BIM 105
Probability and Statistics for Biomedical Engineers

David M. Rocke

Department of Biomedical Engineering

Model Selection

- Suppose we have several possible predictor variables for a response.
- Each of the predictors and the response may be transformed.
- We can try to find a subset of the variables that predict well.
- We may want to include powers or interactions of the variables.
- We need a way to compare and assess models.

Principles of Model Selection

- Whenever an interaction is included, so should be any interaction contained in it.
  - If x_1:x_2 is in the model, then so should be x_1 and x_2.
  - If x_1:x_2:x_3 is in the model, then so should be x_1:x_2, etc.
- Whenever x^2 is in the model then so should be x.
- We want the simplest model that fits the data.
- Occam's Razor.
- Many possible criteria for whether a variable should be kept: the default in MATLAB stepwise is an F-test for the two models.

Total Lung Capacity

- The data set tlc contains observations on 32 lung transplant patients.
  - age in years
  - sex (female=1, male=2)
  - height (cm)
  - tlc = total lung capacity (L)
- We want to investigate the ability of the first three variables to predict total lung capacity.
- This data set was imported as a Table.

```
>> summary(tlc)

    age: 32x1 double
        Values:

            min         11
            median    28.5
            max         52

    sex: 32x1 double
        Values:

            min          1
            median     1.5
            max          2

    height: 32x1 double
        Values:

            min        138
            median     170
            max        189

    tlc: 32x1 double
        Values:

            min         3.4
            median     6.15
            max        9.45
```

Linear Model for tlc Data

```
>> fitlm(tlc)

Linear regression model:
    tlc ~ 1 + age + sex + height

Estimated Coefficients:
                   Estimate        SE        tStat      pValue
                   _________    _________    _______    _________

    (Intercept)      -9.2407      3.4449    -2.6824     0.012123
    age            -0.025025    0.023531    -1.0635      0.29665
    sex              0.69705     0.49944     1.3957      0.17379
    height          0.089546    0.024552     3.6472    0.0010729


Number of observations: 32, Error degrees of freedom: 28
Root Mean Squared Error: 1.16
R-squared: 0.542,  Adjusted R-Squared 0.493
F-statistic vs. constant model: 11.1, p-value = 5.82e-05
```

```
>> fitlm(tlc,'tlc~age*height*sex')

Linear regression model:
    tlc ~ 1 + age*sex + age*height + sex*height + age:sex:height

Estimated Coefficients:
                       Estimate        SE         tStat      pValue
                      __________    _________    ________    _______

    (Intercept)          5.1688       27.113     0.19064    0.85041
    age                 -1.0638       1.1214    -0.94861    0.35227
    sex                 -18.416       17.287     -1.0654    0.29732
    height             0.013239      0.16868    0.078489    0.93809
    age:sex               1.2337      0.80799      1.5269    0.13987
    age:height         0.0056433    0.0067236     0.83933    0.40957
    sex:height           0.10537      0.10455      1.0078    0.32359
    age:sex:height    -0.0068375    0.0046827     -1.4602    0.15721


Number of observations: 32, Error degrees of freedom: 24
Root Mean Squared Error: 1.14
R-squared: 0.618,  Adjusted R-Squared 0.506
F-statistic vs. constant model: 5.54, p-value = 0.000689
```

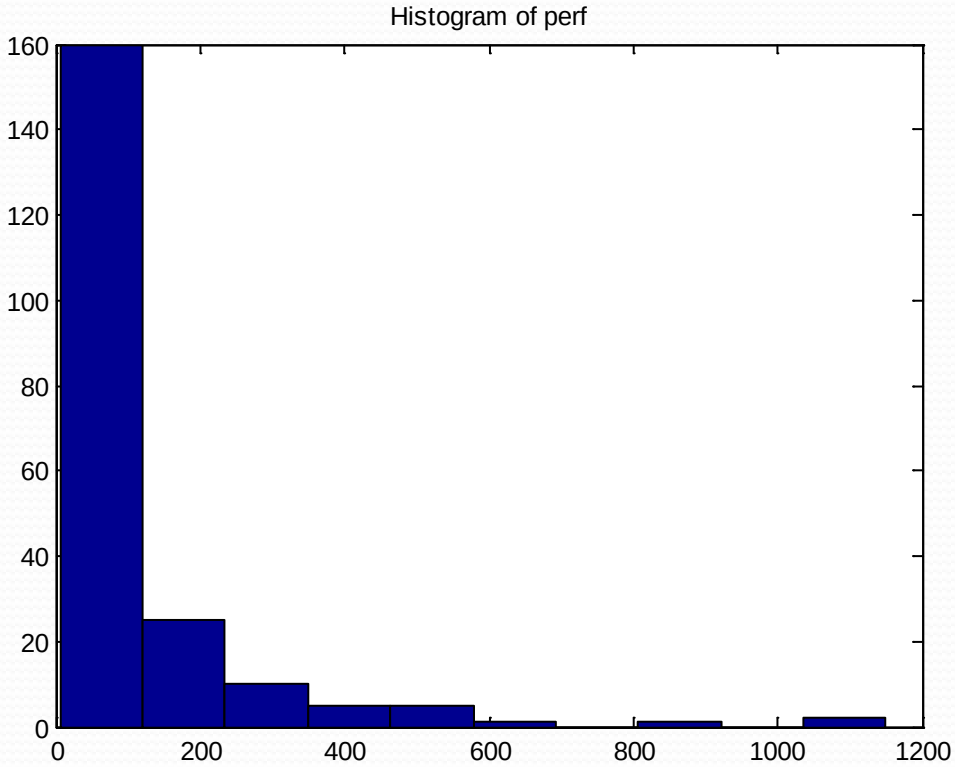
```
>> stepwiselm(tlc)
1. Adding height, FStat = 28.1397, pValue = 9.85334e-06

Linear regression model:
    tlc ~ 1 + height

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    ________    ________    __________

    (Intercept)     -9.7403     2.9911    -3.2564       0.002799
    height         0.094529    0.01782     5.3047    9.8533e-06


Number of observations: 32, Error degrees of freedom: 30
Root Mean Squared Error: 1.19
R-squared: 0.484,  Adjusted R-Squared 0.467
F-statistic vs. constant model: 28.1, p-value = 9.85e-06
```

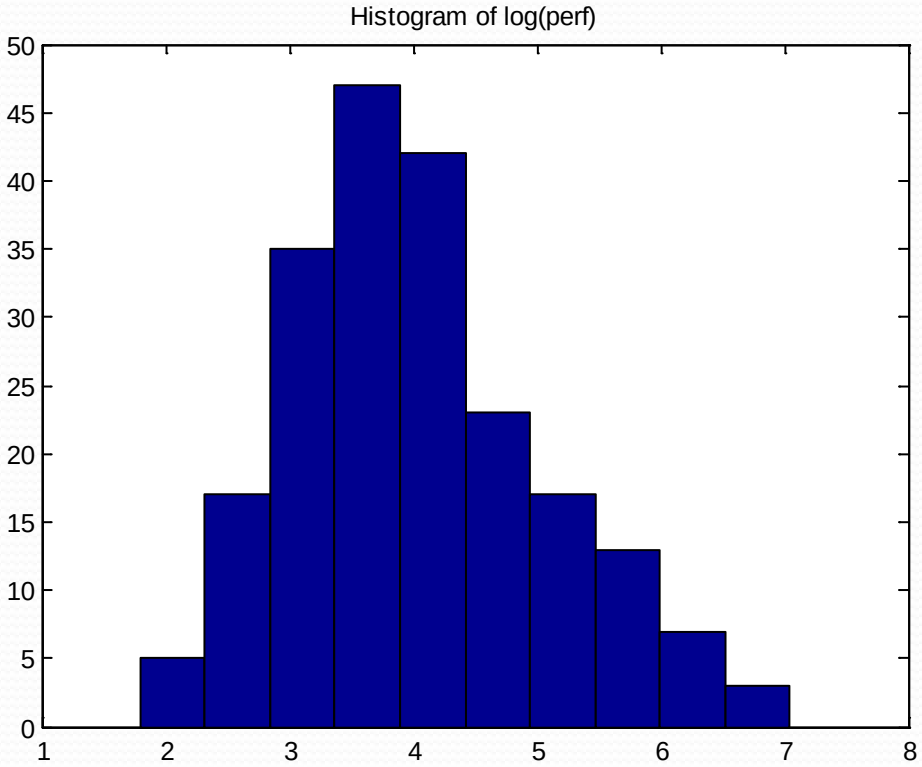
```
>> stepwiselm(tlc,'tlc~age*height*sex')
1. Removing age:sex:height, FStat = 2.1321, pValue = 0.15721
2. Removing sex:height, FStat = 0.26428, pValue = 0.61171
3. Removing age:sex, FStat = 0.59927, pValue = 0.44584
4. Removing age:height, FStat = 1.7378, pValue = 0.1985
5. Removing age, FStat = 1.131, pValue = 0.29665
6. Removing sex, FStat = 2.4182, pValue = 0.13078

Linear regression model:
    tlc ~ 1 + height

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    _______    _______    __________

    (Intercept)     -9.7403     2.9911    -3.2564      0.002799
    height         0.094529    0.01782     5.3047    9.8533e-06


Number of observations: 32, Error degrees of freedom: 30
Root Mean Squared Error: 1.19
R-squared: 0.484,  Adjusted R-Squared 0.467
F-statistic vs. constant model: 28.1, p-value = 9.85e-06
```

