Week 03 R Workshop

Elmer V Villanueva

September 16, 2019

SET YOUR WORKING DIRECTORY!

```
setwd("D:/Dropbox/00 - Working Folder/Teaching/DPH101/2019-2020/Week 03 Summarising Data/R03 R Workshop
```

Load the GLOW500 data.

```
GLOW500_WORK <- read.csv("GLOW500.csv")
```

Make sure that you develop the habit of checking that the file was loaded correctly.

```
str(GLOW500_WORK)
```

```
## 'data.frame':    500 obs. of  15 variables:
##  $ SUB_ID   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SITE_ID  : int  1 4 6 6 1 5 5 1 1 4 ...
##  $ PHY_ID   : int  14 284 305 309 37 299 302 36 8 282 ...
##  $ PRIORFRAC: int  0 0 1 0 0 1 0 1 1 0 ...
##  $ AGE      : int  62 65 88 82 61 67 84 82 86 58 ...
##  $ WEIGHT   : num  70.3 87.1 50.8 62.1 68 68 50.8 40.8 62.6 63.5 ...
##  $ HEIGHT   : int  158 160 157 160 152 161 150 153 156 166 ...
##  $ BMI      : num  28.2 34 20.6 24.3 29.4 ...
##  $ PREMENO  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOMFRAC  : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ ARMASSIST: int  0 0 1 0 0 0 0 0 0 0 ...
##  $ SMOKE    : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ RATERISK : int  2 2 1 1 2 2 1 2 2 1 ...
##  $ FRACSCORE: int  1 2 11 5 1 4 6 7 7 0 ...
##  $ FRACTURE : int  0 0 0 0 0 0 0 0 0 0 ...
```

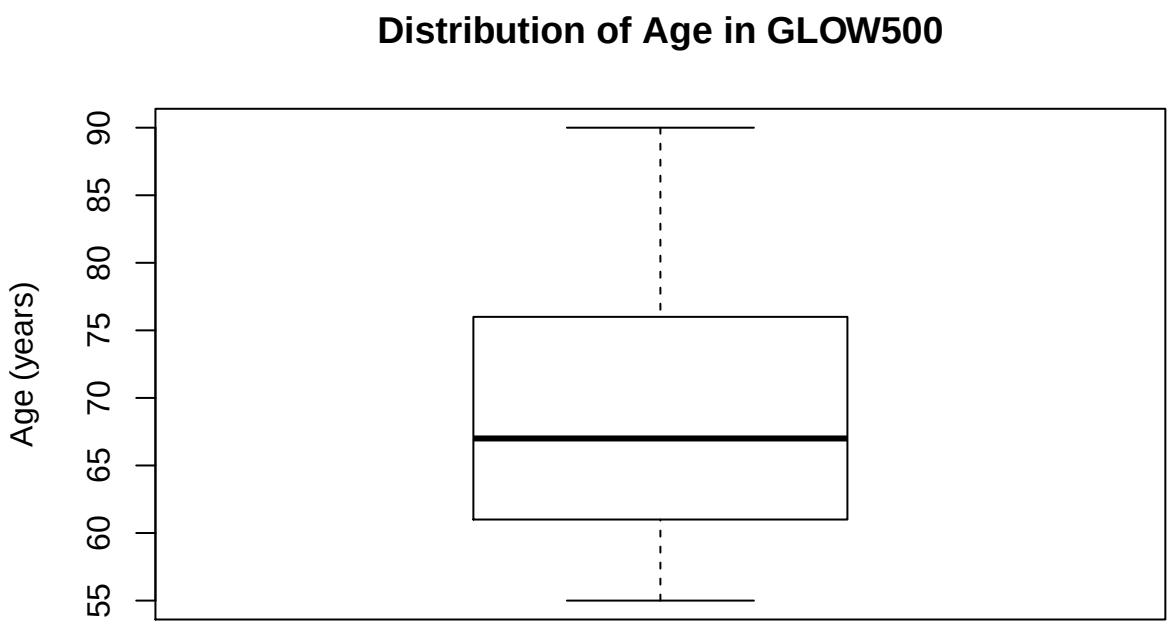
```
head(GLOW500_WORK)
```

```
##   SUB_ID SITE_ID PHY_ID PRIORFRAC AGE WEIGHT HEIGHT      BMI PREMENO
## 1      1       1     14         0  62   70.3    158 28.16055       0
## 2      2       4    284         0  65   87.1    160 34.02344       0
## 3      3       6    305         1  88   50.8    157 20.60936       0
## 4      4       6    309         0  82   62.1    160 24.25781       0
## 5      5       1     37         0  61   68.0    152 29.43213       0
## 6      6       5    299         1  67   68.0    161 26.23356       0
##   MOMFRAC ARMASSIST SMOKE RATERISK FRACSCORE FRACTURE
## 1       0         0     0        2         1        0
## 2       0         0     0        2         2        0
## 3       1         1     0        1        11        0
## 4       0         0     0        1         5        0
## 5       0         0     0        2         1        0
## 6       0         0     1        2         4        0
```

