Correlation Coefficient

Stat 131A, Fall 2018, Prof. Sanchez

Learning Objectives

- Using scatter diagrams to visualize association of two variables
- Using R to "manually" compute the correlation coefficient
- Getting to know the function `cor()`
- Understanding how change of scales affect the correlation

Introduction

In the previous script we talked about how to plot scatter diagrams in R using two different approaches: 1) the basic `plot()` function, and 2) the more advanced graphics package `"ggplot2"`. Knowing how to create scatter diagrams will help us introduce the ideas that have to do with the analysis of two quantitative variables.

Describing and summarizing a single (quantitative) variable is usually the firts step of any data analysis. This should allow you to get to know the data by looking at the distributions of the variables, and reducing the numerical information in the data to a set of measures of center and spread.

After performing a univariate analysis, the next step will usually consist of exploring how two variables may be associated, determine the type of association, how strong is the association (if any), and how to summarize such association.

Anscombe Data Set

In this tutorial we are going to use a special data set known as the *Anscombe* data or *Anscombe's Quartet*. This data was created by Francis Anscombe in the early 1970s to illustrate statistical similarities and differences between four pairs of $x - y$ values. This is one of the many data sets that come in R, and it is available in the object `anscombe`

```
# Anscombe's Quartet
anscombe
```

```
##    x1 x2 x3 x4   y1   y2    y3    y4
## 1  10 10 10  8 8.04 9.14  7.46  6.58
## 2   8  8  8  8 6.95 8.14  6.77  5.76
## 3  13 13 13  8 7.58 8.74 12.74  7.71
## 4   9  9  9  8 8.81 8.77  7.11  8.84
## 5  11 11 11  8 8.33 9.26  7.81  8.47
```

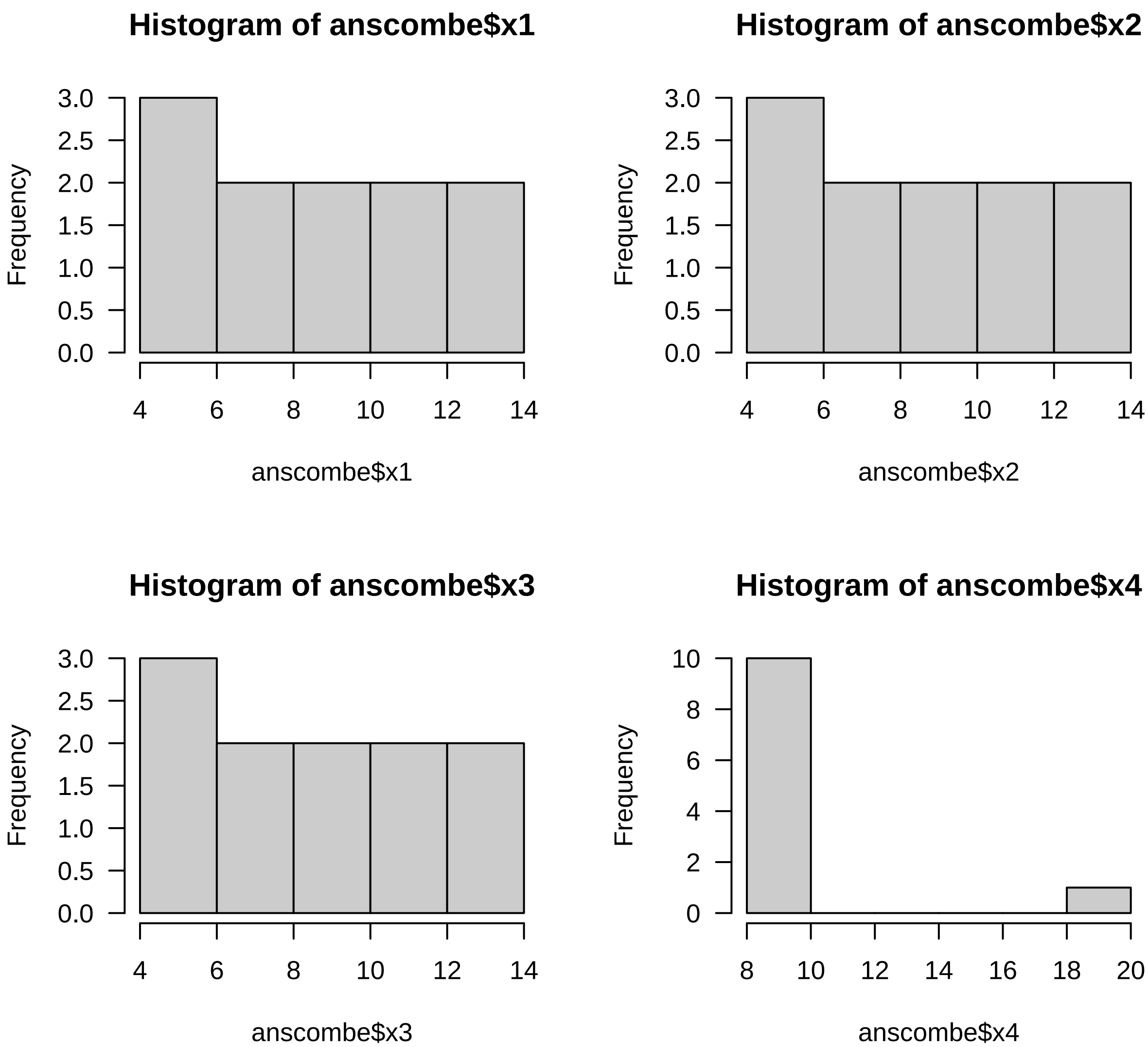
```
## 6   14 14 14  8  9.96 8.10  8.84  7.04
## 7    6  6  6  8  7.24 6.13  6.08  5.25
## 8    4  4  4 19  4.26 3.10  5.39 12.50
## 9   12 12 12  8 10.84 9.13  8.15  5.56
## 10   7  7  7  8  4.82 7.26  6.42  7.91
## 11   5  5  5  8  5.68 4.74  5.73  6.89
```