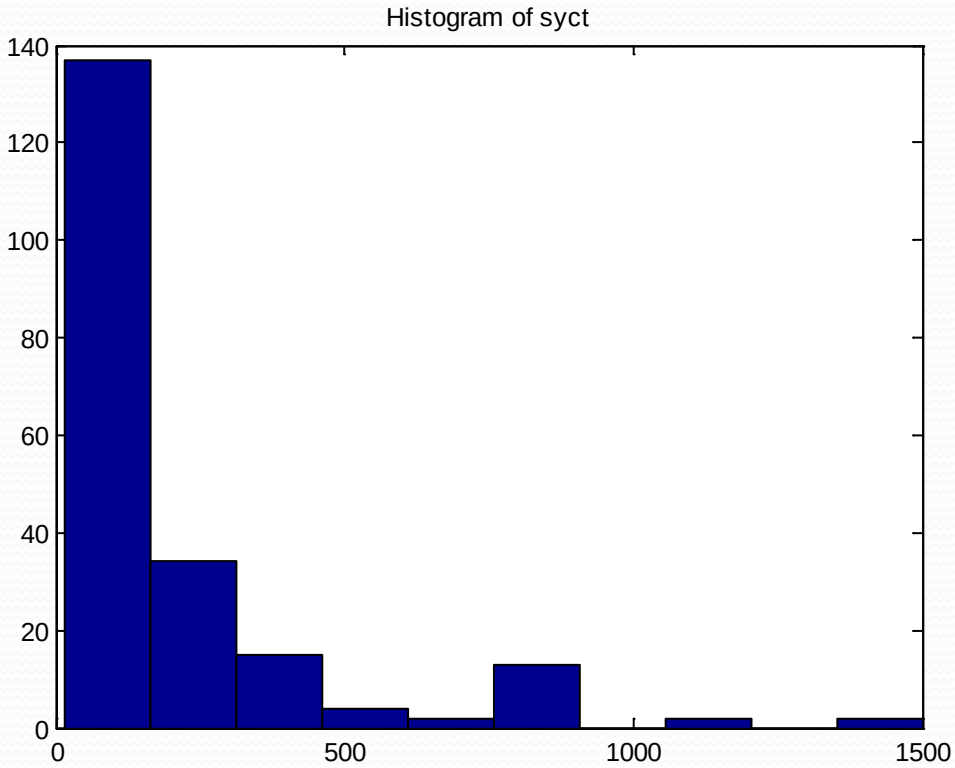
Data Set cpus

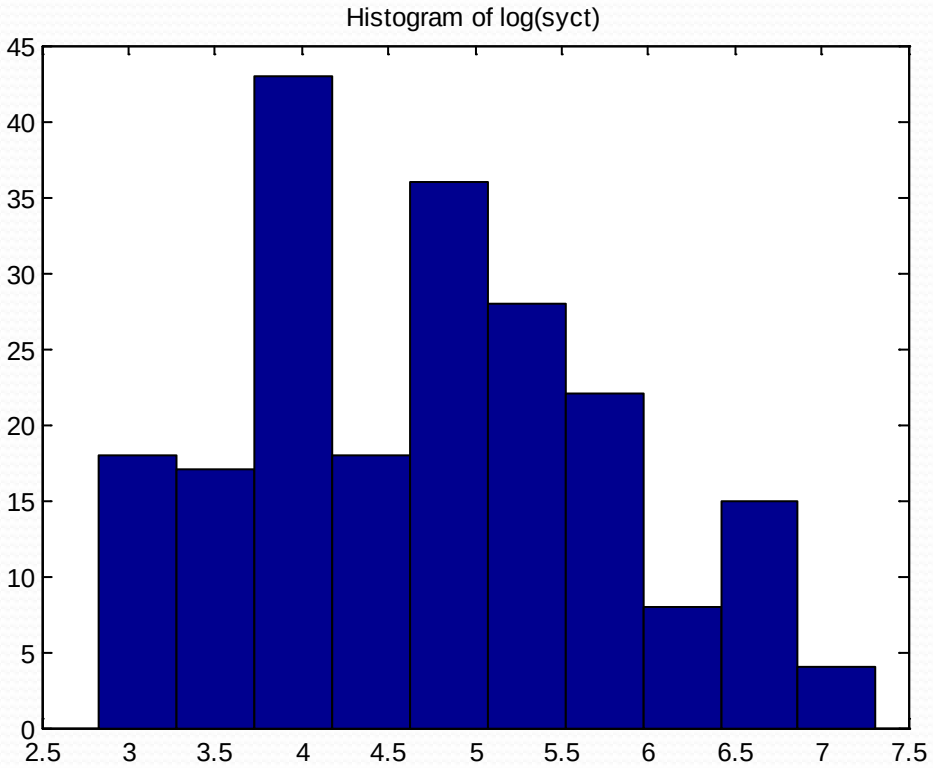
- Data on the performance of 209 computer processors.
  - perf = benchmark performance (response)
  - syct = cycle time in nanoseconds.
  - mmin = minimum main memory
  - mmax maximum main memory
  - cach = cache size
  - chmin = minimum number of channels.
  - chmax = maximum number of channels.

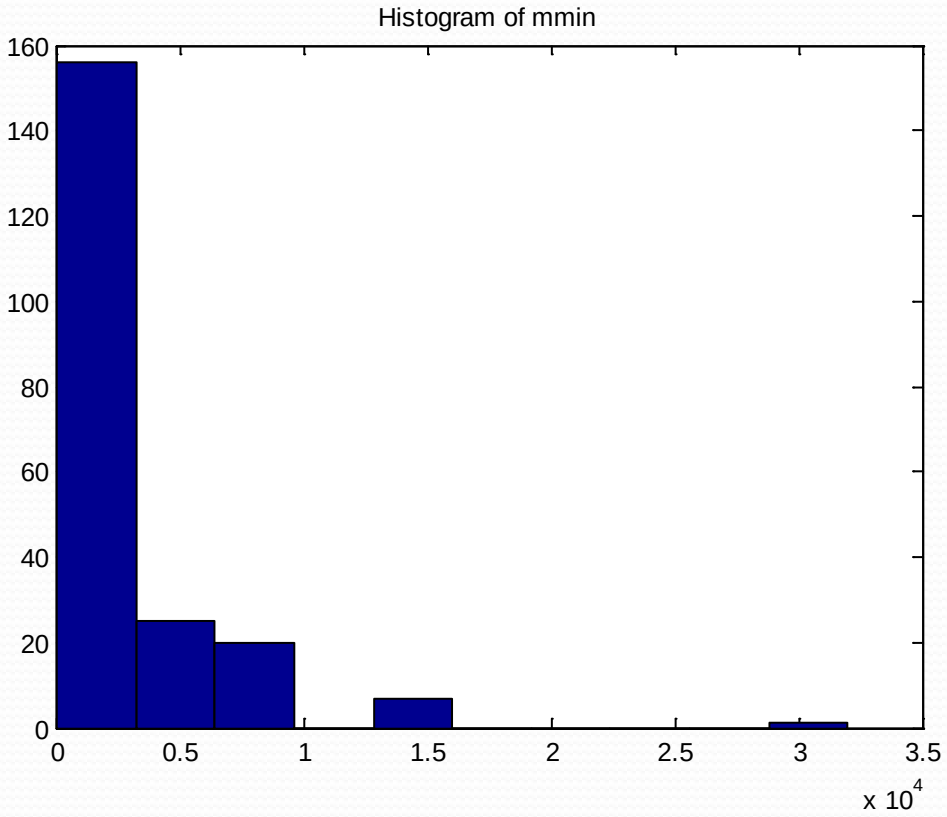
```
>> summary(cpus)
    syct: 209x1 double
            min          17
            median      110
            max        1500
    mmin: 209x1 double
            min           64
            median      2000
            max        32000
    mmax: 209x1 double
            min           64
            median      8000
            max        64000
    cach: 209x1 double
            min           0
            median        8
            max         256
    chmin: 209x1 double
            min           0
            median        2
            max          52
    chmax: 209x1 double
            min           0
            median        8
            max         176

    perf: 209x1 double

            min           6
            median       50
            max        1150
```

