Remember to check that the mosaic package is loaded before starting:

Each of the following statements has a syntax mistake. Write the statements properly and give a sentence saying what was wrong. (Cut and paste the correct statement from R, along with any output that R gives and your sentence saying what was wrong in the original.)

Here's an example:

1 What is wrong with this statement?

```r
> a = fetchData(myfile.csv)
```

ANSWER: It should be one of the following (which are all equivalent):

```r
> a = fetchData("myfile.csv")
> a <- fetchData("myfile.csv")
> a = fetchData('["myfile.csv"]')
> a <- fetchData('["myfile.csv"]')
```

The file name is a character string and therefore should be in quotes. Otherwise it’s treated as an object name, and there is no object called myfile.csv.

Now for the real thing.

2 Say what’s wrong with each of these statements for the purpose given:

a) > # to make x take the value 3 + 5
   > 3 + 5 = x

b) > # to find the square root of 392
   > sqrt[4*98]

c) Why doesn’t this compute the logarithm of 1,234? What does it do?

```r
> log(1,234)
```

[1] 0

d) > # intended to give FALSE TRUE FALSE FALSE
   > sqrt(c(4,16,25,36))==4

e) > # to create a collection of names "apple"
   "berry" "cherry"
   > fruit = c(apple, berry, cherry)

f) > # where x is intended to be assigned the value 20
   > x = 4(3+2)

3 The following statement gives a result that some people are surprised by

```r
> 10e3 == 1000
```

[1] FALSE

Explain why the result was FALSE rather than TRUE.

A 10e3 is 100, not 1000
B 10e3 is 10000, not 1000
C 10e3 is not a valid number
D It should be true; there’s a bug in the software.

4 Here are two ways to write one million:

```r
> 1e6
> 10e5
```

Write 5 more different forms of one-million using scientific notation.

5 The seq() function generates sequences. Use seq() to make the following sequences:

a) the integers from 1 to 10
b) the integers from 5 to 15
c) the integers from 1 to 10, skipping the even ones
d) 10 evenly spaced numbers between 0 and 1