Hypothesis Testing

Why do we need it for an AI project?

Helps answer:

- Is the model better than random?
- Is the difference between variables significant? Is it real?
- Is the correlation strong enough?
- Are two datasets (e.g. training and test) similar?
- Is the training data representative of the whole population?

Null hypothesis and p-values

- Null hypothesis: there is no difference between two (or more) variables or groups
- *p*-value: the probability that the null hypothesis is true

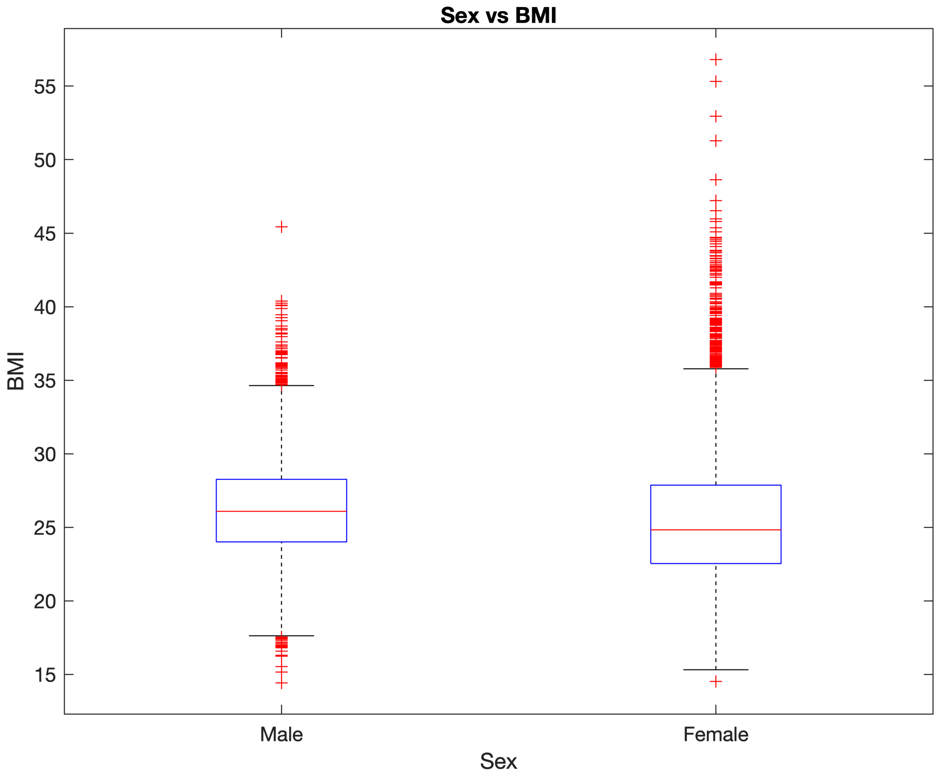
Comparing variables in the Framingham dataset

Begin by loading the Framingham dataset.

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

Create a boxplot to compare BMI betwen males and females.

```matlab
boxplot(fram.BMI, fram.SEX, "Labels", {'Male','Female'})
xlabel("Sex")
ylabel("BMI")
title("Sex vs BMI")
```



Is the difference in BMI between males and females significant statistically?

We can check the significance using a **t-test**. In MATLAB, this is done with the `ttest2` function. The t-test checks if the two distributions (in this case BMI of males and females) have equal means.

The first output, *h*, signifies whether the test rejects or fails to reject the null hypothesis. A value of 1 indicates that the null hypothesis is rejected at a significance level of 5%, and a value of 0 indicates that we fail to reject the null hypothesis.

The second output of `ttest2` will be the *p*-value*.*

```
[h, p_value] = ttest2(fram.BMI(fram.SEX==1), fram.BMI(fram.SEX==2))
```

```
h = 1
p_value = 7.4557e-14
```

Based on the value of *h,* we can reject the null hypothesis. What does the *p*-value say about the two distributions?

If the null hypothesis is true and there is no relationship between sex and BMI, the probability of getting results as extreme as ours is $0.74557 * 10^{-14}$.

T-test assumptions

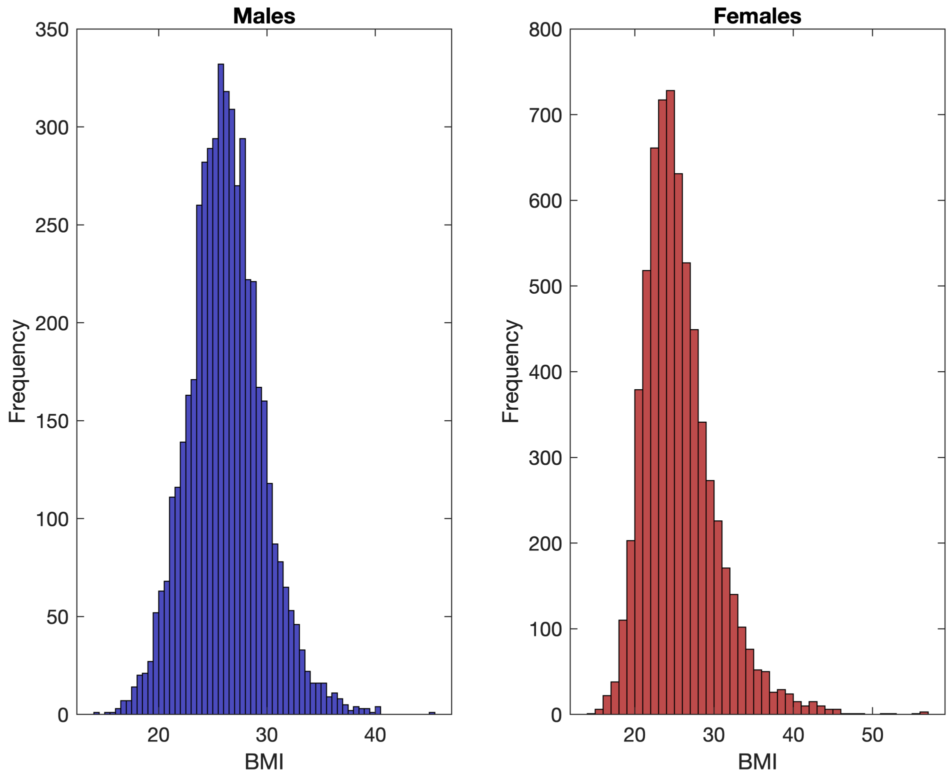
- T-test assumes that distributions are "normal".

- T-test compares the mean and not the median.

How do we know if the data is "normal"?

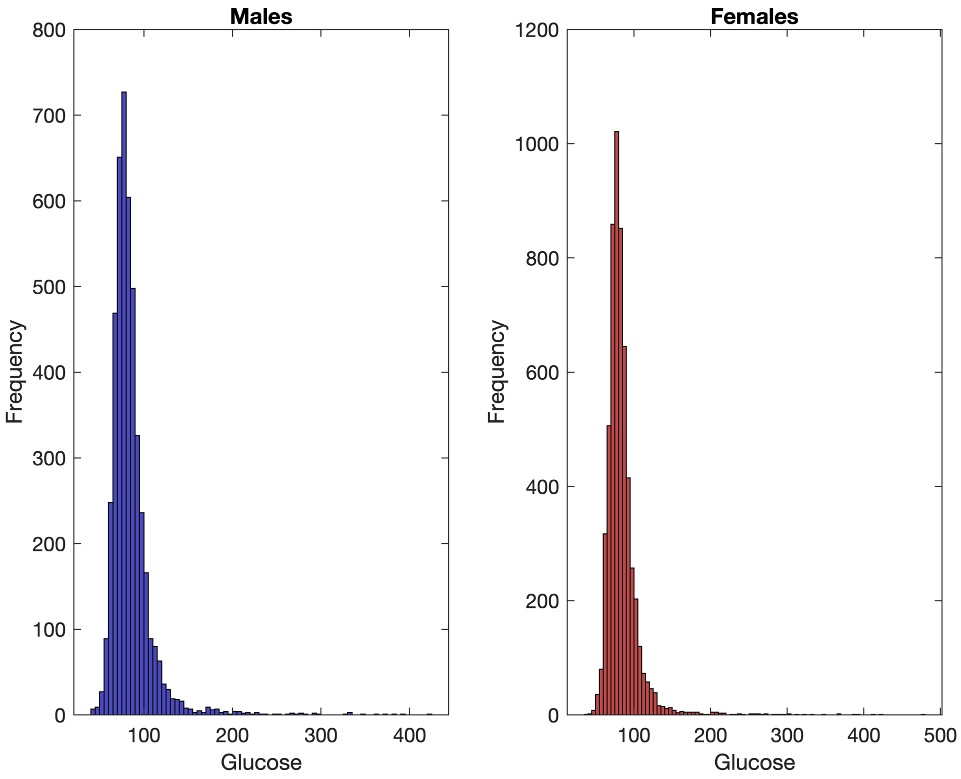
Plot it! Use `histogram` to plot BMI, cholesterol and glucose for males and females.

```
figure;
subplot(1,2,1)
histogram(fram.BMI(fram.SEX==1),"FaceColor","blue");
xlabel('BMI')
ylabel('Frequency')
title('Males')

subplot(1,2,2)
histogram(fram.BMI(fram.SEX==2),"FaceColor","red");
xlabel('BMI')
ylabel('Frequency')
title('Females')
```

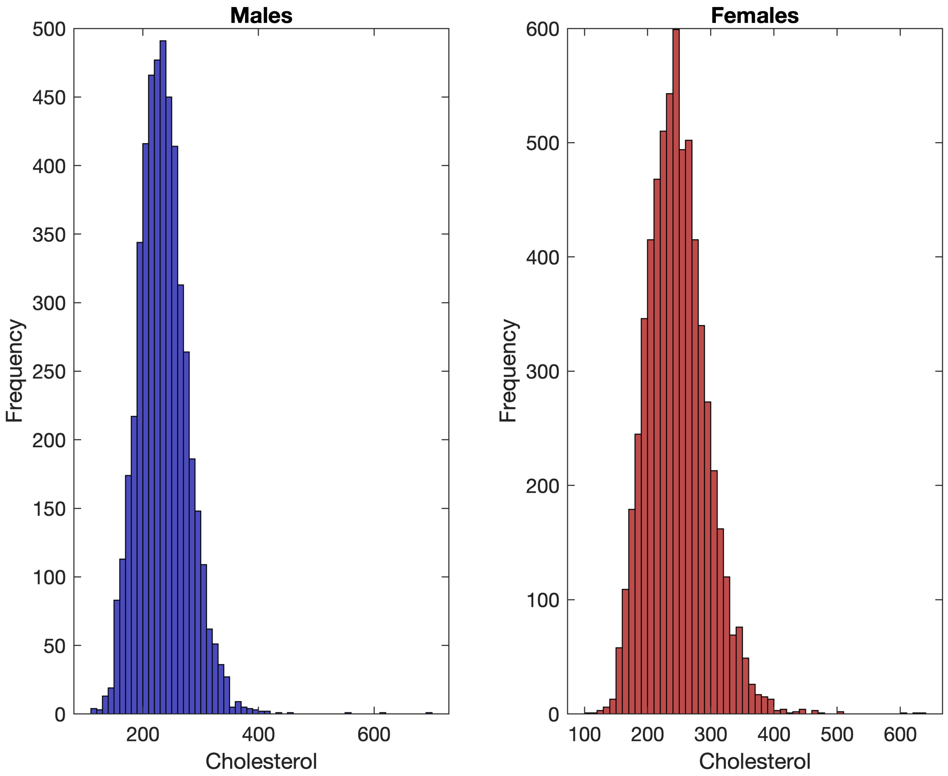


```
figure;
subplot(1,2,1)
histogram(fram.GLUCOSE(fram.SEX==1),"FaceColor","blue");
xlabel('Glucose')
ylabel('Frequency')
title('Males')
```

```
subplot(1,2,2)
histogram(fram.GLUCOSE(fram.SEX==2),"FaceColor","red");
xlabel('Glucose')
ylabel('Frequency')
title('Females')
```