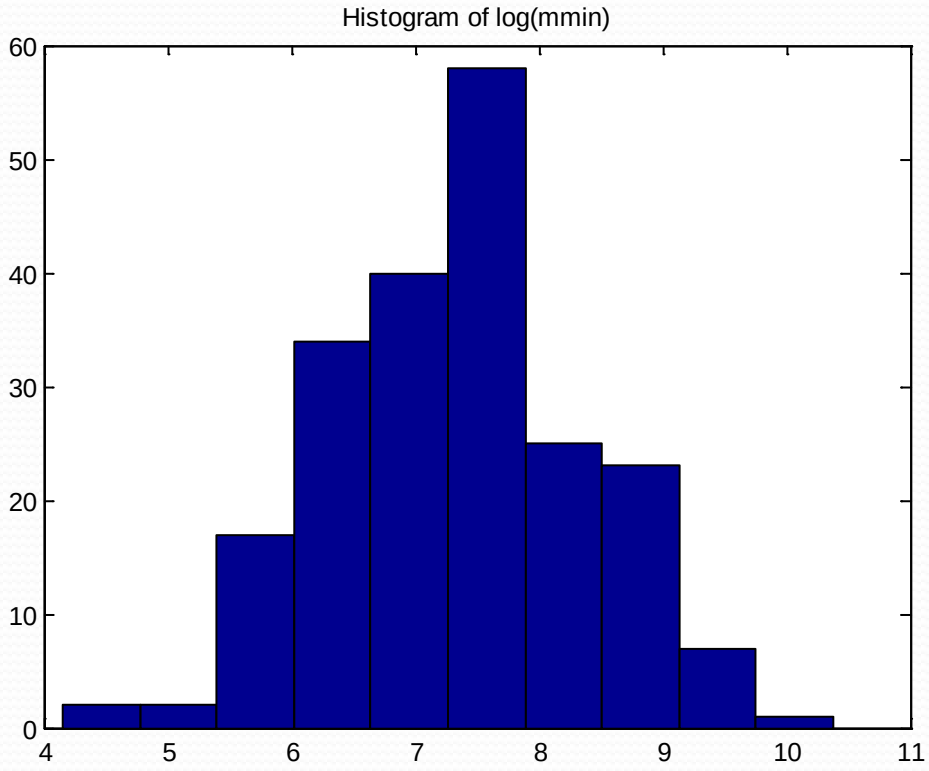

Histogram of perf
160
140
120
100
80
60
40
20
0
0
200
400
600
800
1000
1200

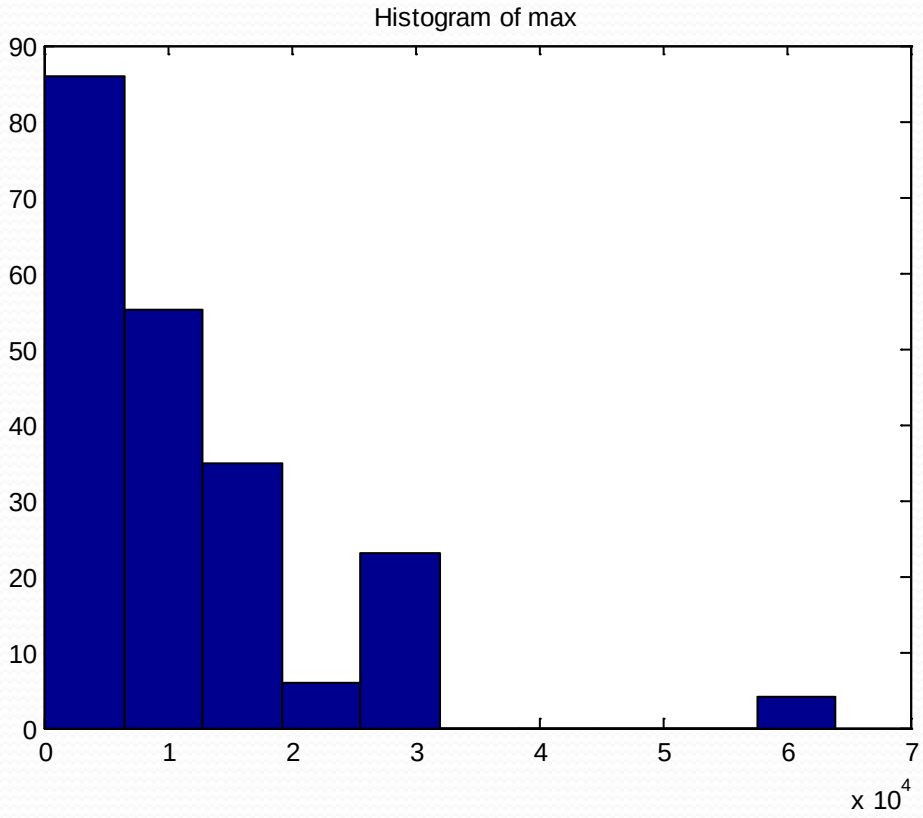

Histogram of log(perf)
50
45
40
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8

