
Question 02

Please watch videos1 and 2 in week 11 lecture assignment. You can download the code which used for S&P from files tab.
Please do the following with your assigned stock.
- Download the data.
- Calculate log returns.
- Calculate volatility measure.
- Calculate volatility over entire length of series for various three different decay factors.
- Plot the results, overlaying the volatility curves on the data, just as was done in the S&P example.

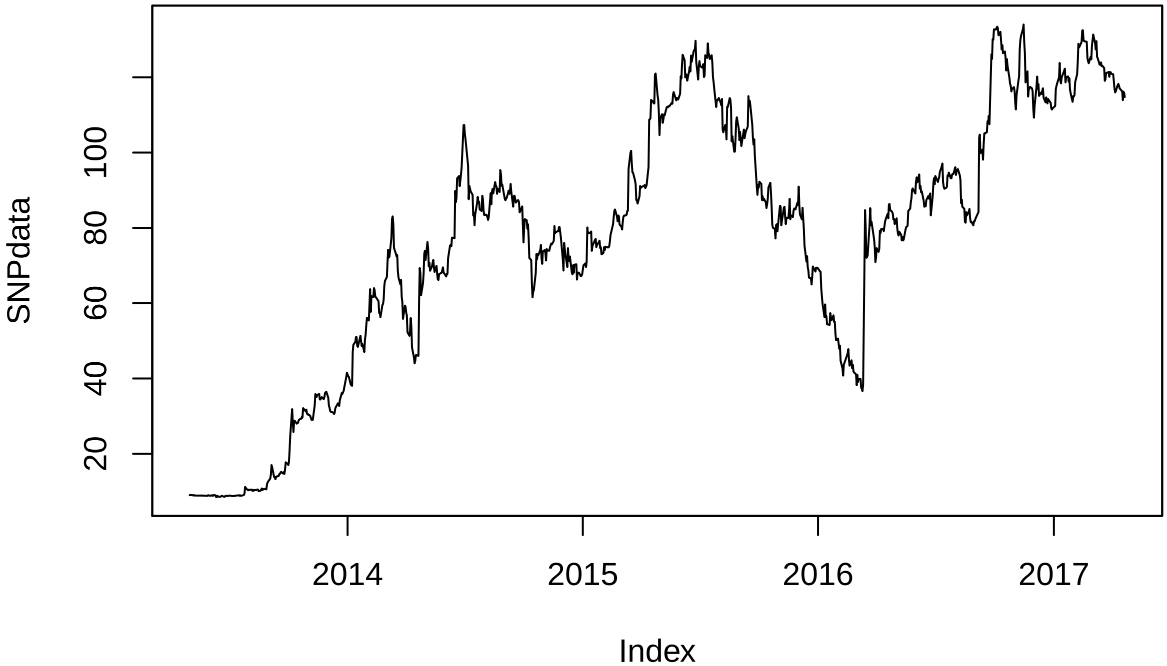
Group	Stock
Patrick & Sunna	GWPH

```
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

library(tseries)

# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ..     volatility function
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

getVol <- function(d, logrets)
{
    var = 0
    lam = 0
    varlist <- c()

    for (r in logrets)
    {
        lam = lam*(1 - 1/d) + 1
        var = (1 - 1/lam)*var + (1/lam)*r^2
        varlist <- c(varlist, var)
    }
    sqrt(varlist)
}


# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    SNPdata <- get.hist.quote('gwph', quote = "Close")
```