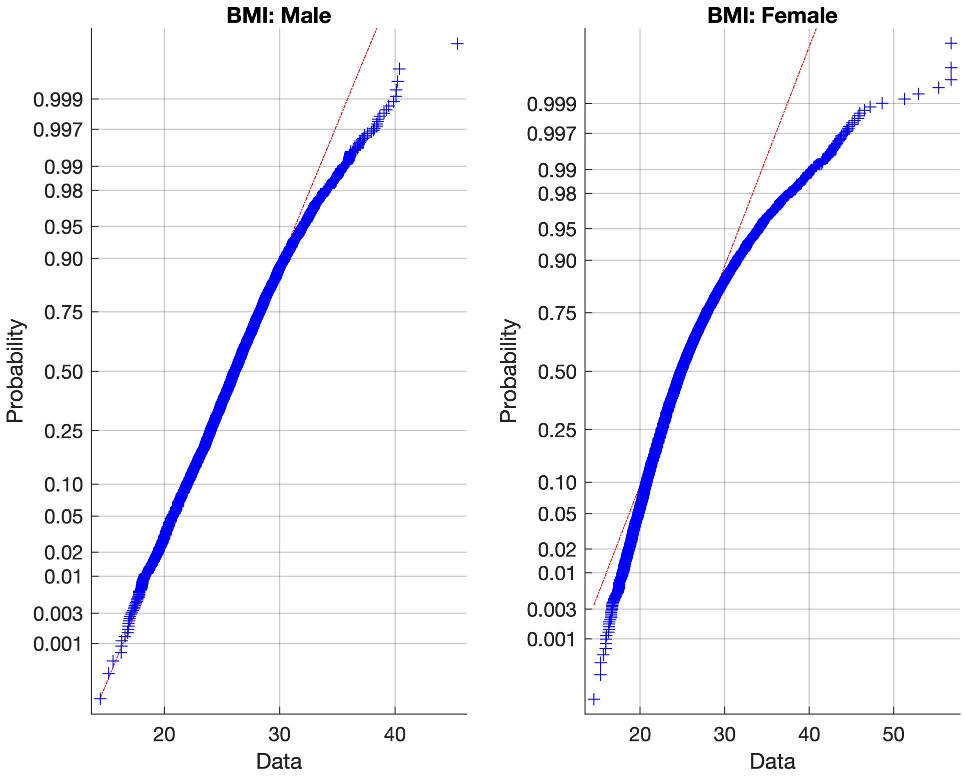


```
figure;
subplot(1,2,1)
histogram(fram.TOTCHOL(fram.SEX==1),"FaceColor","blue");
xlabel('Cholesterol')
ylabel('Frequency')
title('Males')

subplot(1,2,2)
histogram(fram.TOTCHOL(fram.SEX==2),"FaceColor","red");
xlabel('Cholesterol')
ylabel('Frequency')
title('Females')
```

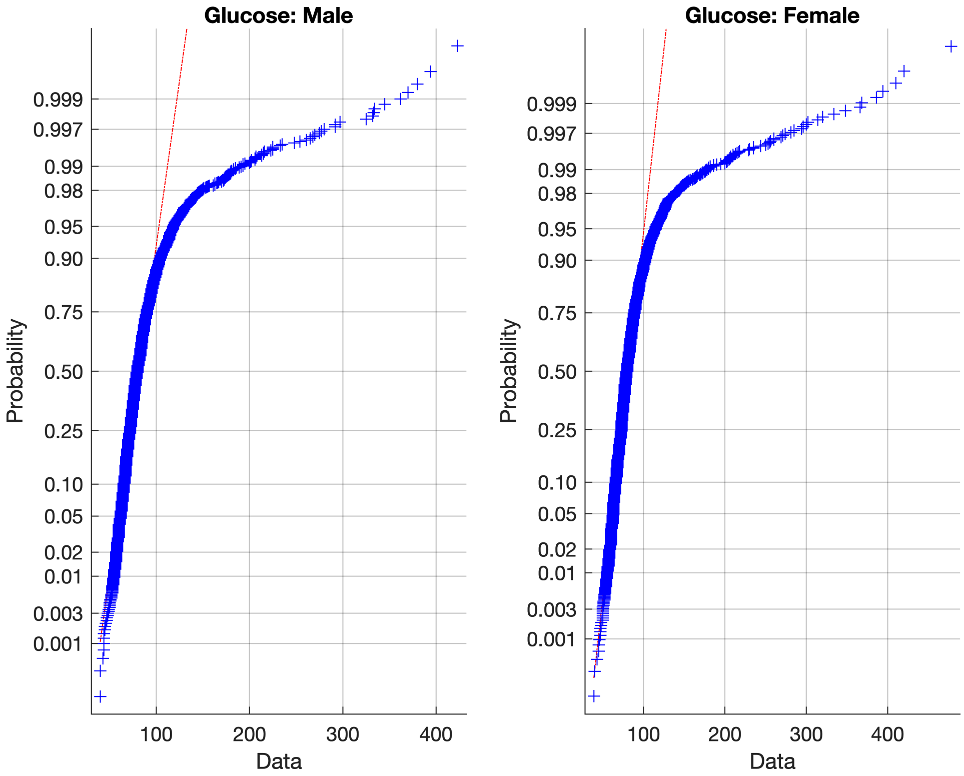


Are the data normally distributed? They look to be fairly normal, but the right tails in all of the plots look a bit longer than the lef tails. To get a better idea about the data normality, we can use `normplot`. This function compares the input distribution to an ideal normal distribution.

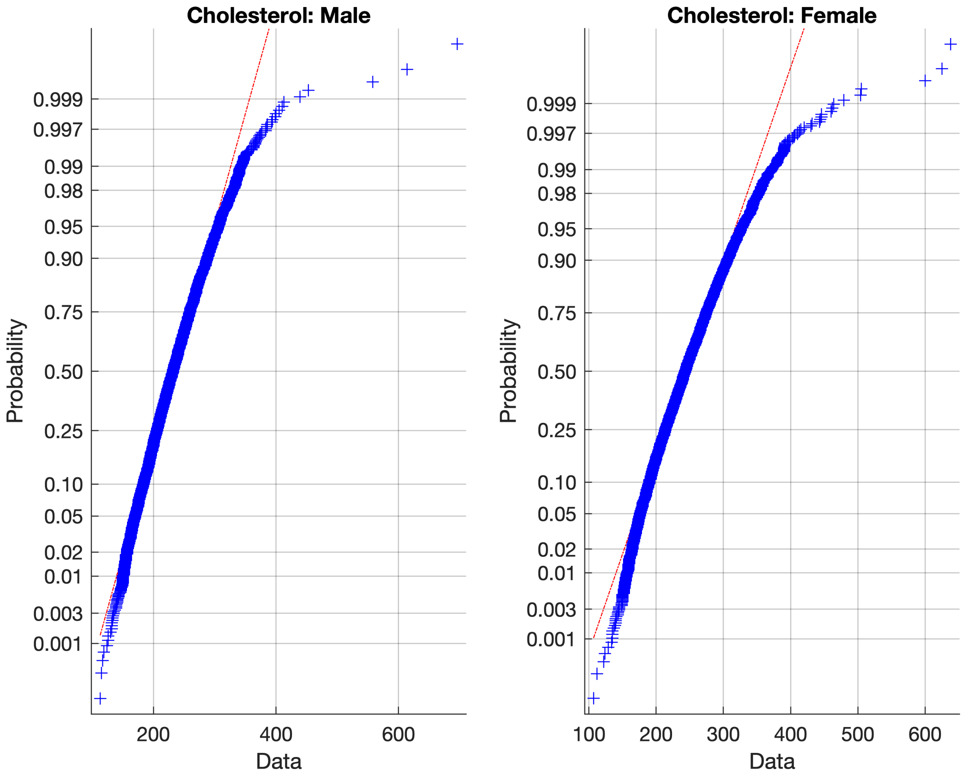
```
figure;
subplot(1,2,1)
normplot(fram.BMI(fram.SEX==1))
title('BMI: Male')

subplot(1,2,2)
normplot(fram.BMI(fram.SEX==2))
title('BMI: Female')
```



```
figure;
subplot(1,2,1)
normplot(fram.GLUCOSE(fram.SEX==1))
title('Glucose: Male')

subplot(1,2,2)
normplot(fram.GLUCOSE(fram.SEX==2))
title('Glucose: Female')
```



```
figure;
subplot(1,2,1)
normplot(fram.TOTCHOL(fram.SEX==1))
title('Cholesterol: Male')

subplot(1,2,2)
normplot(fram.TOTCHOL(fram.SEX==2))
title('Cholesterol: Female')
```



The normality tests confirm that the data is right-skewed in all six cases.

Rank sum test

The **rank sum test** can be used to compare medians for any distribution (not necessarily 'normal')

Compare BMI between males and females using the `ranksum` function. Note that the outputs are in the reverse order compared to the `ttest2` function.

```
[p_rank, h_rank] = ranksum(fram.BMI(fram.SEX==1), fram.BMI(fram.SEX==2))
```

```
p_rank = 1.3124e-44
h_rank = logical
   1
```

With the rank sum test, the null hypothesis is that the two samples are from distributions with equal medians. As with the `ttest2` function, an output of h = 1 means that the null hypothesis is rejected.

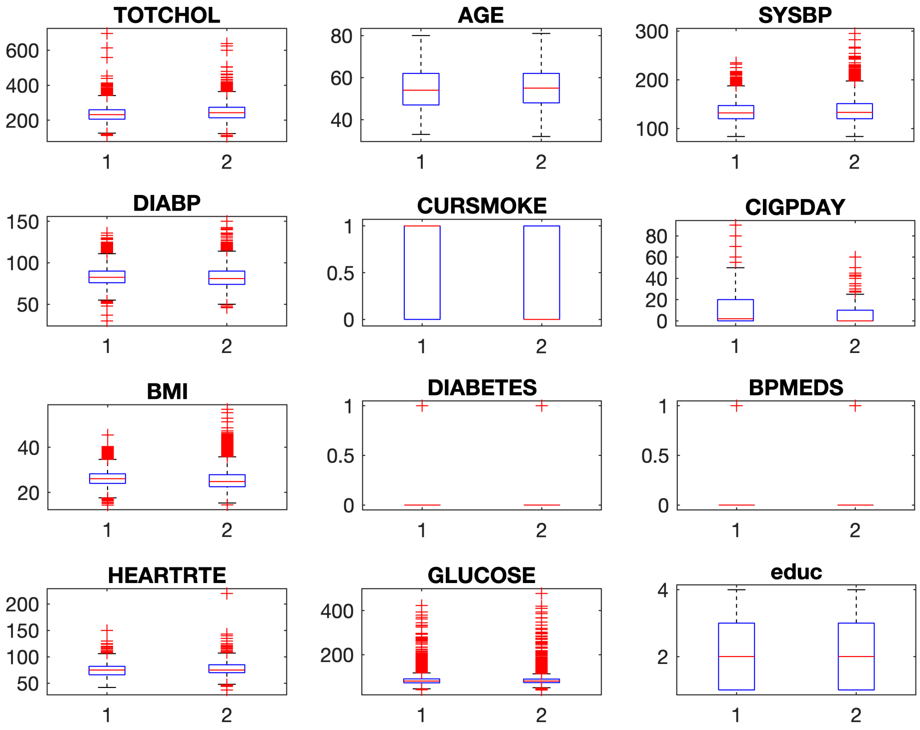
Are these results consistent with those from the t-test?