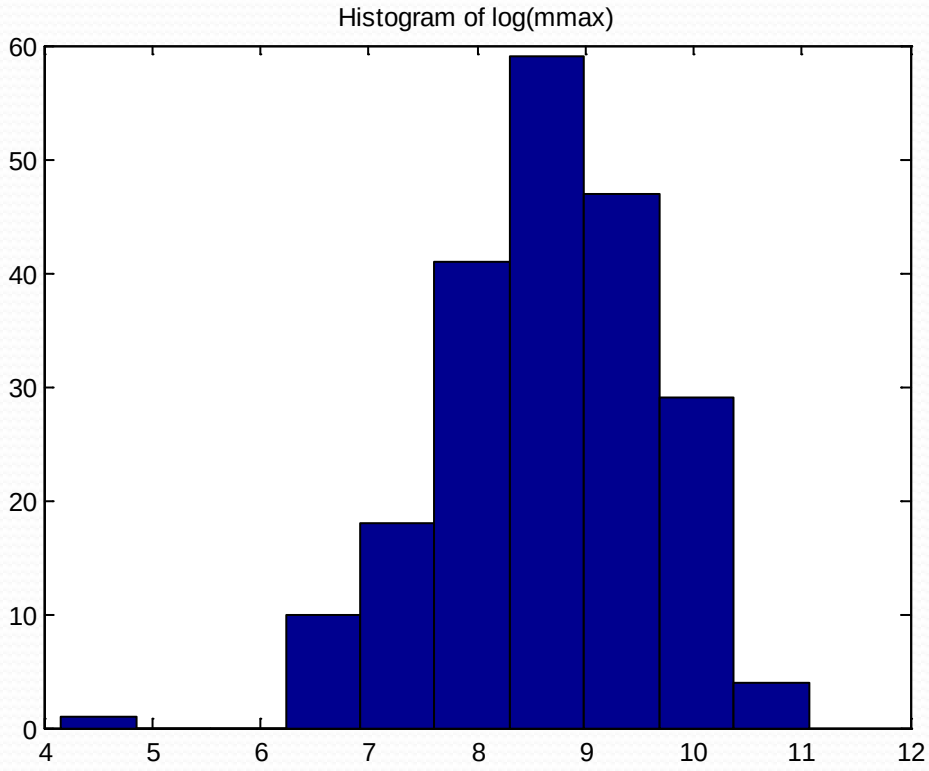

Histogram of syct
140
120
100
80
60
40
20
0
0
500
1000
1500

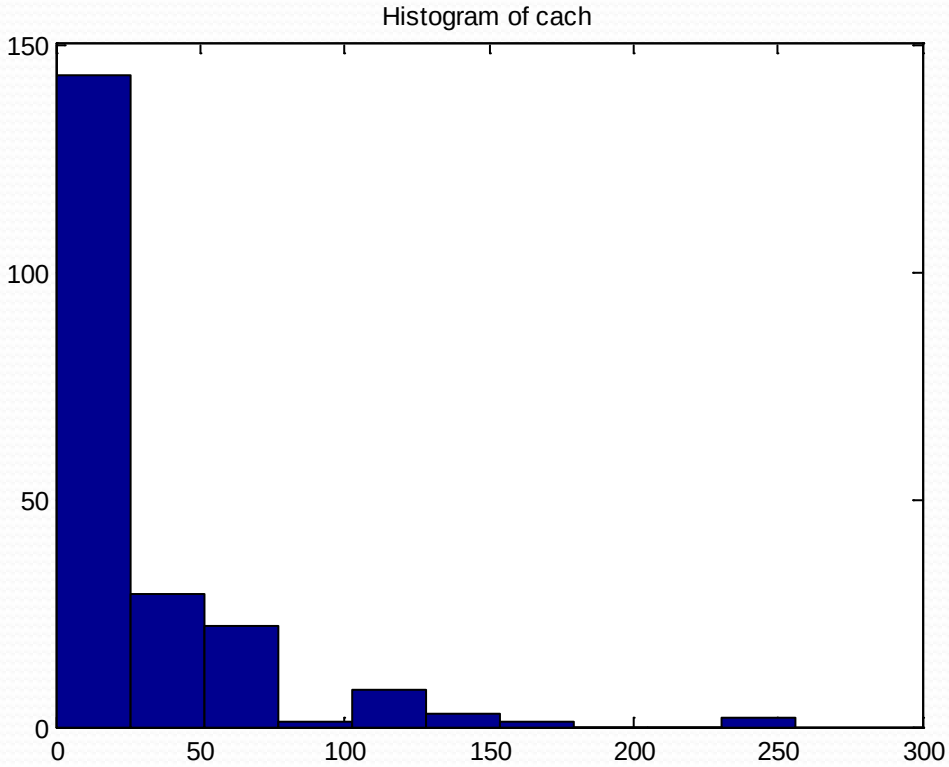

Histogram of log(syct)
45
40
35
30
25
20
15
10
5
0
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5