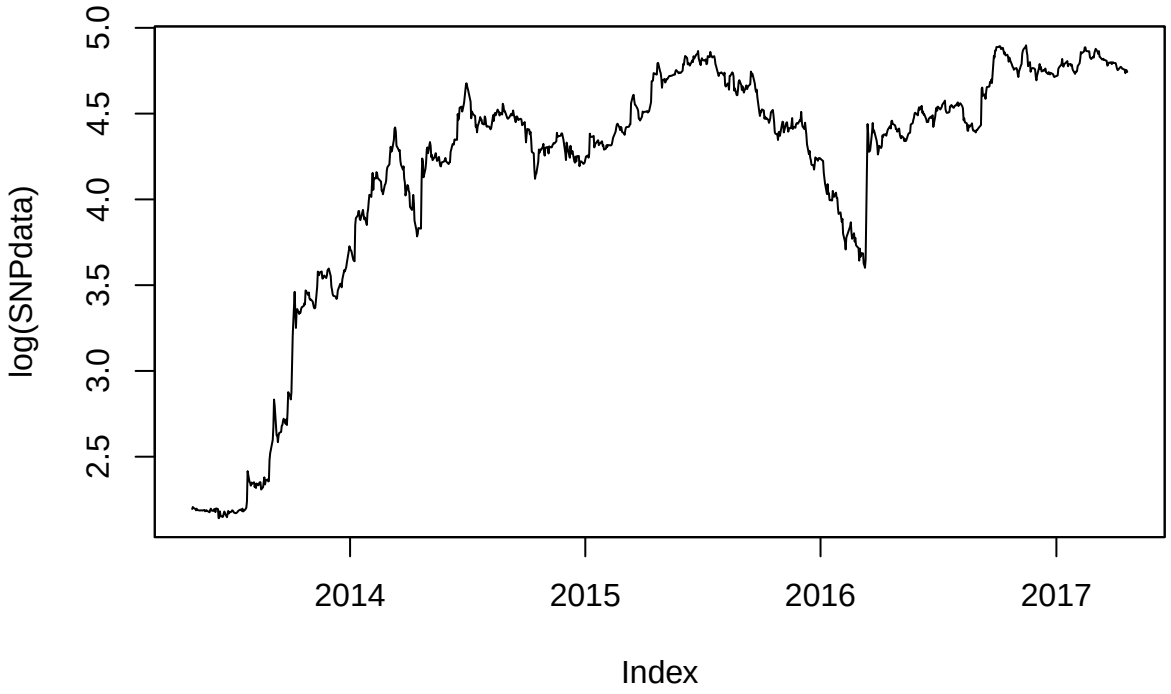
```
## time series starts 2013-05-01
## time series ends   2017-04-21
```

```
    SNPret <- log(lag(SNPdata)) - log(SNPdata)
    SNPvol <- sd(SNPret) * sqrt(250) * 100
```

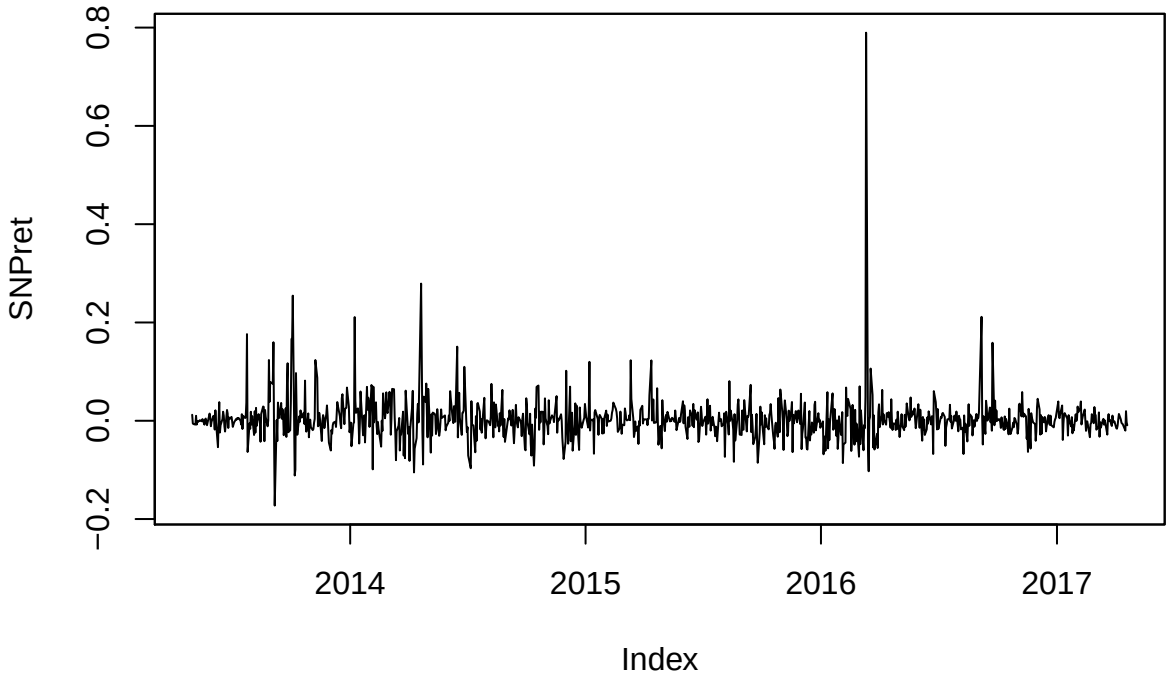
```r
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   recreate figure 6.12 in the text on page 155
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    volest  <- getVol(10, SNPret)
    volest2 <- getVol(30, SNPret)
    volest3 <- getVol(100, SNPret)

    plot(SNPdata)
```



