Worksheet 1

Mykola Chudak (8043157)

2025-05-14

Instructions

To complete this worksheet, add code as needed into the R code chunks given below. First, change the author of this file in Line 3 to your name and student number. To properly see the questions, knit this .rmd file to .pdf and view the output. In order to knit this .rmd file, you will need to save the "equations" image from the Lab 2 folder on UM Learn to your computer, in the same folder as this .rmd file.

To enter the R-based questions, add code as needed into the R code chunks given below the question. Be sure when adding in text responses to never copy and paste symbols from outside of the document. Only use the symbols on your keyboard. **Do not** delete the question text, or modify any other part of the code except for the "author" in Line 3.

After you knit your assignment to PDF, check your code chunks. If your code at any point runs off the page, find the nearest comma, click to the right of it, and press Enter (or Return if you are using a Mac). This will force a break in the code so that it goes onto the next line. All of your code must be readable in the final submission.

It is good practice to knit the document after each new line of code. This will enable you to easily locate any errors reported by R.

You will have a link in your email that takes you to the Crowdmark submission page. Once you have completed the worksheet, knit it to .pdf and upload your output to Crowdmark. (If you do not submit a knit .pdf file, you will be given a grade of zero. You should not submit the .Rmd file.)

This worksheet will be due at 11:59 p.m. on May 14.

Note that Question 1 will be done as a demonstration by your TA in your tutorial. You should follow along and type the correct code in order to receive marks for this question.

The last part of this document will introduce you to LaTeX formatting, which allows you nicely type mathematical equations and symbols. This will not be covered in your tutorial. Rather, a short video will be posted on your tutorial UM Learn page going through the LaTeX portion of this worksheet. You should watch this video before answering Question 3.

Question 1 [1 mark]

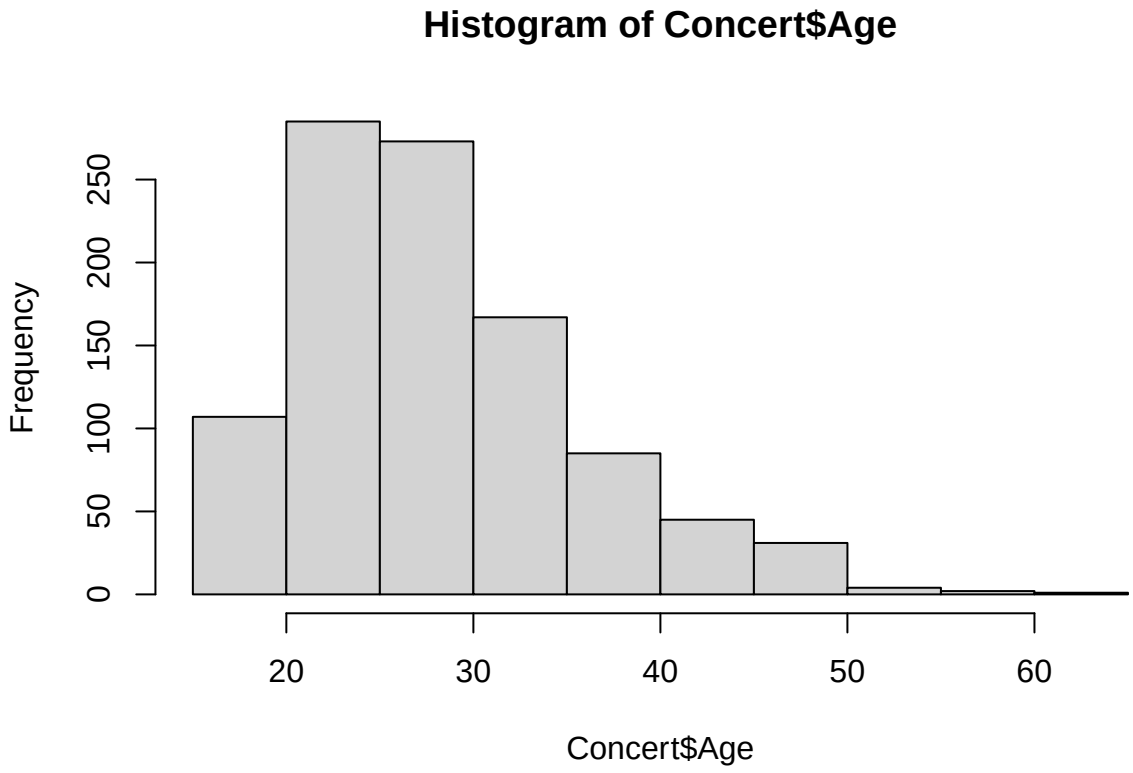
Part (a) The `Concert` data set contains the seating sections, ticket price, age, and favourite music genre for a sample of 1000 people attending a concert. Import it below. (If your code runs off the page for this part, it's ok.)

```
Concert <- read.csv("Concert.csv")
```

Part (b) As we saw in our first tutorial, you can make a simple histogram in `R` by typing `hist(x)`, where `x` is the name your data set.

