MATH255 - Autumn 2023 Computer Lab - Week 8

Note. Question 1(a) is your Lab Preparation exercise for this week. It must be completed and handed in on Moodle as a pdf before the start of your lab. If you can't solve it using R yet, just download the file directly and calculate by hand.

Rather than just looking at data in raw format, it is useful to produce numerical summaries and graphical displays. R is much more convenient for this purpose rather than tedious calculations by hand or calculator.

Useful Formulae

- $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ — sample mean (measure of centre)
- $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$ — sample variance (measure of spread)
- $s = \sqrt{s^2}$ — sample standard deviation (same units of measurement as data)
- $\overline{y} = a + b\overline{x}$, $s_y = |b|s_x$ — mean, standard deviation for linear transformation $y_i = a + bx_i$
- $Q_2 =$ median — middle sorted data value (n odd), average of two middle values (n even)
- $Q_1 =$ lower quartile ($Q_3 =$ upper quartile) — exceeds 25% (75%) of sorted data
- $IQR = Q_3 - Q_1$ — interquartile range (alternative measure of spread)
- (minimum, Q_1, Q_2, Q_3, maximum) — 5 number summary, displayed in boxplot

R Implementation

- Depending on personal preference, you can either use the R package directly, or via the `RStudio` interface. This software is available in computer labs 17.108 and 15.210, and is also available to download and install on your own computer from `https://www.r-project.org`.
- Commands may either be entered directly into the R Console window (cut-and-paste works fine), or run from a script window.
- A small dataset can be entered directly as a vector.

```
x <- c(3,1,4,1,5)            # <- denotes assignment operator
x                            # display value of x
(x <- c(3,1,4,1,5))          # save and display!
```

- Summary statistics can be found one at a time, or you can do several at once. If output is needed for subsequent calculations, rather than just displayed on the screen, then save as a named variable and remember that names are case-sensitive.

```
(Mx <- mean(x))              # save and displaysample mean (average), name Mx
c(var(x),sqrt(var(x)),sd(x)) # standard deviation is square root of variance
```

- For a linear transformation of the form $y_i = a + bx_i$, the new mean can be found by plugging the old mean into the formula. The new standard deviation can be found by multiplying the old standard deviation by $|b|$. This doesn't work for non-linear transformations!

```
c(mean(20-3*x), 20-3*mean(x))      # equivalent calculations
c(sd(20-3*x), abs(-3)*sd(x))        # equivalent calculations
c(mean(log(x)), log(mean(x)))       # not equal; log is non-linear
```

- To compute quartiles in R, use the `quantile` function (not `quartile`). The idea of a p-quantile (or equivalently a $100p$-percentile) is that a proportion p of data values lies below the quantile and a proportion $1-p$ lies above. In particular, the values $p = 0.25$, $p = 0.5$, $p = 0.75$ correspond to the lower quartile, median and upper quartile respectively.

```
quantile(x)        # min, lower quartile, median, upper quartile, max
IQR(x)             # interquartile range, measure of spread
```

- Unfortunately the world cannot agree about the exact definition of sample quartiles. R's `quantile` function can provide 9 different types!. *(You don't need to use alternative methods for assignments or exams, but be aware that computed quartiles may vary according to the method.)*

- For assignments/exams in this subject, use the simple **repeated median** method. Divide the **sorted** data set into two halves, excluding the middle value for odd n. The quartiles Q_1, Q_3 are calculated as medians of the lower and upper half data sets respectively.
For the sorted dataset $\{1, 1, 3, 4, 5\}$, the median (middle value) is 3. As $n = 5$ is odd, exclude the middle value; $Q_1 = 1$ (median of $\{1, 1\}$) is 1 and $Q_3 = (4+5)/2 = 4.5$ (median of $\{4, 5\}$.
The repeated median method sometimes gives different results to `quantile` and/or `fivenum`.

```
fivenum(x)                    # quartiles by repeated median method
xsort <- sort(x)              # sort data in x
median(xsort[1:2])            # lower quartile agrees with fivenum(x)
median(xsort[4:5])            # upper quartile agrees with fivenum(x)
```

- The median and interquartile range (IQR) are alternative measures of centre and spread, much less sensitive to *outliers* (unusual data values) than mean and standard deviation.

```
y <- c(x, 92)                                # add outlier to original sample
c(mean(x),mean(y),median(x),median(y))       # mean changes more than median
c(sd(x),sd(y),IQR(x),IQR(y))                 # std dev changes much more than IQR
y[y<=10]                                     # how to omit data values above 10
```

- For larger datasets, it is more convenient to import a data file rather than entering data directly into R. Download the `system.csv` file from Moodle, and save it where you can find it again! This file contains times in μs (microseconds) between requests for a particular computer system process service.

- If necessary, change the Working Directory to the folder containing `system.csv`. In RStudio go to the `Tools` menu, in Windows R go to the `File` menu, or in Mac R go to the `Session` or `Misc` menu. Select `Change Working Directory`, and select the appropriate folder/directory.

- Read in the data.

```
System <- read.csv("system.csv")       # reads comma separated value file
Time <- System[,1]                     # extracts column 1
```

- In order to generate a boxplot of the system time data, enter

  ```
  boxplot(Time)
  ```

  The box extends between the lower and upper quartiles, with a cross-bar at the median. Some data values have been identified as *outliers*, lying more than 1.5 interquartile ranges beyond the quartiles and displayed as individual points on the graph. "Whiskers" extend from the box to the smallest and largest non-outliers.

- In order to generate a histogram of the system time data, enter

  ```
  hist(Time)      # histogram
  ```

  The vertical axis shows the frequency, *i.e.* the number of observations falling within the corresponding "bin" (interval) on the horizontal axis. Note the extreme skewness to the **right** (*i.e.* concentration of small data values with a long tail extending towards the right).

- The appearance of the histogram can be modified by specifying the number of bins (*e.g.* `hist(Time,16)`), or by specifying bin breakpoints, e.g.

  ```
  hist(Time,seq(0,80000,2500))  # bins from 0 to 2500, 2500 to 5000, ...
  ```

Exercises

1. The following questions are based on the `system` time dataset downloaded from Moodle.
   (a) Verify that the mean is larger than the median. This is expected as the data are strongly skewed to the **right**.
   (b) Use R to compute the upper bound `ub` = (upper quartile) $+1.5*$ IQR. Find any outliers in the dataset which exceed `ub`. (*Hint:* Use syntax like `Time[Time>ub]` .)
   (c) Compute the new mean, median, standard deviation and IQR after omitting the outliers in (d) from the dataset. (*Hint:* Use syntax like `mean(Time[Time<=ub])` .)
   Verify that the median and IQR are less sensitive to outliers than the mean and standard deviation.