EMAT30007 Applied Statistics

Lab 7: Linear Models (Multiple Linear Regression)

Nikolai Bode

This lab has two parts. In the first part, you will work through a multiple linear regression example including model selection. The second half is a group exercise, where you will work in groups on different data sets, performing model selection. Each group will then present their approach to the rest of the class.

(1) Multiple Linear Regression Example

In this part of the lab we'll work through an example in some detail. We will use one of the data sets we already encountered in the last lab, but we will consider more variables as possible predictors. As a reminder, the data was collected to investigate the energy use of appliances in low-energy buildings. The data set is from one house. Readings were taken every 10 minutes for about 4.5 months. Reference for the data:

[Candanedo, L. M., Feldheim, V., Deramaix, D. (2017) "Data driven prediction models of energy use of appliances in a low-energy house". Energy and Buildings, Volume 140, 1 April 2017, Pages 81-97]

You can find the data in the file 'APPLIANCES2.csv'.

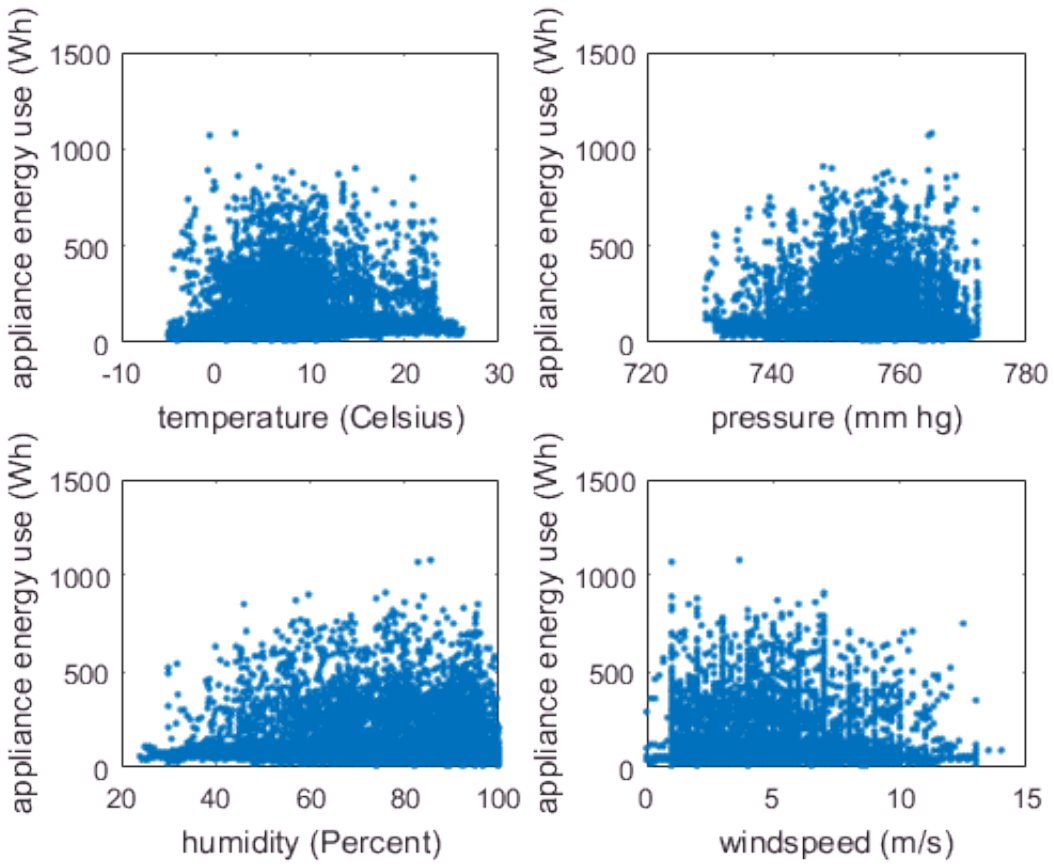
In this file, the response 'Appliances' is included alongside four variables that have been measured independently of the energy usage of the appliances (outside temperature, pressure outside, humidity outside, windspeed outside). The task is to establish which out of these variables (if any) help to predict the energy usage of the appliances in the house.

