data frame  
ZeroTo60.df <- read.delim(PathToZeroTo60,header=TRUE,skip= 0,sep = ",",as.is=TRUE)  
  
#Displaying data in dataframe  
ZeroTo60.df

## Make.and.model Year Time  
## 1 Suzuki GSX-R1000 2006 2.35  
## 2 Suzuki Hayabusa 2002 2.47  
## 3 Yamaha VMAX 2010 2.50  
## 4 Kawasaki ZX-12R 2002 2.59  
## 5 BMW S1000R 2014 2.60  
## 6 BMW S1000RR 2013 2.60  
## 7 Ducati 1199 Panigale 2014 2.60  
## 8 Ducati Diavel 2015 2.60  
## 9 EBR 1190RX 2014 2.60  
## 10 Honda CBR1000RR SP 2013 2.60  
## 11 Kawasaki Ninja ZX-14R 2012 2.60  
## 12 Yamaha YZF-R1 2015 2.60  
## 13 Kawasaki Ninja H2 2015 2.60  
## 14 Kawasaki ZX-10R 2006 2.60  
## 15 Ducati 1198S 2010 2.70  
## 16 Ducati 899 Panigale 2014 2.70  
## 17 Kawasaki ZX-12R 2000 2.70  
## 18 Yamaha FZ1 2001 2.70  
## 19 Yamaha MT-09 (FZ-09) 2013 2.70  
## 20 BMW S1000RR 2011 2.70  
## 21 Suzuki Hayabusa 2010 2.74  
## 22 BMW K1300S 2009 2.79  
## 23 Ducati 959 Panigale 2016 2.80  
## 24 KTM 1190 Adventure 2014 2.80  
## 25 Suzuki GSX-R600 2003 2.80  
## 26 Yamaha YZF-R6 2003 2.80  
## 27 BMW K1300R 2010 2.81  
## 28 BMW K1200R 2006 2.85  
## 29 Suzuki B-King 2008 2.86  
## 30 Yamaha YZF-R1 2010 2.88  
## 31 Suzuki GSX1200W Inazuma 1998 2.88  
## 32 Ducati 1199 Panigale S 2013 2.89  
## 33 BMW S1000XR 2015 2.80  
## 34 BMW R nineT 2016 2.90  
## 35 BMW R1200GS 2014 2.90  
## 36 Ducati Streetfighter 848 2012 2.90  
## 37 Suzuki GSX-R750 2013 2.90  
## 38 Yamaha FZR1000 1990 2.90  
## 39 Yamaha MT-09 (FZ-09) 2015 2.90  
## 40 Yamaha MT-10/FZ-10 2017 2.90  
## 41 Yamaha YZF-R1 2016 2.90  
## 42 Ducati 999 2003 2.90  
## 43 Ducati 749S 2003 2.90  
## 44 Kawasaki ZX-6R 2003 2.90  
## 45 Honda CBR900RR 1994 2.90  
## 46 Suzuki RF900 1994 2.90  
## 47 Kawasaki 1400GTR/Concours 14 2011 2.90  
## 48 Honda CBR1100XX 1999 2.91  
## 49 Ducati Desmosedici RR 2008 2.96  
## 50 Aprilia Tuono RSV-R 2003 3.00  
## 51 Kawasaki GPZ900R 1984 3.00  
## 52 Kawasaki Ninja ZX-9R 1994 3.00  
## 53 Suzuki Bandit 1250S 2007 3.00  
## 54 Suzuki GSX-R1100 1986 3.00  
## 55 BMW R1200RT 2014 3.00  
## 56 Kawasaki Ninja ZX-6R 2006 3.06  
## 57 Suzuki SV1000s 2003 3.08  
## 58 Ducati Multistrada 1200S 2010 3.09  
## 59 Honda CBR600F4i 2003 3.10  
## 60 Buell 1125R 2008 3.15  
## 61 Suzuki TL1000R 1998 3.22  
## 62 Triumph Speed Triple 2006 3.25  
## 63 Honda CBR600 1990 3.30  
## 64 Honda V65 Sabre 1984 3.30  
## 65 Kawasaki ZZ-R600 1990 3.30  
## 66 Triumph Rocket III Roadster 2010 3.30  
## 67 Zero SR 2014 3.30  
## 68 Honda CBR600RR 2003 3.30  
## 69 Yamaha VMAX 1997 3.33  
## 70 Ducati 1299 Panigale S 2015 3.33  
## 71 Suzuki GSX-R1000 2006 3.35  
## 72 Triumph Street Triple R 2009 3.36  
## 73 Harley-Davidson VRSCR 2006 3.40  
## 74 Honda CBR600F 1987 3.40  
## 75 Kawasaki Zephyr / ZR1100 1993 3.40  
## 76 Suzuki TL1000S 1997 3.40  
## 77 Ducati Sport1000 2000 3.44  
## 78 Kawasaki Ninja ZX-7R 1996 3.44  
## 79 Yamaha FZ6 2004 3.44  
## 80 Triumph Daytona 900 1994 3.40  
## 81 Suzuki GSX600F 1990 3.50

## Part b

Which entries include ’Ninja` in the model name?