Interpreting *p*-values

P-values alone can be misleading; always use it in conjunction with a plot and other statistical metrics (e.g. fold change)

Plot the distribution between males and females for features 3 thru 14 using `boxplot` and `subplot`.

```
figure;
for i = 3:14
    subplot(4,3,i-2)
    boxplot(fram.(i),fram.SEX)
    title(fram.Properties.VariableNames(i))
end
```



By looking at the boxplots, can you guess if the *p*-values will be less than 0.05?

Calculate the *p*-values using both t-test and ranksum test.

```
j = 1;
for i = 3:14
    [h_ttest(j), p_ttest(j)] = ttest2(fram.(i)(fram.SEX==1), fram.(i)(fram.SEX==2));
    [p_rank(j), h_rank(j)] = ranksum(fram.(i)(fram.SEX==1), fram.(i)(fram.SEX==2));
    j = j + 1;
end
```

```
t = splitvars(table([h_ttest', p_ttest', h_rank', p_rank']));
t.Properties.VariableNames = {'h_ttest', 'p_ttest', 'h_rank', 'p_rank'};
t.Properties.RowNames = fram.Properties.VariableNames(3:14)
```

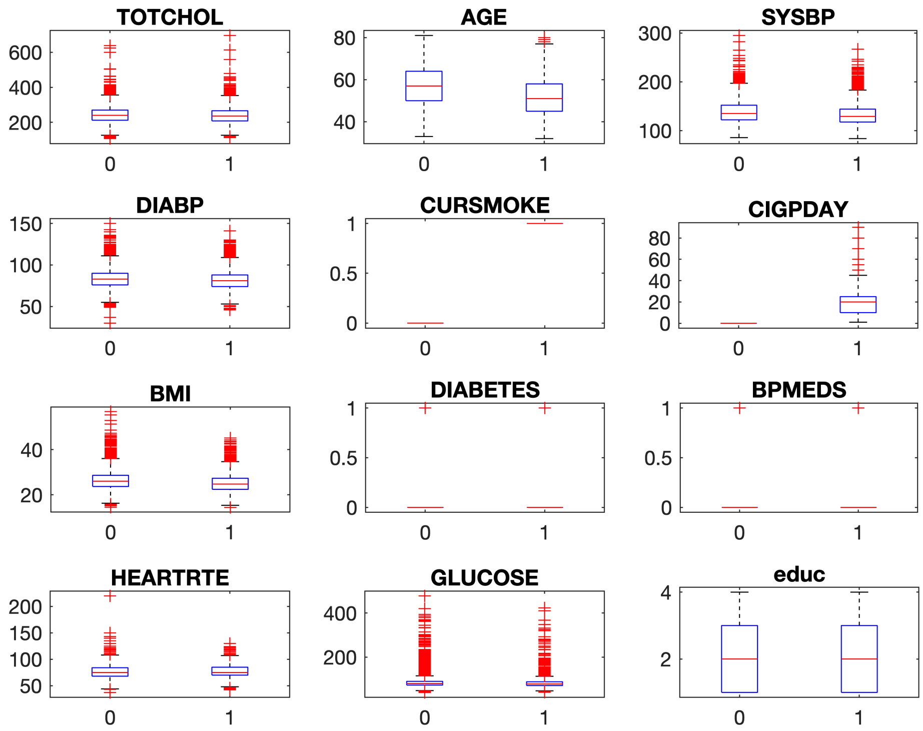
```
t = 12×4 table
```

	h_ttest	p_ttest	h_rank	p_rank
1 TOTCHOL	1	0.0000	1	0.0000
2 AGE	1	0.0033	1	0.0036
3 SYSBP	1	0.0000	1	0.0041
4 DIABP	1	0.0000	1	0.0000
5 CURSMOKE	1	0.0000	1	0.0000
6 CIGPDAY	1	0.0000	1	0.0000
7 BMI	1	0.0000	1	0.0000
8 DIABETES	1	0.0307	1	0.0307
9 BPMEDS	1	0.0000	1	0.0000
10 HEARTRTE	1	0.0000	1	0.0000
11 GLUCOSE	0	0.1041	0	0.4793
12 educ	0	0.1090	0	0.4608

According to both tests, only Glucose and Education do not meet the threshold *p*-value of 0.05. Therefore, we fail to reject the null hypothesis for those two variables.

Repeat the same process, this time comparing smokers and non-smokers.

```
j = 1;
figure;
for i = 3:14
    % fill subplot with boxplots
    subplot(4,3,j);
    boxplot(fram.(i),fram.CURSMOKE);
    title(fram.Properties.VariableNames(i))

    % store h and p-values
    [h_ttest(j), p_ttest(j)] = ttest2(fram.(i)(fram.CURSMOKE==0), fram.(i)(fram.CURSMOK
    [p_rank(j), h_rank(j)] = ranksum(fram.(i)(fram.CURSMOKE==0), fram.(i)(fram.CURSMOKE

    j = j + 1;
end
```



```
t2 = splitvars(table([h_ttest', p_ttest', h_rank', p_rank']));
t2.Properties.VariableNames = {'h_ttest', 'p_ttest', 'h_rank', 'p_rank'};
t2.Properties.RowNames = fram.Properties.VariableNames(3:14)
```

```
t2 = 12×4 table
```

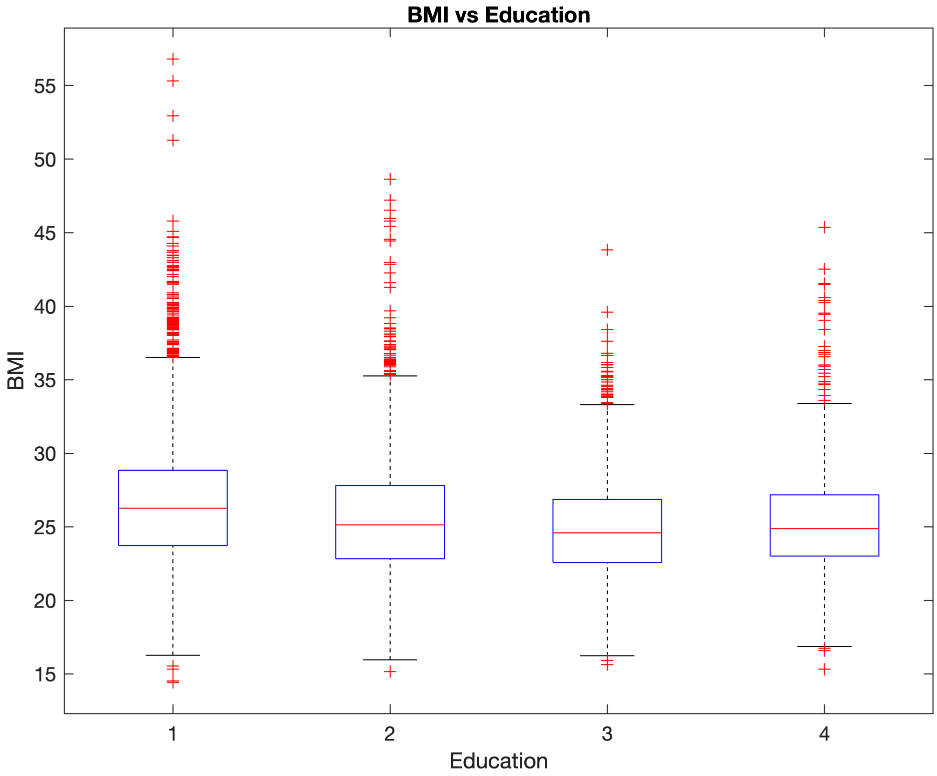
	h_ttest	p_ttest	h_rank	p_rank
1 TOTCHOL	1	0.0000	1	0.0000
2 AGE	1	0.0000	1	0.0000
3 SYSBP	1	0.0000	1	0.0000
4 DIABP	1	0.0000	1	0.0000
5 CURSMOKE	1	0	1	0
6 CIGPDAY	1	0	1	0
7 BMI	1	0.0000	1	0.0000
8 DIABETES	1	0.0000	1	0.0000
9 BPMEDS	1	0.0000	1	0.0000
10 HEARTRTE	1	0.0000	1	0.0000
11 GLUCOSE	1	0.0000	1	0.0000
12 educ	0	0.3852	0	0.0763

In this case, we reject the null hypothesis for all variables except Education.

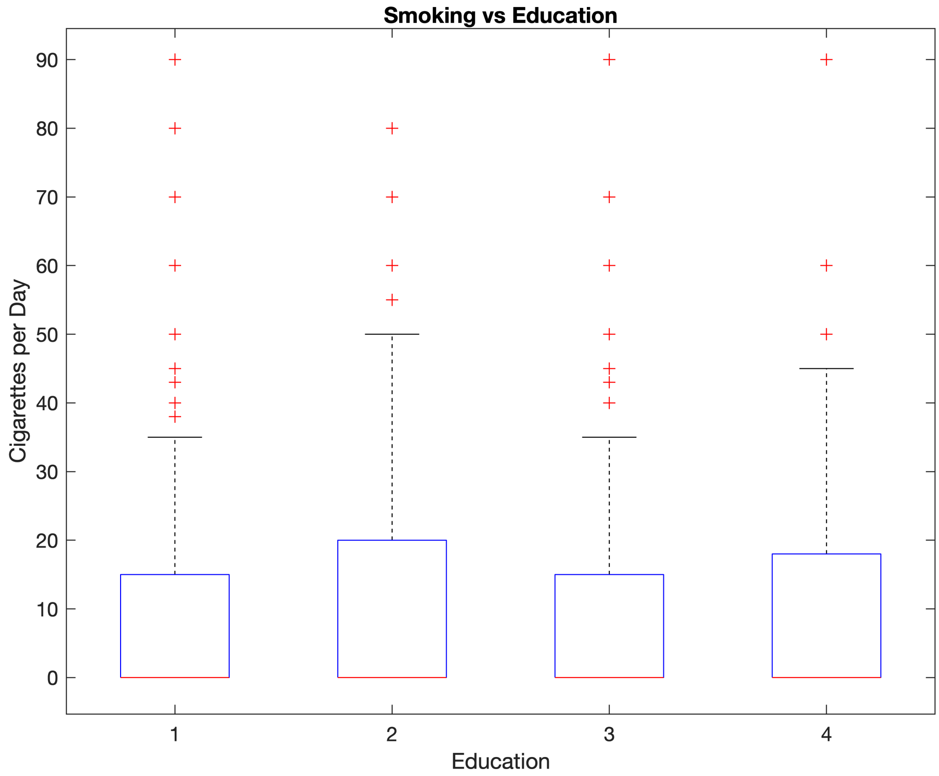
Comparing multiple distributions using Anova

Compare BMI, smoking and diabetes based on education level using boxplots.