Below, make a basic histogram of the `Age` variable .

```
hist(Concert$Age)
```



Part (c) We can add more details to this graph by using extra *arguments*.

The additional arguments of a function must stay within the brackets of the function, must be separated by commas, and must be referenced by name. For the `hist` function, the most common arguments are `breaks`, `main`, `xlab`, `ylab`, and `col`. To supply these arguments to `hist`, we would use the syntax `hist(x, breaks = ..., main = ..., xlab = ..., ylab = ..., and col = ...)`. In this function:

- The vector `x` is the data set we provide.

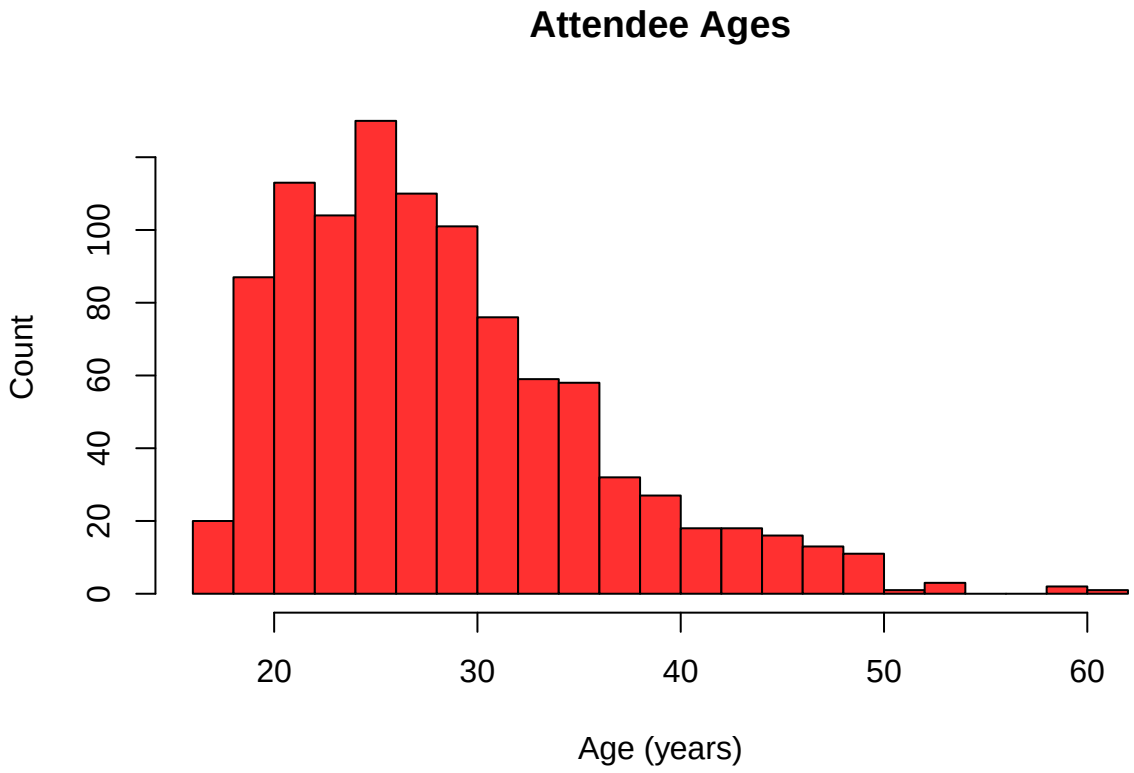
- The argument `breaks` allows us to suggest a number of bars for the histogram. Note that R will often not give you the exact number of bars that you ask for. Instead, it will give you the nearest number of bars that it determines to be "nice". We will not go into how this calculation is made.
- The argument `main` allows us to give a name to the histogram. Note that, since you will be supplying a set of characters as your argument, this will need to be wrapped in quotation marks. That is, to set a title of **My Histogram Title**, you would type `main = "My Histogram Title"`.
- The argument `xlab` allows us to give a label to the x-axis of the histogram.
- The argument `ylab` allows us to give a label to the y-axis of the histogram.
- The argument `col` allows us to set the colours of the bars of the histogram. For a list of all available colours in `R`, follow this link here. Just like for `main`, `xlab`, and `ylab`, you will have to surround the colour name in quotation marks.

If you do not specify any give argument, it will be left at its default value.

Below, create another histogram of the `Age` variable. Set this histogram to have 30 breaks, a main title of "Attendee Ages", an x-axis label of "Age (years)", a y-axis label of "Count", and a colour of `firebrick1`.