The data frame `anscombe` contains 8 variables: 4 `x`'s and 4 `y`'s. The way you should handle these variables is: `x1` with `y1`, `x2` with `y2`, and so on.

Histograms

Let's begin a univariate analysis by looking at the histograms of the `x` variables:

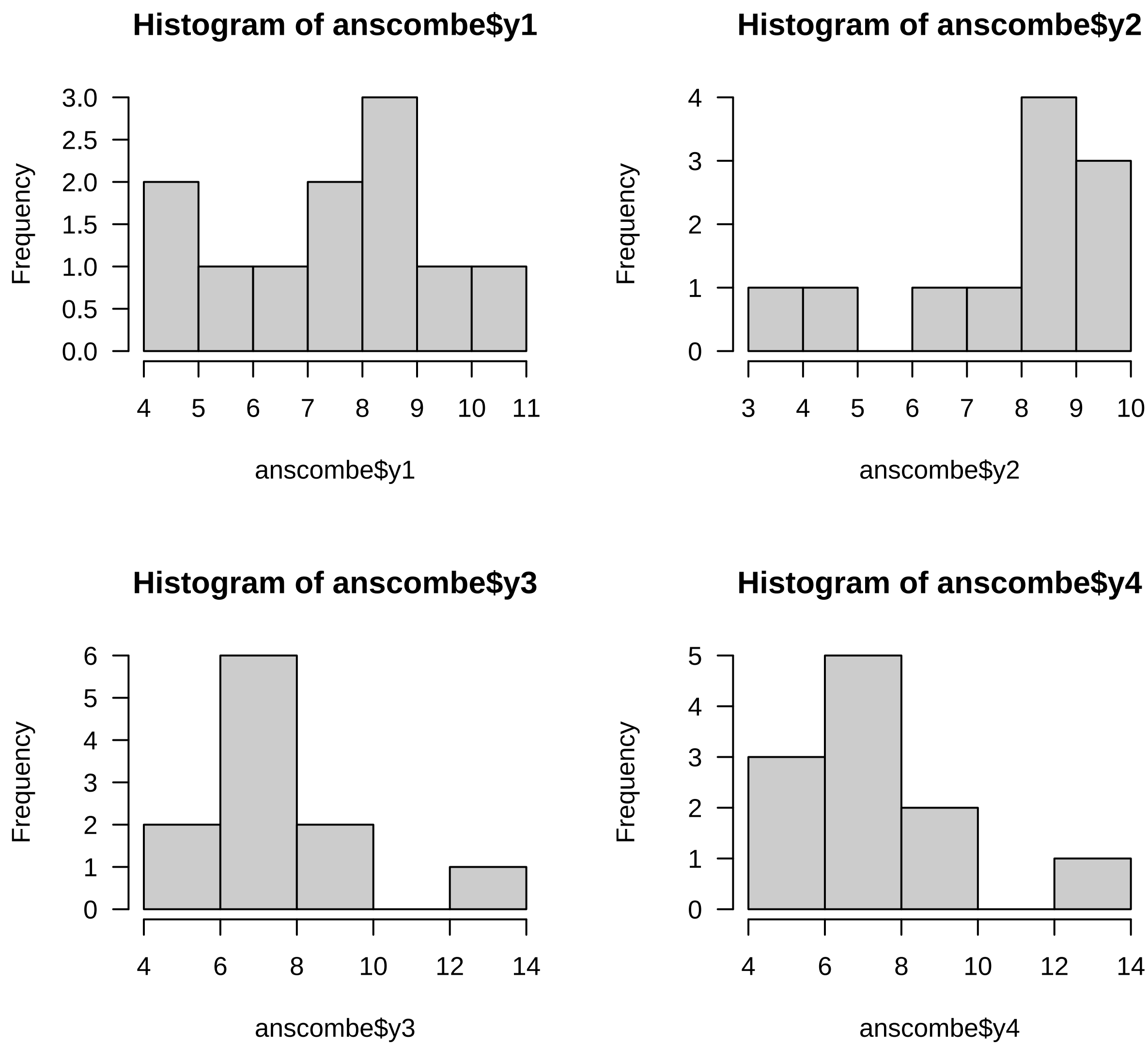
```r
# historgams of x-variables in 2x2 layout
op = par(mfrow = c(2, 2))
hist(anscombe$x1, col = 'gray80', las = 1)
hist(anscombe$x2, col = 'gray80', las = 1)
hist(anscombe$x3, col = 'gray80', las = 1)
hist(anscombe$x4, col = 'gray80', las = 1)
par(op)
```



Note that `x1`, and `x2`, and `x3` have the exact same histogram. If you look at the data frame, this is explained by the fact that these variables have the same values. In contrast, `x4` has almost all of its vallues equal to 8, except for one value of 19.

Now let's look at the histograms of the `y` variables:

```
# historgams of y-variables in 2x2 layout
op = par(mfrow = c(2, 2))
hist(anscombe$y1, col = 'gray80', las = 1)
hist(anscombe$y2, col = 'gray80', las = 1)
hist(anscombe$y3, col = 'gray80', las = 1)
hist(anscombe$y4, col = 'gray80', las = 1)
par(op)
```



Measures of Center and Spread

To get various summary statistics, you can use the function `summary()`

```
# basic summary of x-variables
summary(anscombe[ ,1:4])
```

```
##        x1             x2             x3             x4    
##  Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8  
##  1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8  
##  Median : 9.0   Median : 9.0   Median : 9.0   Median : 8  
##  Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9  
##  3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8  
##  Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19  
```

```r
# SD+ of x-variables
apply(anscombe[, 1:4], MARGIN = 2, FUN = sd)
```

```
##       x1       x2       x3       x4
## 3.316625 3.316625 3.316625 3.316625
```

Again, note that the `summary()` output for `x1`, and `x2`, and `x3` is the same. As for the standard deviation (SD^+), all `x`-variables have identical values. To calculate all the standard deviations at once, we are using the function `apply()`. This function allows you to *apply* a function, e.g. `sd()`, to the columns (`MARGIN = 2`) of the input data `anscombe[, 1:4]`.