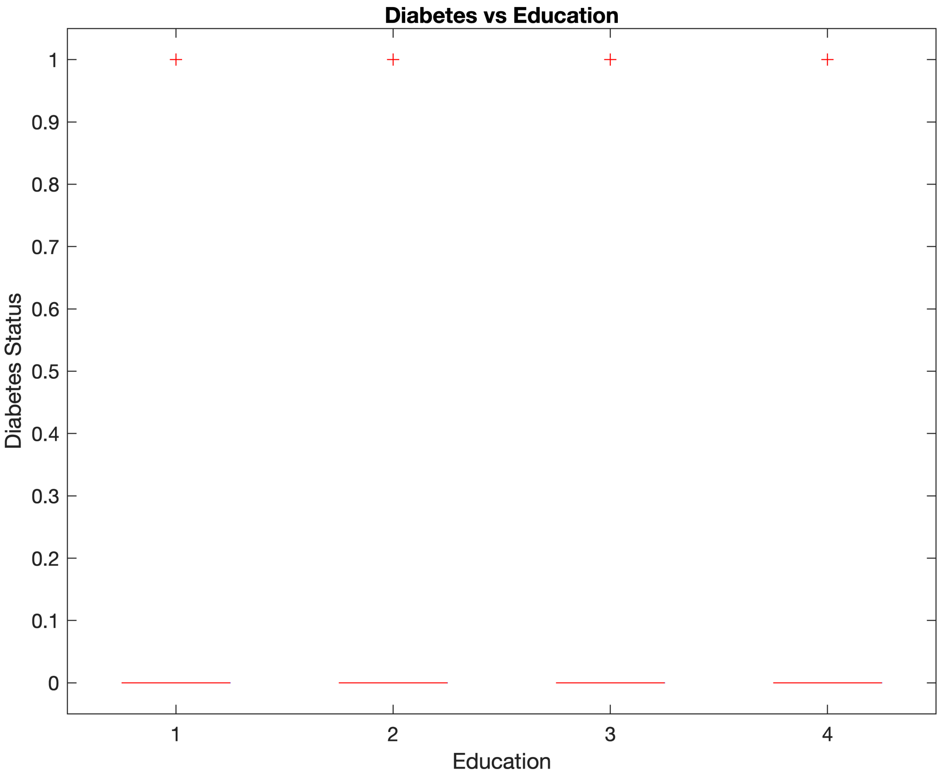
```
figure;
boxplot(fram.BMI, fram.educ)
title('BMI vs Education')
xlabel('Education')
ylabel('BMI')
```



```
figure;
boxplot(fram.CIGPDAY, fram.educ)
title('Smoking vs Education')
xlabel('Education')
ylabel('Cigarettes per Day')
```



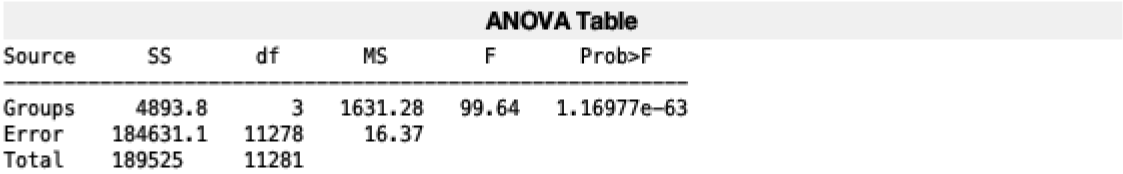
```
figure;
boxplot(fram.DIABETES, fram.educ)
title('Diabetes vs Education')
xlabel('Education')
ylabel('Diabetes Status')
```



Are the differences between education levels significant for these three variables? We can use Anova to calculate the *p*-value.

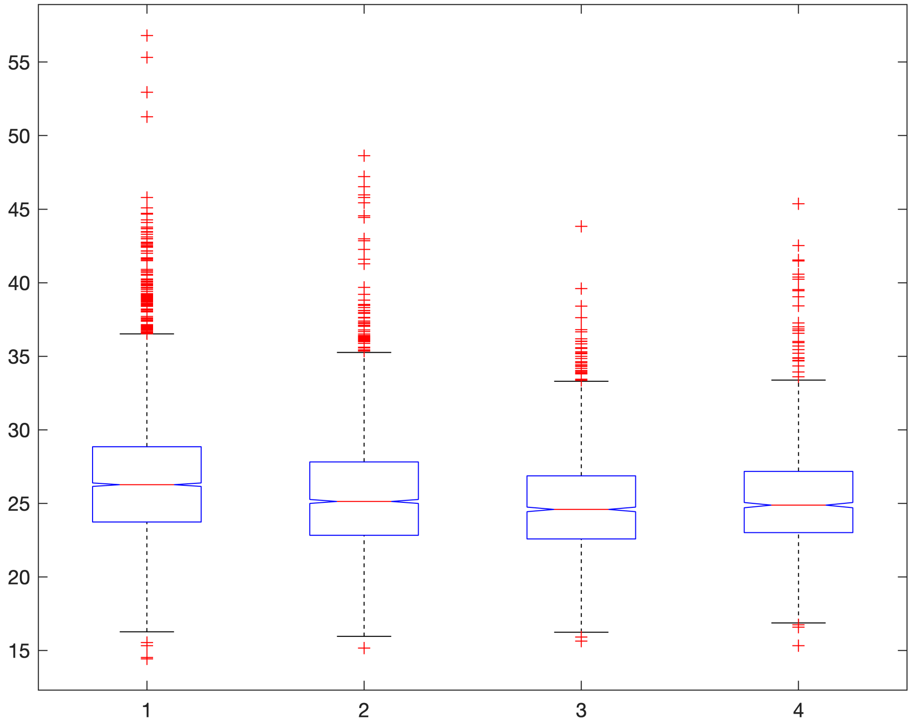
Anova extends the t-test to multiple comparisons (also assumes the distributions are 'normal' and it compares the mean).

```
pvalue = anova1(fram.BMI, fram.educ)
```



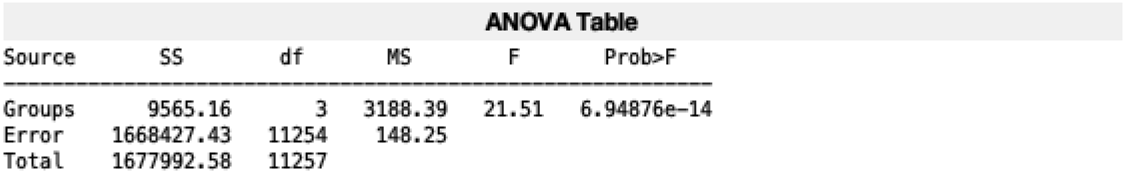
ANOVA Table

```
Source        SS        df        MS        F        Prob>F
-----------------------------------------------------------
Groups      4893.8        3    1631.28   99.64    1.16977e-63
Error     184631.1    11278      16.37
Total     189525      11281
```



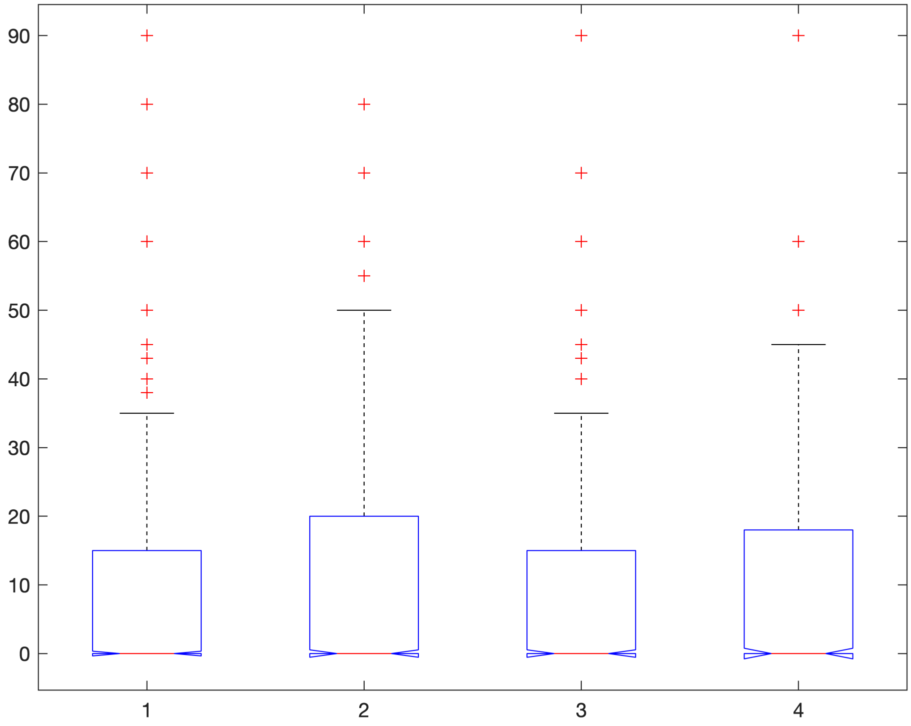
```
pvalue = 1.1698e-63
```

```
pvalue = anova1(fram.CIGPDAY, fram.educ)
```



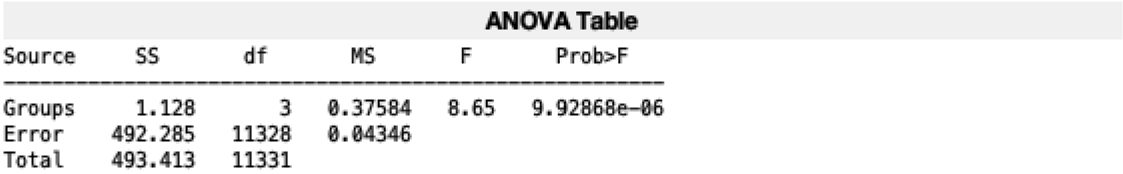
ANOVA Table

```
Source        SS         df       MS        F      Prob>F
-------------------------------------------------------------
Groups       9565.16       3   3188.39   21.51   6.94876e-14
Error     1668427.43   11254    148.25
Total     1677992.58   11257
```

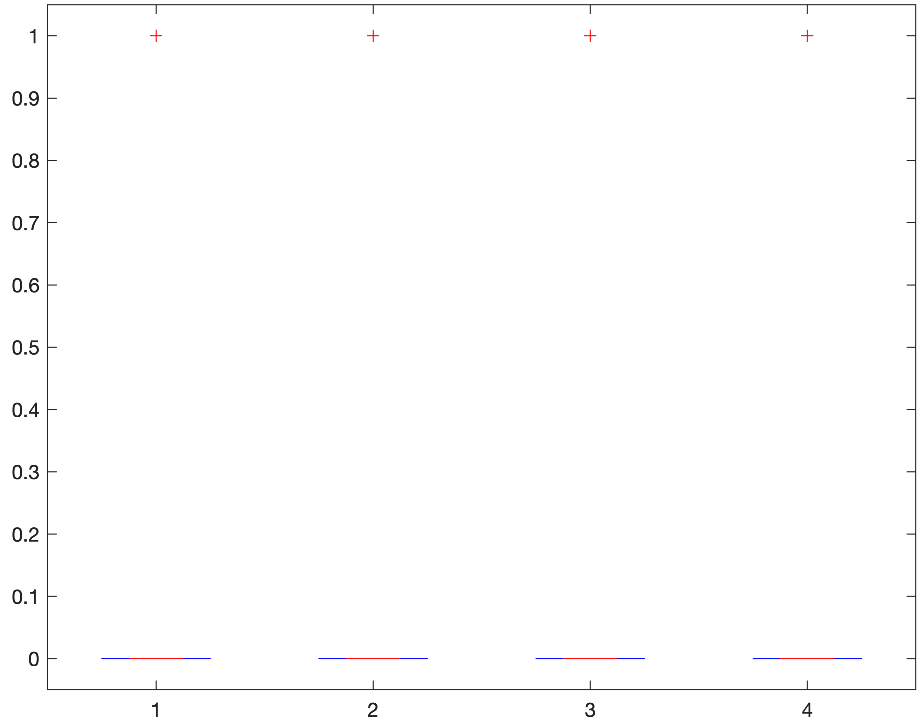


```
pvalue = 6.9488e-14
```

```
pvalue = anova1(fram.DIABETES, fram.educ)
```



```
                              ANOVA Table
Source       SS       df       MS        F       Prob>F
---------------------------------------------------------
Groups      1.128        3   0.37584   8.65   9.92868e-06
Error     492.285    11328   0.04346
Total     493.413    11331
```