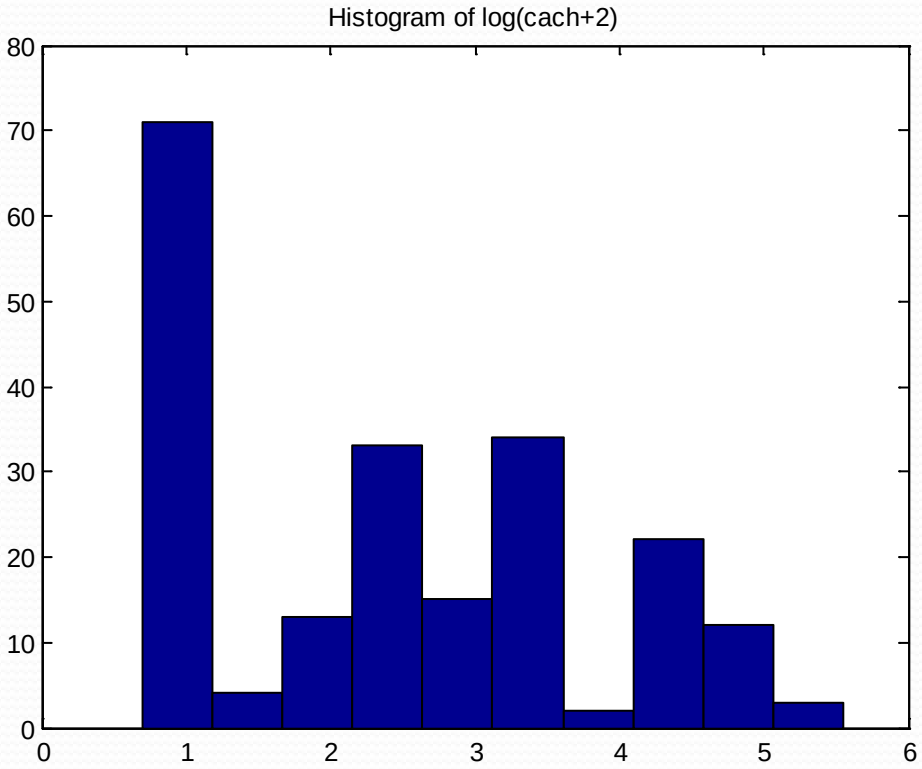

Histogram of mmin
160
140
120
100
80
60
40
20
0
0
0.5
1
1.5
2
2.5
3
3.5
x 10^4

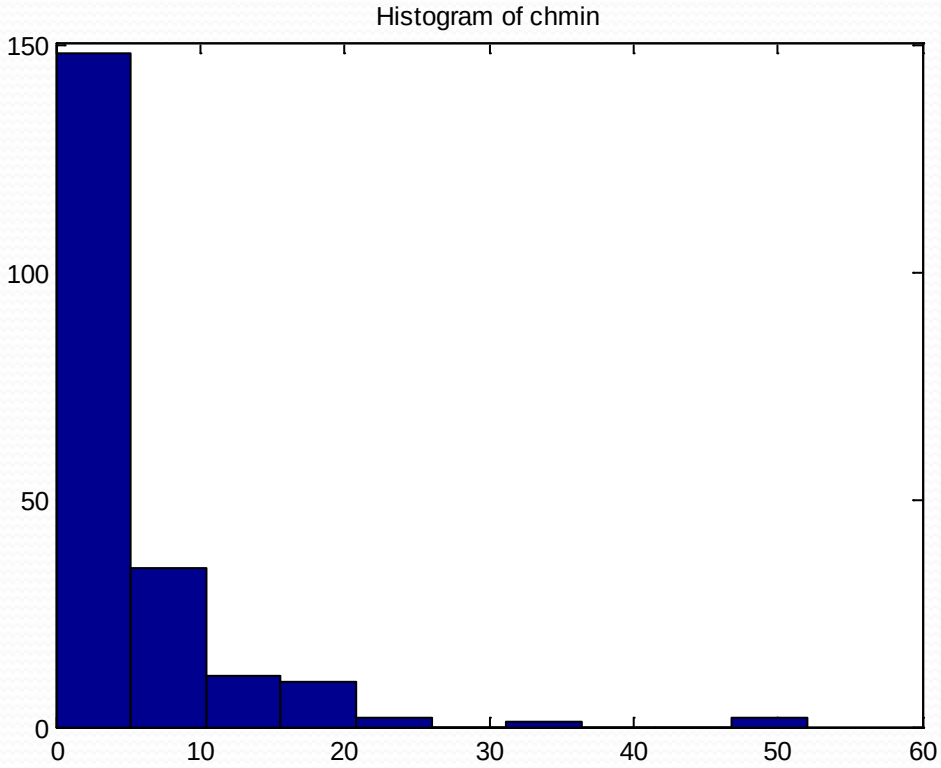

Histogram of log(mmin)
60
50
40
30
20
10
0
4
5
6
7
8
9
10
11

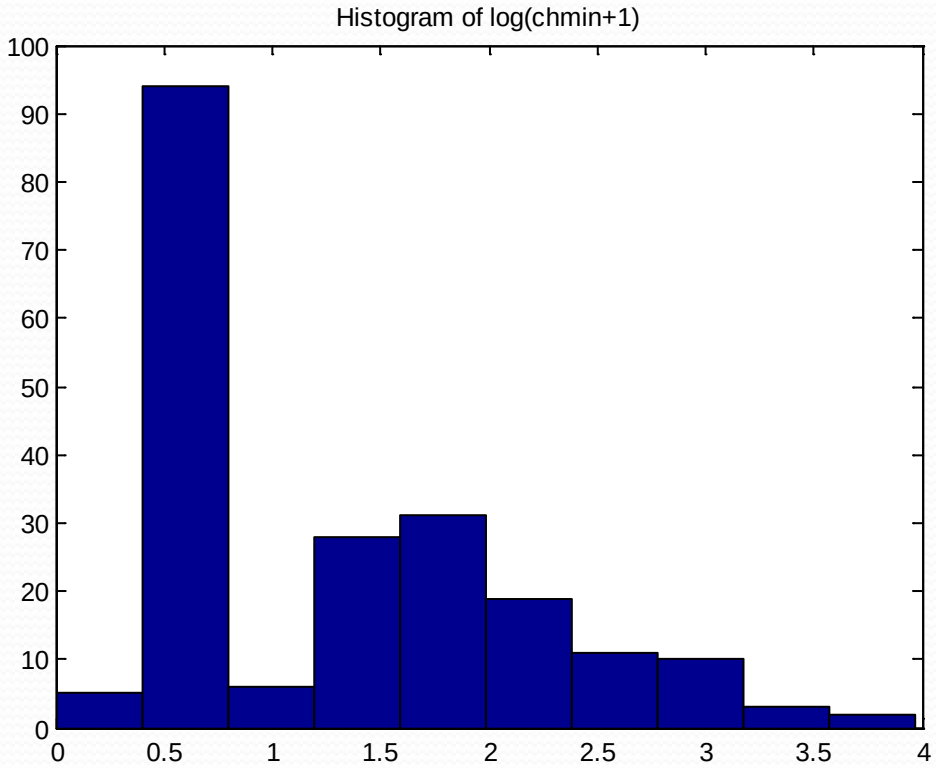

Histogram of max
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
x 10^4

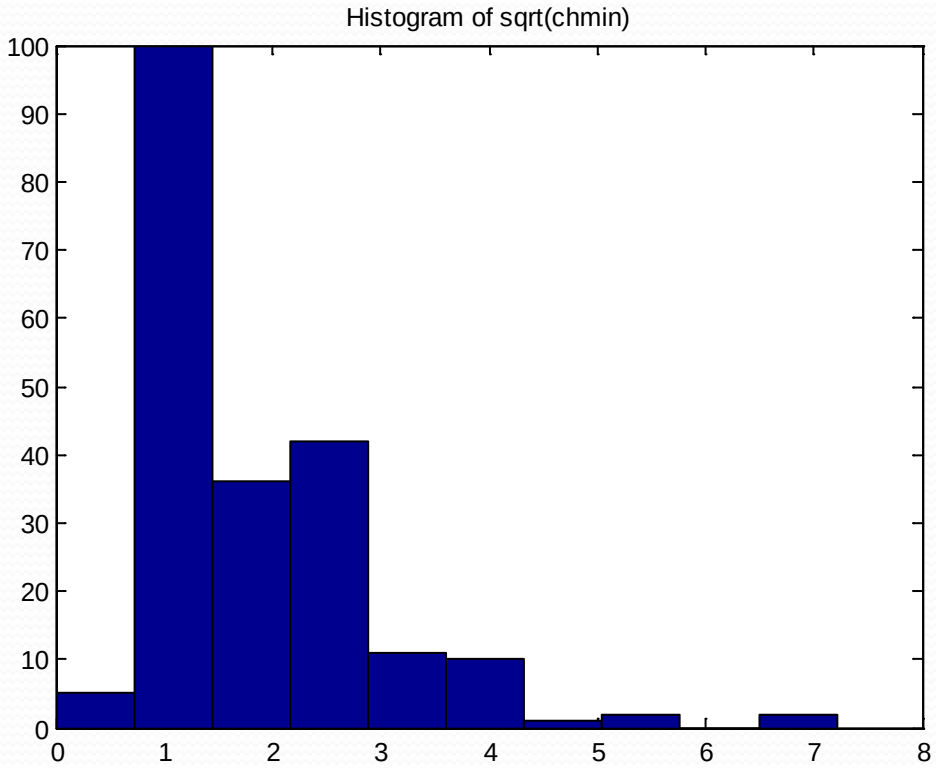

Histogram of log(mmax)
60
50
40
30
20
10
0
4
5
6
7
8
9
10
11
12

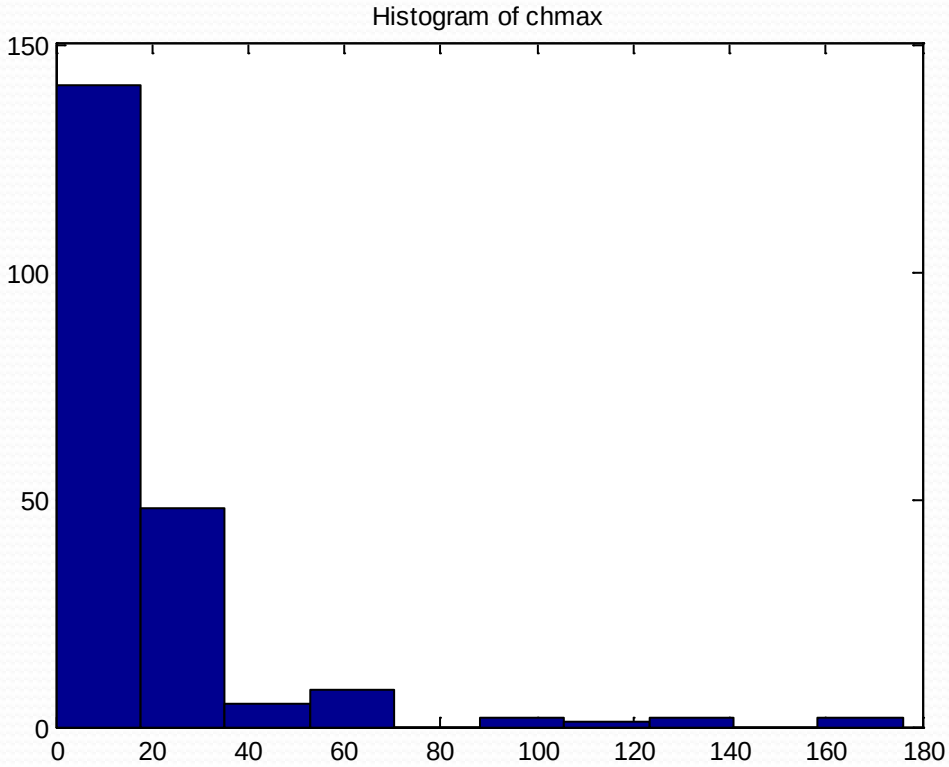

Histogram of cach
150
100
50
0
0
50
100
150
200
250
300

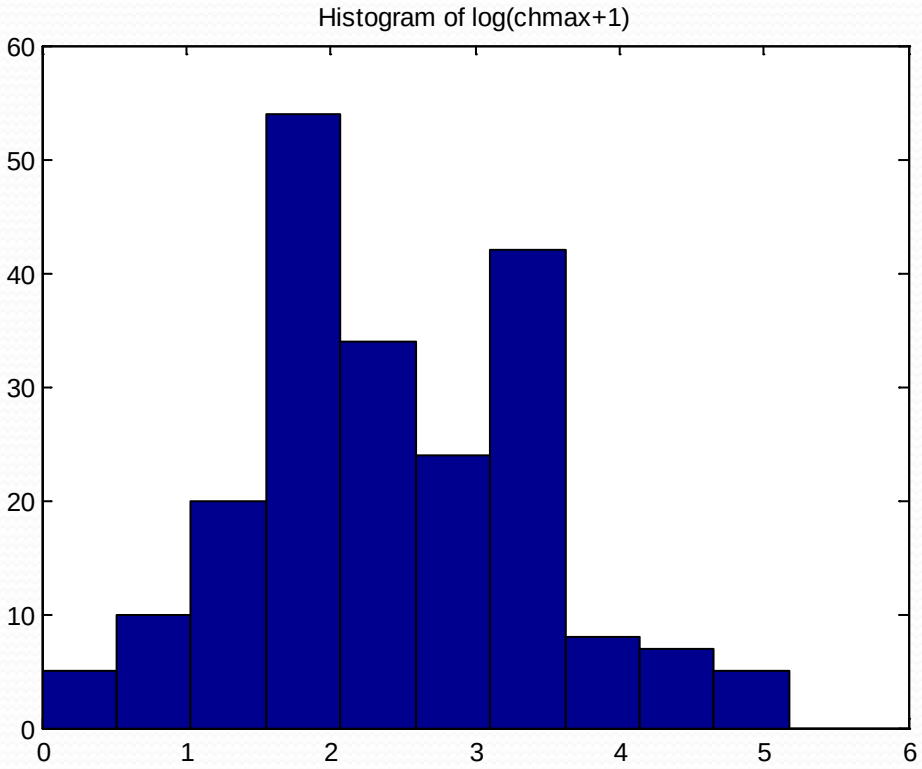

Histogram of log(cach+2)
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6


Histogram of chmin
150
100
50
0
0
10
20
30
40
50
60


Histogram of log(chmin+1)
100
90
80
70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
2.5
3
3.5
4


Histogram of sqrt(chmin)
100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8