Now let's get the summary indicators and standard deviation for the `y` variables:

```r
# basic summary of y-variables
summary(anscombe[ ,5:8])
```

```
##        y1               y2              y3              y4        
##  Min.   : 4.260   Min.   :3.100   Min.   : 5.39   Min.   : 5.250  
##  1st Qu.: 6.315   1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170  
##  Median : 7.580   Median :8.140   Median : 7.11   Median : 7.040  
##  Mean   : 7.501   Mean   :7.501   Mean   : 7.50   Mean   : 7.501  
##  3rd Qu.: 8.570   3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190  
##  Max.   :10.840   Max.   :9.260   Max.   :12.74   Max.   :12.500  
```

```r
# SD+ of y-variables
apply(anscombe[, 5:8], MARGIN = 2, FUN = sd)
```

```
##       y1       y2       y3       y4
## 2.031568 2.031657 2.030424 2.030579
```

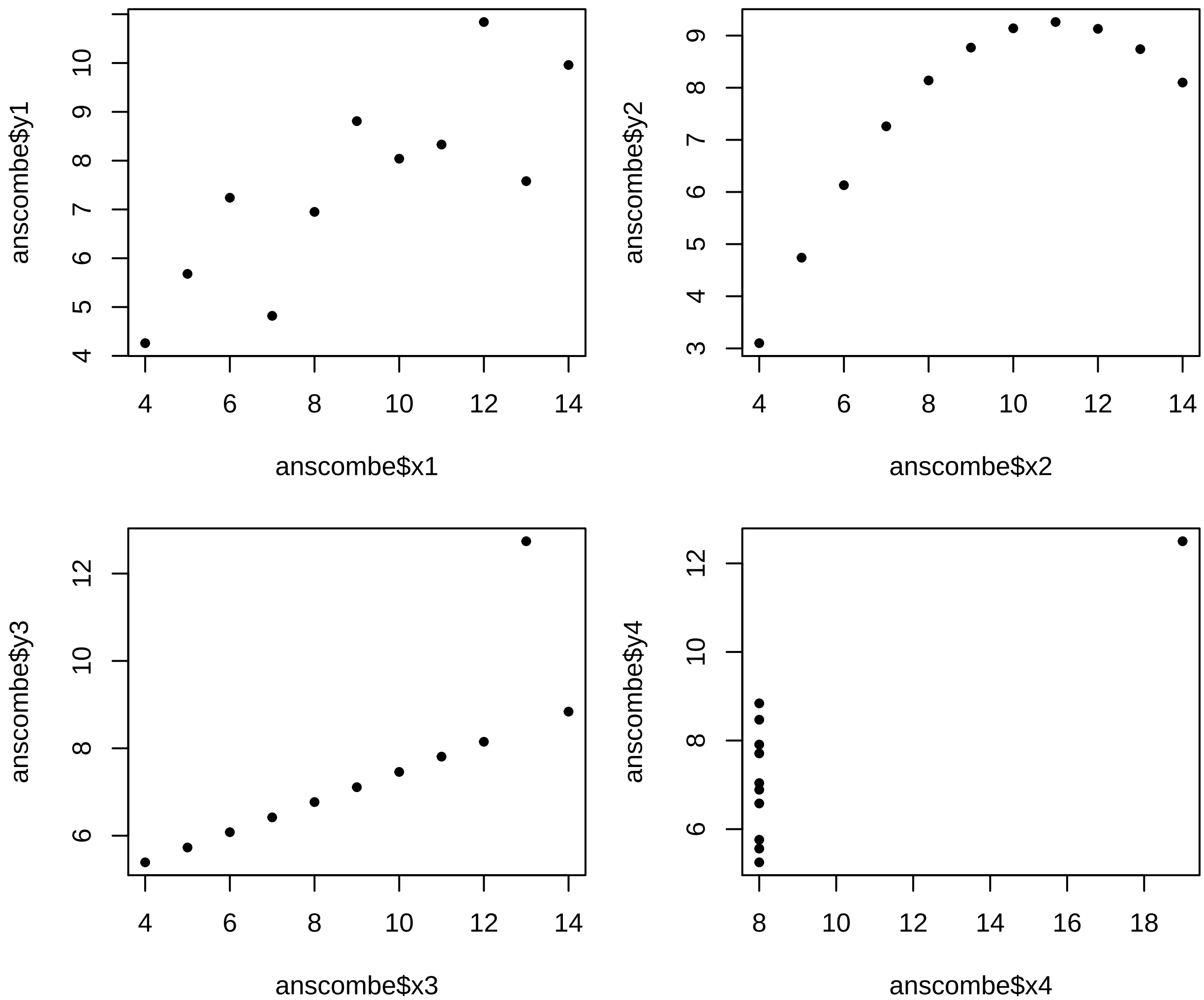
Can you notice anything special? Here's a hint: look at the averages and SDs. All four `y` variables have pretty much the same averages and SDs. But they have different ranges, quartiles, and medians. And if you take a peek at their histograms, their distirbutions also have different shapes.