If your code runs off the page, make sure you break the line by pressing Enter/Return after a comma.

```
hist(Concert$Age, breaks = 30,
     main = "Attendee Ages", xlab = "Age (years)", ylab = "Count", col = "firebrick1")
```



Part (d) Calculate the mean and median age for this data set.

```r
mean(Concert$Age)
```

```
## [1] 28.767
```

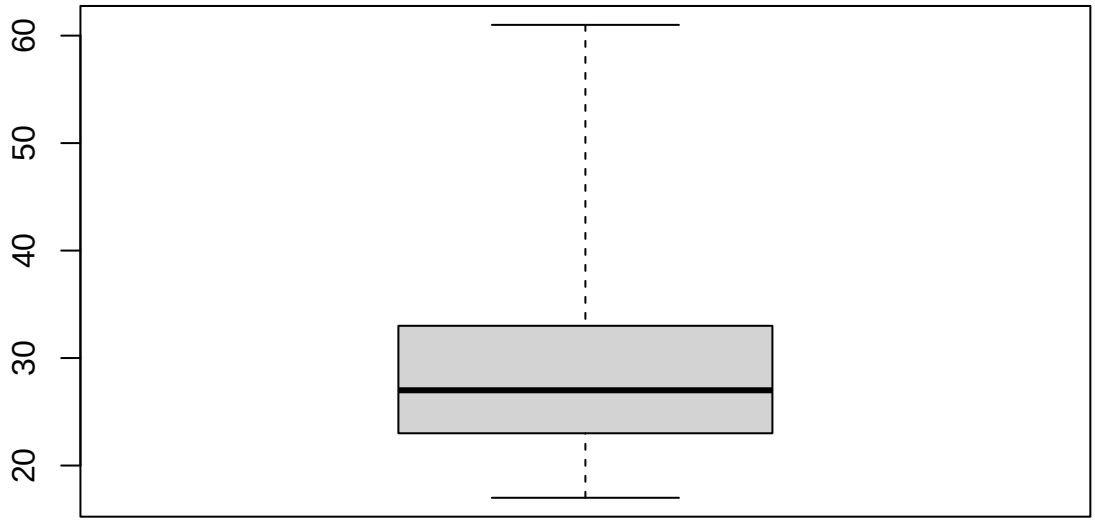
```r
median(Concert$Age)
```

```
## [1] 27
```

Part (e) `R` may also be used to create boxplots, using the function `boxplot`. To make a simple quantile boxplot in `R`, we type `boxplot(x, range = 0)`, where `x` is the name of our data set.

Below, make a basic quantile boxplot of the Ages in this data set.

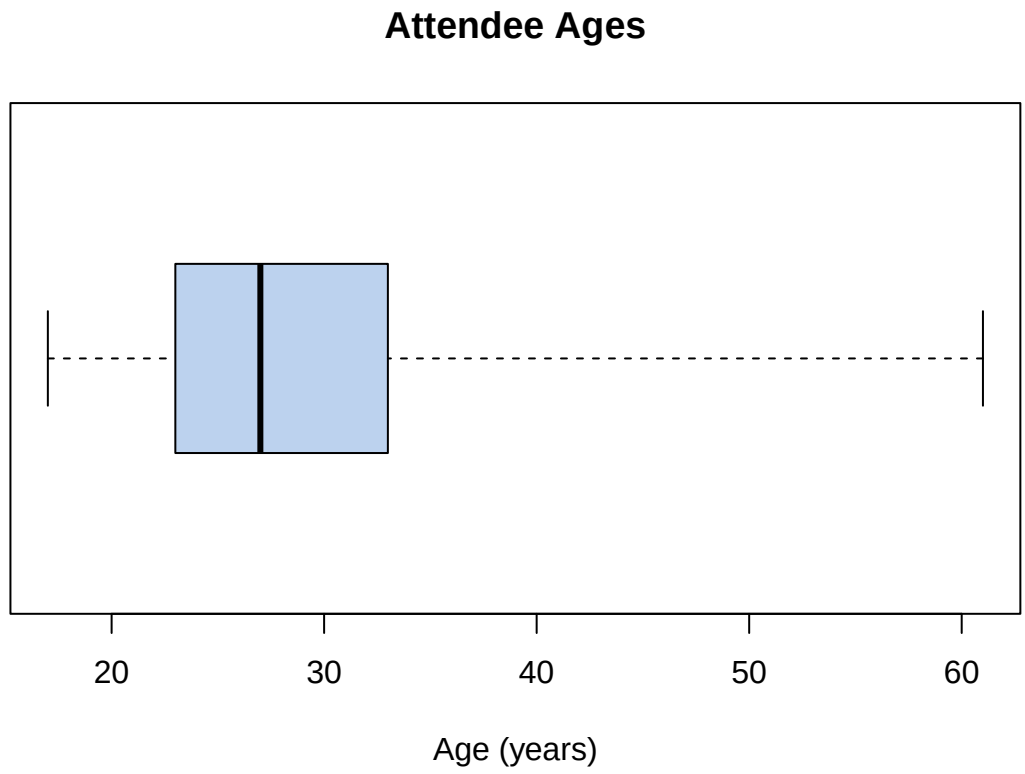
```r
boxplot(Concert$Age, range = 0)
```