Histogram of chmax
150
100
50
0
0
20
40
60
80
100
120
140
160
180


Histogram of log(chmax+1)
60
50
40
30
20
10
0
0
1
2
3
4
5
6

```
>> cpus2 = table(log(cpus.syct),log(cpus.mmin),log(cpus.mmax),log(cpus.cach+2),
    log(cpus.chmin+1),log(cpus.chmax+1),log(cpus.perf),
    'VariableNames',{'lsyct', 'lmmin', 'lmmax', 'lcach', 'lchmin', 'lchmax', 'lperf'} )

>> fitlm(cpus2)

Linear regression model:
    lperf ~ 1 + lsyct + lmmin + lmmax + lcach + lchmin + lchmax

Estimated Coefficients:
                   Estimate        SE        tStat        pValue
                   ________    ________    ________    __________

    (Intercept)      -1.3527     0.53856     -2.5117      0.012797
    lsyct          0.0033807    0.047601    0.071022       0.94345
    lmmin            0.18792    0.048365      3.8855    0.00013838
    lmmax            0.32143    0.047068       6.829    9.8191e-11
    lcach            0.23918    0.029901       7.999      9.59e-14
    lchmin           0.17949    0.059834      2.9999     0.0030405
    lchmax           0.13167    0.044035      2.9901     0.0031349


Number of observations: 209, Error degrees of freedom: 202
Root Mean Squared Error: 0.434
R-squared: 0.834,  Adjusted R-Squared 0.829
F-statistic vs. constant model: 169, p-value = 7.86e-76
```

```
>> stepwiselm(cpus2)
1. Adding lmmax, FStat = 359.965, pValue = 3.554256e-47
2. Adding lcach, FStat = 121.2385, pValue = 1.794327e-22
3. Adding lchmin, FStat = 49.0826, pValue = 3.45972e-11
4. Adding lmmin, FStat = 13.5988, pValue = 0.000290186
5. Adding lmmin:lmmax, FStat = 22.8081, pValue = 3.42459e-06
6. Adding lchmax, FStat = 12.0322, pValue = 0.000639034
7. Adding lchmin:lchmax, FStat = 7.2802, pValue = 0.0075648
8. Adding lmmax:lchmin, FStat = 12.0451, pValue = 0.000636092

Linear regression model:
    lperf ~ 1 + lcach + lmmin*lmmax + lmmax*lchmin + lchmin*lchmax

Estimated Coefficients:
                     Estimate        SE        tStat       pValue
                    _________    ________    _______    __________

    (Intercept)        5.6085      1.2427     4.5131    1.0891e-05
    lmmin             -1.0769     0.21943    -4.9079    1.9061e-06
    lmmax            -0.41148     0.13883    -2.9639     0.0034064
    lcach             0.20124    0.027085     7.4301    3.0882e-12
    lchmin             1.2346     0.42736      2.889     0.0042912
    lchmax          -0.079517    0.063678    -1.2487       0.21322
    lmmin:lmmax        0.14528    0.024385     5.9576    1.1365e-08
    lmmax:lchmin      -0.18178    0.052376    -3.4706    0.00063609
    lchmin:lchmax      0.17682    0.041043     4.3083    2.5782e-05


Number of observations: 209, Error degrees of freedom: 200
Root Mean Squared Error: 0.392
R-squared: 0.866,  Adjusted R-Squared 0.86
F-statistic vs. constant model: 161, p-value = 7.19e-83
```

Model Selection

- Many models will make predictions of similar quality.
- In such cases, there is no one best model, but one can still select the simplest model that seems to do a good job.
- Stepwise regression can be helpful, but the variables should first be placed on the right scale (possibly taking logs).
- If one starts with many variables, stepwise regression will usually select some even if none of the variables actually is related to the response, so caution is often called for (see pages 635–638 in the text).

Factorial Experiments

- Regression type models can work with quantitative responses (and also qualitative ones) and predictors that are either quantitative or qualitative.
- Sometimes, the predictors and the response on a given unit are observed together as when we measure characteristics of a patient and the outcome of treatment.
- Sometimes the treatments are chosen by the experimenter and then the response is observed, as when some patients get a new treatment and others standard of care.

One-Factor Experiments

- We apply two or more treatments to experimental units, choosing the treatment for each unit by using random numbers.
- There needs to be in this case at least two units with each treatment or there is no basis for comparison.
- With exactly two treatments, this can be analyzed by the two-sample t-test.
- If there are more than two treatments, we use one-way ANOVA.
- Ideally, the number of units on each treatment is the same, but this cannot always be assured. When it is, the design is called *balanced*.

Red Cell Folate Study

- Red cell folate is a measure of folic acid (vitamin B9). It can be disrupted by anesthesia with nitrous oxide (N_2O).
- This study compared operations under three conditions:
  - N_2O (50%) +O_2 (50%) for 24 hours continuously up to and including the operation.
  - N_2O (50%) +O_2 (50%) only during the operation.
  - O_2 at 30%-50% before the operation, but no N_2O before the operation.
- There were 22 patients allocated 8/9/5 to the three treatments (unbalanced).
- The MATLAB function `fitlm` will be able to tell that ventilation is a factor because it does not consist of numbers. If it does, you have to tell it which variables are categorical. You can use `nominal` to convert numbers to categories of the Name-Value pair `'CategoricalVars'` in `fitlm`.

```
>> folatelm = fitlm(folate,'folate~ventilation')

Estimated Coefficients:
                                Estimate      SE       tStat       pValue
                                ________    ______    _______    __________

    (Intercept)                  316.62     16.164     19.588    4.6492e-14
    ventilation_"N2O+O2--op"    -60.181     22.216    -2.7089       0.01392
    ventilation_"O2--24h"       -38.625     26.064    -1.4819       0.15476


Number of observations: 22, Error degrees of freedom: 19
Root Mean Squared Error: 45.7
R-squared: 0.281,  Adjusted R-Squared 0.205
F-statistic vs. constant model: 3.71, p-value = 0.0436

These coefficients are comparisons between each of the two listed
treatments and the omitted comparison level, which is "N2O+O2--24h". This
is not a test of whether the factor as a whole is important. The F-test is
a valid test of the factor as a whole and is MS(ventilation)/MS(error) as
given on the next slide in more detail.
```

```
>> anova(folatelm)

                SumSq    DF    MeanSq    F         pValue
    ventilation 15516     2    7757.9    3.7113    0.043589
    Error       39716    19    2090.3
```

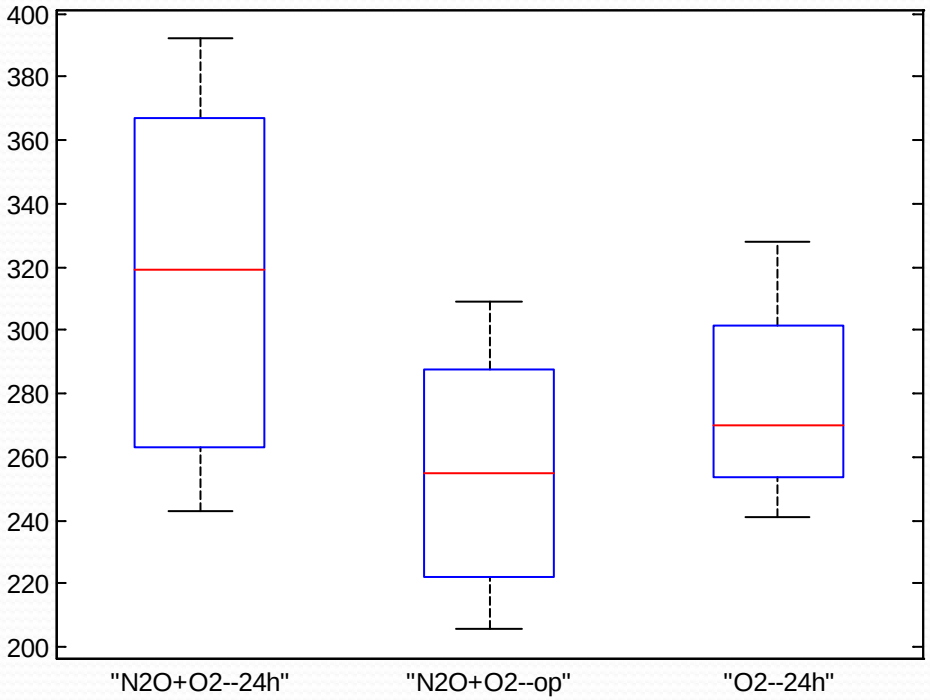
This test shows that ventilation has an effect on folate levels which is statistically significant at the 5% level. Note that the F ratio MS(ventilation)/MS(error) has df 2 and 19.

