Statistical Analysis - Part I

Part I

- Correlation - two variables
- Correlation - multiple variables
- Visualizing correlations

Part II

- Linear and multi-linear regression
- Logistic regression
- Polynomial regression

Correlation - two variables

Helps identify if a linear relationship exists between two variables

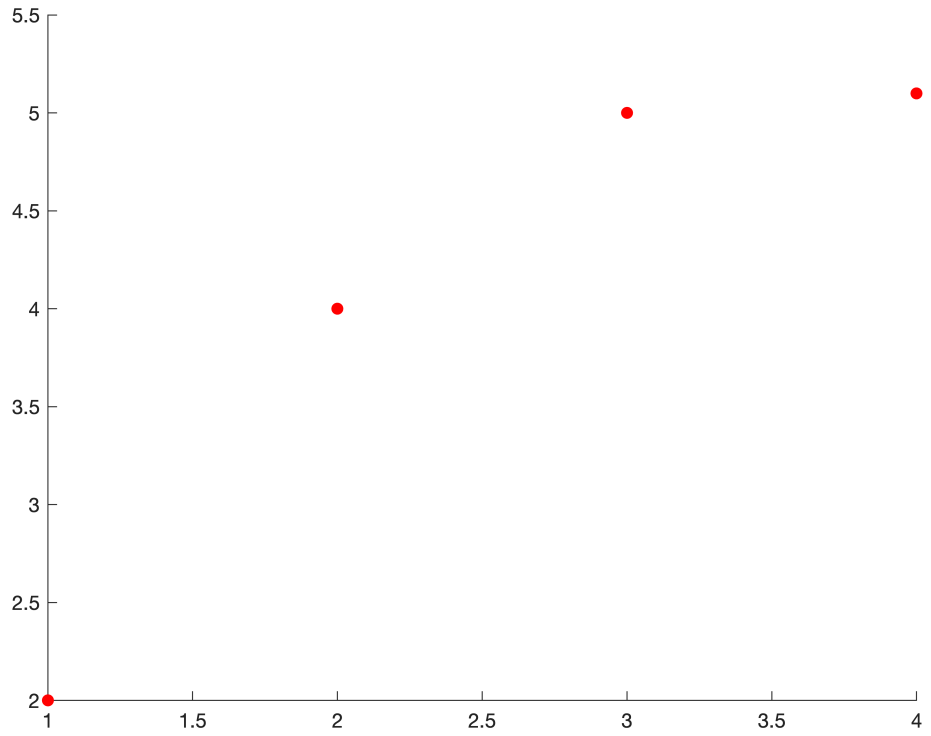
Different types of correlation - Pearson's and Spearman (rank)

- **Pearson's** - measures the strength and direction of a linear relationship between two variables
- **Spearman** - measures the strength and direction of a monotonic relationship between two variables

Both of these measures of correlation are on a scale of -1 to 1.

Consider the two arrays, *A* and *B*:

```
A = [1, 2, 3, 4];
B = [2, 4, 5, 5.1];
scatter(A,B,'red','filled')
```



Based on the above plot, what values would you guess for the Pearson's and Spearman correlation estimates?

We can use the `corr` function, which outputs the pairwise correlation between the columns of the two matrices. With the `corr` function, we have the option to specify Pearson or Spearman correlation by using the `'type'` option. Note that we have to transpose the arrays in order to arrange the data into columns.

```
r = corr(A',B',"type","Pearson")
```

```
r = 0.9244
```

```
r_spear = corr(A',B',"type","Spearman")
```

```
r_spear = 1
```

As expected, the Pearson correlation value is fairly high as the two variables appear to have a strong linear relationship.

The output of 1 for the Spearman correlation also makes sense because the two arrays are completely **monotonic**, meaning that as one variable increases, so does the other.

Once we have *r,* we can easily calculate the **coefficient of determination,** r^2. The coefficient of determination measures the proportion of variance in the dependent variable that can be explained by the independent variable.

```
r_squared = r^2
```

```
r_squared = 0.8545
```

Correlation - Multiple Variables

We can also examine the correlation between more than two variables. Load the Framingham data set for use in the following examples.

```
fram = readtable("../frmgham.xls");
```

Use the `corr` function to find the correlation between systolic and diastolic blood pressure. Note that the default correlation type for `corr` is `"Pearson"`.

```
corr(fram.SYSBP, fram.DIABP)
```

```
ans = 0.7116
```

When the `corr` function is used on more than two variables, the output is a matrix of the correlation coefficients. Find the correlation between systolic blood pressure, diastolic blood pressure, and age.

```
corr([fram.SYSBP, fram.DIABP, fram.AGE])
```

```
ans = 3×3
    1.0000    0.7116    0.3890
    0.7116    1.0000    0.0693
    0.3890    0.0693    1.0000
```

Each value in the matrix corresponds to the correlation between the variables in that particular row and column. For example, the correlation between diastolic blood pressure and age is 0.0693.

