File for the CAPM exercise on real data

Starter code for the CAPM exercise. We will load the data and run one of the regressions to show how to do it in Julia.

Loading the data

We start by loading the `RDatasets` package to get the data.

```
using RDatasets
```

Loading the *capm* dataset.

capm =

	RFood	RDur	RCon	RMRF	RF
1	-4.59	0.87	-6.84	-6.99	0.33
2	2.62	3.46	2.78	0.99	0.29
3	-1.67	-2.28	-0.48	-1.46	0.35
4	0.86	2.41	-2.02	-1.7	0.19
5	7.34	6.33	3.69	3.08	0.27
6	4.99	-1.26	2.05	2.09	0.24
7	-1.52	-5.09	-3.79	-2.23	0.13
8	3.96	4.38	-1.08	2.85	0.17
9	-3.98	-4.23	-4.71	-6.0	0.16
10	0.99	1.17	-1.44	-0.7	0.22
more					
516	-1.02	-4.89	-5.13	-5.42	0.11

```
capm = dataset("Ecdat", "capm")
```

Defining the dependent variable and regressors, where we use the `ones()` function to generate the intercept.

```
Y =
 [-4.92, 2.33, -2.02, 0.67, 7.07, 4.75, -1.65, 3.79, -4.14, 0.77, 9.09, 3.96, 4.56, 4.39,
```

```
Y = capm.RFood.-capm.RF
```

```
X1 =
 [-7.32, 0.7, -1.81, -1.89, 2.81, 1.85, -2.36, 2.68, -6.16, -0.92, 4.59, 4.52, 6.04, 3.4,
```

```
X1 = capm.RMRF.-capm.RF
```

```
X0 =
 [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
```

```
X0 = ones(size(X1,1))
```

We construct the matrix of regressors by

$$X = [\iota,\ r_{mt} - r_f]$$

```
X = 516×2 Matrix{Float64}:
     1.0   -7.32
     1.0    0.7
     1.0   -1.81
     1.0   -1.89
     1.0    2.81
     1.0    1.85
     1.0   -2.36
     ⋮
     1.0   -8.41
     1.0    0.5
     1.0  -10.24
     1.0    7.16
     1.0    5.84
     1.0   -5.53
```

```
X = [X0 X1]
```

OLS estimation

$$\hat{\beta} = (X'X)^{-1}(X'Y)$$

```
beta_ols =   [0.236624, 0.783144]
```

```
beta_ols = (X'*X)\X'*Y
```

Note that there are small differences with the results in the book. It may be due to small errors at capturing the data.

Variance estimation

$$\hat{s^2} = \frac{1}{(n-k)}\sum_i^n (Y_i - X_i\hat{\beta})^2$$

so that

$$\widehat{Var}(\hat{\beta}) = \hat{s^2}(X'X)^{-1}$$

```
sigma_ols = 8.312701328241321
```

```
sigma_ols = (Y-X*beta_ols)'*(Y-X*beta_ols)/(size(X1,1)-2)
```

```
var_beta = 2×2 Matrix{Float64}:
            0.0161126    4.60399e-5
            4.60399e-5   0.000794801
```

```
var_beta = sigma_ols*inv(X'*X)
```

```
std_int = 0.1269352323795595
```

```
std_int = sqrt(var_beta[1,1])
```

```
std_beta = 0.028192210122901688
```

```
std_beta = sqrt(var_beta[2,2])
```