One-way Analysis Of Variance (ANOVA)

- One-way ANOVA is a statistical method to test the claim that the means of **two or more (*k*)** samples x_i with sample size n_i (i=1 … k) are equal.
- The test makes the following assumptions:
  - the shape of the population from which the samples are drawn should be normal
  - all samples have equal variance
  - all samples should be drawn independently
- Again there are two hypothesis:
  - The null hypothesis H_0 states that the means $\bar{x}_i$ with $i = 1 \dots k$ are equal.
  - The alternative hypothesis H_1 states that the means $\bar{x}_i$ with $i = 1 \dots k$ are not equal.
- One-way ANOVA is a generalization of the two sample t-test.

One-way ANOVA is a generalization of the two sample t-test.

- One-way ANOVA is a generalization of the two sample t-test.

```
n = 9
geneA = np.random.normal(size=n,loc=0.26,scale=0.01)
geneB = np.random.normal(size=n,loc=0.28,scale=0.01)
print "two sample t-test:"
print stats.ttest_ind(geneA,geneB)
print "one-way ANOVA:"
print stats.f_oneway(geneA,geneB)
```

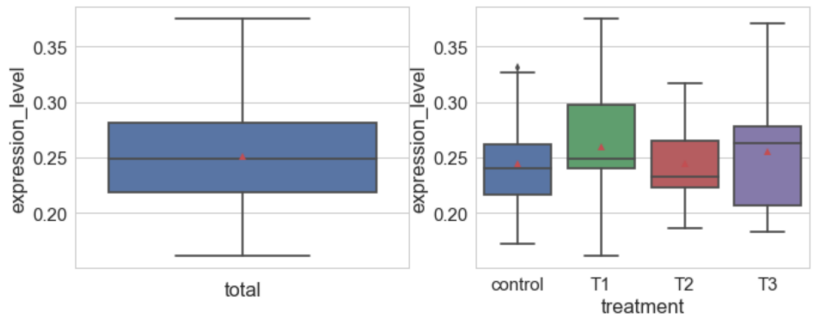
```
two sample t-test:
Ttest_indResult(statistic=-4.022327404376181, pvalue=0.0009847119072361069)
one-way ANOVA:
F_onewayResult(statistic=16.179117747995637, pvalue=0.0009847119072361058)
```

One-way Analysis Of Variance (ANOVA)

```
n = 4
sample_control  = np.random.normal(size=n,loc=0.26,scale=0.05)
sample_T1 = np.random.normal(size=n,loc=0.26,scale=0.05)
sample_T2 = np.random.normal(size=n,loc=0.26,scale=0.05)
sample_T3 = np.random.normal(size=n,loc=0.26,scale=0.05)
```

	treatment	expression_level
0	control	0.276610
1	control	0.268073
2	control	0.256852
3	control	0.249141
4	T1	0.252675
5	T1	0.247875
6	T1	0.280871
7	T1	0.261644
8	T2	0.281502
9	T2	0.257326
10	T2	0.271810
11	T2	0.281779
12	T3	0.256650
13	T3	0.270311
14	T3	0.249154
15	T3	0.246365

One-way Analysis Of Variance (ANOVA)



	treatment	expression_level
0	control	0.276610
1	control	0.268073
2	control	0.256852
3	control	0.249141
4	T1	0.252675
5	T1	0.247875
6	T1	0.280871
7	T1	0.261644
8	T2	0.281502
9	T2	0.257326
10	T2	0.271810
11	T2	0.281779
12	T3	0.256650
13	T3	0.270311
14	T3	0.249154
15	T3	0.246365

One-way Analysis Of Variance (ANOVA)

- ANOVA computes a **F-statistic** that is simply a ratio of two variances.
- Where the two sample t-test T-statistic looks at the difference in mean between two samples, the ANOVA F-statistics looks at the ratio of two variances.
- The first variance in the F-statistic is the variation between the *k*=4 sample means. It is computed as what is known as the treatment sum of squares SST:

$$SST = \sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x}_{total})$$

- The second variation is the variation within the sample. This is computed as the sum of squares of the error SSE:

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)$$

One-way Analysis Of Variance (ANOVA)