This same process can be used for any number of variables. We will now analyze the variables in columns 2 through 11 of the Framingham data set. First, extract these columns from the full data set.

```
fram_2_11 = fram(:,2:11);
```

What variables are included in these ten columns?

```
var_names = fram_2_11.Properties.VariableNames;
var_names'
```

```
ans = 10×1 cell array
    {'SEX'     }
    {'TOTCHOL' }
    {'AGE'     }
    {'SYSBP'   }
    {'DIABP'   }
    {'CURSMOKE'}
    {'CIGPDAY' }
    {'BMI'     }
    {'DIABETES'}
```

```
    {'BPMEDS'  }
```

In order for the `corr` function to work correctly, we need to remove the observations containing NaN. One option is to create a new matrix with the `rmmissing` function, which removes all rows with missing data:

```
fram_noNaN = rmmissing(fram_2_11);
r = corr(fram_noNaN.Variables)
```

```
r = 10×10
    1.0000    0.1309    0.0267    0.0438   -0.0526   -0.1562   -0.2514   -0.0740 ···
    0.1309    1.0000    0.1655    0.1580    0.1360   -0.0483   -0.0372    0.0843
    0.0267    0.1655    1.0000    0.3832    0.0576   -0.2522   -0.2253    0.0612
    0.0438    0.1580    0.3832    1.0000    0.7108   -0.1438   -0.1011    0.2707
   -0.0526    0.1360    0.0576    0.7108    1.0000   -0.0750   -0.0298    0.3340
   -0.1562   -0.0483   -0.2522   -0.1438   -0.0750    1.0000    0.7806   -0.1648
   -0.2514   -0.0372   -0.2253   -0.1011   -0.0298    0.7806    1.0000   -0.0996
   -0.0740    0.0843    0.0612    0.2707    0.3340   -0.1648   -0.0996    1.0000
   -0.0232    0.0060    0.1323    0.1362    0.0229   -0.0429   -0.0451    0.0889
    0.0838    0.0612    0.2002    0.3287    0.2267   -0.0896   -0.0845    0.0984
```

Another option is to specify for `corr` to only analyze complete rows:

```
r = corr(fram_2_11.Variables,"rows","complete")
```

```
r = 10×10
    1.0000    0.1309    0.0267    0.0438   -0.0526   -0.1562   -0.2514   -0.0740 ···
    0.1309    1.0000    0.1655    0.1580    0.1360   -0.0483   -0.0372    0.0843
    0.0267    0.1655    1.0000    0.3832    0.0576   -0.2522   -0.2253    0.0612
    0.0438    0.1580    0.3832    1.0000    0.7108   -0.1438   -0.1011    0.2707
   -0.0526    0.1360    0.0576    0.7108    1.0000   -0.0750   -0.0298    0.3340
   -0.1562   -0.0483   -0.2522   -0.1438   -0.0750    1.0000    0.7806   -0.1648
   -0.2514   -0.0372   -0.2253   -0.1011   -0.0298    0.7806    1.0000   -0.0996
   -0.0740    0.0843    0.0612    0.2707    0.3340   -0.1648   -0.0996    1.0000
   -0.0232    0.0060    0.1323    0.1362    0.0229   -0.0429   -0.0451    0.0889
    0.0838    0.0612    0.2002    0.3287    0.2267   -0.0896   -0.0845    0.0984
```

These two methods should produce the same results.

Now use the `maxk` function to find the five highest correlation features (positive or negative) for each variable as well as the index. We will specify for the function to return the five largest values for each column, the indices of these vaues, and to use absolute values to find the maximums.

```
[max_corr, I] = maxk(r,5,"ComparisonMethod","abs")
```

```
max_corr = 5×10
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000 ···
   -0.2514    0.1655    0.3832    0.7108    0.7108    0.7806    0.7806    0.3340
   -0.1562    0.1580   -0.2522    0.3832    0.3340   -0.2522   -0.2514    0.2707
    0.1309    0.1360   -0.2253    0.3287    0.2267   -0.1648   -0.2253   -0.1648
    0.0838    0.1309    0.2002    0.2707    0.1360   -0.1562   -0.1011   -0.0996
I = 5×10
     1     2     3     4     5     6     7     8     9    10
     7     3     4     5     4     7     6     5     4     4
     6     4     6     3     8     3     1     4     3     5
     2     5     7    10    10     8     3     6     8     3
    10     1    10     8     2     1     4     7    10     8
```

We can make the top ranking features easier to interpret by creating tables with the variable names, as opposed to the indices.

```
corr_table = array2table(max_corr, "VariableNames",var_names);
I_table = array2table(var_names(I),"VariableNames",var_names)
```

```
I_table = 5×10 table
```

...