Just like for the `hist` function, we can use the `main`, `xlab`, `ylab`, and `col` arguments. We can also add the argument `horizontal = TRUE` to display a horizontal boxplot rather than a vertical one. If we leave this argument out, it will stay as a vertical boxplot.

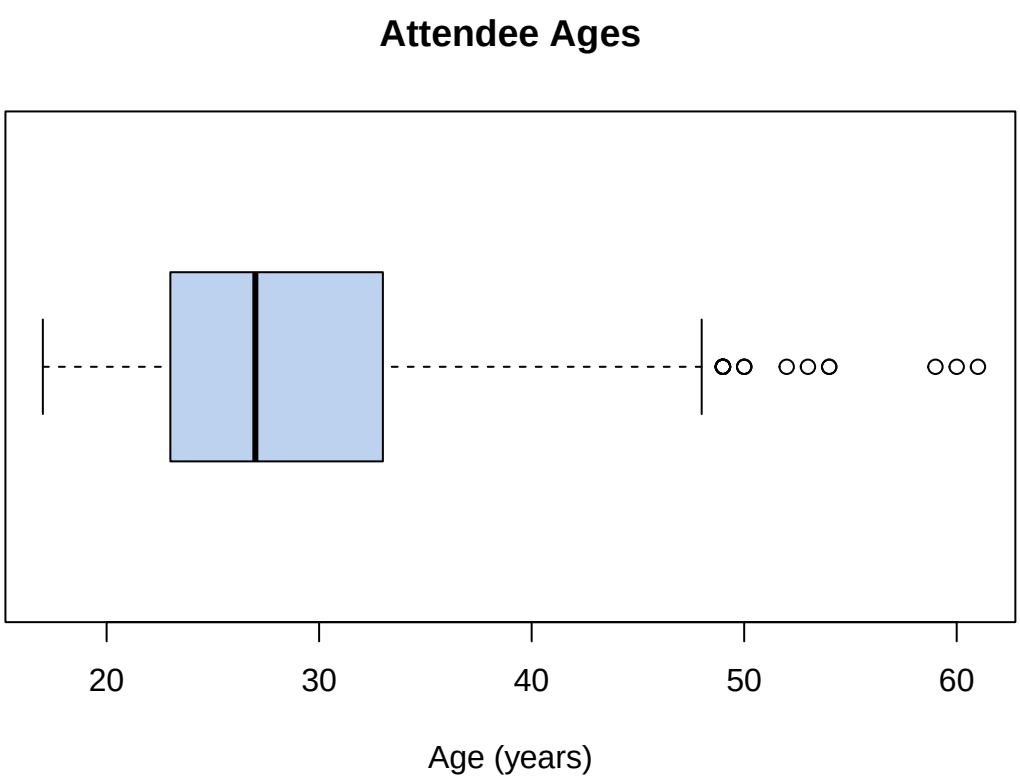
Part (f) Below, create a horizontal quantile boxplot of the concert goers' ages. Set this boxplot to have a title of "Attendee Ages", an x-axis label of "Age (years)", and a colour of `lightsteelblue2`.

```
boxplot(Concert$Age, range = 0, main = "Attendee Ages", xlab = "Age (years)",
        col = "lightsteelblue2", horizontal = TRUE)
```



Part (g) If we instead want an outlier boxplot, we can remove the argument `range = 0`. Recreate the above graph, but as an outlier boxplot instead of a quantile boxplot.

```
boxplot(Concert$Age, main = "Attendee Ages", xlab = "Age (years)",
        col = "lightsteelblue2", horizontal = TRUE)
```