- The variance of the expression level in the dataset (not taking treatment into account) is computed by the total sum of squares TSS:

$$TSS = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_{total})^2$$

- So, ANOVA partitions the total variance (TSS) into a part caused by the treatment (SST) and a part the is not explained (SSE), i.e. TSS = SST + SEE.
- explained / unexplained
- between group / within group

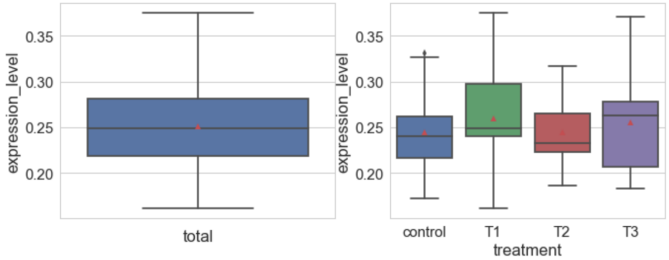
One-way Analysis Of Variance (ANOVA)

- The F-statistic computed by ANOVA is the ratio of SST and SSE (with some variance normalization):

$$F = \frac{SST/(k-1)}{SSE/(n_t - k)}$$

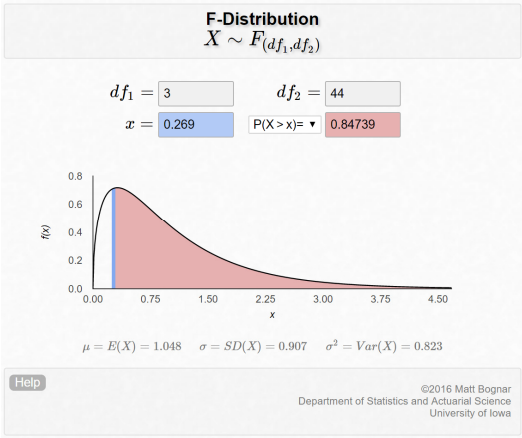
- So, ANOVA partitions the total variance (TSS) into a part caused by the treatment (SST) and a part that is with unknown source (SSE).
- (explained / unexplained) or (between group / within group)
- If the null hypothesis is true then the F-statistic follows an F-distribution with df_1=k-1 and df_2=n_t-k degrees of freedom.

One-way Analysis Of Variance (ANOVA)



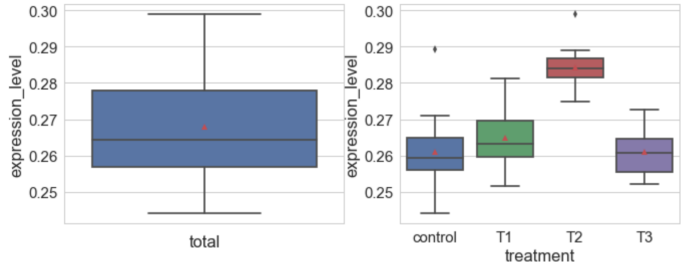

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
mod = ols('expression_level ~ treatment',
                data=df).fit()
aov_table = sm.stats.anova_lm(mod, typ=2)
print aov_table
```

```
              sum_sq    df         F    PR(>F)
treatment   0.001968   3.0  0.269032  0.847367
Residual    0.107293  44.0       NaN       NaN
```



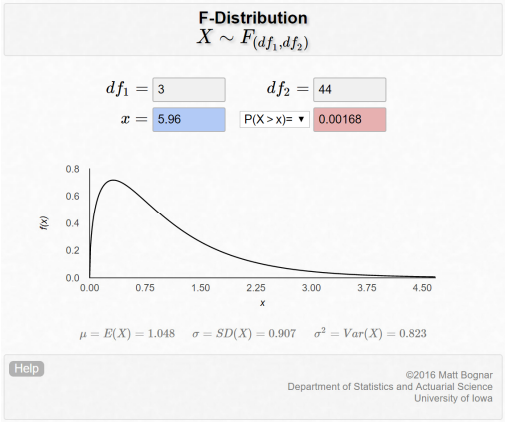
http://homepage.divms.uiowa.edu/~mbognar/applets/f.html

One-way Analysis Of Variance (ANOVA)



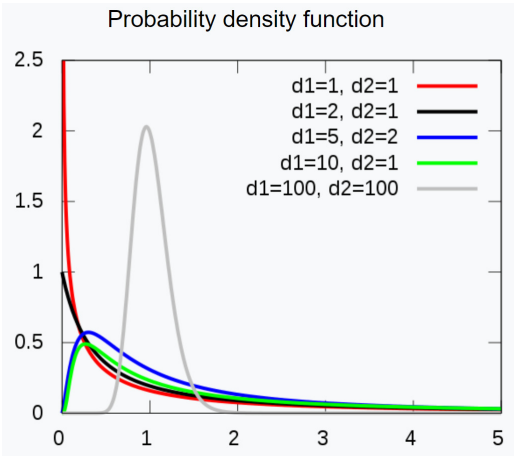