But which treatments are better?

This can be analyzed graphically with

```
>> boxplot(folate.folate,folate.ventilation)
```

And statistically with a more complex procedure for *multiple comparisons*, that is comparing each of the three procedures with the other two.


400
380
360
340
320
300
280
260
240
220
200
"N2O+O2--24h"
"N2O+O2--op"
"O2--24h"

Multiple Comparisons

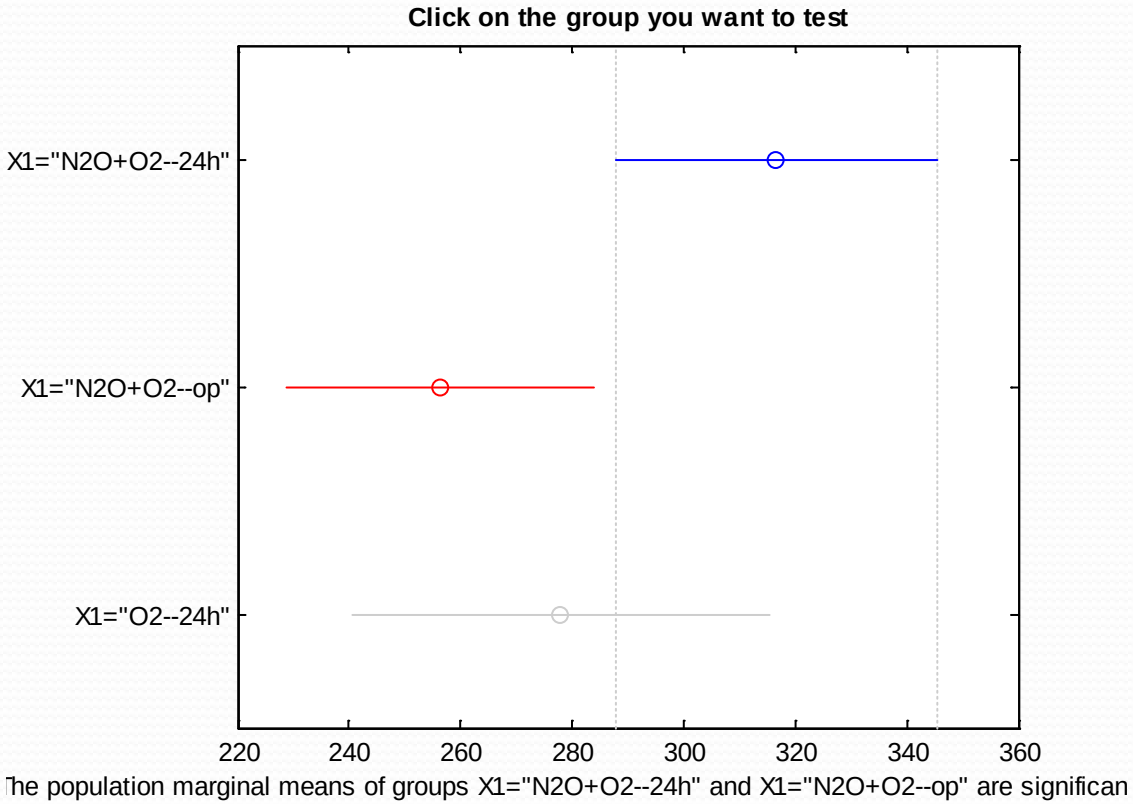
```
>> vent2 = num2cell(folate.ventilation,1)
>> [p,table,stats] = anovan(folate.folate,vent2)
p =
    0.0436
table =

    'Source'    'Sum Sq.'       'd.f.'    'Singular?'    'Mean Sq.'      'F'          'Prob>F'
    'X1'        [1.5516e+04]    [   2]    [        0]    [7.7579e+03]    [3.7113]     [0.0436]
    'Error'     [3.9716e+04]    [  19]    [        0]    [2.0903e+03]          []           []
    'Total'     [5.5232e+04]    [  21]    [        0]                []           []           []

stats =

         source: 'anovan'
          resid: [22x1 double]
         coeffs: [4x1 double]
            Rtr: [3x3 double]
      rowbasis: [3x4 double]
            dfe: 19
            mse: 2.0903e+03
    nullproject: [4x3 double]
          terms: 1
         nlevels: 3
     continuous: 0
         vmeans: 0
       termcols: [2x1 double]
     coeffnames: {4x1 cell}
           vars: [4x1 double]
       varnames: {'X1'}
       grpnames: {{3x1 cell}}
```

The `anovan` command does one or multiway ANOVA, but it needs the grouping variable to be in a cell array. The resulting ANOVA table and other statistics are the same as using `fitlm` and then `anova`, but we need the stats structure for the multiple comparisons.

```
>> multcompare(stats)

    1.0000    2.0000    3.7421   60.1806  116.6190    0.0355
    1.0000    3.0000  -27.5904   38.6250  104.8404    0.3215
    2.0000    3.0000  -86.3406  -21.5556   43.2295    0.6802
```

The first two columns show the groups being compared, the fourth column the difference, and the flanking third and fifth columns show a 95% CI adjusted for multiple comparisons. If this 95% CI does not include 0, then the groups are significantly different. In this case, this is only 1 vs. 2. The last column shows the associated p-value.

```
>> celldisp(getfield(stats,'grpnames'))

ans{1}{1} =
"N2O+O2--24h"

ans{1}{2} =
"N2O+O2--op"

ans{1}{3} =
"O2--24h"
```


Click on the group you want to test
X1="N2O+O2--24h"
X1="N2O+O2--op"
X1="O2--24h"
220
240
260
280
300
320
340
360
The population marginal means of groups X1="N2O+O2--24h" and X1="N2O+O2--op" are significan

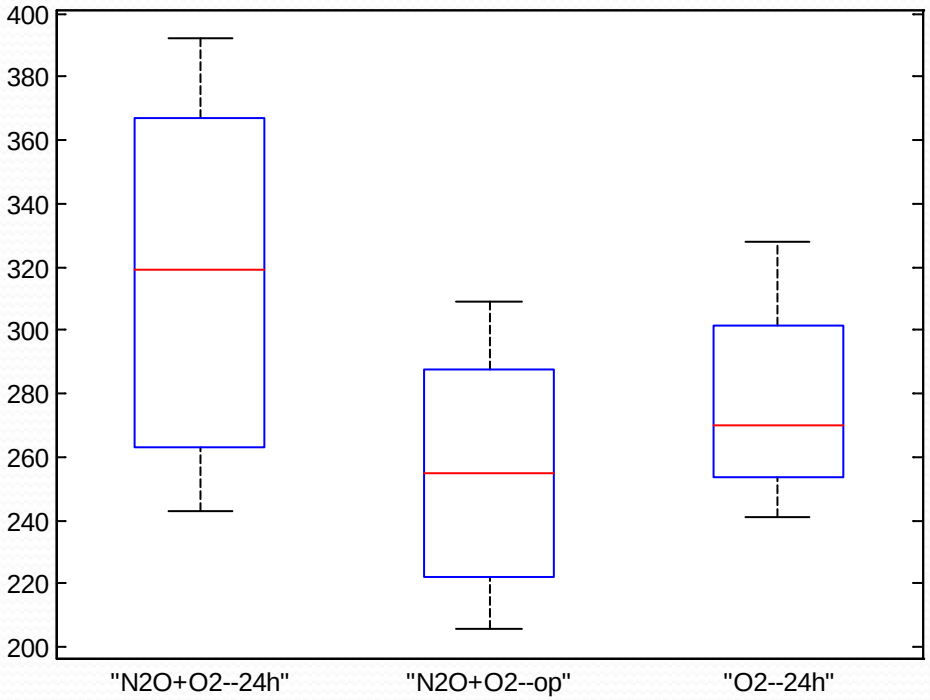
Multiple Comparisons

- `multcompare` can use different comparisons metrics.
- The default is the Tukey HSD or honest significant difference which is based on the studentized range, and attempts to declare any one or more differences significant only 5% of the time if all of the true group means are actually the same.
- An alternative is the least significant difference (lsd) which should only be used if the F-test is significant (protected lsd), but gives narrower intervals.

```
>> multcompare(stats)

ans =

    1.0000    2.0000    3.7421   60.1806  116.6190    0.0355
    1.0000    3.0000  -27.5904   38.6250  104.8404    0.3215
    2.0000    3.0000  -86.3406  -21.5556   43.2295    0.6802

>> multcompare(stats,'ctype','lsd')

ans =

    1.0000    2.0000   13.6821   60.1806  106.6791    0.0139
    1.0000    3.0000  -15.9285   38.6250   93.1785    0.1548
    2.0000    3.0000  -74.9306  -21.5556   31.8195    0.4085

The intervals are narrower, but the results are unchanged in this case.
```

Conclusions

- The form of ventilation appears to affect the folate level.
- The definitive conclusion of the study is that "N2O+O2--24h“ is better than "N2O+O2--op“, with "O2--24h“ in the middle and not definitively different from either one.
- A main assumption of ANOVA is that the groups have the same variance, and the boxplot does not strongly challenge that assumption.