	SEX	TOTCHOL	AGE	SYSBP	DIABP	CURSMOKE	CIGPDAY	BMI
1	'SEX'	'TOTCHOL'	'AGE'	'SYSBP'	'DIABP'	'CURSMOKE'	'CIGPDAY'	'BMI'
2	'CIGPDAY'	'AGE'	'SYSBP'	'DIABP'	'SYSBP'	'CIGPDAY'	'CURSMOKE'	'DIABP'
3	'CURSMOKE'	'SYSBP'	'CURSMOKE'	'AGE'	'BMI'	'AGE'	'SEX'	'SYSBP'
4	'TOTCHOL'	'DIABP'	'CIGPDAY'	'BPMEDS'	'BPMEDS'	'BMI'	'AGE'	'CURSMOKE'
5	'BPMEDS'	'SEX'	'BPMEDS'	'BMI'	'TOTCHOL'	'SEX'	'SYSBP'	'CIGPDAY'

Looking at these two tables, what are some of the variables with the highest correlation? Aside from the 1's in the correlation coefficient matrix (which refer to the variable's correlation with itself), what are the three highest values and what features do they correspond to?

- Correlation = 0.7806 - CIGPDAY, CURSMOKE
- Correlation = 0.7108 - SYSBP, DIABP
- Correlation = 0.3832 - SYSBP, AGE

Does it makes sense that these variables have strong linear relationships? Are there any other variables that you would expect to have high correlation coefficients?

Repeat the same process on columns 2-11 of the Framingham data set, but this time using rank correlation.

```
r_rank = corr(fram_noNaN.Variables,'type','Spearman');
[max_corr_rank, I_rank] = maxk(r_rank,5,"ComparisonMethod","abs");
max_corr_rank = array2table(max_corr_rank,"VariableNames",var_names)
```

```
max_corr_rank = 5×10 table
```

...

	SEX	TOTCHOL	AGE	SYSBP	DIABP	CURSMOKE	CIGPDAY	BMI
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	-0.2149	0.1833	0.3914	0.7084	0.7084	0.9509	0.9509	0.3235
3	-0.1562	0.1721	-0.2551	0.3914	0.3235	-0.2520	-0.2551	0.2666
4	-0.1350	0.1471	-0.2520	0.2958	0.2126	-0.1710	-0.2149	-0.1710
5	0.1270	0.1270	0.1981	0.2666	0.1471	-0.1562	-0.1483	-0.1483

```
I_rank = array2table(var_names(I_rank),"VariableNames",var_names)
```

I_rank = 5×10 table

...

	SEX	TOTCHOL	AGE	SYSBP	DIABP	CURSMOKE	CIGPDAY	BMI
1	'SEX'	'TOTCHOL'	'AGE'	'SYSBP'	'DIABP'	'CURSMOKE'	'CIGPDAY'	'BMI'
2	'CIGPDAY'	'AGE'	'SYSBP'	'DIABP'	'SYSBP'	'CIGPDAY'	'CURSMOKE'	'DIABP'
3	'CURSMOKE'	'SYSBP'	'CIGPDAY'	'AGE'	'BMI'	'AGE'	'AGE'	'SYSBP'
4	'BMI'	'DIABP'	'CURSMOKE'	'BPMEDS'	'BPMEDS'	'BMI'	'SEX'	'CURSMOKE'
5	'TOTCHOL'	'SEX'	'BPMEDS'	'BMI'	'TOTCHOL'	'SEX'	'BMI'	'CIGPDAY'

Do the same features have the strongest correlation coefficients?

- Correlation = 0.9509 - CIGPDAY, CURSMOKE
- Correlation = 0.7084 - SYSBP, DIABP
- Correlation = 0.3914 - SYSBP, AGE