```
mod = ols('expression_level ~ treatment',
                data=df).fit()
aov_table = sm.stats.anova_lm(mod, typ=2)
print aov_table
```

	sum_sq	df	F	PR(>F)
treatment	0.001384	3.0	5.945948	0.001702
Residual	0.003413	44.0	NaN	NaN



http://homepage.divms.uiowa.edu/~mbognar/applets/f.html

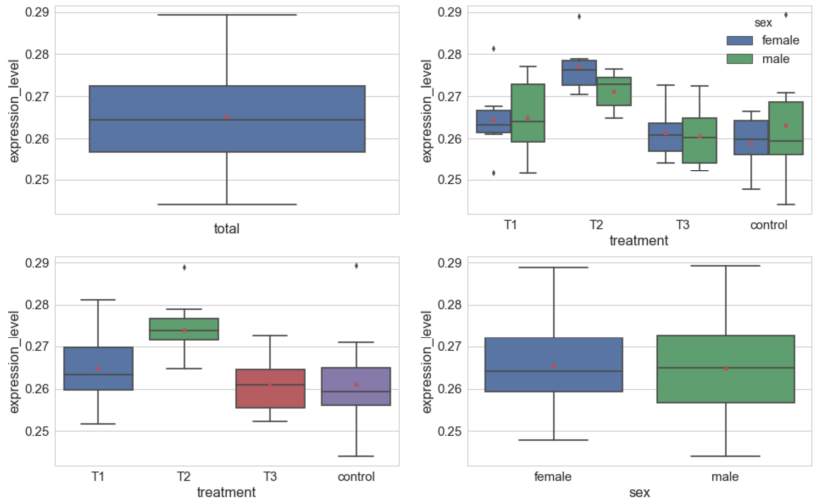
One-way Analysis Of Variance (ANOVA)



https://en.wikipedia.org/wiki/F-distribution

Two-way Analysis Of Variance (ANOVA)

- In two-way ANOVA there are two factors.



	treatment	sex	expression_level
0	control	female	0.276610
1	control	male	0.268073
2	control	female	0.256852
3	control	male	0.249141
4	T1	female	0.252675
5	T1	male	0.247875
6	T1	female	0.280871
7	T1	male	0.261644
8	T2	female	0.281502
9	T2	male	0.257326
10	T2	female	0.271810
11	T2	male	0.281779
12	T3	female	0.256650
13	T3	male	0.270311
14	T3	female	0.249154
15	T3	male	0.246365

Two-way Analysis Of Variance (ANOVA)

```
mod = ols('expression_level ~ treatment',
                data=df).fit()
aov_table = sm.stats.anova_lm(mod, typ=2)
print aov_table
```

```
              sum_sq    df          F    PR(>F)
treatment   0.001384   3.0   5.945948  0.001702
Residual    0.003413  44.0        NaN       NaN
```

```
mod = ols('expression_level ~ sex',
                data=df).fit()
aov_table = sm.stats.anova_lm(mod, typ=2)
print aov_table
```