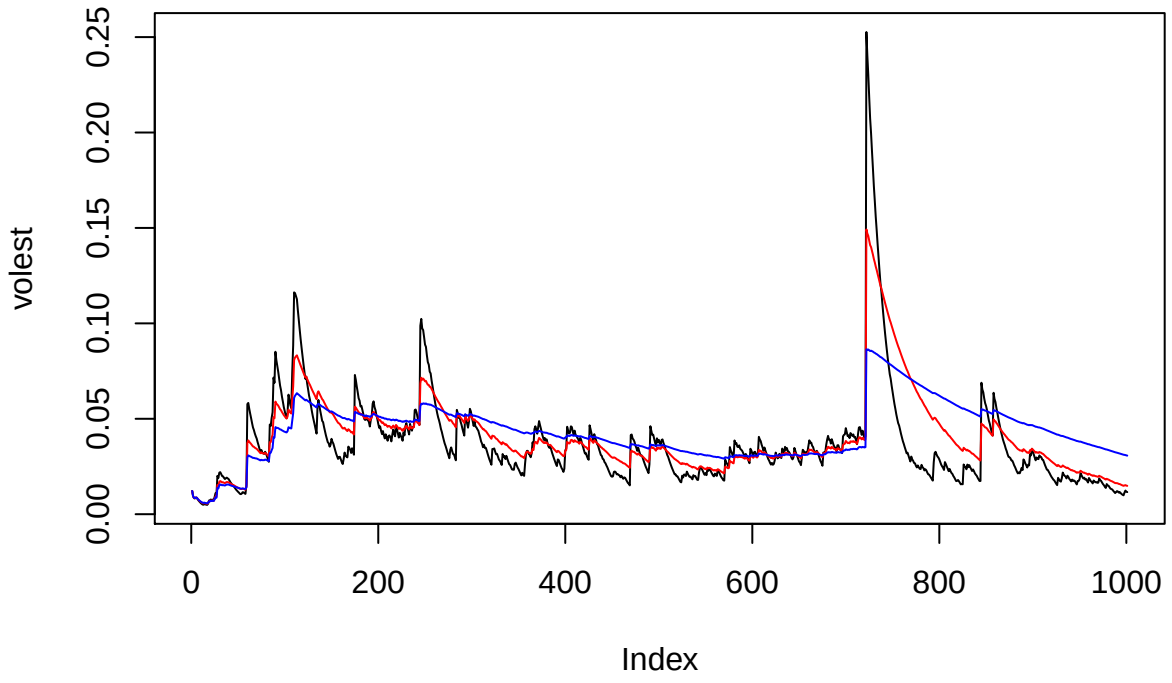

```r
    plot(log(SNPdata))
```




```
plot(SNPret)
```




```
plot(volest, type = "l")
lines(volest2, type = "l", col = "red")
lines(volest3, type = "l", col = "blue")
```



Question 03

The built-in data set called Orange in R is about the growth of orange trees. The Orange data frame has 3 columns of records of the growth of orange trees.
Variable description
* Tree : an ordered factor indicating the tree on which the measurement is made. The ordering is according to increasing maximum diameter.
* age : a numeric vector giving the age of the tree (days since 1968/12/31)
* circumference : a numeric vector of trunk circumferences (mm). This is probably "circumference at breast height", a standard measurement in forestry.