Boxplots

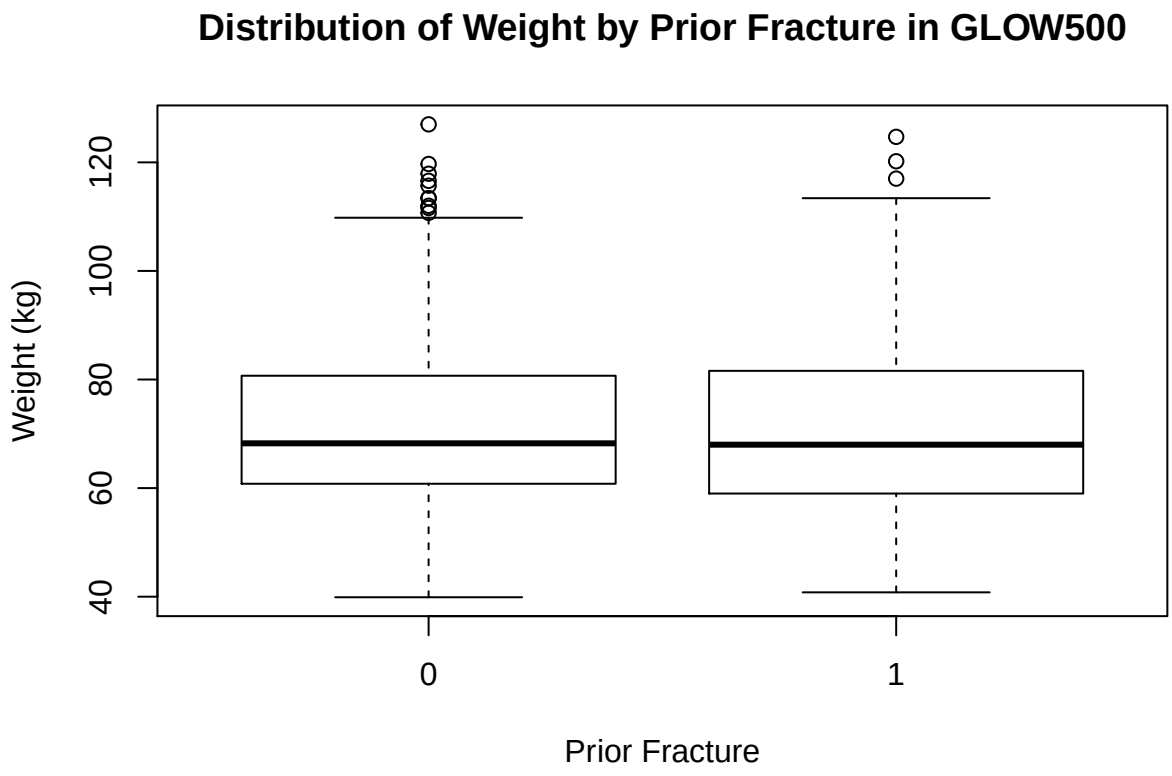
Let's produce a boxplot of `AGE`.

```
boxplot(GLOW500_WORK$AGE,
        ylab = "Age (years)",
        main = "Distribution of Age in GLOW500")
```