```
              sum_sq    df          F    PR(>F)
sex         0.000003   1.0   0.033092   0.85645
Residual    0.004793  46.0        NaN       NaN
```

	treatment	sex	expression_level
0	control	female	0.276610
1	control	male	0.268073
2	control	female	0.256852
3	control	male	0.249141
4	T1	female	0.252675
5	T1	male	0.247875
6	T1	female	0.280871
7	T1	male	0.261644
8	T2	female	0.281502
9	T2	male	0.257326
10	T2	female	0.271810
11	T2	male	0.281779
12	T3	female	0.256650
13	T3	male	0.270311
14	T3	female	0.249154
15	T3	male	0.246365

Two-way Analysis Of Variance (ANOVA)

- In two-way ANOVA both factors are taken into account when computing the F-statistic:

$$TSS = SS_{treatment} + SS_{sex} + SSE$$

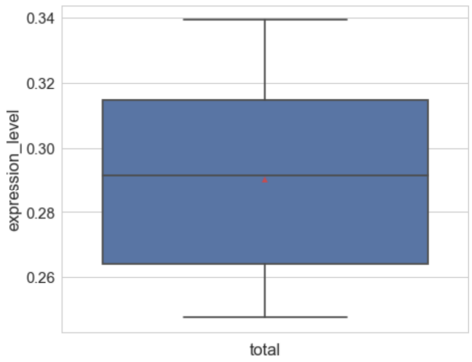
```
mod = ols('expression_level ~ treatment + sex',
                data=df).fit()
aov_table = sm.stats.anova_lm(mod, typ=2)
print aov_table
```

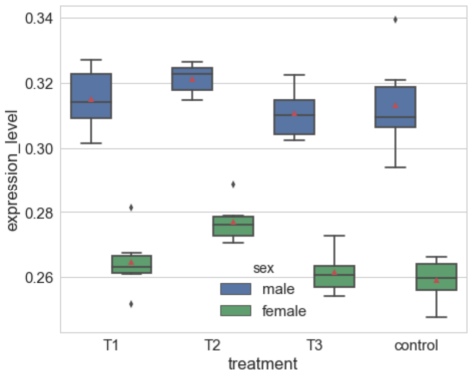
```
              sum_sq     df          F    PR(>F)
treatment   0.001384    3.0   5.816689  0.001986
sex         0.000003    1.0   0.043488  0.835794
Residual    0.003410   43.0        NaN       NaN
```

	treatment	sex	expression_level
0	control	female	0.276610
1	control	male	0.268073
2	control	female	0.256852
3	control	male	0.249141
4	T1	female	0.252675
5	T1	male	0.247875
6	T1	female	0.280871
7	T1	male	0.261644
8	T2	female	0.281502
9	T2	male	0.257326
10	T2	female	0.271810
11	T2	male	0.281779
12	T3	female	0.256650
13	T3	male	0.270311