#Assigning Make and Model data to "Motorcycles"" from ZeroTO60 dataframe.  
Motorcycles <- as.character(ZeroTo60.df$Make.and.model)  
Motorcycles

## [1] "Suzuki GSX-R1000" "Suzuki Hayabusa"   
## [3] "Yamaha VMAX" "Kawasaki ZX-12R"   
## [5] "BMW S1000R" "BMW S1000RR"   
## [7] "Ducati 1199 Panigale" "Ducati Diavel"   
## [9] "EBR 1190RX" "Honda CBR1000RR SP"   
## [11] "Kawasaki Ninja ZX-14R" "Yamaha YZF-R1"   
## [13] "Kawasaki Ninja H2" "Kawasaki ZX-10R"   
## [15] "Ducati 1198S" "Ducati 899 Panigale"   
## [17] "Kawasaki ZX-12R" "Yamaha FZ1"   
## [19] "Yamaha MT-09 (FZ-09)" "BMW S1000RR"   
## [21] "Suzuki Hayabusa" "BMW K1300S"   
## [23] "Ducati 959 Panigale" "KTM 1190 Adventure"   
## [25] "Suzuki GSX-R600" "Yamaha YZF-R6"   
## [27] "BMW K1300R" "BMW K1200R"   
## [29] "Suzuki B-King" "Yamaha YZF-R1"   
## [31] "Suzuki GSX1200W Inazuma" "Ducati 1199 Panigale S"   
## [33] "BMW S1000XR" "BMW R nineT"   
## [35] "BMW R1200GS" "Ducati Streetfighter 848"   
## [37] "Suzuki GSX-R750" "Yamaha FZR1000"   
## [39] "Yamaha MT-09 (FZ-09)" "Yamaha MT-10/FZ-10"   
## [41] "Yamaha YZF-R1" "Ducati 999"   
## [43] "Ducati 749S" "Kawasaki ZX-6R"   
## [45] "Honda CBR900RR" "Suzuki RF900"   
## [47] "Kawasaki 1400GTR/Concours 14" "Honda CBR1100XX"   
## [49] "Ducati Desmosedici RR" "Aprilia Tuono RSV-R"   
## [51] "Kawasaki GPZ900R" "Kawasaki Ninja ZX-9R"   
## [53] "Suzuki Bandit 1250S" "Suzuki GSX-R1100"   
## [55] "BMW R1200RT" "Kawasaki Ninja ZX-6R"   
## [57] "Suzuki SV1000s" "Ducati Multistrada 1200S"   
## [59] "Honda CBR600F4i" "Buell 1125R"   
## [61] "Suzuki TL1000R" "Triumph Speed Triple"   
## [63] "Honda CBR600" "Honda V65 Sabre"   
## [65] "Kawasaki ZZ-R600" "Triumph Rocket III Roadster"   
## [67] "Zero SR" "Honda CBR600RR"   
## [69] "Yamaha VMAX" "Ducati 1299 Panigale S"   
## [71] "Suzuki GSX-R1000" "Triumph Street Triple R"   
## [73] "Harley-Davidson VRSCR" "Honda CBR600F"   
## [75] "Kawasaki Zephyr / ZR1100" "Suzuki TL1000S"   
## [77] "Ducati Sport1000" "Kawasaki Ninja ZX-7R"   
## [79] "Yamaha FZ6" "Triumph Daytona 900"   
## [81] "Suzuki GSX600F"

#This will find all motorcycles with the name "Ninja in the model name"  
Ninjas <- Motorcycles[grepl("?Ninja?",Motorcycles )]  
Ninjas

## [1] "Kawasaki Ninja ZX-14R" "Kawasaki Ninja H2" "Kawasaki Ninja ZX-9R"   
## [4] "Kawasaki Ninja ZX-6R" "Kawasaki Ninja ZX-7R"

## Part c

List the motorcycle with model names ending with ‘R’ (for racing? I suppose)?

#This will find all motorcycles with model names ending in 'R'  
Racing <- Motorcycles[grepl("R$",Motorcycles )]  
Racing

## [1] "Kawasaki ZX-12R" "BMW S1000R"   
## [3] "BMW S1000RR" "Kawasaki Ninja ZX-14R"   
## [5] "Kawasaki ZX-10R" "Kawasaki ZX-12R"   
## [7] "BMW S1000RR" "BMW K1300R"   
## [9] "BMW K1200R" "BMW S1000XR"   
## [11] "Kawasaki ZX-6R" "Honda CBR900RR"   
## [13] "Ducati Desmosedici RR" "Aprilia Tuono RSV-R"   
## [15] "Kawasaki GPZ900R" "Kawasaki Ninja ZX-9R"   
## [17] "Kawasaki Ninja ZX-6R" "Buell 1125R"   
## [19] "Suzuki TL1000R" "Zero SR"   
## [21] "Honda CBR600RR" "Triumph Street Triple R"  
## [23] "Harley-Davidson VRSCR" "Kawasaki Ninja ZX-7R"

## Part d

List the motorcycles that might be smaller than ‘liter’ bikes (engine size < 1000 cc), based on their name. First, exclude motorcycles with 1 in the name (these will mostly be 1000+ numbers). From that set of names, select those with numbers in range 2-9 in their names.