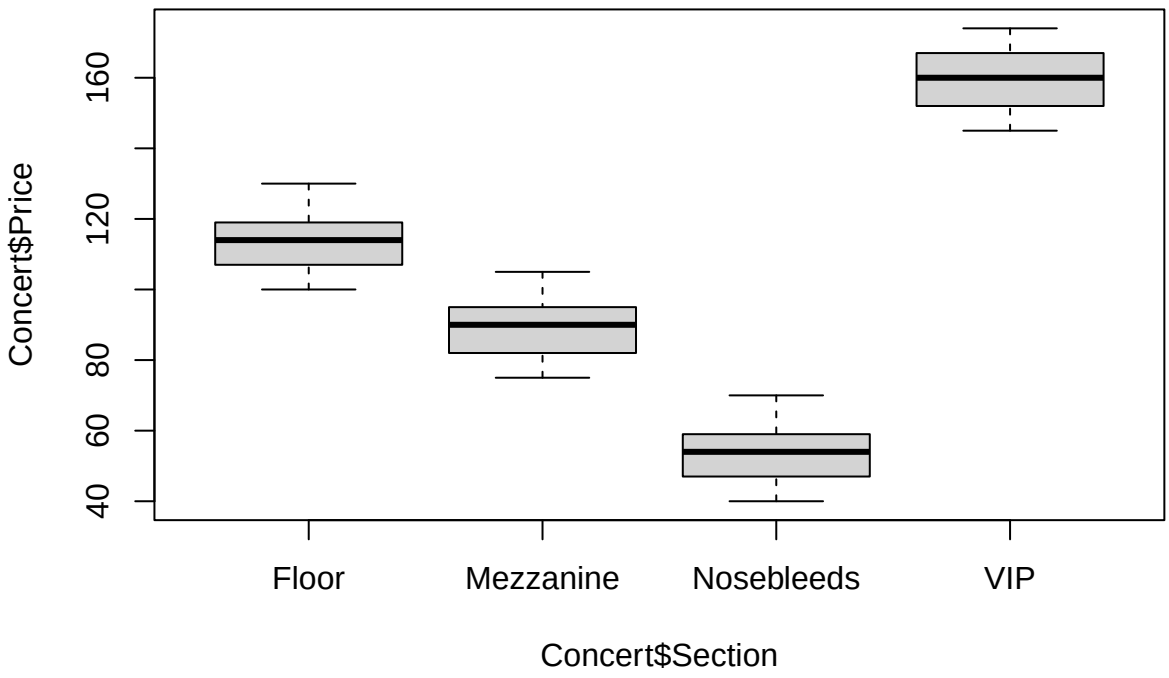


Part (h) We can make side-by-side boxplots of the subgroups in this data set by using R's tilde notation.

To do this, your first argument will be of the form `Y ~ X`, where `X` is the variable that defines the groups, and `Y` is the variable that contains your data measurements. For example, suppose you wanted to compare GPAs of U of M and U of W students. If you had a data set named `Data` with a variable called `University` and a variable called `GPA`, you could make side-by-side boxplots comparing GPAs for the two universities by typing `boxplot(Data$GPA ~ Data$University)`.

Below, create side-by-side boxplots comparing the ticket prices for people sitting in the various sections. (Note that in this case, there are no outliers in ticket prices for any of the four sections, so you will get the same graph, regardless of whether you make quantile boxplots or outlier boxplots.)

```
boxplot(Concert$Price ~ Concert$Section)
```



Part (i) We can find the five-number summary for a variable using the function `fivenum`. Find the five-number summary of ticket prices.

```
fivenum(Concert$Price)
```

```
## [1]  40  56  77 100 174
```

Part (j) Now we can calculate the interquartile range of ticket prices, either by using `R` as a calculator with the quartiles from the five-number summary above, or with the function `IQR`. Calculate the interquartile range of ticket prices using each of these methods.*

```
IQR <- 100 - 56
IQR
```

```
## [1] 44
```

```
IQR(Concert$Price)
```

```
## [1] 44
```

*Note that in this case, `R` gives the same value for the interquartile range using both methods. However, `R` uses a slightly different algorithm to calculate the quartiles using the `fivenum` function than it does using the `IQR` function, so don't be surprised if you get slightly different values for some data sets using the two different methods.

Part (k) Calculate the sample variance and standard deviation of ticket prices.

```
var(Concert$Price)
```

```
## [1] 900.8127
```

```
sd(Concert$Price)
```

```
## [1] 30.01354
```