14	T3	female	0.249154
15	T3	male	0.246365

Two-way Analysis Of Variance (ANOVA)

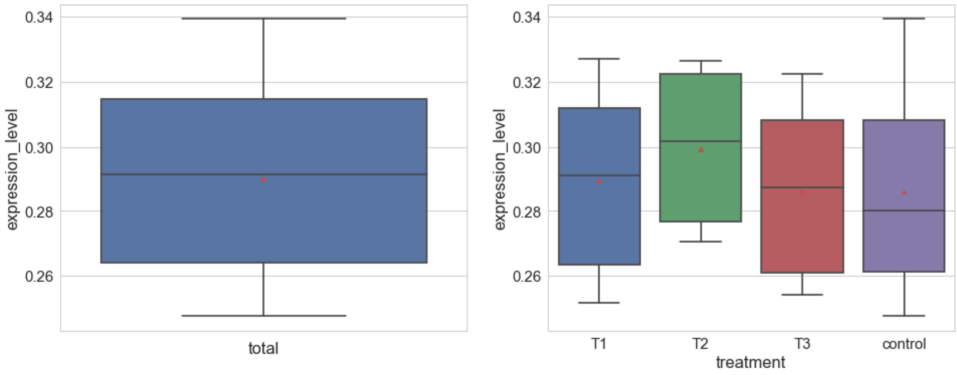
- A large effect of factor “sex” on the gene expression level is added:





Two-way Analysis Of Variance (ANOVA)

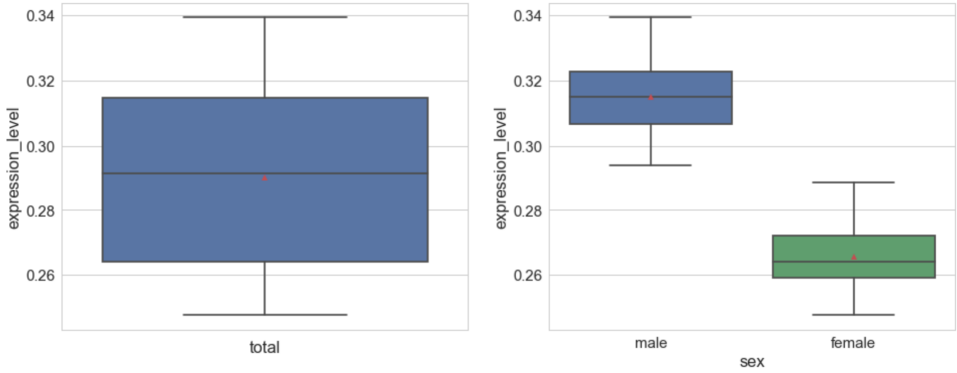
- A large effect of factor “sex” on the gene expression level is added:



	sum_sq	df	F	PR(>F)
treatment	0.001384	3.0	0.619269	0.606255
Residual	0.032770	44.0	NaN	NaN

Two-way Analysis Of Variance (ANOVA)

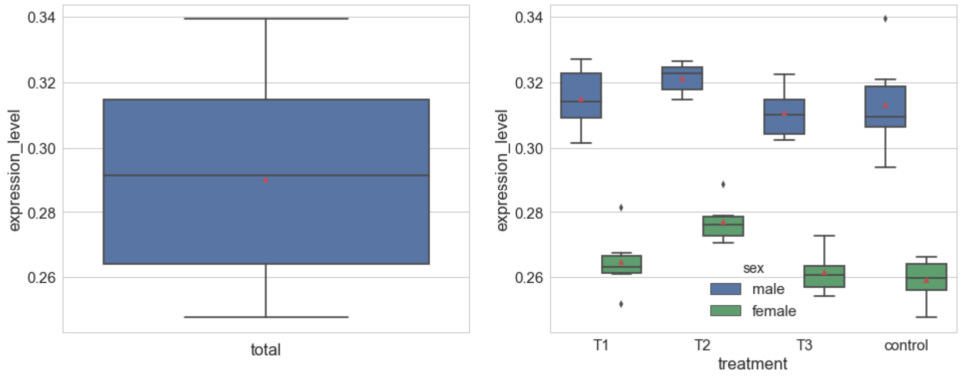
- A large effect of factor “sex” on the gene expression level is added:



	sum_sq	df	F	PR(>F)
sex	0.029360	1.0	281.771448	3.053987e-21
Residual	0.004793	46.0	NaN	NaN

Two-way Analysis Of Variance (ANOVA)

- A large effect of factor “sex” on the gene expression level is added:

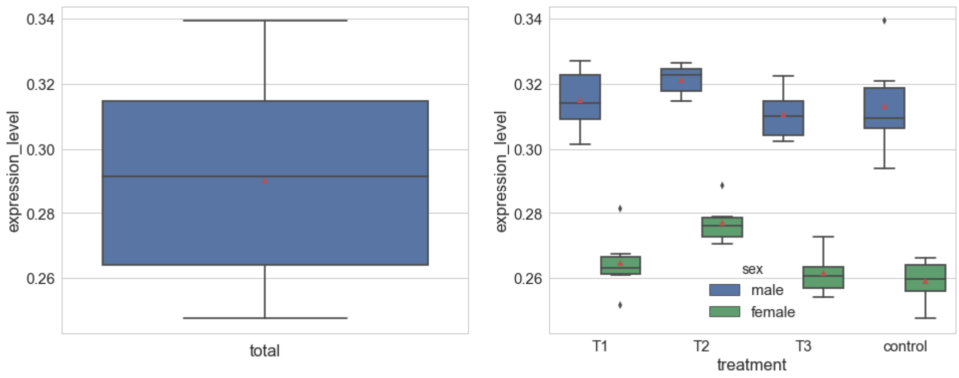


	sum_sq	df	F	PR(>F)
treatment	0.001384	3.0	5.816689	1.986050e-03
sex	0.029360	1.0	370.284849	9.454352e-23
Residual	0.003410	43.0	NaN	NaN

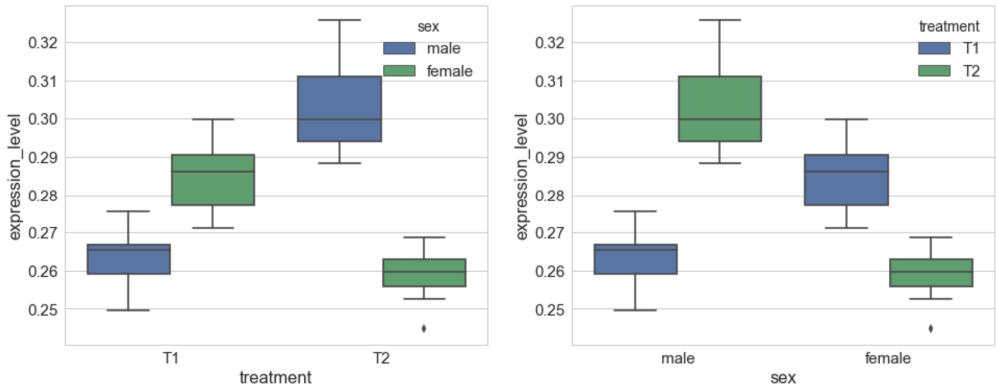
Two-way Analysis Of Variance (ANOVA)

- The main difference with one-way ANOVA is the computation of the sum of squares of the error SSE which is computed in two-way ANOVA as

$$SSE = \sum_{l=1}^{r} \sum_{j=1}^{k_s} \sum_{i=1}^{k_t} (x_{ijl} - \bar{x}_{ij})^2$$

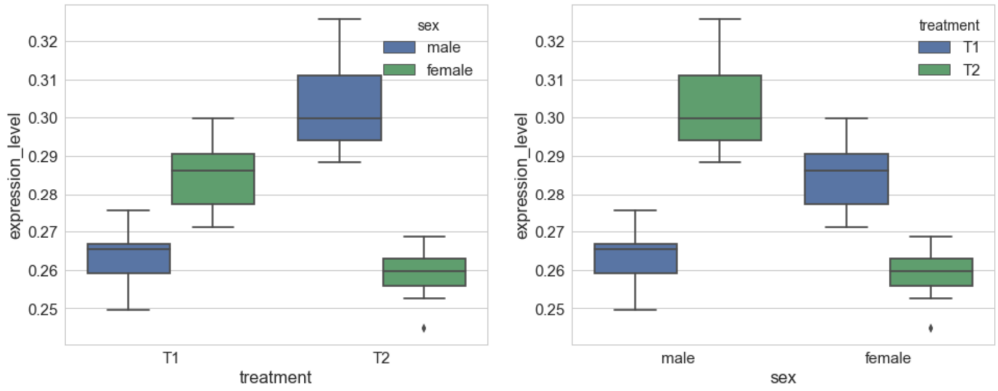


Two-way Analysis Of Variance (ANOVA)

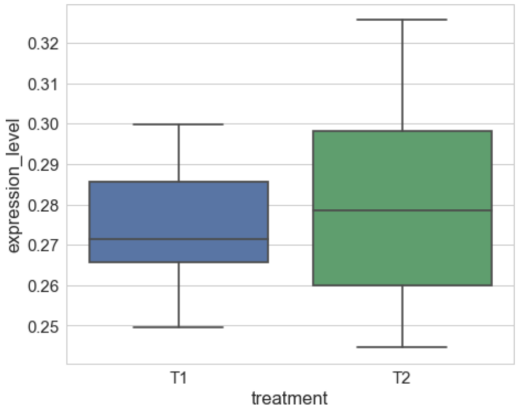


Two-way Analysis Of Variance (ANOVA)

