

# MAS115 R programming, Homework Solutions 2

*Prof. P. G. Blackwell*

2017-18

## Q1 Toothgrowth in guinea pigs

- a) Use `?ToothGrowth` or `help(ToothGrowth)`. No written answer is needed.
- b) By using the `str()` command we can see that the dose variable is stored as a numeric:

```
str(ToothGrowth) # to look at the structure of the object
```

```
'data.frame': 60 obs. of 3 variables:
 $ len : num  4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
 $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 ...
 $ dose: num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

- c) To create a table showing the number of guinea pigs that took each supplement, we need to use the `table()` command, mentioned in §1.2.1 of the Lab 2 handout. This counts the number of observations taking each level of the factor passed to it.

```
table(ToothGrowth$supp)
```

```
OJ VC
30 30
```

- d) We can create boxplots using the `boxplot()` command, specifying in the formula `len ~ supp` that we want to plot `len` split by the factor `supp` in the dataset `ToothGrowth`. This is done with the following command; I have added some useful axis labels and a title to my plot. The output is shown in Figure 1.

```
boxplot(len ~ supp, data = ToothGrowth,
xlab = "Supplement", ylab = "Tooth Length",
main = "Plot of tooth length according to supplement" )
```

- e) To find the mean tooth length of those guinea pigs which had their supplement delivered from Orange Juice we first need to filter out the relevant individuals and then find the mean of their tooth length. There are several different ways this can be achieved. One way, written in stages, is as follows.

```
OJcases <- ToothGrowth$supp == "OJ"
OJlengths <- ToothGrowth[OJcases,"len"]
mean(OJlengths)
```

```
[1] 20.66333
```

You should look at the values of the intermediate variables (which I have called `OJcases` and `OJlengths`) and make sure you can understand what is happening at each step. This can be done in a single step:

```
mean(ToothGrowth[ToothGrowth$supp == "OJ", "len"])
```

```
[1] 20.66333
```

- f) A similar filtering can be done to find the maximum tooth length of those individuals that had a dose of 2mg. Alternatively, in either case we could refer to the column of the data frame by number instead of name:

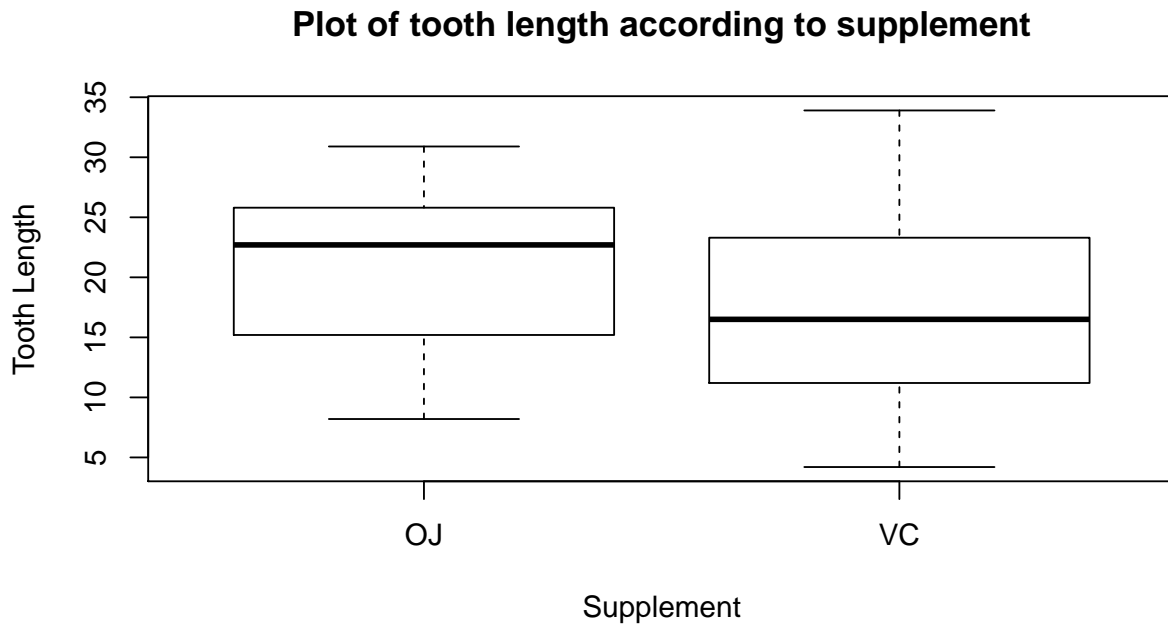


Figure 1: Plot of the length of guinea pig teeth dependent upon whether a Vitamin C supplement was delivered using Orange Juice (OJ) or ascorbic acid (VC).

```
max(ToothGrowth[ToothGrowth$dose == 2,1])
```

```
[1] 33.9
```

## Q2 Modelling lightbulb failures

- a) From the help file (`?rexp`) we see that we need to specify the rate parameter when creating realisations of an exponential random variable. To get 100 random times with  $E[X] = 50$  we need the following.

```
FailTime <- rexp(100, rate = 0.02) # Mean is 1/rate = 50
head(FailTime) # Show the first few values, e.g. for homework submission!
```

```
[1] 43.705936 6.240565 131.978675 11.461634 62.276370 31.930034
```

- b) To split this up into factor groups we use the `cut()` command specifying how we would like the groups split up.

```
FailGroup <- cut(FailTime, c(0, 10, 30, 70, Inf))
```

Note the use of `Inf` to specify the group  $(70, \infty)$ . To create a table counting the number in each group we can type

```
table(FailGroup)
```

```
FailGroup
(0,10] (10,30] (30,70] (70,Inf]
     20      34      28      18
```

*Note:* You will get different numbers in each group than my solutions since it is creating random failure times.