Correlation & causation (practice in R)

Problem 1: Testing for marginal correlation

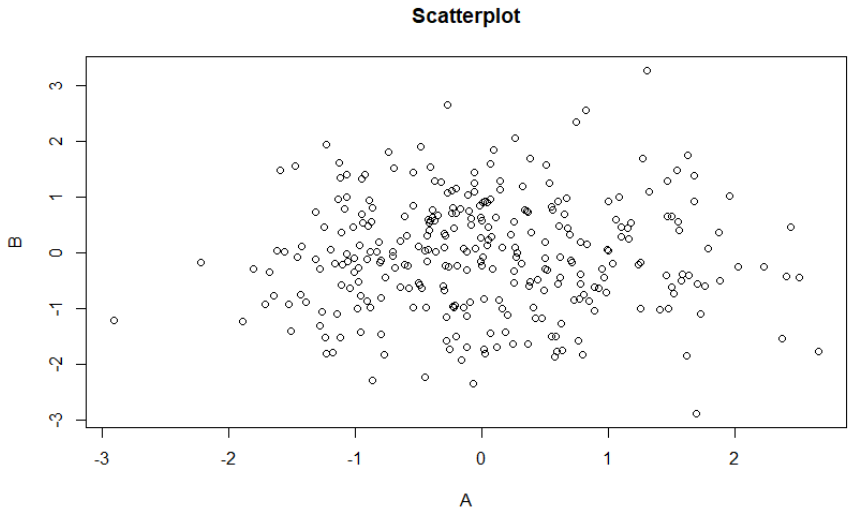
The first thing we need is to load the data into RStudio:

```
# upload data
path = '/Users/ninag/Documents/AlgorithmsAndBigDataInChemistry/R/correlation_causation_practice/data.rds'
data = readRDS(path)
```

The next step is to apply the attach function to the data so that A and B can be called without accessing data$. Then display observations A and B as a scatterplot:

```
# make a plot
attach(data)
plot(A, B, main = 'Scatterplot', xlab = 'A', ylab = 'B')
```

Output:



So we don't see correlation. This is consistent with the fact that in Figure 1 they are not connected.

Now we need to check the correlation between A and B:

```
# test correlation between A and B
correlatin.result = cor.test(data$A, data$B, method = 'pearson', althernative = 'two.sided')

# print results
correlatin.result$estimate
correlatin.result$p.value
```

Correlation = 0.01169715, it is close to zero.

P-value = 0.840103, it is very high (usually we use 0,05).

Conclusion: there is no correlation.

Problem 2: Testing for partial correlation

To check the partial correlation of A and B we use linear regression and find the residuals:

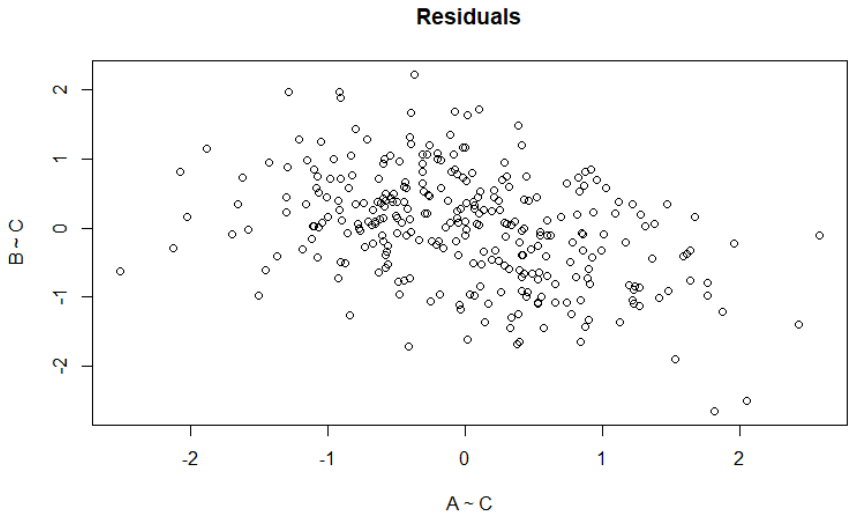
```
# regression A to C
ac.regression = lm(A ~ C, data = data)
ac.residuals = residuals(ac.regression)

# regression B to C
bc.regression = lm(B ~ C, data = data)
bc.residuals = residuals(bc.regression)
```

Then display residuals A ~ C and B ~ C as a scatterplot:

```
# make a plot of residuals
plot(ac.residuals, bc.residuals, main = 'Residuals', xlab = 'A ~ C', ylab = 'B ~ C')
```

Output:



By the look of the cloud of points, we can assume that it is slightly negatively correlated. Let's also check the correlation:

```
# verify this with the cor test function
cor.test(ac.residuals, bc.residuals)
```

Correlation = -0.3992521, it is what we expected

P-value = 6.6e-13, so we can be sure in our results

Conclusion: As a result, there is a slight negative correlation here. This is consistent with the fact that in Figure 1 A - C and B - C are connected.

