Advanced Bioinformatics 2020 assessment

[Candidate ID: 9535]

1st April 2020

## Task 3.1

#### Using the sum() function and : operator, the sum of all integers between 5 and 55 was coded as shown:

# Sum of 5 to 55 sequence:  
sum(5:55)

## [1] 1530

## Task 3.2

#### The following code was used to produce the “sumfun” function, enabling one input parameter to calculate the sum of all integers between 5 and n. The function is used to do the calculation for n = 10, n = 20, and n = 100 with the results presented below:

# Function sumfun created as requested by the following code:  
sumfun <- function(n) {sum(seq(from=5,to=n,by=1))}

* The following code was run for n = 6 as presented:

# Output for sumfun if n=6  
sumfun(6)

## [1] 11

* The following code was run for n = 20 as presented:

# Output for sumfun if n=20  
sumfun(20)

## [1] 200

* The following code was run for n = 100 as presented:

# Output for sumfun if n=100  
sumfun(100)

## [1] 5040

## Task 3.3

#### An R script using a “for loop” statement to calculate the Fibonacci series for the first 12 entries, where the first two steps in the sequence are 1, 1 is shown below:

# Fibonacci is explained as nth= (n-1)th + (n-2)th. Using "x" instead of n (as n used prior) and a "for loop" statement.  
Fibonacci <- function(x) {  
if(x<=1) {  
 return(x)  
 } else {  
 return(Fibonacci(x-1) + Fibonacci(x-2))  
 }}  
# First 12 entries to print starting from 1 to 12:  
for(entries in 1:(12)) {   
 print(Fibonacci(entries))  
 }

## [1] 1  
## [1] 1  
## [1] 2  
## [1] 3  
## [1] 5  
## [1] 8  
## [1] 13  
## [1] 21  
## [1] 34  
## [1] 55  
## [1] 89  
## [1] 144

## Task 3.4

#### With the mtcars dataset bundled with R, ggplot is used to generate a box of miles per gallon (in the variable mpg) as a function of the number of gears (in the variable gear), with the fill aesthetic to colour bars by number of gears as shown:

* The following code was run to install ggplot:

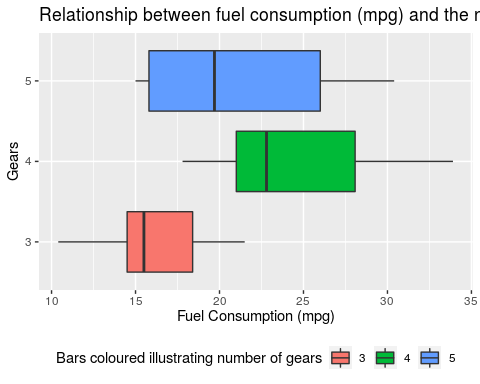
# Install ggplot and load:  
install.packages("ggplot2")  
library(ggplot2)

* The following code was run to load mtcars dataset:

# Load mtcars datset:  
data(mtcars)

* The code that generated the box plot as well as the generated plot is shown below:

# Boxplot with fill aesthetic, added titles and legend placed at the bottom for better viewing:  
  
ggplot(mtcars, aes(mpg, as.factor(gear), fill= as.factor(gear))) + geom\_boxplot() + ggtitle("Relationship between fuel consumption (mpg) and the number of gears") + xlab("Fuel Consumption (mpg)") + ylab("Gears") + labs(fill = "Bars coloured illustrating number of gears") + theme(legend.position="bottom")



## Task 3.5

#### The following code fits a linear relationship between speed and breaking distance in the variable distance and provides details for the fitted slope and intercept of the line, and their standard errors. The units of each variable are also listed and how they were procured.

# Load cars datset  
data(cars)  
  
# Code for variable distance and use of lm  
distance <- lm(dist~speed, data = cars)  
  
# Code to display a set of descriptive statistics (including the fitted slope and intercept of the line, and their standard errors):  
summary(distance)

##   
## Call:  
## lm(formula = dist ~ speed, data = cars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.069 -9.525 -2.272 9.215 43.201   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -17.5791 6.7584 -2.601 0.0123 \*   
## speed 3.9324 0.4155 9.464 1.49e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.38 on 48 degrees of freedom  
## Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438   
## F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

###### The slope is 3.93 with a standard error of 0.42. The intercept is at -17.58 with a standard error of 6.76. *(Values to 2 decimal places)*