$$TSS = SS_{treatment} + SS_{sex} + SS_{treatment:sex} + SSE$$



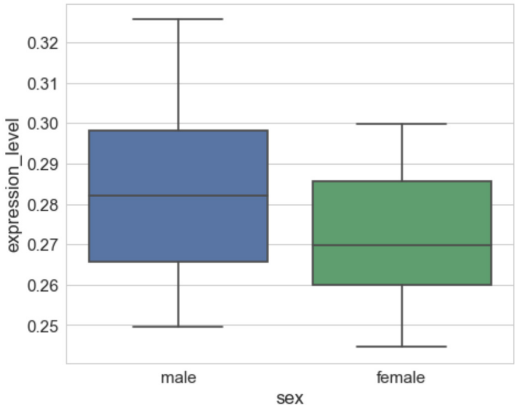
Two-way Analysis Of Variance (ANOVA)



one-way ANOVA for treatment:

	sum_sq	df	F	PR(>F)
treatment	0.000575	1.0	1.37193	0.247512
Residual	0.019265	46.0	NaN	NaN

Two-way Analysis Of Variance (ANOVA)

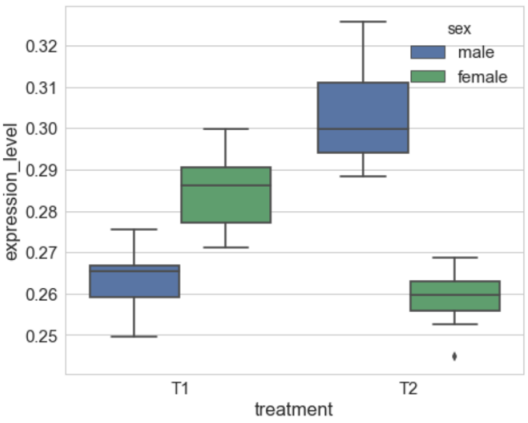


one-way ANOVA for sex:

	sum_sq	df	F	PR(>F)
sex	0.001612	1.0	4.068529	0.049547
Residual	0.018228	46.0	NaN	NaN

Two-way Analysis Of Variance (ANOVA)

- If there is an interaction effect we should be very critical about the conclusions we make.



two-way ANOVA without interaction term:

	sum_sq	df	F	PR(>F)
treatment	0.000575	1.0	1.464674	0.232507
sex	0.001612	1.0	4.109628	0.048588
Residual	0.017653	45.0	NaN	NaN

two-way ANOVA with interaction term:

	sum_sq	df	F	PR(>F)
treatment	0.000575	1.0	6.116762	1.732354e-02
sex	0.001612	1.0	17.162605	1.532358e-04
treatment:sex	0.013520	1.0	143.928763	1.823620e-15
Residual	0.004133	44.0	NaN	NaN

Power analysis

- The **power** of a hypothesis test is defined as the probability of rejecting the null hypothesis in case the alternative hypothesis is true.
- In power analysis the power is defined as 1 - β where β is the probability of accepting the null hypothesis even though the alternative hypothesis is true.
- The **effect size**. It is the minimum size of the difference between the null hypothesis and the alternative hypothesis that we hope to detect.

Power analysis: population variance