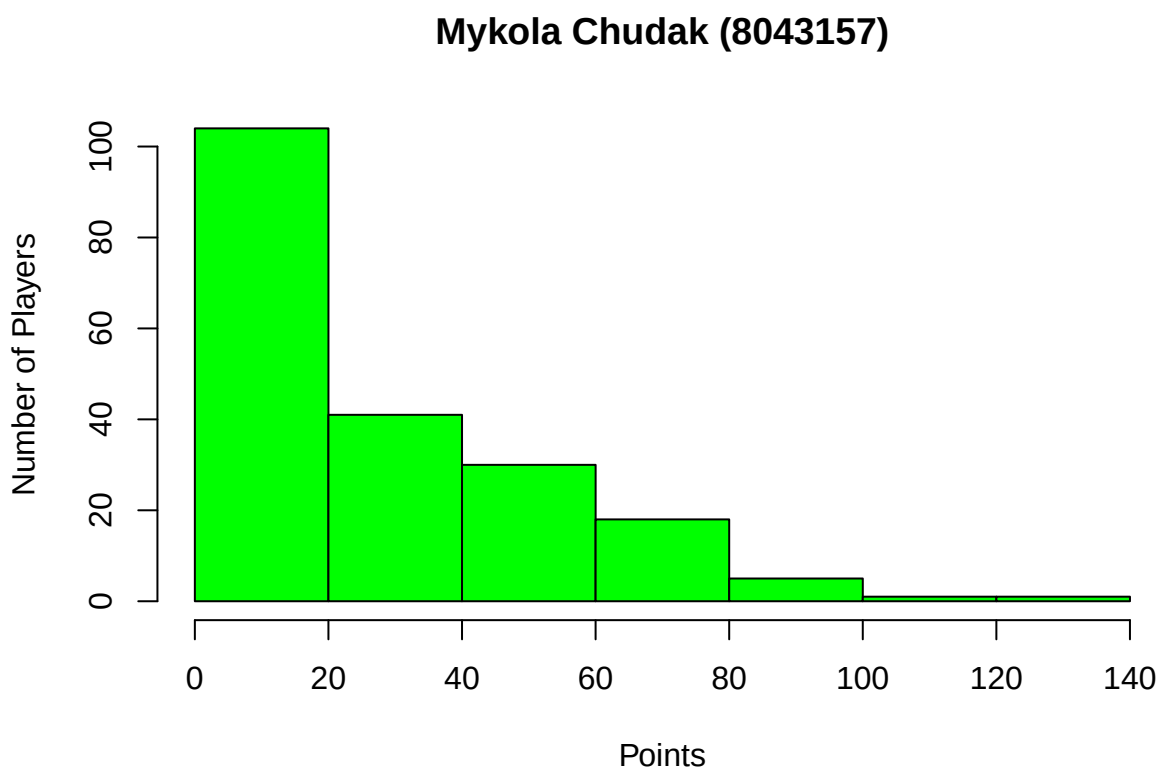
Question 2 [6 marks]

Part (a) The `NHL` data set contains the values of 18 variables for a sample of 200 National Hockey League (NHL) players last year. Import it below.

```
NHL <- read.csv("NHL.csv")
```

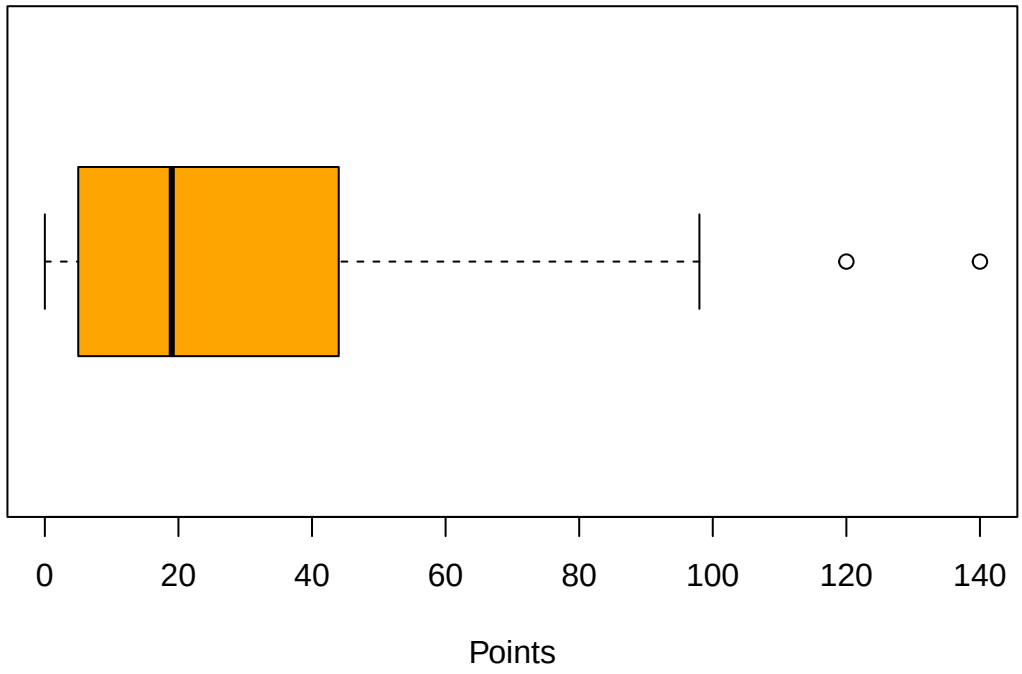
Part (b) Create a histogram of the `Points` variable. Set this histogram to have 8 breaks, an x-axis label of "Points", a y-axis label of "Number of Players", and a colour of `green`. Set the title of the graph to your name and student number.

```
hist(NHL$Points, breaks = 8, xlab = "Points", ylab = "Number of Players", col = "green",
     main = "Mykola Chudak (8043157)")
```