400
380
360
340
320
300
280
260
240
220
200
"N2O+O2--24h"
"N2O+O2--op"
"O2--24h"

Two Factor Experiments

- In a two factor experiment, there are two sets of treatments and each experimental unit gets one treatment from each set.
- We can evaluate the effects of each factor separately, and also the interaction.

Coking Data

- This experiment is on time to coking (making coke from coal) in an experiment in which oven width and temperature were varied.
  - width = a factor with levels 4, 8, and 12 giving the oven width in inches.
  - temp = a factor with levels 1600 and 1900, giving the oven temperature in degrees Fahrenheit.
  - time = a numeric variable, time to coking
- This is a balanced two-way experiment with three replicates under each set of conditions (3x2 = 6 conditions), so n = 18.

Main Effects and Interactions

- The main effect of width is the change in time as width changes, averaged over temperatures. For example,

```
time(width = 12) – time(width = 8),
```

  which are differences of simple averages.
- The main effect of temperature is the change in time as temperature changes, averaged over widths.
- The interaction of width and time can be thought of as the change in the time between high and low temperature as width changes. For example,

```
 [time(width=12, temp=1900) – time(width=12, temp=1600)]
–[time(width=8, temp=1900) – time(width=8, temp=1600)]
```

  which are differences of differences.

```
>> cokinglm = fitlm(coking,'time~width*temp','CategoricalVars',[1,2])
Linear regression model:
    time ~ 1 + width*temp
Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	3.0667	0.30399	10.088	3.2569e-07
width_8	4.1	0.4299	9.5371	5.962e-07
width_12	7.7333	0.4299	17.989	4.7896e-10
temp_1900	-0.76667	0.4299	-1.7834	0.099819
width_8:temp_1900	-0.86667	0.60797	-1.4255	0.1795
width_12:temp_1900	-2.7	0.60797	-4.441	0.00080545

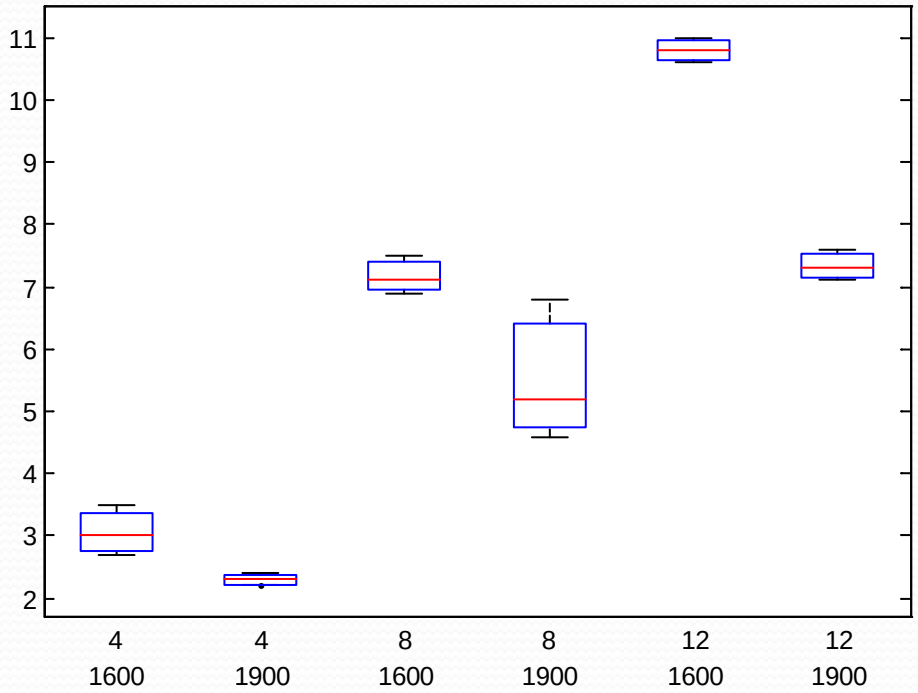
```
Number of observations: 18, Error degrees of freedom: 12
Root Mean Squared Error: 0.527
R-squared: 0.978,  Adjusted R-Squared 0.968
F-statistic vs. constant model: 105, p-value = 1.74e-09
```

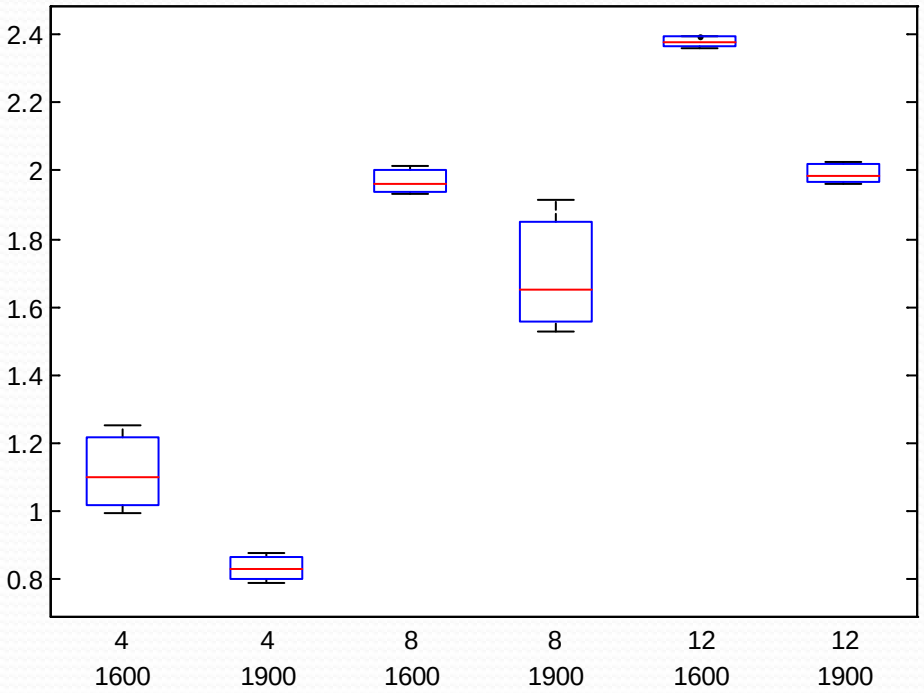
```
>> anova(cokinglm)
```

	SumSq	DF	MeanSq	F	pValue
width	123.14	2	61.572	222.1	3.3123e-10
temp	17.209	1	17.209	62.076	4.3942e-06
width:temp	5.7011	2	2.8506	10.283	0.0025036
Error	3.3267	12	0.27722		

The interaction term is significant, which means that the effect of temperature is different at different levels of width. This makes it hard to interpret the main effects of width and temperature.

```
>> boxplot(coking.time,[coking.width coking.temp])
>> boxplot(log(coking.time),[coking.width coking.temp])
```


11
10
9
8
7
6
5
4
3
2
4
1600
4
1900
8
1600
8
1900
12
1600
12
1900


2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
4
1600
4
1900
8
1600
8
1900
12
1600
12
1900

```
>> ltime = log(coking.time)
>> coking2 = [coking table(ltime)]
>> coking2.time = []

coking2 =

    width    temp     ltime
    _____    ____    _______

     4       1600     1.2528
     4       1600     1.0986
     4       1600    0.99325
    ..........................

>> coking2lm = fitlm(coking2,'ltime~width*temp','CategoricalVars',[1,2])
>> anova(coking2lm)

                SumSq      DF     MeanSq        F         pValue
               ________    __    ________    _______    __________

    width         4.6648     2      2.3324     224.99    3.0714e-10
    temp         0.44332     1     0.44332     42.764    2.7718e-05
    width:temp  0.012252     2    0.006126    0.59094       0.56915
    Error         0.1244    12    0.010367
```

On the log scale, only the main effects are significant, which makes the interpretation much easier.