Exercise 1.3

Indexing by logical statements

useful commands: `==`, `!=`, `>`, `<`, `&`, `|`, `sum()`, `which()`, `table()`, `!`

* 1. Create your own logical vector with three TRUEs and three FALSEs

```
a = c(TRUE, TRUE, FALSE, FALSE, TRUE, FALSE)
a ## let's print to screen and make sure it is stored in this variable
```

```
## [1]  TRUE  TRUE FALSE FALSE  TRUE FALSE
```

* 2. Produce a vector of the index number of the true's

```
which(a)  ## which gives you the index of TRUE values automatically
```

```
## [1] 1 2 5
```

```
which(a == TRUE)   ## but sometimes it's reassuring to state exactly what you're doing
```

```
## [1] 1 2 5
```

3. Produce a second vector which indexes the numbers of the falses

```
which(!a)
```

```
## [1] 3 4 6
```

```
which(a == FALSE)
```

```
## [1] 3 4 6
```

Go back to the iris dataset

```
data(iris)  ## this reloads the data set in case you've closed R since using iris
```

4. How many irises have sepals less than 5.5 cm long?

```
sum(iris[,'Sepal.Length']<5.5)  ## remember TRUE's are 1 and FALSE's are 0
```

```
## [1] 52
```

```
length(which(iris[,'Sepal.Length']<5.5))  ## here, which() will only return the index of TRUE values, s
```

```
## [1] 52
```

* 5. Which iris individual has the largest petal length? What is the width of it's petal?

```
max(iris[,'Petal.Length'])  ## this gives us the length of the longest petal
```

```
## [1] 6.9
```

```
which(iris[,'Petal.Length'] == max(iris[,'Petal.Length']))  ## this gives us the index of the individual
```

```
## [1] 119
```

```
iris[,'Petal.Width'][which(iris[,'Petal.Length'] == max(iris[,'Petal.Length']))] ## now we're subsettin
```

```
## [1] 2.3
```

```
## another way to do this would be to use the index of the individual with the longest petal to pick ro
iris[which(iris[,'Petal.Length'] == max(iris[,'Petal.Length'])) , 'Petal.Width']
```

```
## [1] 2.3
```

6. How many of the irises are in this dataset belong to the species versicolor?

```
sum(iris[,'Species']=='versicolor')
```

```
## [1] 50
```

```
table(iris[,'Species']) ## this could get us close
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

* 7. How many irises have petals longer than 6cm?

```
sum(iris[,'Petal.Length'] > 6)
```

```
## [1] 9
```

8. Create a vector of species name for each iris with sepals longer than 6cm.

```
iris[,'Species'][iris[,'Sepal.Length']>6]
```

```
##  [1] versicolor versicolor versicolor versicolor versicolor versicolor
##  [7] versicolor versicolor versicolor versicolor versicolor versicolor
## [13] versicolor versicolor versicolor versicolor versicolor versicolor
## [19] versicolor versicolor virginica  virginica  virginica  virginica
## [25] virginica  virginica  virginica  virginica  virginica  virginica
## [31] virginica  virginica  virginica  virginica  virginica  virginica
## [37] virginica  virginica  virginica  virginica  virginica  virginica
## [43] virginica  virginica  virginica  virginica  virginica  virginica
## [49] virginica  virginica  virginica  virginica  virginica  virginica
## [55] virginica  virginica  virginica  virginica  virginica  virginica
## [61] virginica
## Levels: setosa versicolor virginica
```

```
iris[iris[,'Sepal.Length']>6, 'Species'] ## alternatively, we can put the logical vector in the row par
```

```
##  [1] versicolor versicolor versicolor versicolor versicolor versicolor
##  [7] versicolor versicolor versicolor versicolor versicolor versicolor
## [13] versicolor versicolor versicolor versicolor versicolor versicolor
## [19] versicolor versicolor virginica  virginica  virginica  virginica
## [25] virginica  virginica  virginica  virginica  virginica  virginica
## [31] virginica  virginica  virginica  virginica  virginica  virginica
## [37] virginica  virginica  virginica  virginica  virginica  virginica
## [43] virginica  virginica  virginica  virginica  virginica  virginica
## [49] virginica  virginica  virginica  virginica  virginica  virginica
## [55] virginica  virginica  virginica  virginica  virginica  virginica
## [61] virginica
## Levels: setosa versicolor virginica
```

* 9. How many irises have sepals shorter than 5cm, but wider than 3cm?

```
sum( iris[,'Sepal.Length'] < 5 & iris[,'Sepal.Width'] > 3 )
```

```
## [1] 13
```

10. How many irises have petals narrower than 0.2cm or shorter than 1.5cm?

```
sum( iris[,'Petal.Width'] < 0.2 | iris[,'Petal.Length'] < 1.5 )
```

```
## [1] 26
```

11. What is the average width of setosa iris sepals that are longer than 5cm?

```
mean( iris[,'Sepal.Width'][iris[,'Sepal.Length'] > 5]) ## convince yourself the second part is a logica
```

```
## [1] 3.048305
```

```
mean( iris[iris[,'Sepal.Length'] > 5, 'Sepal.Width']) ## again, we can alternatively subset using logic
```

```
## [1] 3.048305
```

12. What is the difference between the longest and shortest petal lengths of the species virginica?

```
max(iris[,'Petal.Length'][iris[,'Species']=='virginica']) - min(iris[,'Petal.Length'][iris[,'Species']==
```

```
## [1] 2.4
```

13. What proportion of flowers in the dataset have petals wider than 1cm?

```
sum(iris[,'Petal.Width'] > 1 ) / nrow(iris) ## here, we're counting up how many are wider than 1 cm, an
```

```
## [1] 0.62
```

14. Create a new column within your dataframe, called sepalCategory, and set all values equal to 'long'

```
iris[,'sepalCategory'] = 'long'  ## this sets ever entry in the column equal to 'long'
```

- Subset short values of this column, and set their values to 'short' (Short sepals are those less than 5.5 cm)

```
iris[,'sepalCategory'][iris[,'Sepal.Length']<5.5] = 'short'  ## this sets only those entries that match
```

- How many plants with short sepals are there? How many long?

```
table(iris[,'sepalCategory'])
```

```
##
##  long short
##    98    52
```