Scatter Diagrams

The real interest in the Anscombe data set has to do with studying the association between each pair of $x - y$ values. The best way to start exploring pairwise associations is by looking at the scatter diagrams of each pair of points. How would you describe the shapes and patterns in each plot?

```r
# scatter diagrams in 2x2 layout
op = par(mfrow = c(2, 2), mar = c(4.5, 4, 1, 1))
plot(anscombe$x1, anscombe$y1, pch = 20)
plot(anscombe$x2, anscombe$y2, pch = 20)
plot(anscombe$x3, anscombe$y3, pch = 20)
```

```r
plot(anscombe$x4, anscombe$y4, pch = 20)
par(op)
```



- The first set `x1` and `y1` shows some degree of linear association. Although the dots do not lie on a line, we can say that they follow a linear pattern.
- The second set clearly has a non-linear pattern; instead, the dots follow some type of curve (perhaps quadratic) or a polynomial of degree greater than 1.
- The third set is almost perfectly linear except for the observation corresponding to $x = 13$ which falls outside the pattern of the rest of y values.
- The fourth set is similar to the third one in the sense that there is one observation (an outlier?) that does not follow the pattern of the other values. Most dots follow a vertical line at $x = 8$ except for the dot at $x = 19$.

Correlation Coefficient

In addition to the visual inspection of the scatter diagrams, statisticians use a summary measure to quantify the degree of *linear association* between two quantitative variables: the **coefficient of correlation**.

One way to obtain the correlation coefficient of two variables x and y is as the average of the product of x and y in standard units.

Let's consider `x1` and `y1` from the `anscobe` data set, and use R to "manually" calculate the correlation coefficient. This involves obtaining the average and the standard deviation SD, and then converting values to standard units:

```
# number of observations
n = nrow(anscombe)

# x1 in SU
x1_avg = mean(anscombe$x1)
x1_sd = sqrt((n-1)/n) * sd(anscombe$x1)
x1su = (anscombe$x1 - x1_avg) / x1_sd

# y1 in SU
y1_avg = mean(anscombe$y1)
y1_sd = sqrt((n-1)/n) * sd(anscombe$y1)
y1su = (anscombe$y1 - y1_avg) / y1_sd

# correlation: average of products
mean(x1su * y1su)
```

```
## [1] 0.8164205
```

Here's some good news. You don't really need to "manually" calcualte the correlation coefficient. R actually has a function to compute the correlation of two variables: `cor()`

```
# correlation coefficient
cor(anscombe$x1, anscombe$y1)
```

```
## [1] 0.8164205
```

Now let's get the correlation coefficients for all four pairs of variables:

```
cor(anscombe$x1, anscombe$y1)
```

```
## [1] 0.8164205
```

```
cor(anscombe$x2, anscombe$y2)
```

```
## [1] 0.8162365
```

```r
cor(anscombe$x3, anscombe$y3)
```

```
## [1] 0.8162867
```

```r
cor(anscombe$x4, anscombe$y4)
```

```
## [1] 0.8165214
```

Any surprises? As you can tell, all four pairs of x, y variables have basically the same correlation of 0.816. But not all of them have scatter diagrams in which the points clustered around a line.