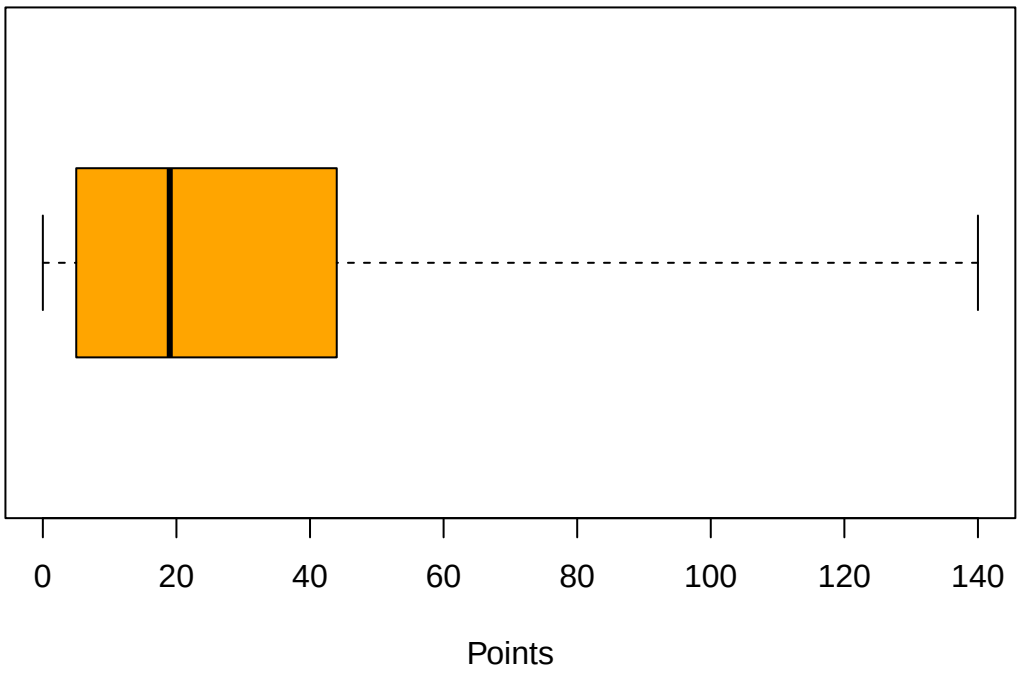


Part (c) Create a horizontal outlier boxplot and a horizontal quantile boxplot (in that order) of the `Points` variable. Label the x-axis as "Points" and set the color to `orange`.

```
boxplot(NHL$Points, horizontal = TRUE, xlab = "Points", col = "orange")
```




```
boxplot(NHL$Points, range = 0, horizontal = TRUE, xlab = "Points", col = "orange")
```



Part (d) Calculate the average and standard deviation of the `Assists` variable for this data set.

```
mean(NHL$Assists)
```

```
## [1] 17.22
```

```
sd(NHL$Assists)
```

```
## [1] 16.87559
```

Part (e)

Calculate the five-number summary of the `Goals` variable.

```
fivenum(NHL$Goals)
```

```
## [1]  0.0  1.0  5.5 16.0 51.0
```

Part (f)

Above what value would a player's goals be considered an outlier? (i.e., Calculate the upper fence.) Use R as a calculator for this part.

```r
Q3 <- fivenum(NHL$Goals)[4]
Q1 <- fivenum(NHL$Goals)[1]
Q3 + (1.5 * (Q3 - Q1))
```

```
## [1] 40
```

TeX Introduction

In the worksheets and assignments for this class, you will at times be asked to show your work. This can be done in RMarkdown, with the typsetting system known as TeX.

TeX (pronounced "teck") is a typsesetting system designed by computer scientist and mathematician Donald Knuth, first used in 1978. It is widely used throughout academia (through the software system LaTeX, pronounced "lay-teck" or "lah-teck") to typeset mathematical expressions.

When you installed TinyTeX, what you were installing was a package (known as a LaTeX distribution) that allows for the production of files using the TeX system.

In RMarkdown, all statements in the TeX system must be delimited with either a single set or a double set of dollar signs. Note that TeX output will **only** work in an RMarkdown (.Rmd) file. It will not display in the console, or an R Script.

When you use a double set of dollar signs, this is known as display math mode. All text that goes inside the double set of dollar signs will be read as TeX code. It will go onto a new line and it will be centered.

