Topic 2: Exercise 1

Daniel Alonso

November 28th, 2020

```
library(dplyr)
library(Rcpp)
```

Importing libraries

```
d <- read.csv("../../datasets/Colleges.csv")
```

Importing data as described by exercise

```
d$Private <- ifelse(d$Private == "Yes", 1, 0)
```

Replacing binary variable Private with 1 and 0

```
data <- d %>% dplyr::select('Private','Apps','Accept','Enroll','F.Undergrad')
```

Selecting columns

```
cov_matrix <- cov(data)
cov_matrix
#>                  Private          Apps        Accept       Enroll  F.Undergrad
#> Private        0.1986559     -745.3552     -519.2042    -235.1942    -1330.764
#> Apps        -745.3552439 14978459.5301  8949859.8119 3045255.9876 15289702.474
#> Accept      -519.2042169  8949859.8119  6007959.6988 2076267.7627 10393582.435
#> Enroll      -235.1942393  3045255.9876  2076267.7627  863368.3923  4347529.884
#> F.Undergrad -1330.7637175 15289702.4742 10393582.4355 4347529.8841 23526579.326
```

Calculating covariances

```
corr_matrix <- cov2cor(cov_matrix)
corr_matrix
#>                Private       Apps     Accept     Enroll F.Undergrad
#> Private      1.0000000 -0.4320947 -0.4752520 -0.5679078  -0.6155605
#> Apps        -0.4320947  1.0000000  0.9434506  0.8468221   0.8144906
#> Accept      -0.4752520  0.9434506  1.0000000  0.9116367   0.8742233
#> Enroll      -0.5679078  0.8468221  0.9116367  1.0000000   0.9646397
#> F.Undergrad -0.6155605  0.8144906  0.8742233  0.9646397   1.0000000
```

Calculating correlations

Experimenting a little bit with the private variable Let's try changing the Yes to 0 and the No to 1 and checking the covariances and correlations

```
d <- read.csv("../../datasets/Colleges.csv")
d$Private <- ifelse(d$Private == "Yes", 0, 1)
data <- d %>% dplyr::select('Private','Apps','Accept','Enroll','F.Undergrad')
```

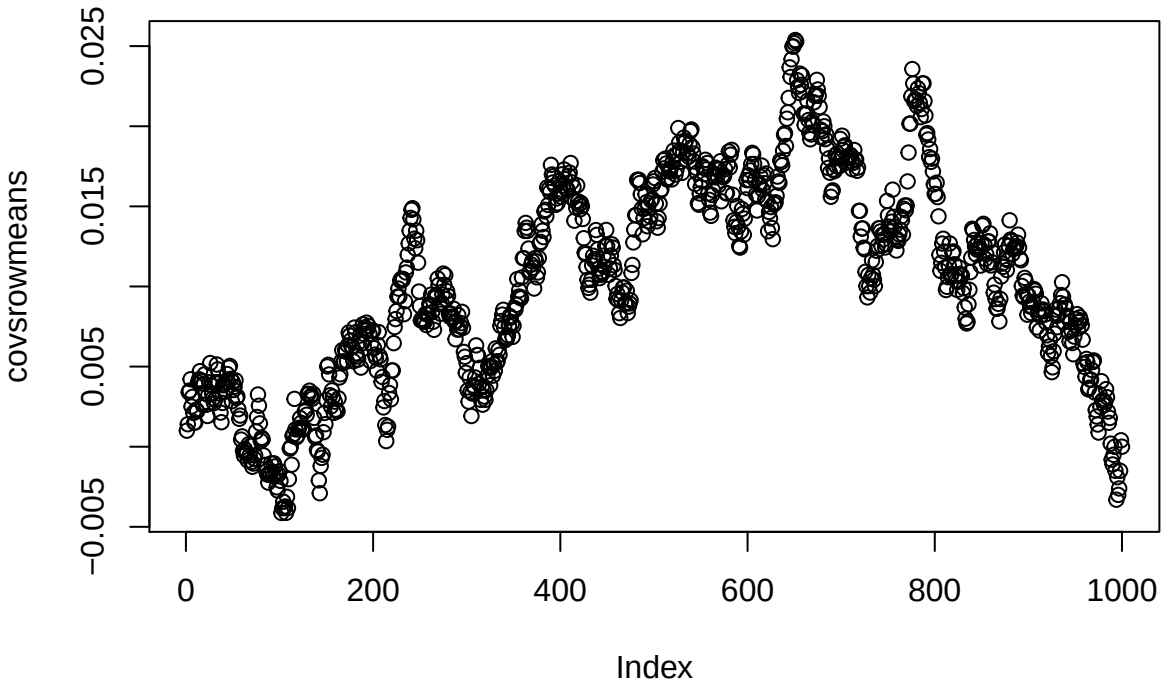
```
cov_matrix <- cov(data)
cov_matrix
#>                  Private         Apps       Accept       Enroll  F.Undergrad
#> Private        0.1986559 7.453552e+02 5.192042e+02     235.1942     1330.764
#> Apps         745.3552439 1.497846e+07 8.949860e+06 3045255.9876 15289702.474
#> Accept       519.2042169 8.949860e+06 6.007960e+06 2076267.7627 10393582.435
#> Enroll       235.1942393 3.045256e+06 2.076268e+06  863368.3923  4347529.884
#> F.Undergrad 1330.7637175 1.528970e+07 1.039358e+07 4347529.8841 23526579.326
corr_matrix <- cov2cor(cov_matrix)
corr_matrix
#>                Private      Apps    Accept    Enroll F.Undergrad
#> Private      1.0000000 0.4320947 0.4752520 0.5679078   0.6155605
#> Apps         0.4320947 1.0000000 0.9434506 0.8468221   0.8144906
#> Accept       0.4752520 0.9434506 1.0000000 0.9116367   0.8742233
#> Enroll       0.5679078 0.8468221 0.9116367 1.0000000   0.9646397
#> F.Undergrad 0.6155605 0.8144906 0.8742233 0.9646397   1.0000000
```