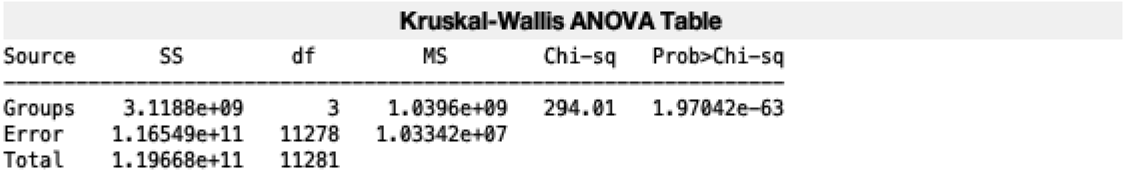


```
pvalue = 9.9287e-06
```

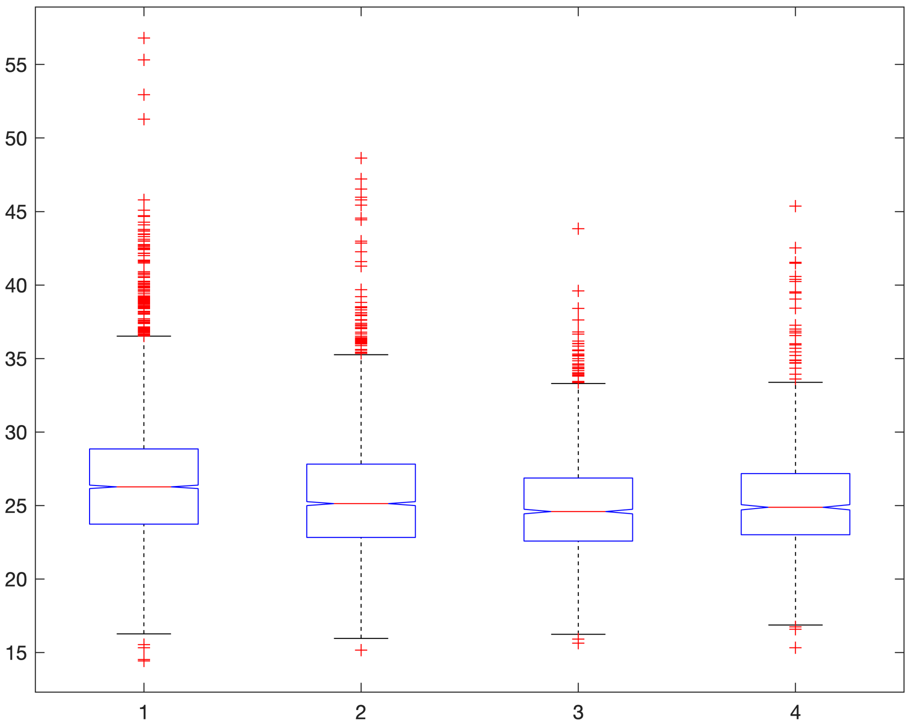
Based on the p-values, we can infer that the means of each educational group are different in regards to BMI, cigarettes per day and diabetes.

The **Kruskal-Wallis test** is the equivalent of the rank-sum test for multiple comparisons.

```
pvalue = kruskalwallis(fram.BMI, fram.educ)
```

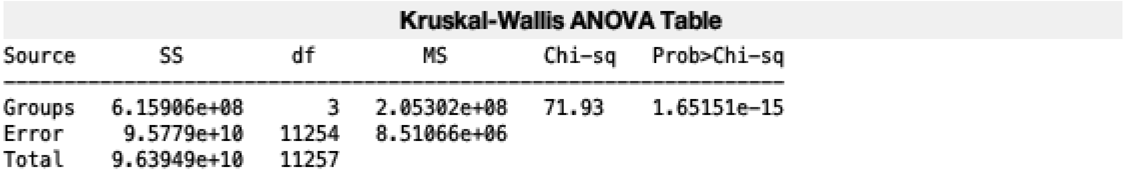


```
                      Kruskal-Wallis ANOVA Table
Source       SS          df          MS         Chi-sq   Prob>Chi-sq
---------------------------------------------------------------------
Groups    3.1188e+09        3    1.0396e+09    294.01    1.97042e-63
Error    1.16549e+11    11278  1.03342e+07
Total    1.19668e+11    11281
```

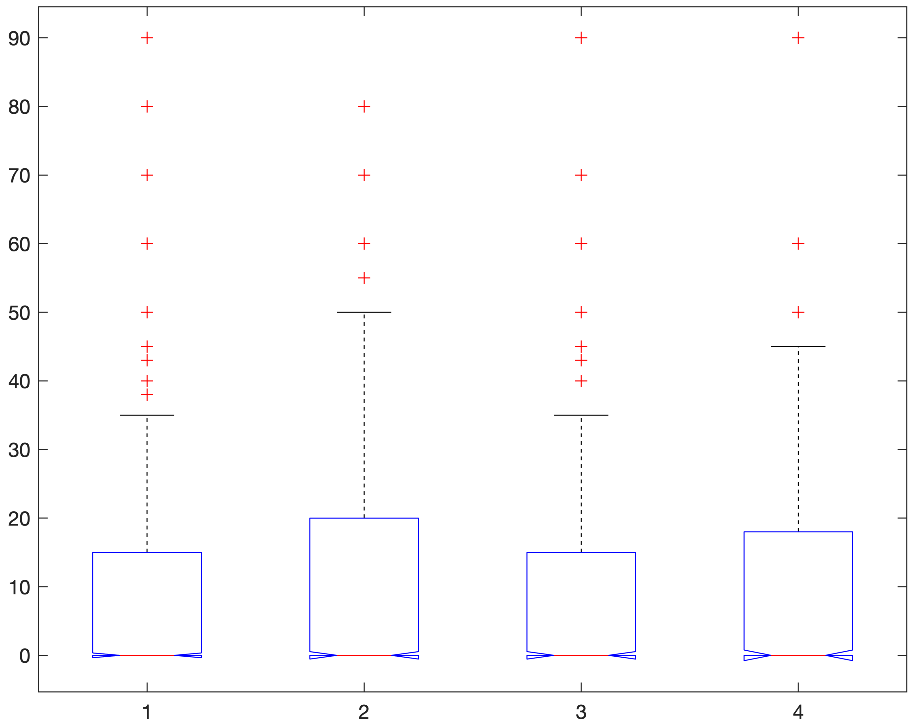


```
pvalue = 1.9704e-63
```

```
pvalue = kruskalwallis(fram.CIGPDAY, fram.educ)
```

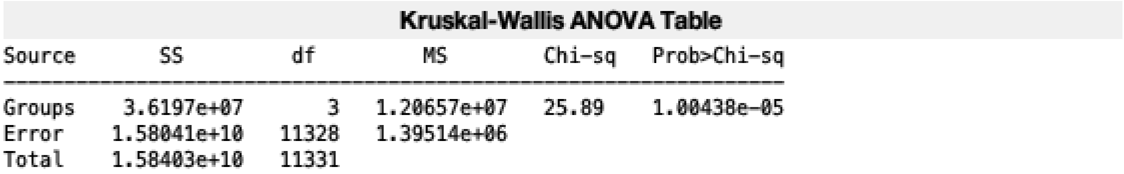


```
                      Kruskal-Wallis ANOVA Table
Source        SS         df        MS          Chi-sq    Prob>Chi-sq
-------------------------------------------------------------------
Groups   6.15906e+08      3    2.05302e+08    71.93     1.65151e-15
Error     9.5779e+10  11254    8.51066e+06
Total    9.63949e+10  11257
```

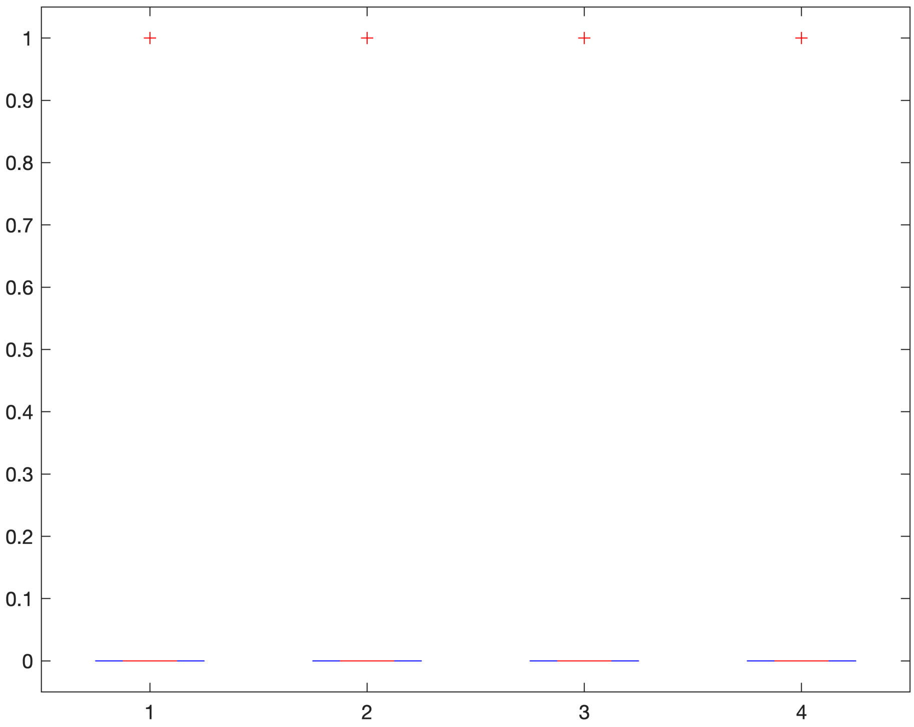


```
pvalue = 1.6515e-15
```

```
pvalue = kruskalwallis(fram.DIABETES, fram.educ)
```



```
                         Kruskal-Wallis ANOVA Table
Source        SS          df        MS          Chi-sq    Prob>Chi-sq
-------------------------------------------------------------------
Groups     3.6197e+07       3    1.20657e+07    25.89     1.00438e-05
Error     1.58041e+10   11328    1.39514e+06
Total     1.58403e+10   11331
```



```
pvalue = 1.0044e-05
```

Again, the very low p-values indicate that the mean values across the education levels are different and we can reject the null hypothesis for the three variables.