Problem 3: Running the PC algorithm

To perform this task, install and download the pcalg package, as well as all that was required for further visualization:

```
# install packages
if (!require('BiocManager', quietly = TRUE))
  install.packages('BiocManager')

BiocManager::install('RBGL')

BiocManager::install('graph')

BiocManager::install('Rgraphviz')

install.packages('pcalg')

# PC algorithm
library(pcalg)
```

Convert the data into the required format:

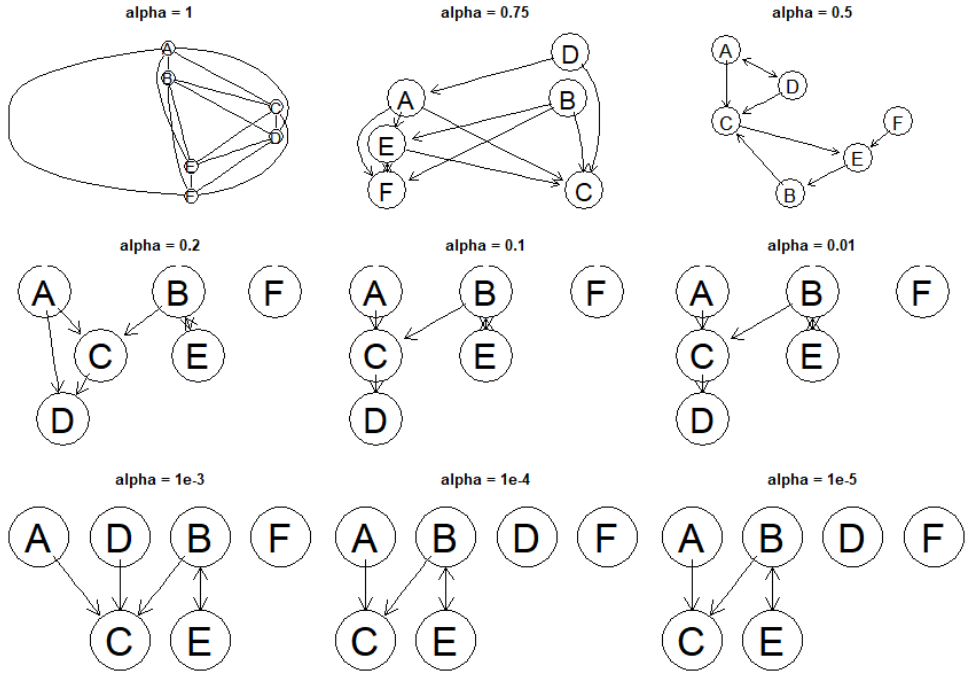
```
# prepare the data in the specific format
suffStat = list(C = cor(data), n = nrow(data))
```

We run the algorithm for different alpha values and visualize the resulting graphs:

```
# run the algorithm at different significance levels
pc.result.1 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 1, labels = colnames(data), verbose = TRUE)
pc.result.2 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.75, labels = colnames(data), verbose = TRUE)
pc.result.3 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.5, labels = colnames(data), verbose = TRUE)
pc.result.4 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.2, labels = colnames(data), verbose = TRUE)
pc.result.5 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.1, labels = colnames(data), verbose = TRUE)
pc.result.6 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.01, labels = colnames(data), verbose = TRUE)
pc.result.7 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.001, labels = colnames(data), verbose = TRUE)
pc.result.8 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.0001, labels = colnames(data), verbose = TRUE)
pc.result.9 = pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.00001, labels = colnames(data), verbose = TRUE)

# make plots
par(mfrow = c(3, 3))
plot(pc.result.1, main = 'alpha = 1')
plot(pc.result.2, main = 'alpha = 0.75')
plot(pc.result.3, main = 'alpha = 0.5')
plot(pc.result.4, main = 'alpha = 0.2')
plot(pc.result.5, main = 'alpha = 0.1')
plot(pc.result.6, main = 'alpha = 0.01')
plot(pc.result.7, main = 'alpha = 1e-3')
plot(pc.result.8, main = 'alpha = 1e-4')
plot(pc.result.9, main = 'alpha = 1e-5')
```

Output:



We see that for large alpha the graph is fully connected.

With decreasing level of significance more and more edges are removed.

We also see that at alpha up to 1e-4 the connection between C and D is removed while the connection between B and E is still two-sided.

Conclusion: accordingly, it is not possible to achieve the same dependence as in Figure 1.