```
m = 1000
alpha = 0.01
n = 9
diff = 0.02
for sigma in [0.01,0.015,0.02,0.025,0.03,0.04,0.05]:
    power = 0.
    for i in range(m):
        geneA = np.random.normal(size=n,loc=0.26,scale=sigma)
        geneB = np.random.normal(size=n,loc=0.26+diff,scale=sigma)
        pvalue = stats.ttest_ind(geneA,geneB).pvalue
        if pvalue < alpha:
            power += 1.
    print "sample size: %i / effect size: %.2f / population std: %.3f / power: %.2f" % \
    (n,diff,sigma,power/m)
```

```
sample size: 9 / effect size: 0.02 / population std: 0.010 / power: 0.88
sample size: 9 / effect size: 0.02 / population std: 0.015 / power: 0.52
sample size: 9 / effect size: 0.02 / population std: 0.020 / power: 0.26
sample size: 9 / effect size: 0.02 / population std: 0.025 / power: 0.13
sample size: 9 / effect size: 0.02 / population std: 0.030 / power: 0.10
sample size: 9 / effect size: 0.02 / population std: 0.040 / power: 0.06
sample size: 9 / effect size: 0.02 / population std: 0.050 / power: 0.03
```

Power analysis: sample size

```
sigma = 0.01
diff = 0.02
for n in range(2,15):
    power = 0.
    for i in range(m):
        geneA = np.random.normal(size=n,loc=0.26,scale=sigma)
        geneB = np.random.normal(size=n,loc=0.26+diff,scale=sigma)
        pvalue = stats.ttest_ind(geneA,geneB).pvalue
        if pvalue < alpha:
            power += 1.
    print "sample size: %i / effect size: %.2f / population std: %.3f / power: %.2f" % \
    (n,diff,sigma,power/m)
```