Multiple hypothesis testing

- When analyzing big data sets, we usually (knowingly or unknowingly) test for multiple hypotheses.
- Every time we test a hypothesis, there is a small but finite probability that the results may be significant due to random chance. This is quantified by the p-value.
- If we test multiple hypotheses, then it is likely that some of the hypotheses with significant *p*-values may be false positives.
- Multiple hypothesis correction helps to avoid such false discoveries.
- This is a signficant problem in big data analysis.

Hypothesis: Males and females have different BMI

- A *p*-value of 0.05 implies that there is only a 5% chance that the difference between two variables is due to random coincidence.
- **Multiple hypothesis scenario 1**: If the Framingham experiment was repeated 100 times, in 5 trials you may find significant difference.
- **Multiple hypothesis scenario 2**: Males and females have different properties.
- Some comparisons may have low *p*-values out of random chance.

Multiple hypothesis correction

- **Bonferroni Correction** - divide the *p*-values by the number of trials.
- For example, for a single hypothesis, a *p*-value cutoff of 0.05 is usually used. If you test 100 hypotheses, then the equivalent p-value theshold for significance should be 0.05 / 100 = 0.0005 to correct for multiple hypotheses using the Bonferroni approach.
- The Bonferroni approach is very conservative.

Multiple hypothesis correction using FDR

- **Benjamin Hochberg correction** - False discovery rate (FDR) is designed to control the proportion of false positives among the set of rejected hypotheses.
- This approach provides a *p*-value for the *p*-value (called the q-value). It tells you how many false discoveries (false positives) will ovver with a *p*-value threshold.
- A q-value of 0.05 for a given *p*-value means that 5% of hypotheses with the same *p*-value are likely false positives.

Correcting for multiple hypotheses in transcriptomics data

- **Transcriptomics** - the activity of close to 30,000 gene transcripts are measured
- Scientists routinely compare transcriptome of normal and cancer patients to find biomarkers for cancer diagnosis.
- With a *p*-value of 0.05, we would expect to see 30,000 x 0.05 = 1,500 genes to be statistically significant by random chance!

Testing if a gene is differentially active in normal and cancer patients

Start by loading the cancer transcriptomics data.

```
data = readtable("../cancer_RNAseq_data.xlsx")
```

```
Warning: Table variable names were modified to make them valid MATLAB identifiers. The original
 names are saved in the VariableDescriptions property.
data = 100×1001 table
```

...

	Patient	MUC6	SPAG17	CCL20	ADCY2	TP63	SOX2	C20orf151
1	'Patient-1'	1.2829	1.0530	3.7592	9.9001	4.5369	3.4636	6.1759
2	'Patient-2'	4.7240	4.0634	9.2203	1.7275	6.2634	12.8483	5.9222
3	'Patient-3'	0.9613	1.5331	4.1353	7.7829	9.5298	7.0667	7.6014
4	'Patient-4'	1.5561	0.9784	3.6230	8.4578	9.9743	6.6894	7.4379
5	'Patient-5'	1.6375	0	9.3092	0.7687	1.2675	0.7687	9.6459
6	'Patient-6'	12.7285	0	8.6408	2.9676	1.5604	1.9723	8.4408

	Patient	MUC6	SPAG17	CCL20	ADCY2	TP63	SOX2	C20orf151
7	'Patient-7'	1.4742	6.6573	4.1756	3.4539	4.9649	6.1654	7.3249
8	'Patient-8'	0.3963	2.9330	6.4371	2.4396	5.1605	8.1635	6.3050
9	'Patient-9'	11.7109	2.6544	1.4674	9.4088	8.0004	6.3611	7.9703
10	'Patient-10'	6.0515	2.1210	6.7576	6.1110	2.2953	6.7798	6.1569
11	'Patient-11'	2.8874	8.2162	7.1199	8.7787	3.3828	0	0
12	'Patient-12'	4.6684	5.4078	6.5474	2.4440	3.9481	0.4543	0
13	'Patient-13'	9.2933	6.8461	11.2201	1.4076	6.9462	10.3884	6.6224
14	'Patient-14'	1.1686	1.9679	0.8734	2.2461	0.5018	1.1686	9.2973
15	'Patient-15'	2.4138	1.0581	4.4543	7.3706	8.8585	5.2401	7.6158
16	'Patient-16'	5.3194	2.0517	5.0205	7.9215	7.6634	5.3051	7.3029
17	'Patient-17'	0	6.8037	9.9297	8.1864	3.7773	0.8640	0
18	'Patient-18'	0.7866	3.9541	3.5698	4.1729	3.5251	0	4.6183
19	'Patient-19'	12.8532	2.7065	1.9811	0	0.4526	1.5072	7.0939
20	'Patient-20'	8.1701	0.7904	9.5091	5.1260	2.2164	0.7904	7.9259
21	'Patient-21'	2.7061	0.6906	4.7761	6.4422	8.3259	7.1012	7.9954
22	'Patient-22'	4.1323	0.5213	9.1728	4.2021	1.4545	1.4545	8.7557
23	'Patient-23'	11.1106	0.6678	2.1800	8.3561	9.0146	6.5636	6.8985
24	'Patient-24'	2.3696	0.7148	5.5792	5.7263	4.9306	0	0
25	'Patient-25'	1.2280	0.5336	0.9222	8.6653	4.8675	3.1056	7.0309
26	'Patient-26'	7.5730	0	3.5342	8.5839	9.5001	5.5100	8.0532
27	'Patient-27'	0	7.5454	6.1315	0.2837	2.5846	0	0
28	'Patient-28'	0	5.1087	1.0947	3.1619	5.8095	6.2033	7.7180
29	'Patient-29'	2.2521	0	6.3451	5.7125	3.6914	4.2624	2.0484
30	'Patient-30'	1.9583	1.2887	9.6166	4.6742	2.9512	1.2887	8.2594
31	'Patient-31'	2.2279	3.9761	7.0286	4.0845	3.0650	9.3594	7.4494
32	'Patient-32'	5.4383	6.9964	1.1442	2.2113	7.0009	2.6426	7.5025
33	'Patient-33'	1.8956	7.1491	8.3672	0.6268	1.3964	0	0.6268
34	'Patient-34'	3.4609	4.9031	9.1420	2.1167	7.9078	4.7325	5.7355
35	'Patient-35'	2.9222	0.7917	9.0226	0.7917	3.0551	0	7.2512
36	'Patient-36'	2.9357	1.2632	3.8025	4.8408	6.2596	1.4596	8.3263
37	'Patient-37'	2.2755	0.8904	7.8199	5.0999	1.6481	2.5098	8.3077
38	'Patient-38'	5.1751	0	2.2123	10.0487	9.1294	5.5741	7.8528
39	'Patient-39'	0.9197	7.3433	2.9425	3.1008	1.8777	1.4768	2.8563
40	'Patient-40'	4.7372	6.4962	3.2946	3.1148	5.9253	0	6.2570

	Patient	MUC6	SPAG17	CCL20	ADCY2	TP63	SOX2	C20orf151
41	'Patient-41'	1.3497	3.3634	9.4808	3.4167	2.4972	1.3497	0
42	'Patient-42'	9.5300	8.4137	6.7804	1.8669	4.7398	6.6882	7.4635
43	'Patient-43'	2.7378	0.4906	10.2723	3.7608	1.1474	0.8560	8.9866
44	'Patient-44'	2.1793	0.4774	8.6045	7.5277	10.1441	6.9421	7.4682
45	'Patient-45'	1.9164	1.9164	9.2534	1.6870	3.0381	7.7384	8.7237
46	'Patient-46'	11.8891	2.6021	4.0192	9.1066	8.5223	6.1182	8.2468
47	'Patient-47'	2.0020	8.9853	0.8943	9.0167	3.8816	4.7877	7.8686
48	'Patient-48'	5.5824	8.3253	1.5631	9.7829	3.3127	0.6402	0
49	'Patient-49'	1.9143	7.9151	7.7689	2.9347	7.5519	2.5813	6.6458
50	'Patient-50'	3.1469	6.8772	5.2519	7.5995	6.1990	5.2353	8.1491
51	'Patient-51'	1.2160	1.8663	8.5266	4.7440	1.5775	1.5775	9.0121
52	'Patient-52'	1.6888	3.4603	8.3324	2.8309	5.5926	1.6888	9.7675
53	'Patient-53'	3.1146	1.9703	7.3126	4.9503	4.6273	4.8625	6.9061
54	'Patient-54'	0	3.1285	3.8075	5.8507	9.5939	0	5.9379
55	'Patient-55'	6.5351	8.8662	8.8811	6.5681	13.0514	10.4728	4.9809
56	'Patient-56'	1.7915	0	6.8360	4.7335	3.6800	1.7915	6.9553
57	'Patient-57'	0.9382	2.7981	10.8312	3.7894	2.6998	2.4805	7.6589
58	'Patient-58'	4.7371	2.9364	5.3717	9.2091	3.7902	0.5610	0
59	'Patient-59'	4.6332	5.6194	7.3031	5.3278	6.3289	6.1197	6.5929
60	'Patient-60'	2.8449	8.5413	5.8719	0.9646	3.8400	0	0
61	'Patient-61'	1.4948	1.0642	4.1773	7.4378	1.0642	7.1020	7.5568
62	'Patient-62'	14.1608	2.8525	4.2706	1.9739	5.3668	0.7925	0
63	'Patient-63'	1.5923	2.4871	6.1194	9.0713	3.2249	0.8982	6.4866
64	'Patient-64'	11.8038	4.5241	6.2342	4.0720	7.2830	8.3950	5.2532
65	'Patient-65'	7.1800	8.4264	2.5655	5.3506	8.7508	0.5333	7.1023
66	'Patient-66'	2.0909	1.2041	0	8.1952	4.5741	2.4761	7.6381
67	'Patient-67'	0.5705	1.8748	2.4874	6.7925	7.3139	3.8417	6.2558
68	'Patient-68'	3.4963	5.3371	8.1935	0.4252	2.5359	0.7533	0
69	'Patient-69'	3.7244	7.3790	10.8803	2.1669	4.3111	5.8727	3.9691
70	'Patient-70'	0	4.7184	5.2184	7.5090	6.7865	5.7199	6.9266
71	'Patient-71'	8.4787	1.4402	5.5522	0.7531	10.1159	9.3467	6.7538
72	'Patient-72'	0.3665	7.4732	0.3665	7.0188	1.9604	0.9011	6.0955
73	'Patient-73'	0.6455	1.0900	9.0645	3.4051	2.9583	2.1329	7.2178
74	'Patient-74'	1.7345	5.2275	7.8997	6.5798	3.7033	0.8286	0