1.1 Preliminary analysis

It is always a good idea to have a look at the raw data to see if this shows some trends. Read in the data and plot scatterplots of the possible predictors against the energy usage of appliances.

```
%% read data:
delimiterIn = ',';
A = importdata('APPLIANCES2.csv',delimiterIn);

%% scatterplots of data
appliances = A.data(:,1);
temperature = A.data(:,2);
pressure = A.data(:,3);
humidity = A.data(:,4);
windspeed = A.data(:,5);

clf
subplot(2,2,1)
plot(temperature,appliances,'.')
```

```
xlabel('temperature (Celsius)')
ylabel('appliance energy use (Wh)')

subplot(2,2,2)
plot(pressure,appliances,'.')
xlabel('pressure (mm hg)')
ylabel('appliance energy use (Wh)')

subplot(2,2,3)
plot(humidity,appliances,'.')
xlabel('humidity (Percent)')
ylabel('appliance energy use (Wh)')

subplot(2,2,4)
plot(windspeed,appliances,'.')
xlabel('windspeed (m/s)')
ylabel('appliance energy use (Wh)')
```



With this many data points, it can be hard to tell what's going on in plots of raw data. So we proceed to statistical analysis using a multiple linear regression model.

1.2. Multiple linear regression

When performing statistical model fitting on data that includes more than one possible predictor, there are different approaches. We could either start with minimal models that only consider one predictor at a

time or we could fit a full model that includes all available predictors. Either approach is valid and which one to use typically depends on the type of analysis. For an exploratory analysis as we perform it here, it is often convenient to start with a full model.

Fit a full model to the data. The code in Matlab for multiple linear regression is a straighforward extension of what we do for simple linear regression. In `fitlm()`, simply add more predictors to the model, separating them with a '+'.

```
% create a table for the data:
data = table(temperature,pressure,humidity,windspeed,appliances,'VariableNames',...
    {'temperature','pressure','humidity','windspeed','appliances'});
% show first few rows of table
data(1:5,:)
```

```
ans =
    temperature    pressure    humidity    windspeed    appliances
    ___________    ________    ________    _________    __________

     6.6           733.5       92              7       60
    6.48           733.6       92          6.6667      60
    6.37           733.7       92          6.3333      50
    6.25           733.8       92               6      50
    6.13           733.9       92          5.6667      60
```