Submit your final R code and necessary plots for each part.

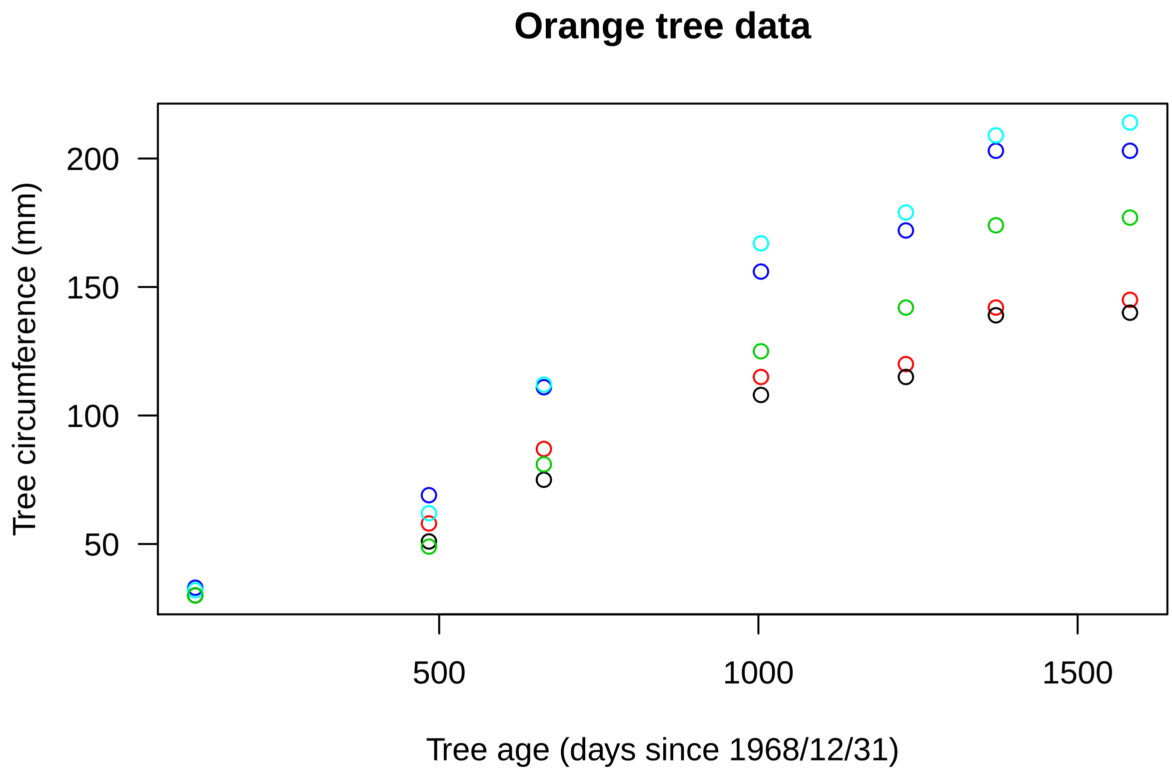
```r
data (Orange)

# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   a) Calculate the mean and the median of the trunk circumferences
# ...       for different size of the trees. (Tree)
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

col_drops <- c("age", "tree")
df_orange_tmp <- Orange[ , !(names(Orange) %in% col_drops)]
aggregate(df_orange_tmp[, 2], list(df_orange_tmp$Tree), mean)
```

```
##   Group.1         x
## 1       3  94.00000
## 2       1  99.57143
## 3       5 111.14286
## 4       2 135.28571
## 5       4 139.28571
```