	Patient	MUC6	SPAG17	CCL20	ADCY2	TP63	SOX2	C20orf151
75	'Patient-75'	9.0588	2.5951	4.0394	6.4012	3.6623	1.5316	7.3431
76	'Patient-76'	11.0687	7.2370	0	6.8023	8.3459	9.2499	5.2471
77	'Patient-77'	4.3202	0	2.4050	7.2179	7.0527	0.4415	7.6469
78	'Patient-78'	5.4795	3.3866	3.3123	8.1021	8.0285	1.3654	7.8027
79	'Patient-79'	4.5850	8.4217	5.9244	6.7970	5.3633	1.0214	0
80	'Patient-80'	1.9824	0.8823	9.1849	3.7254	0.5075	0.5075	8.9480
81	'Patient-81'	1.0131	7.3114	11.6219	8.4184	5.3475	0	0
82	'Patient-82'	3.3759	2.3161	1.5801	6.5969	9.7286	3.9001	7.8229
83	'Patient-83'	4.5717	0	4.3976	10.1679	10.5701	6.5613	7.2802
84	'Patient-84'	13.8436	0.8528	11.5261	1.9339	3.7541	0	6.1765
85	'Patient-85'	4.6208	10.5204	4.2082	7.2251	10.0706	9.9329	6.1661
86	'Patient-86'	3.9334	0.6315	1.9053	7.8996	5.9532	4.7740	8.0890
87	'Patient-87'	1.0876	0	11.9928	1.7006	3.9398	6.1620	7.4660
88	'Patient-88'	1.5694	0.4788	11.2629	3.1492	3.0152	1.3642	8.8938
89	'Patient-89'	1.0044	0.8111	4.5002	0.5878	4.3303	3.5316	6.8944
90	'Patient-90'	2.4142	2.8065	7.6620	2.1147	2.5842	1.8738	7.4253
91	'Patient-91'	10.1921	5.7562	12.0531	3.5373	4.9154	11.0234	7.3548
92	'Patient-92'	10.4596	0.7686	2.3845	7.8121	7.0429	7.5340	6.6769
93	'Patient-93'	7.2817	0.7294	10.8264	5.0112	1.7236	3.6454	9.0964
94	'Patient-94'	3.3252	5.8801	9.5102	5.4671	3.4295	0	0.4604
95	'Patient-95'	2.9032	7.8982	2.9869	0	8.2385	1.5904	5.1187
96	'Patient-96'	1.1907	7.4929	9.3443	3.8332	3.2556	1.1907	0
97	'Patient-97'	4.4430	8.3053	3.0072	3.8042	2.6294	1.8455	3.2513
98	'Patient-98'	10.2098	6.8068	0.8928	8.1065	9.7567	6.0563	7.7438
99	'Patient-99'	0	4.9080	7.4216	1.0394	10.1187	3.2393	5.2049
100	'Patient-100'	3.5983	7.5620	13.5027	1.9175	2.4930	10.0080	7.9555

- This is a small subset; it contains only 1,000 genes and 100 patients.
- Let's assume that the first 50 patients are cancer type A and the next 50 are normal.
- Hypothesis: Gene 1 (MUC6) is higher in cancer vs. normal.
- Null hypothesis: Gene 1 is not different in both.
- This can be tested using a t-test (calculate *p*-value).

```
[h, p_value] = ttest2(data.MUC6(1:50), data.MUC6(51:100))
```

```
h = 0
p_value = 0.6871
```

Finding all genes differentilly active in normal and cancer patients

- Multiple hypothesis: any gene that is different between cancer and normal.
- This can also be calculated using a t-test, but *p*-values need to be corrected for multiple hypothesis using the Benjamin-Hochberg procedure.

Start by finding the *p*-values for all 1,000 genes.

```
for i = 2:1001
    [h(i), p_value(i)] = ttest2(data.(i)(1:50), data.(i)(51:100));
end
```

The MATLAB function `mafdr` finds the FDR for multiple hypothesis testing.

```
FDR = mafdr(p_value);
```

Next use Bonferroni's method (multiply *p*-values by 1,000) and repeat the FDR calculation.

```
p_corrected = p_value*1000;
```

How many genes are statistically different at both FDR < 0.05 and *p*-value < 0.05 using both methods? Compare the number of signficant genes from the corrected and uncorrected lists.

```
sum(FDR < 0.05)
```

```
ans = 0
```

```
sum(p_corrected < 0.05)
```

```
ans = 0
```

```
sum(p_value < 0.05)
```

```
ans = 77
```

Multiple hypothesis correction with correlations and *p*-values

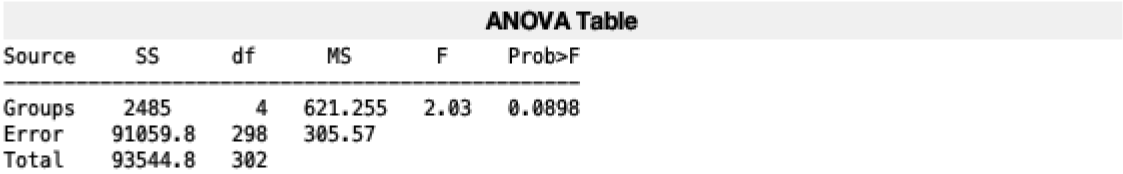
We will investigate several hypotheses regarding the correlation between dataset features and heart disease.

Load the Cleveland heart dataset.

```
data2 = readtable('../cleveland_data_revised.xlsx');
```

Hypothesis - blood pressure is correlated with severity.

```
p_value = anova1(data2.restingBP, data2.diseaseSeverity)
```



ANOVA Table

```
Source     SS       df       MS        F      Prob>F
--------------------------------------------------
Groups    2485       4    621.255   2.03   0.0898
Error    91059.8   298    305.57
Total    93544.8   302
```

```
p_value = 0.0898
```

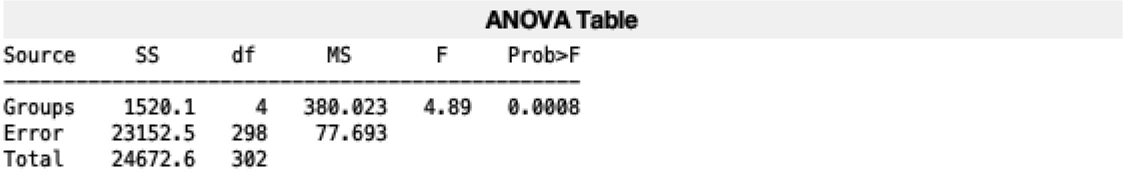
The anova test indicates that there is not a strong correlation between blood pressure and disease severity.

Multiple hypothesis - any feature is correlated with severity

```
p = zeros(13,1)
```

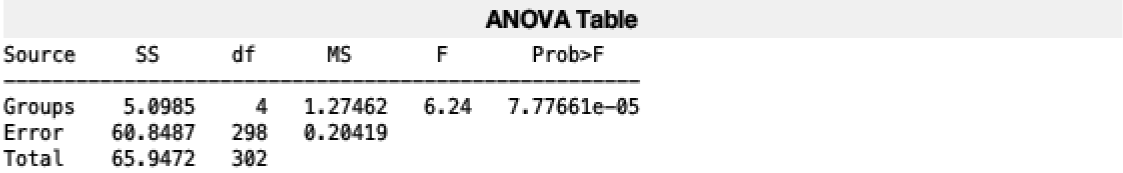
```
p = 13×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

```
for i = 1:13
    p(i) = anova1(data2.(i), data2.diseaseSeverity);
end
```



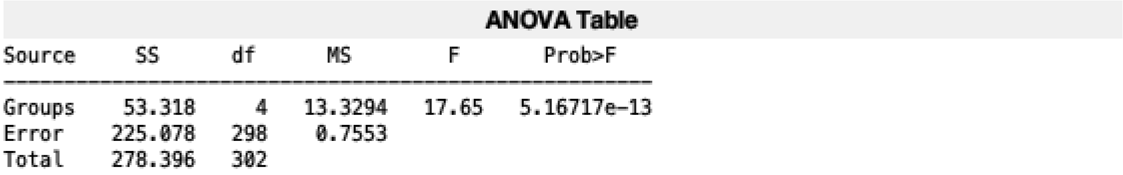
ANOVA Table

```
Source     SS      df      MS       F     Prob>F
------------------------------------------------
Groups    1520.1     4   380.023   4.89   0.0008
Error    23152.5   298    77.693
Total    24672.6   302
```



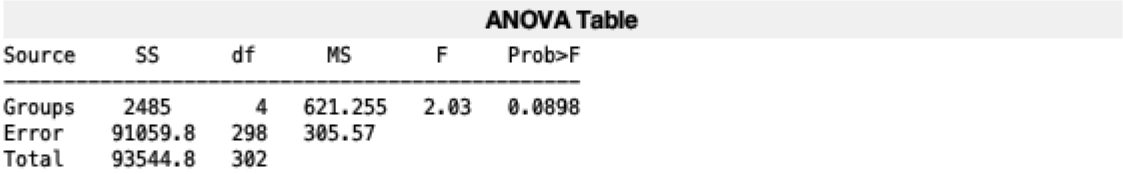
ANOVA Table

```
Source     SS      df      MS       F       Prob>F
-----------------------------------------------------
Groups    5.0985     4   1.27462   6.24   7.77661e-05
Error    60.8487   298   0.20419
Total    65.9472   302
```



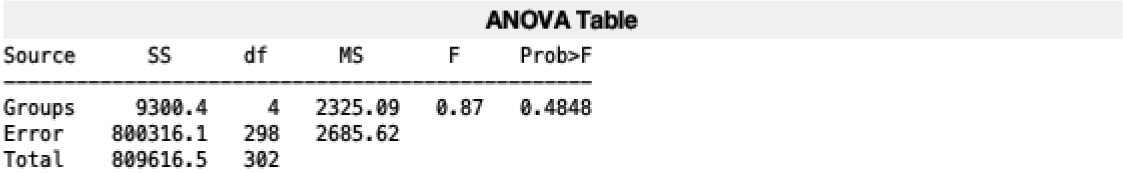
ANOVA Table

```
Source     SS       df      MS        F       Prob>F
-------------------------------------------------------
Groups    53.318     4    13.3294   17.65   5.16717e-13
Error    225.078   298     0.7553
Total    278.396   302
```



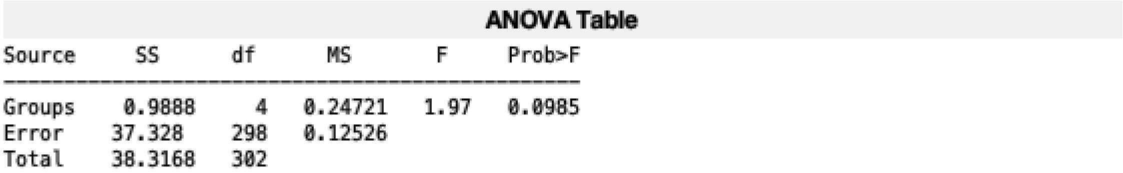
ANOVA Table

```
Source     SS       df      MS       F      Prob>F
-------------------------------------------------
Groups    2485       4    621.255   2.03   0.0898
Error    91059.8   298    305.57
Total    93544.8   302
```



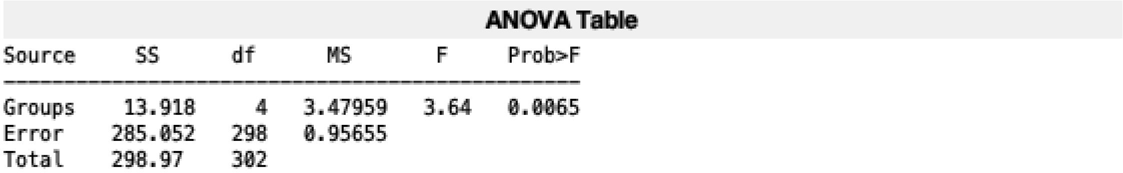
ANOVA Table