#Creates list of models 'Small' with '1' anywhere in the name for use in the upcoming filter  
Small <- grepl(".1", Motorcycles)  
  
#Creates list of models 'Smaller' that excludes the models from the previous list  
Smaller <- Motorcycles[!Small]  
Smaller

## [1] "Suzuki Hayabusa" "Yamaha VMAX"   
## [3] "Ducati Diavel" "Kawasaki Ninja H2"   
## [5] "Ducati 899 Panigale" "Yamaha MT-09 (FZ-09)"   
## [7] "Suzuki Hayabusa" "Ducati 959 Panigale"   
## [9] "Suzuki GSX-R600" "Yamaha YZF-R6"   
## [11] "Suzuki B-King" "BMW R nineT"   
## [13] "Ducati Streetfighter 848" "Suzuki GSX-R750"   
## [15] "Yamaha MT-09 (FZ-09)" "Ducati 999"   
## [17] "Ducati 749S" "Kawasaki ZX-6R"   
## [19] "Honda CBR900RR" "Suzuki RF900"   
## [21] "Ducati Desmosedici RR" "Aprilia Tuono RSV-R"   
## [23] "Kawasaki GPZ900R" "Kawasaki Ninja ZX-9R"   
## [25] "Kawasaki Ninja ZX-6R" "Honda CBR600F4i"   
## [27] "Triumph Speed Triple" "Honda CBR600"   
## [29] "Honda V65 Sabre" "Kawasaki ZZ-R600"   
## [31] "Triumph Rocket III Roadster" "Zero SR"   
## [33] "Honda CBR600RR" "Yamaha VMAX"   
## [35] "Triumph Street Triple R" "Harley-Davidson VRSCR"   
## [37] "Honda CBR600F" "Kawasaki Ninja ZX-7R"   
## [39] "Yamaha FZ6" "Triumph Daytona 900"   
## [41] "Suzuki GSX600F"

#Further filters the previous list 'Smaller' to only include model names with the numbers 2-9 in them  
Smallest <- Smaller[grepl(".[2-9]",Smaller)]  
Smallest

## [1] "Kawasaki Ninja H2" "Ducati 899 Panigale"   
## [3] "Yamaha MT-09 (FZ-09)" "Ducati 959 Panigale"   
## [5] "Suzuki GSX-R600" "Yamaha YZF-R6"   
## [7] "Ducati Streetfighter 848" "Suzuki GSX-R750"   
## [9] "Yamaha MT-09 (FZ-09)" "Ducati 999"   
## [11] "Ducati 749S" "Kawasaki ZX-6R"   
## [13] "Honda CBR900RR" "Suzuki RF900"   
## [15] "Kawasaki GPZ900R" "Kawasaki Ninja ZX-9R"   
## [17] "Kawasaki Ninja ZX-6R" "Honda CBR600F4i"   
## [19] "Honda CBR600" "Honda V65 Sabre"   
## [21] "Kawasaki ZZ-R600" "Honda CBR600RR"   
## [23] "Honda CBR600F" "Kawasaki Ninja ZX-7R"   
## [25] "Yamaha FZ6" "Triumph Daytona 900"   
## [27] "Suzuki GSX600F"

# Exercise 6

Use the quarter.mile.csv file from Homework 5, read this into a data table. Plot each combination of columns (excluding Make and Model) with MPH as the dependent variable.

## Part a.

Create a list of independent variable names by excluding ‘MPH’ (this will be our independent variable), ‘Make’ and ‘Model’ (there are too few observations for many of these entries) from a vector containing the column names from quarter.mile.csv

## Part b.

Iterate over the independent variable names. For each name, concatenate the name with MPH using the delimiter '~'. This will be the string correspond to the formula notation for a plot.

Plot each combination of column by calling the plot with each formula string. You may need to use as.formula in R. There should be 4 plots.

## Part c.

Concatenate the independent variables names in to single string, delimited by ‘+’, then concatenate this string with ‘MPH’ and ‘~’. Name this string multivariate.model and perform a multivariate AOV by executing the code (change the eval flag). Edit the data expression to match the name of your table.

anova(lm(as.formula(multivariate.model),data=fastest.dat))

If you use SAS, you may need to rename one column after import to remove spaces in the name.

I’ve provided PROC SQL code that will read the column names of your imported table into a list of macro variables. Write a macro to iterate over these macro variables to produce the plots described for the R exercises. As you iterate, skip the excluded columns described above.

Also write a macro to iterate over the column names to create a model statement for PROC GLM: I’ve provided a template and a global macro variable GLMModel. Set this global variable from inside your macro.

The model statement should resolve to something like model MPH = .... followed by your column names, each separated by a space.

Run the PROC GLM statement inside the comments.