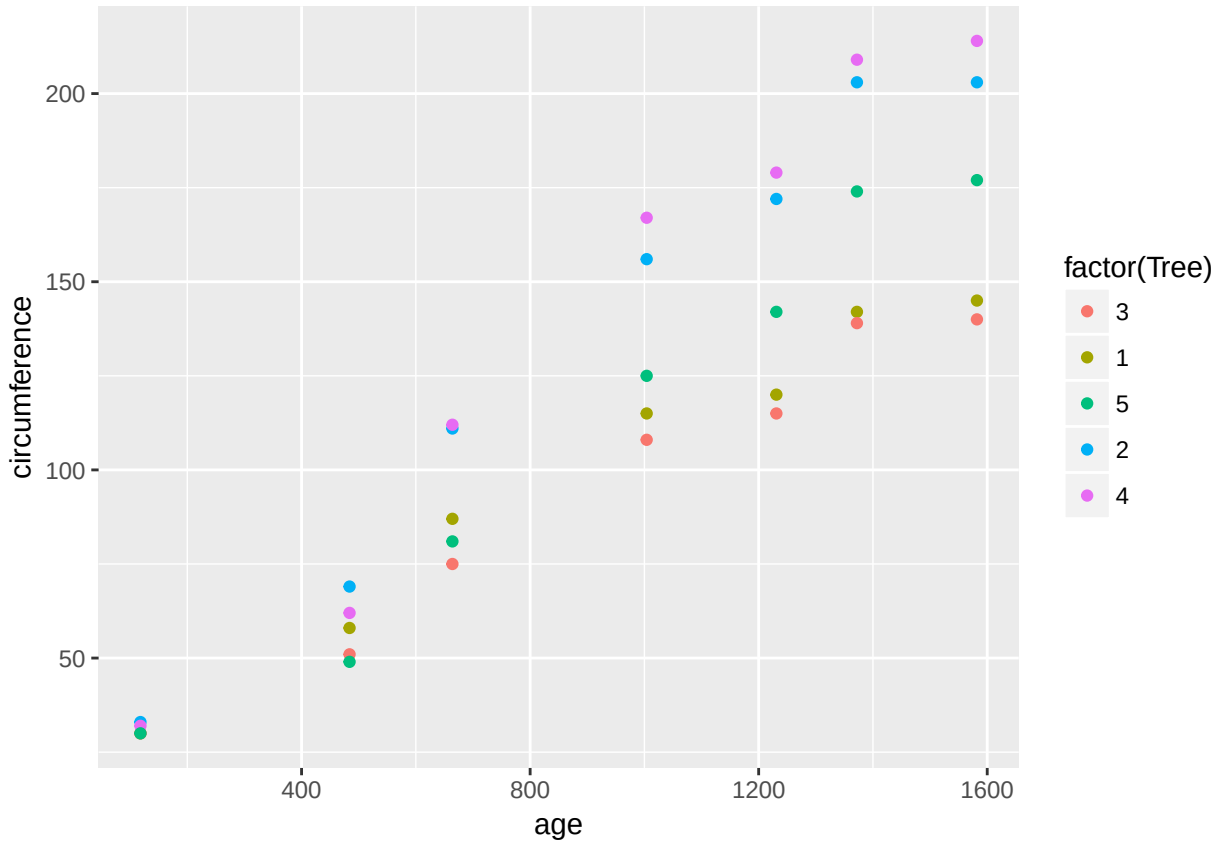
```r
aggregate(df_orange_tmp[, 2], list(df_orange_tmp$Tree), median)
```

```
##   Group.1   x
## 1       3 108
## 2       1 115
## 3       5 125
## 4       2 156
## 5       4 167
```

```r
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...b) Make a scatter plot of the trunk circumferences against the age of the tree.
# ...   Use different plotting symbols for different size of trees.
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    plot(circumference ~ age, data = Orange,
        xlab = "Tree age (days since 1968/12/31)",
        ylab = "Tree circumference (mm)", las = 1,
        main = "Orange tree data",
        col = Tree)
```