```
sample size: 2 / effect size: 0.02 / population std: 0.010 / power: 0.05
sample size: 3 / effect size: 0.02 / population std: 0.010 / power: 0.16
sample size: 4 / effect size: 0.02 / population std: 0.010 / power: 0.33
sample size: 5 / effect size: 0.02 / population std: 0.010 / power: 0.47
sample size: 6 / effect size: 0.02 / population std: 0.010 / power: 0.65
sample size: 7 / effect size: 0.02 / population std: 0.010 / power: 0.73
sample size: 8 / effect size: 0.02 / population std: 0.010 / power: 0.84
sample size: 9 / effect size: 0.02 / population std: 0.010 / power: 0.91
sample size: 10 / effect size: 0.02 / population std: 0.010 / power: 0.93
sample size: 11 / effect size: 0.02 / population std: 0.010 / power: 0.96
sample size: 12 / effect size: 0.02 / population std: 0.010 / power: 0.98
sample size: 13 / effect size: 0.02 / population std: 0.010 / power: 0.99
sample size: 14 / effect size: 0.02 / population std: 0.010 / power: 0.99
```

Power analysis: effect size

```
sigma = 0.01
n= 9
for diff in [0.001,0.005,0.01,0.02,0.03,0.04]:
    power = 0.
    for i in range(m):
        geneA = np.random.normal(size=n,loc=0.26,scale=sigma)
        geneB = np.random.normal(size=n,loc=0.26+diff,scale=sigma)
        pvalue = stats.ttest_ind(geneA,geneB).pvalue
        if pvalue < alpha:
            power += 1.
    print "sample size: %i / effect size: %.2f / population std: %.3f / power: %.2f" % \
    (n,diff,sigma,power/m)
```

```
sample size: 9 / effect size: 0.00 / population std: 0.010 / power: 0.01
sample size: 9 / effect size: 0.01 / population std: 0.010 / power: 0.06
sample size: 9 / effect size: 0.01 / population std: 0.010 / power: 0.24
sample size: 9 / effect size: 0.02 / population std: 0.010 / power: 0.89
sample size: 9 / effect size: 0.03 / population std: 0.010 / power: 1.00
sample size: 9 / effect size: 0.04 / population std: 0.010 / power: 1.00
```

Power analysis

- The power of a hypothesis test depends on these four numbers:
  - the effect size
  - the population variance
  - the sample size
  - the significance level
- Given the effect size, the population variance, the significance level and the expected minimum power we can compute the minimum samples size required to reach that power.

Sample size calculations

- Sample size calculation for unpaired t-test:

  Suppose you want to compare the mean in one group to the mean in another (i.e. carry out an unpaired t-test). The number, n, required in each group is given by

$$n = f(\alpha, \beta) \cdot \frac{2s^2}{\delta^2}$$

  Where:
  α is the significance level (using a two-sided test) — i.e. your cut-off for regarding the result as statistically significant.
  $1 - \beta$ is the power of your test.
  $f(\alpha, \beta)$ is a value calculated from α and β
  δ is the smallest difference in means that you regard as being important to be able to detect.
  s is the standard deviation of whatever it is we're measuring — this will need to be estimated from previous studies.

$f(\alpha, \beta)$ for the most commonly used values for α and β

	β			
α	0.05	0.1	0.2	0.5
0.05	13.0	10.5	7.9	3.8
0.01	17.8	14.9	11.7	6.6

Power analysis

```
import statsmodels.stats.power as smp
sigma = 0.01
diff = 0.02
for power in [0.99,0.95,0.9,0.8]:
    print "Minimum sample size required for power = %f: %f" \
       % (power,smp.TTestIndPower().solve_power(diff/sigma, power=power, alpha=alpha, alternative='two-sided'))
```

```
Minimum sample size required for power = 0.990000: 13.805334
Minimum sample size required for power = 0.950000: 10.710971
Minimum sample size required for power = 0.900000: 9.251530
Minimum sample size required for power = 0.800000: 7.660540
```

Power analysis

```
n = 11
power = 0.
for i in range(m):
    geneA = np.random.normal(size=n,loc=0.26,scale=0.01)
    geneB = np.random.normal(size=n,loc=0.26+diff,scale=0.01)
    pvalue = stats.ttest_ind(geneA,geneB).pvalue
    if pvalue < alpha:
        power += 1.
print "Power for sample size %i, mean difference %f and population sigma %f: %f" % \
(n,diff,sigma,power/(float(m)))
```

```
Power for sample size 11, mean difference 0.020000 and population sigma 0.010000: 0.956000
```

Power analysis