The take home message is that the correlation coefficient can be misleading in the presence of outliers or non-linear association.

Properties of the Correlation Coefficient

One of the properties of the correlation coefficient is that it is a symmetric measure. By this we mean that the order of the variables is not important. You can interchange between x and y, and the correlation between them is unchanged:

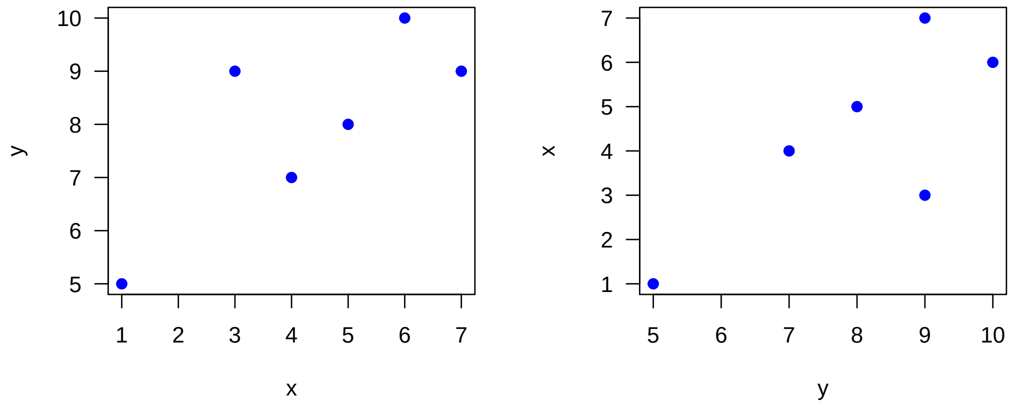
$$cor(x, y) = cor(y, x)$$

To illustrate this property, let's create two variables:

```r
# two variables
x = c(1, 3, 4, 5, 7, 6)
y = c(5, 9, 7, 8, 9, 10)
```

```r
op = par(mfrow = c(1,2))
plot(x, y, pch = 20, col = "blue", las = 1, cex = 1.5)
plot(y, x, pch = 20, col = "blue", las = 1, cex = 1.5)
par(op)
```

```r
op = par(mfrow = c(1,2))
plot(x, y, pch = 20, col = "blue", las = 1, cex = 1.5)
plot(y, x, pch = 20, col = "blue", las = 1, cex = 1.5)
```




```
par(op)
```

The scatter diagram changes depending on what variable is on each axis. However, the correlation coefficient in both cases is the same:

```
# symmetric
cor(x, y)
```

```
## [1] 0.7763238
```

```
cor(y, x)
```