```
Source       SS      df       MS       F     Prob>F
---------------------------------------------------
Groups     9300.4     4    2325.09   0.87   0.4848
Error    800316.1   298    2685.62
Total    809616.5   302
```



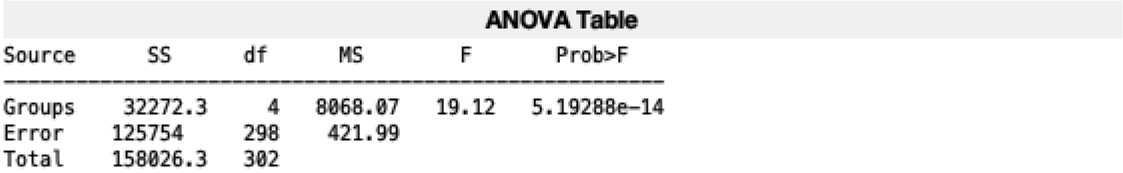
ANOVA Table

```
Source      SS      df       MS       F     Prob>F
--------------------------------------------------
Groups    0.9888     4   0.24721   1.97   0.0985
Error    37.328    298   0.12526
Total    38.3168   302
```



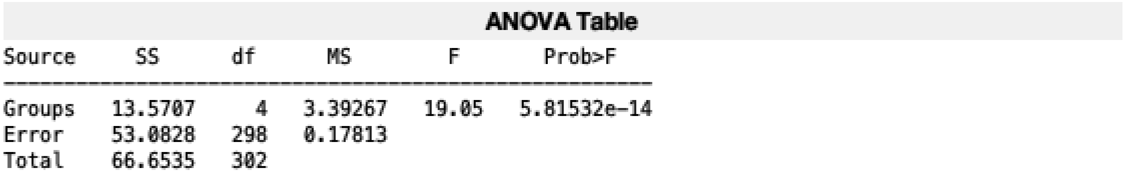
ANOVA Table

```
Source     SS      df      MS       F     Prob>F
------------------------------------------------
Groups    13.918     4   3.47959   3.64   0.0065
Error    285.052   298   0.95655
Total    298.97    302
```



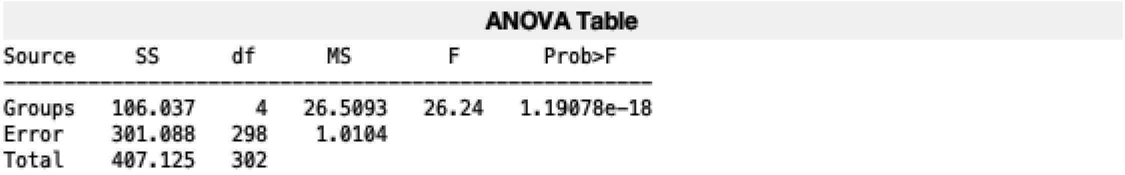
ANOVA Table

```
Source      SS       df      MS        F       Prob>F
-------------------------------------------------------
Groups    32272.3      4   8068.07   19.12   5.19288e-14
Error    125754      298    421.99
Total    158026.3    302
```



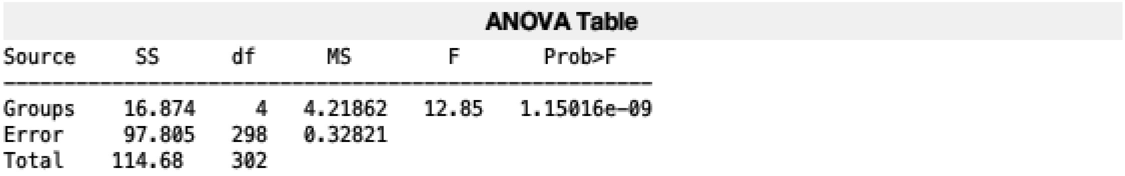
ANOVA Table

```
Source    SS       df     MS        F        Prob>F
------------------------------------------------------

Groups   13.5707    4   3.39267   19.05   5.81532e-14
Error    53.0828  298   0.17813
Total    66.6535  302
```



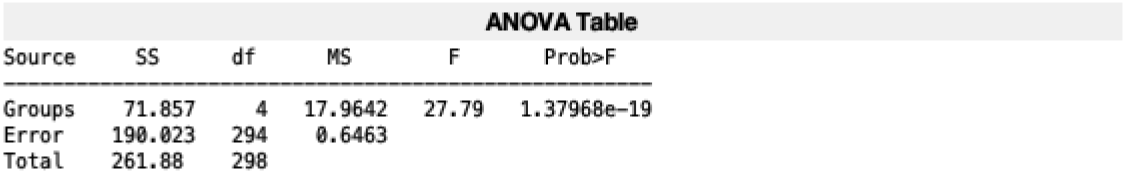
ANOVA Table

```
Source    SS       df     MS        F        Prob>F
------------------------------------------------------

Groups   106.037    4   26.5093   26.24   1.19078e-18
Error    301.088  298    1.0104
Total    407.125  302
```



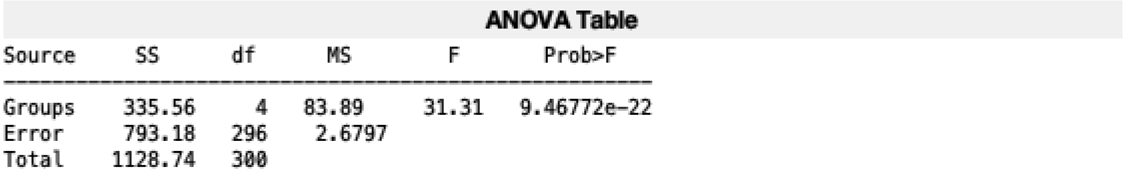
ANOVA Table

```
Source    SS      df     MS        F      Prob>F
-------------------------------------------------------
Groups    16.874    4   4.21862   12.85   1.15016e-09
Error     97.805  298   0.32821
Total    114.68   302
```



ANOVA Table

```
Source    SS      df     MS        F      Prob>F
-------------------------------------------------------
Groups    71.857    4   17.9642   27.79   1.37968e-19
Error    190.023  294    0.6463
Total    261.88   298
```



ANOVA Table

```
Source       SS      df      MS         F       Prob>F
-------------------------------------------------------
Groups     335.56     4    83.89      31.31    9.46772e-22
Error      793.18   296     2.6797
Total     1128.74   300
```

Correct the *p*-values using the Bonferroni and Benjamin Hochberg methods.

```
p_corrected = p*height(data2);
FDR = mafdr(p);
```

Which features are correlated with heart disease severity based on each method?

```
t = table(p,p_corrected,FDR);
t.Properties.RowNames = data2.Properties.VariableNames(1:13);
t.Properties.VariableNames = {'p_value','Bonferroni', 'FDR'}
```

```
t = 13×3 table
```

	p_value	Bonferroni	FDR
1 Age	0.0008	0.2367	2.0829e-06
2 SEX	0.0001	0.0236	2.3323e-07
3 chestPain	0.0000	0.0000	2.0662e-15
4 restingBP	0.0898	27.1958	1.9577e-04
5 cholest	0.4848	146.9004	8.9477e-04
6 highBloodSugar	0.0985	29.8442	1.9693e-04
7 ECG	0.0065	1.9778	1.5660e-05
8 maxHR	0.0000	0.0000	3.1148e-16

	p_value	Bonferroni	FDR
9 angina	0.0000	0.0000	2.7905e-16
10 ECGDepression	0.0000	0.0000	9.5233e-21
11 slopeST	0.0000	0.0000	3.9422e-12
12 fluoroscopy	0.0000	0.0000	1.6551e-21
13 thalliumTest	0.0000	0.0000	2.2715e-23

Evaluating AI model accuracy by comparison with a random distribution

- Null hypothesis: Model predictions are similar to random guess.
- To test this, we can create a model based on random guess and then compare with the machine-learning model using t-test or ranksum test.
- The same approach can be used for comparing between AI models.

Evaluating accuracy of a logistic regression model

Build a logistic regression model to predict Stroke in the Framingham dataset. We will use Age, BMI, Heart Rate, Diastolic BP, Glucose and Cigarettes per Day as predictors. In addition to building the model, find the accuracy, precision and recall.

```
% Create matrix with features of interest and remove rows with NaNs
fram_revised = rmmissing([fram.AGE,fram.BMI,fram.HEARTRTE,fram.DIABP,fram.GLUCOSE ,fran

% Create X and Y variables (and scale X)
X = zscore(fram_revised(:,1:6));
Y = fram_revised(:,7);

% Create train & test sets for hold out validation
[Xtrain, Ytrain, Xtest, Ytest] = trainTestSplit(X,Y,0.7);
Ytrain = categorical(Ytrain);

% Create model w/ training sets.
b = mnrfit(Xtrain,Ytrain);

% Calculate probability for each category, round up to nearest integer
% and calculate accuracy using Ypred and Ytest
probability = mnrval(b,Xtest);
Ypred_round = round(probability(:,2));

% issue with there being zero predictions in Ypred_round -> causes
% NaN output for precision
sum(Ypred_round==1)
```

```
ans = 5
```

```
accuracy = sum(Ypred_round == Ytest)/length(Ypred_round)
accuracy = 0.9105
```

```
precision = sum(Ypred_round==1 & Ytest==1)/sum(Ypred_round==1)
```

```
precision = 0.4000
```

```
recall = sum(Ypred_round==1 & Ytest==1)/sum(Ytest==1)
```

```
recall = 0.0074
```

Null Hypothesis: Model predictions are similar to random guess.

Create a distribution of random guesses - use the `randperm` function to create a random prediction for the test set.

```
random_pred = fram.STROKE(randperm(length(fram.STROKE)));
```

This creates a random prediction with the same number of stroke and normal patients - this represents a guess assuming that you the fraction of stroke patients in a population.

Get the accuracy of the random prediction.

```
accuracy_guess = sum(random_pred == fram.STROKE)/length(fram.STROKE)
```

```
accuracy_guess = 0.8316
```

Repeat 100 times and get 100 accuracies.

```
accuracy_guess = zeros(100,1);
for i = 1:100
    random_pred = fram.STROKE(randperm(length(fram.STROKE)));
    accuracy_guess(i,1) = sum(random_pred == fram.STROKE)/length(fram.STROKE);
end
```

Plot the distribution of random accuracies.

```
histogram(accuracy_guess,15);
xlabel("Random Accu]racy")
ylabel("Frequency")
title("Random Accuracy Histogram")
```

Use t-test to compare the accuracy of random guess with actual accuracy.

```
[h_acc, p_acc] = ttest2(accuracy, accuracy_guess)
```

```
h_acc = 1
p_acc = 1.7708e-68
```

Repeat for precision and recall.

```
precision_guess = zeros(100,1);
recall_guess = zeros(100,1);
for i = 1:100
    random_pred = fram.STROKE(randperm(length(fram.STROKE)));
    precision_guess(i,1) = sum(random_pred==1 & fram.STROKE==1)/sum(random_pred==1);
    recall_guess(i,1) = sum(random_pred==1 & fram.STROKE==1)/sum(fram.STROKE==1);
end
```

```
histogram(precision_guess,15);
xlabel("Random Precision")
ylabel("Frequency")
title("Random Precision Histogram")
```

```
histogram(recall_guess,15);
xlabel("Random Recall")
ylabel("Frequency")
title("Random Recall Histogram")
```

```
[h_prec, p_prec] = ttest2(precision, precision_guess)
```

```
h_prec = 1
p_prec = 1.1575e-64
```

```
[h_recall, p_recall] = ttest2(recall, recall_guess)
```

```
h_recall = 1
p_recall = 7.8718e-20
```