PCA Exercises

Part 1 - Descriptive statistics and correlation analysis

Exercises

1. Prove that $cov(x, x) = var(x)$
2. Prove that $cor(x, x) = 1$
3. In a finite sample, what is the difference between: the average difference between an individual and the mean VS the standard deviation (yes it's not the same)?
4. Compute the Pearson and Spearman correlation coefficient on the different variables from the iris dataset. Is there any differences? Why?
5. With R, create two samples: one where the Pearson coefficient correlation is equal to 1 (but with 2 distinct variables) and one where it's between [-0.1 ; 0.1]. Use a sample size >100.
6. What is the condition on the type of the variables used in order to compute the Pearson (or Spearman) coefficient correlation (it's implicitly said in the course)?
7. What are the condition(s) that make possible that `cov(x,y)` is equal to `cor(x,y)`?
8. Why the Pearson correlation coefficient isn't enough to say that there is a causal link between the variables?
9. What happens when doing `cor(X)`, where X is a matrix of quantitative variables?
10. *With your own words*, make a short description of what correlation is.
11. Based on the sample below, explain *with your own words* why 2 variables created the same way are not correlated.

```r
x = rnorm(n=1000)
y = rnorm(n=1000)
cor.test(x,y)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = -0.33022, df = 998, p-value = 0.7413
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07239858  0.05157409
## sample estimates:
##         cor
## -0.01045241
```

12. Create a scatter plot with the sample size in the x-axis and the Pearson correlation coefficient in the y-axis for a range of 10 to 500 for the sample size. For this, you have to use the `generate()` function below. Do the same thing with the p-value in the y-axis. What do you see? What are the implications?

```r
generate = function(sample_size){

  #generate sample
  x = rnorm(n=sample_size, sd=15)
  y = 1.001*x + rnorm(n=sample_size, sd=30)

  #compute correlation for the sample
  correlation = cor.test(x,y)
  p = correlation$p.value
  r = correlation$estimate

  return(r) #change it to p for the second plot
}
```