Below, we will try our first line of TeX code to write "x + y = z" in display mode:

$$x + y = z$$

We can see the typeface used for the symbols `x`, `y`, and `+` differ from the font used outside of math mode (known as paragraph mode). When going into display mode in RMarkdown, you should move this to a new line of code.

If we want the equation to show in the same line as our text, we can use a single set of dollar signs, which is called in-line math mode.

If we want to write "x + y = z" in in-line math mode, we type the equation between dollar single signs as follows: $x + y = z$.

We will now see how to create more complex mathematical expressions. **Remember that all TeX formatting must go inside either a set of single or double dollar signs**.

Superscripts & Subscripts

TeX can easily produce superscripts and subscripts.

Superscript text is text that appears in a smaller form, raised above the standard text. We can superscript a symbol by using the caret symbol (^) found on your keyboard. For example, to write the Pythagorean theorem in display mode, we type `x^2 + y^2 = z^2` as follows:

$$x^2 + y^2 = z^2$$

Note that if you want to put multiple symbols into superscript, you must surround them with a set of curly brackets. For example, to type the product rule for exponents, we type `x^{a+b} = x^a x^b` as follows:

$$x^{a+b} = x^a x^b$$

Subscript text is text that appears in a smaller form, lowered below the standard text. We can subscript a symbol by using the underscore (the _ symbol) found on your keyboard. For example, you may remember the equation of a line being written in the form "y = mx + b" from math class. In Statistics, we instead use the form y = b0 + b1x (where the 0 and 1 are written as subscripts). We write the equation of this line in TeX as $y = b_0 + b_1 x$.

Greek Letters

In Mathematics and Statistics, we make extensive use of Greek symbols. To type these symbols in TeX, we start with a backslash, and then provide the name of the Greek letter. For example, the Greek letters alpha, beta and delta are written as `\alpha`, `\beta` and `\delta` as follows: α, β, δ. The standard symbol for the population mean is written as `\mu` (μ), and the standard symbol for the population standard deviation is written as `\sigma` (σ). The population variance is therefore σ^2.

Square Roots

To write a square root in TeX, we type `\sqrt{}`, where the argument is contained between the curly brackets.

For example, to write the square root of 15 in TeX, we would type `\sqrt{15}` as follows: $\sqrt{15}$.

Bars and Hats

To put a bar over a symbol in TeX, we type `\bar{}`, where the argument goes between the curly brackets. For example, to write the sample mean of x (i.e., x-bar) in TeX, we would type `\bar{x}` as follows: $\bar{x}$.

To put a hat over a symbol in TeX (which we will need later in the course), we type `\hat{}`, where the argument goes between the curly brackets. For example, to write a y with a hat over top of it in TeX, we would type `\hat{y}` as follows: $\hat{y}$.

Fractions

To type a fraction in TeX, we type `\frac{}{}`, where the numerator goes in the first set of curly brackets, and the denominator goes in the second. For example, to create the fraction x/y, we type `\frac{x}{y}` as follows: $\frac{x}{y}$.

If we want the fraction to appear larger so both the numerator and denominator are the same size as the rest of the text, we can type `\dfrac{}{}`. For example, we can type the fraction 3/8 as follows:

$$\dfrac{3}{8}$$

Parentheses

To create round brackets in TeX, we can simply type the standard brackets as (). However, these brackets will not scale with your output. For example, if you tried to put these brackets around a fraction, it would display as the output below:

$$(\frac{x}{y})$$

You can see that this does not look very nice. The best solution to this problem is to use brackets that automatically scale to the size of what's between them. To do so, you can instead use `\left(` and `\right)`. Note that if you do not use both of these together, TeX will return an error.

$$\left(\frac{x}{y}\right)$$

We can see this looks much better.

Sums

To display a sum, we can use the `\sum` symbol. For example, to display the sum of a data set, we would type `\sum x_i` as follows: $\sum x_i$. (You can add the indices i=1 to n, but we won't worry about that code; in this course, any summation will be understood to be over all data values.)

Question 3 [3 marks]

In this question, you will be asked to type three simple equations using TeX formatting. Knit this document now to see what the equations should look like.

The volume of a sphere is calculated as:

$$\frac{4}{3}\pi r^3$$

The quadratic equation is:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The sample correlation r between two variables X and Y is:

$$r = \frac{1}{(n-1)s_x s_y}\sum(x_i - \bar{x})(y_i - \bar{y})$$

Part (a) [1 mark] Type the equation for the volume of a sphere below using TeX formatting.

$$\frac{4}{3}\pi r^3$$

Part (b) [1 mark] Type the quadratic formula below using TeX formatting. (To produce the plus/minus symbol, type `\pm`.)

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Part (c) [1 mark] Type the correlation formula below using TeX formatting.

$$r = \frac{1}{(n-1)s_x s_y} \sum (x_i - \bar{x})(y_i - \bar{y})$$