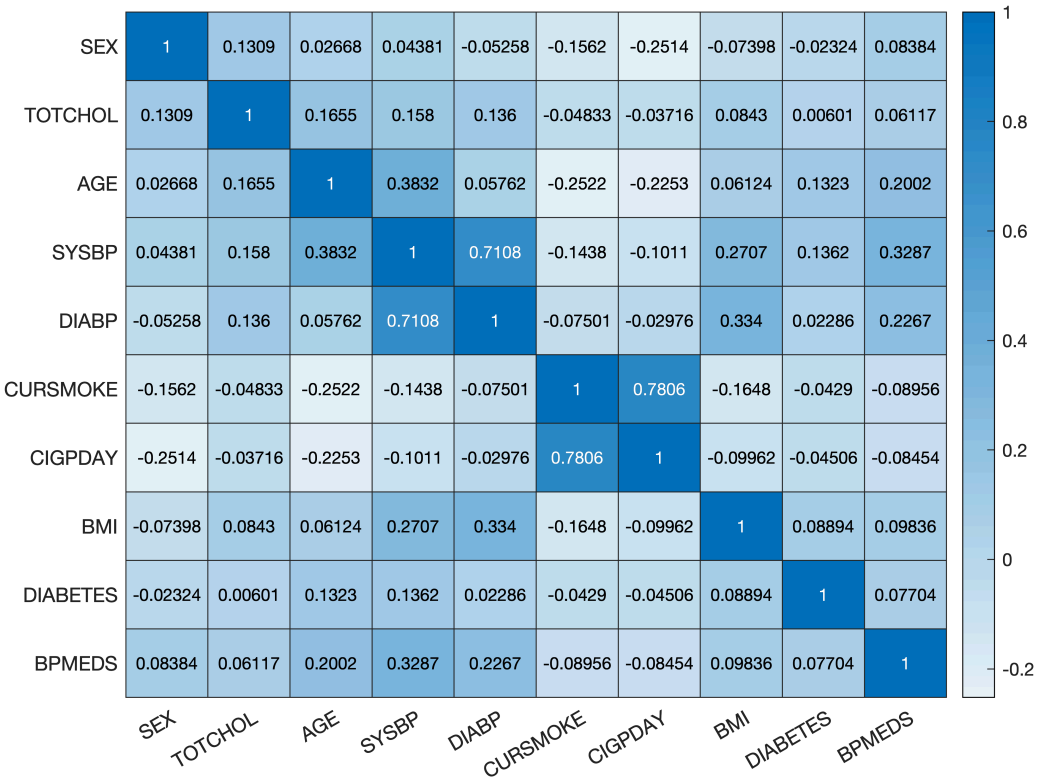
While the correlation coefficients are different from the Spearman's rank test, the variables with the strongest linear relationships are the same.

Visualizing Correlations

Heatmaps use color-coding to compare data, making it easy to observe variable relationships and patterns. Heatmaps are a useful tool for visualizing pairwise correlations. The `heatmap` command in MATLAB accepts both tables and matrices as inputs.

Create a heatmap for the pairwise correllations of our ten variables.

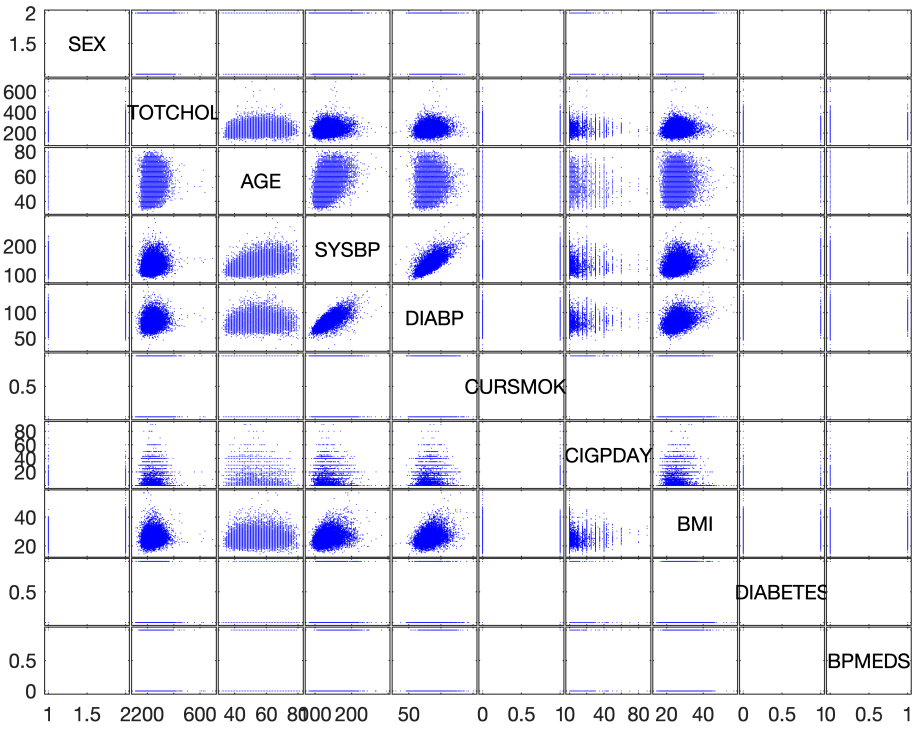
```
h = heatmap(r);
h.XDisplayLabels = var_names;
h.YDisplayLabels = var_names;
```



The heatmap makes it very easy to pick out the highest and lowest correlated features.

Another tool in MATLAB for visualizing raw data correlations is `gplotmatrix`. This command creates a matrix of scatter plots for each variable in the data set against every other variable.

```
gplotmatrix(fram_2_11.Variables,[],[],[],[],[],[],'variable',var_names);
```


SEX
TOTCHOL
AGE
SYSBP
DIABP
CURSMOK
CIGPDAY
BMI
DIABETES
BPMEDS