Let's produce a boxplot of `WEIGHT` by prior fracture status (`PRIORFRAC`).

```
boxplot(WEIGHT ~ PRIORFRAC, data = GLOW500_WORK,
        ylab = "Weight (kg)",
        xlab = "Prior Fracture",
        main = "Distribution of Weight by Prior Fracture in GLOW500")
```

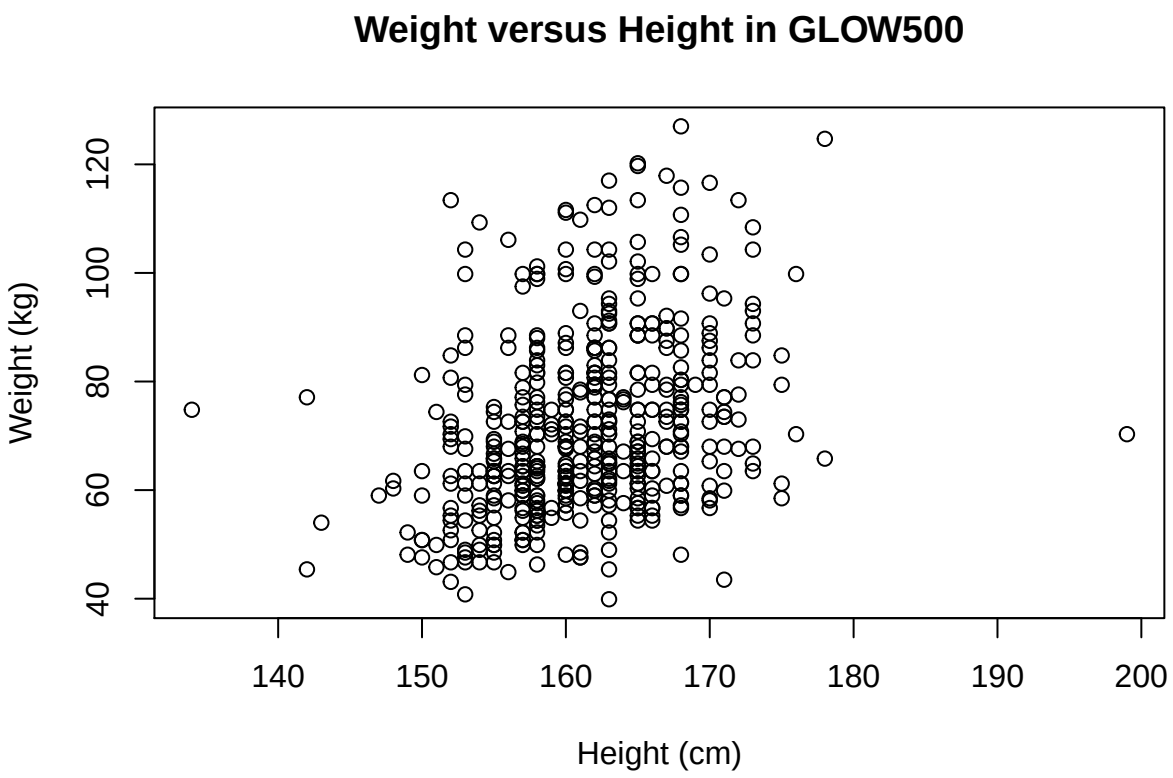


Why is the prior fracture labels appearing as zero and one?