We get the same numbers with reversed signs.

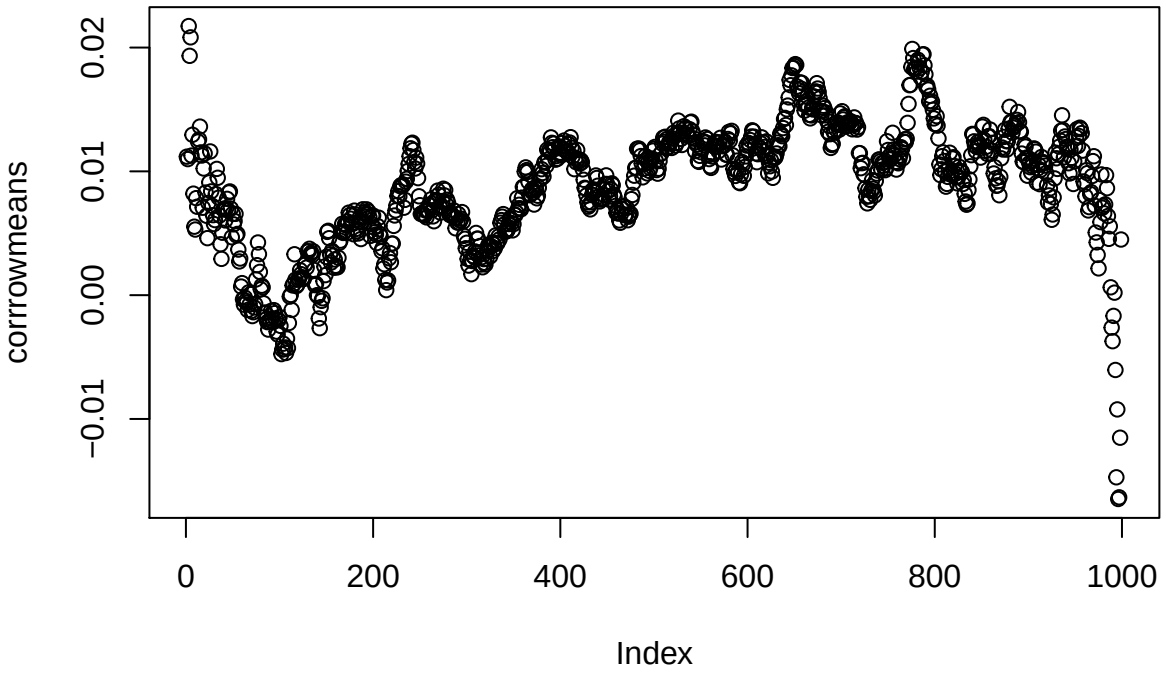
Let's play with the amount of 1s and 0s in Private and compare it to a simulated variable with only positive values in order to see how the covariance and correlation change and plot it.

```
using Random
using Statistics
using Plots

function simulation_binaries(nrows, simulations)
    covs = zeros(Float64, nrows, simulations)
    corr = zeros(Float64, nrows, simulations)
    for s in 1:simulations
        pvtapps = zeros(Float64, nrows, 2)
        pvtapps[:,2] = rand(1:10,nrows)
        for i in 1:nrows
            pvtapps[:,1] = vcat(zeros(nrows-i,1), ones(i,1))
            covs[i,s] = cov(pvtapps[:,1],pvtapps[:,2])
            corr[i,s] = cor(pvtapps[:,1],pvtapps[:,2])
        end
    end
    covsrowmeans = zeros(Float64, nrows)
    corrrowmeans = zeros(Float64, nrows)
    for i in 1:nrows
        covsrowmeans[i] = mean(covs[i,:])
        corrrowmeans[i] = mean(corr[i,:])
    end
    return covsrowmeans, corrrowmeans
end
#> simulation_binaries (generic function with 1 method)

(covsrowmeans, corrrowmeans) = simulation_binaries(1000,10)
#> ([0.0010017017017017015, 0.0014028028028028028, 0.003405505505505505, 0.003506306306306306, 0.004207
```

```
covsrowmeans <- JuliaCall::julia_eval("covsrowmeans")
plot(covsrowmeans)
```




```
corrrowmeans <- JuliaCall::julia_eval("corrrowmeans")
plot(corrrowmeans)
```



What information does the sample covariance provide?

We know that because the Private variable (binary variable) has only 2 possible values, its covariance with other variables is always going to be relatively small and will not provide much information.

What information does the sample correlation provide?