# To identify the units for the variables, the entered code provided the information:  
help(cars)  
  
# Measurement units: speed (MPH) and stopping distance (ft.) as listed in the cars documentation.

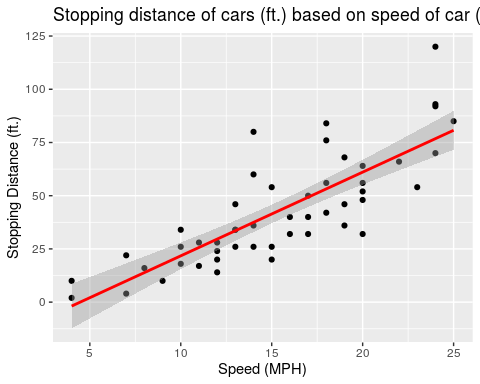
###### The units for the dataset was obtained from the *cars* help document. By using the “help()” function. The speed was in **miles per hour**. The distance in **feet**.

## Task 3.6

#### Used ggplot to plot the data points from Task 6\* and the linear fit as shown:

###### \* *I am assuming this task is referring to the one previous to this (Task 3.5) rather than itself*

# Following on from 3.5 to plot data points and the linear fit using ggplot. I also coloured the line red and placed headings on the axis as well as a title.  
  
ggplot(cars, aes(x = speed, y = dist)) +  
 geom\_point () +  
 stat\_smooth(method ="lm", formula = "y~x", col ="red") +  
 ggtitle("Stopping distance of cars (ft.) based on speed of car (mph)") + xlab("Speed (MPH)") + ylab("Stopping Distance (ft.)")



## Task 3.7

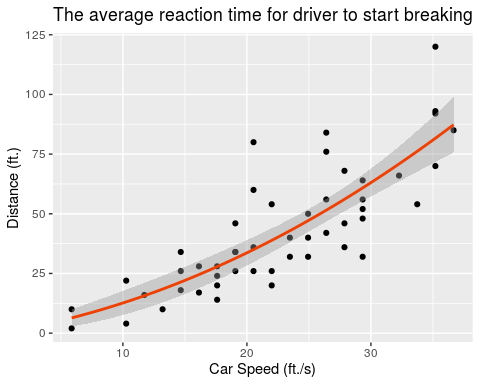
#### The code used to generate the plot (with explanations) to answer the question is as follows with the addition of the generated plot and stating whether reasonable results have been achieved, are as listed:

# The speed list from cars datset in mph converted to ft/s (conversion factor is 5280/3600), in order to estimate reaction time in seconds :  
speed\_ft <- cars$speed \* (5280/3600)  
  
# Equation for line including the assumption OB Distance ∝ speed^2:  
line<-lm(dist ~ 0 + speed\_ft + I(speed\_ft^2),data = cars)  
  
# Call summary information:  
summary(line)

##   
## Call:  
## lm(formula = dist ~ 0 + speed\_ft + I(speed\_ft^2), data = cars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.836 -9.071 -3.152 4.570 44.986   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## speed\_ft 0.84479 0.38180 2.213 0.03171 \*   
## I(speed\_ft^2) 0.04190 0.01366 3.067 0.00355 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.02 on 48 degrees of freedom  
## Multiple R-squared: 0.9133, Adjusted R-squared: 0.9097   
## F-statistic: 252.8 on 2 and 48 DF, p-value: < 2.2e-16

###### Yes, the result is reasonable as 0.84479 seconds is a realistic average reaction time. However, the reaction time may be influenced by additional variables. Considering that we do not know the age of the drivers, reaction time increases as the driver is older. The breaking mechanisms in these 1920s period cars may not be as responsive as they are today. So there may be a delay in the mechanism engaging the brakes. Road condition for each data point may have been slightly different. We are assuming “once breaking commences, braking distance is proportional to the square of the speed”. This is somewhat realistic. Although, yes, the reaction time seems reasonable it is based on multiple assumptions, and may be a lot quicker or slower. With online sources stating a reaction speed between 0.7-1.5 seconds.

# Denote what x and y are:  
y <- cars$dist  
x <- cars$speed\_ft  
  
# Generate Plot (with titles and the line coloured orange-red and placed headings on the axis as well as a title) with the stated formula:  
ggplot(cars, aes(speed\_ft, dist)) + geom\_point() + geom\_smooth(method='lm', formula="y~0+x+I(x^2)", col ="orangered2") + labs(title = "The average reaction time for driver to start breaking", x="Car Speed (ft./s)", y="Distance (ft.)")



# The formula "y~0+x+I(x^2)" is a linear model that fits a quadratic formula with zero offset.