Scatterplot

Let's look at the relationship between `WEIGHT` and `HEIGHT`

```
plot(WEIGHT ~ HEIGHT, data = GLOW500_WORK,
     ylab = "Weight (kg)",
     xlab = "Height (cm)",
     main = "Weight versus Height in GLOW500")
```



Simple numerical summaries

Let's produce simple numerical summaries of `HEIGHT`.

```
summary(GLOW500_WORK$HEIGHT)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   134.0   157.0   161.5   161.4   165.0   199.0
```

Geometric mean

R doesn't have a built-in function for the geometric mean. However, there is a function in the `EnvStats` package that we can use.

```
if (!require("EnvStats")) install.packages("EnvStats", repos = "https://mirrors.tuna.tsinghua.edu.cn/CR
```

```
## Loading required package: EnvStats
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```
library("EnvStats")
```

Let's calculate the geometric mean of `WEIGHT`.

```
geoMean(GLOW500_WORK$WEIGHT)
```

```
## [1] 70.06875
```

There is an easier way, however, that we can do this without installing and loading a package.

```
exp(mean(log(GLOW500_WORK$WEIGHT)))
```

```
## [1] 70.06875
```

Harmonic mean

R doesn't have a built-in function to calculate the harmonic mean, but there is a function in the `lmomco` package that we can use.

```
if (!require("lmomco")) install.packages("lmomco", repos = "https://mirrors.tuna.tsinghua.edu.cn/CRAN/")
```

```
## Loading required package: lmomco
```

```
library("lmomco")
```

Let's calculate the harmonic mean of `HEIGHT`.

```
harmonic.mean(GLOW500_WORK$HEIGHT)
```

```
## $harmean
## [1] 161.115
##
## $correction
## [1] 1
##
## $source
## [1] "harmonic.mean"
```

As before, there is an easier way to calcualte this without needing to load a package.

```
1/mean(1/GLOW500_WORK$HEIGHT)
```

```
## [1] 161.115
```

Quantiles

Let's calcualte the 3rd, 45th, 59th and 93rd percentile of `AGE`.

```
quantile(GLOW500_WORK$AGE,
         probs = c(0.03, 0.45, 0.59, 0.93))
```

```
##  3% 45% 59% 93%
##  56  66  70  83
```

Let's calculate the interquartile range of `AGE`.

```
quantile(GLOW500_WORK$AGE,
         probs = c(0.25, 0.75))
```

```
## 25% 75%
##  61  76
```

Standard deviation

Calculate the standard deviation of `HEIGHT`.

```
sd(GLOW500_WORK$HEIGHT)
```

```
## [1] 6.355493
```

Variance

Calculate the variance of `WEIGHT`.

```
var(GLOW500_WORK$WEIGHT)
```

```
## [1] 270.1418
```

Coefficient of variation

Calculate the coefficeint of variation of `HEIGHT`.

```
sd(GLOW500_WORK$HEIGHT)/mean(GLOW500_WORK$HEIGHT)
```

```
## [1] 0.03938606
```

Manipulating data

Let's create a small data frame.

```
ID <- c(1, 2, 3, 4, 5)
PETALS <- c(30, 35, 26, 23, 41)
COLOR <- c("Red", "White", "White", "Red", "Red")
ROSE <- data.frame(ID, PETALS, COLOR)
str(ROSE)
```

```
## 'data.frame':    5 obs. of  3 variables:
##  $ ID    : num  1 2 3 4 5
##  $ PETALS: num  30 35 26 23 41
##  $ COLOR : Factor w/ 2 levels "Red","White": 1 2 2 1 1
```

```
head(ROSE)
```

```
##   ID PETALS COLOR
## 1  1     30   Red
## 2  2     35 White
## 3  3     26 White
## 4  4     23   Red
## 5  5     41   Red
```

Let's arrange the data in ascending order of the number of petals.

```
ROSE[order(ROSE$PETALS),]
```

```
##   ID PETALS COLOR
## 4  4     23   Red
## 3  3     26 White
## 1  1     30   Red
## 2  2     35 White
## 5  5     41   Red
```

Let's produce a dataset containing only data from the red roses.

```r
ROSE.RED <- subset(ROSE, COLOR=="Red")
head(ROSE.RED)
```

```
##   ID PETALS COLOR
## 1  1     30   Red
## 4  4     23   Red
## 5  5     41   Red
```

THE END