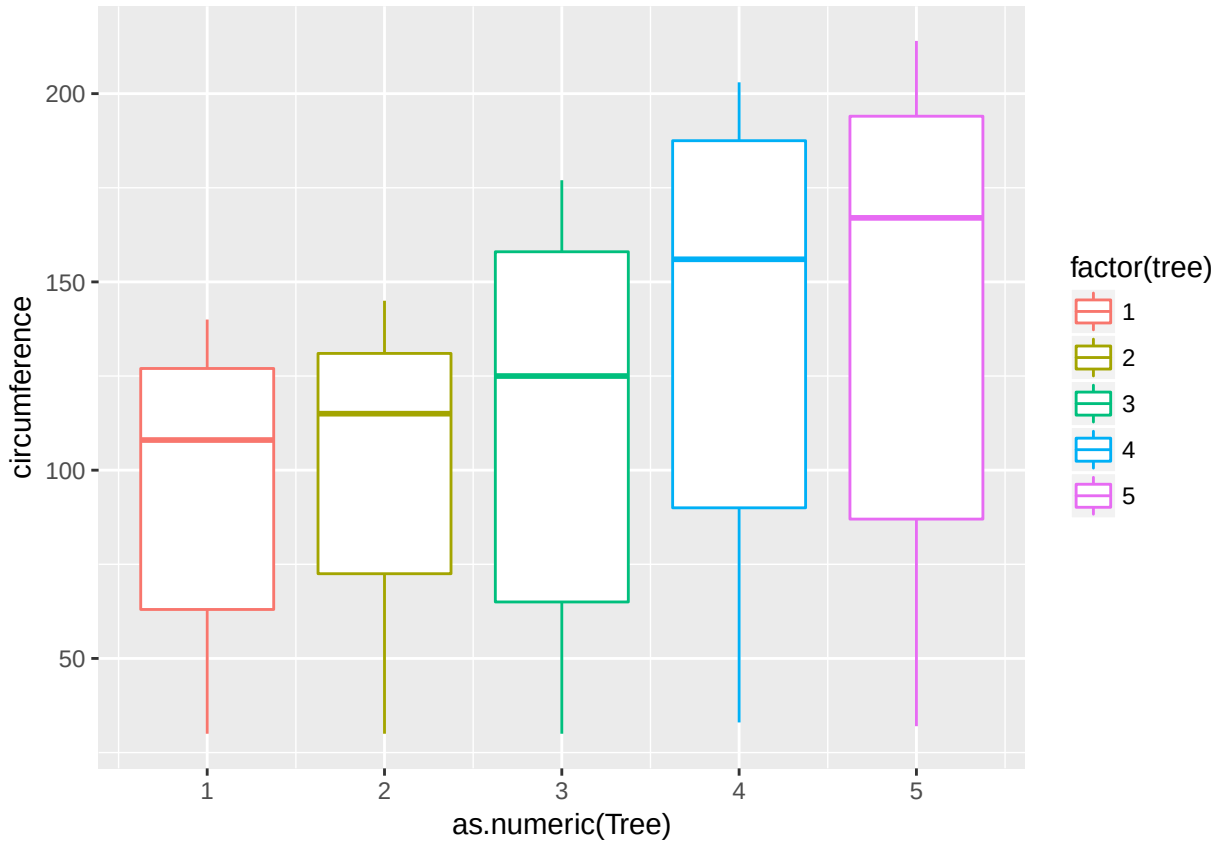



```r
    p <- ggplot(Orange, aes(age, circumference))
    p + geom_point(aes(color = factor(Tree)))
```




```
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ...   c) Display the trunk circumferences on a comparative boxplot against tree.
# ...   Be sure you order the boxplots in the increasing order of maximum diameter.
# ...   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

```
    Orange$tree <- as.numeric(Orange$Tree)

    p <- ggplot(Orange, aes(x = as.numeric(Tree), y = circumference))
    p + geom_boxplot(aes(color = factor(tree)))
```


200
150
100
50
circumference
1
2
3
4
5
as.numeric(Tree)
factor(tree)
1
2
3
4
5