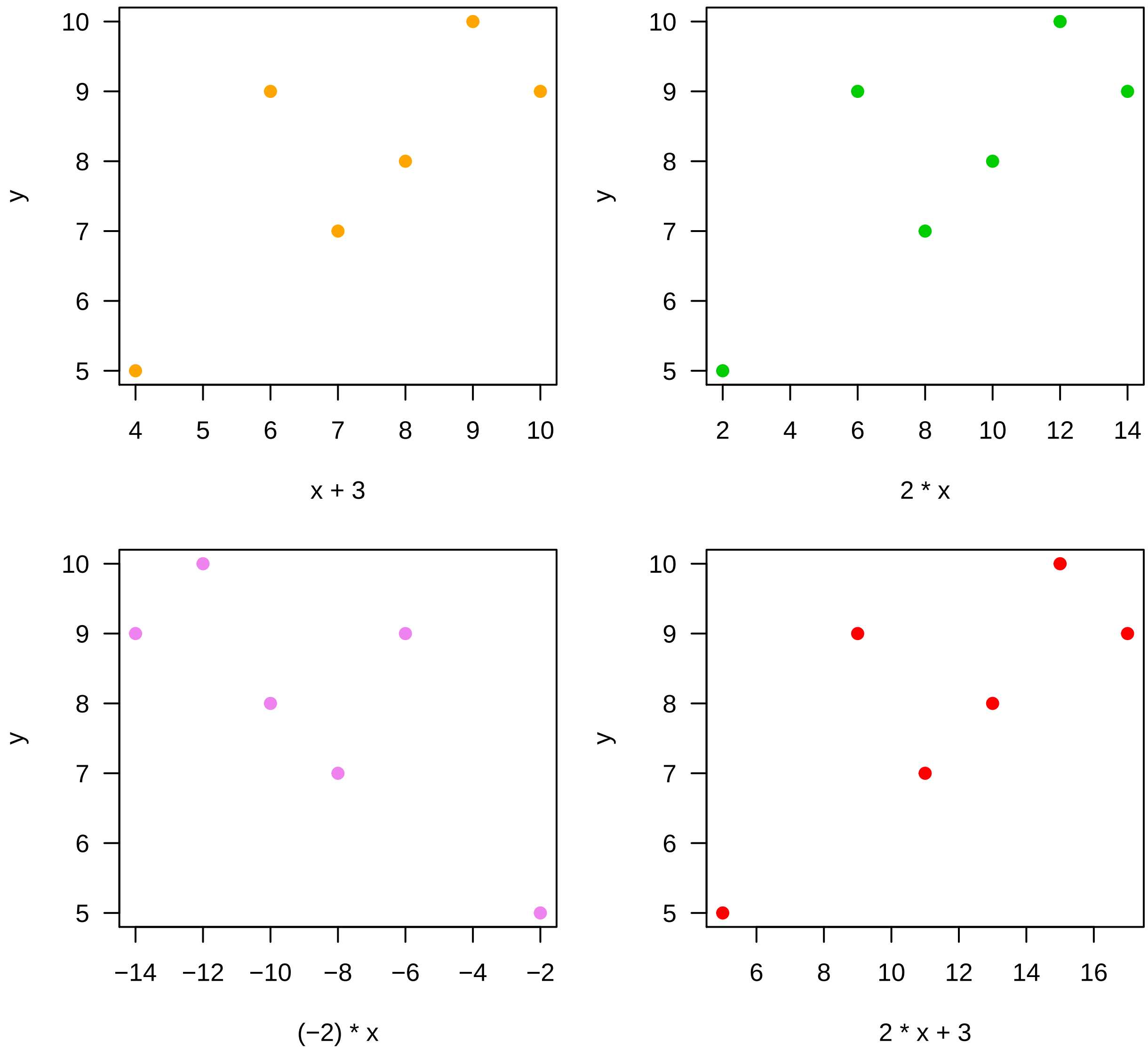
```
## [1] 0.7763238
```

Change of Scale

The other properties of the correlation coefficient have to do with what the FPP book calls *change of scale*. To be more precise, the considered change of scales involve **linear** change of scales (i.e. linear transformation). Typical operations that result in a linear change of scale are:

- Adding a scalar: $x + 3, y$
- Multiplying times a positive scalar: $2x, y$
- Multiplying times a negative scalar: $-2x, y$
- Adding and multiplying: $2x + 3, y$

```
# scatter diagrams in 2x2 layout
op = par(mfrow = c(2, 2), mar = c(4.5, 4, 1, 1))
plot(x + 3, y, pch = 20, col = "orange", las = 1, cex = 1.5)
plot(2 * x, y, pch = 20, col = "green3", las = 1, cex = 1.5)
plot((-2) * x, y, pch = 20, col = "violet", las = 1, cex = 1.5)
plot(2 * x + 3, y, pch = 20, col = "red", las = 1, cex = 1.5)
par(op)
```




```r
cor(x, y)
cor(x + 3, y)
cor(2 * x, y)
cor(-2 * x, y)
cor(2 * x + 3, y)
```

```
## [1] 0.7763238

## [1] 0.7763238

## [1] 0.7763238

## [1] -0.7763238

## [1] 0.7763238
```

Wat can you conclude from the change of scales? In which case the correlation coefficient is affected by such changes?