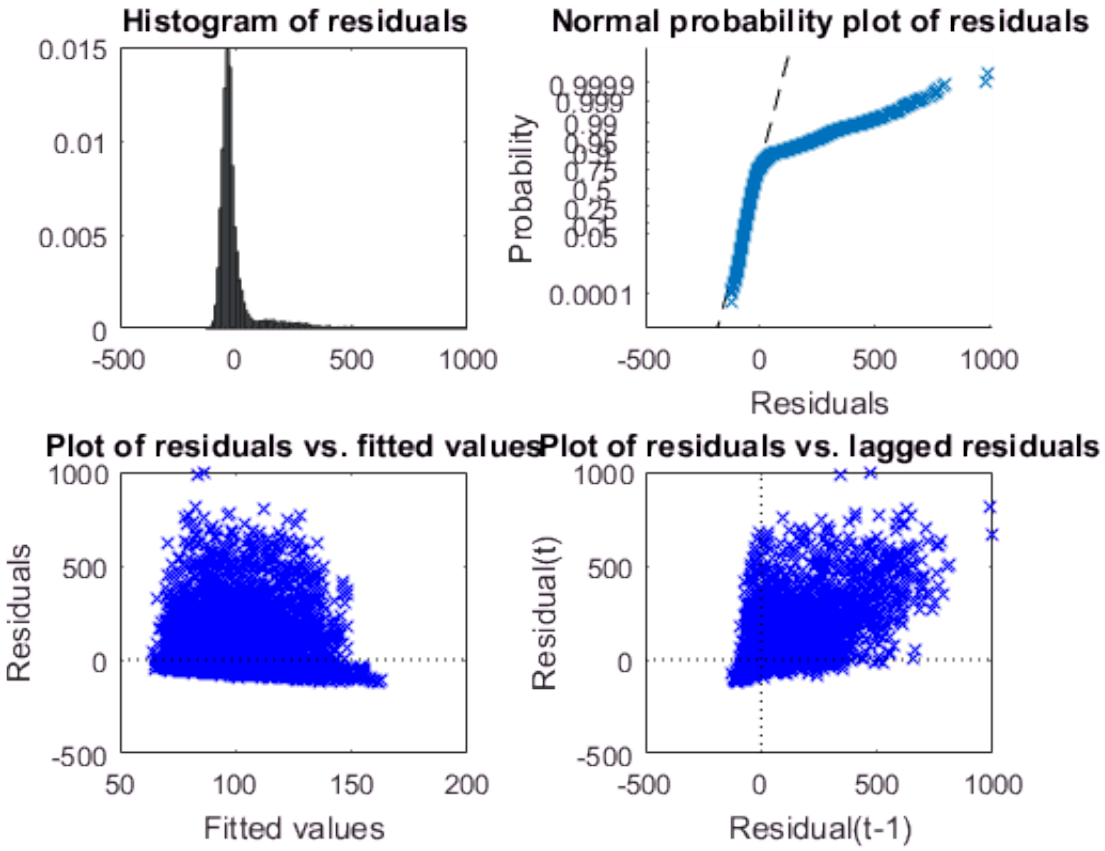
```
% start with a full model that includes all predictors:
m1 = fitlm(data, 'appliances~temperature+pressure+humidity+windspeed')
```

```
m1 =

Linear regression model:
    appliances ~ 1 + temperature + pressure + humidity + windspeed

Estimated Coefficients:
                    Estimate        SE         tStat        pValue
                   __________    ________    _________    __________

    (Intercept)        545.65      80.196        6.804    1.0468e-11
    temperature    -0.0052315     0.17051    -0.030681       0.97552
    pressure         -0.49835     0.10358      -4.8114    1.5102e-06
    humidity          -1.0074    0.060943       -16.53    5.7247e-61
    windspeed           2.212     0.30902        7.158    8.4738e-13


Number of observations: 19735, Error degrees of freedom: 19730
Root Mean Squared Error: 101
R-squared: 0.0281,  Adjusted R-Squared 0.0279
F-statistic vs. constant model: 143, p-value = 1.56e-120
```

This model fit tells us quite a few things already. Consider r-squared. This is quite low (you should find `R-squared=0.0281`), suggesting that only a small proportion of the variance in the responce (appliance energy usage) is explained by the model. However, the F-test suggests that our model is better than a constant model (i.e. one that only includes the intercept, `p-value=1.56e-120`). The t-tests are specific to the parameters linked with the different variables. Before we consider these in more detail, look at the residual plots for this model fit (use the same plots as in lab 6).

```
% Check residual plots for model m1:
```

```
% distribution of residuals:
clf
subplot(2,2,1)
plotResiduals(m1)
% Q-Q plot to check normality
subplot(2,2,2)
plotResiduals(m1,'probability')
% residuals versus fitted values
subplot(2,2,3)
plotResiduals(m1,'fitted')
% auto-correlation (via lagged residuals)
subplot(2,2,4)
plotResiduals(m1,'lagged')
```




```
```

These plots look pretty catastrophic - very similar to what we saw in lab 6. Really, at this stage, we should reconsider fundamental aspects of our modelling approach or seek additional variables that might explain the features we see in the data.

However, to illustrate model selection, we will continue with this type of model for the data.

1.3. Model selection

Have another look at the table produced by the model fit above. Remember that the t-tests investigate the null hypothesis that the parameter associated with a variable in the regression model equals zero.

For temperature it looks like we cannot reject the null hypothesis (interesting, considering that in lab 6 temperature had an effect...!!!).

So let's fit a new model to the data that does not include the predictor temperature.

```
% reduced model excluding temperature
m2 = fitlm(data, 'appliances~pressure+humidity+windspeed')
```

```
m2 =

Linear regression model:
    appliances ~ 1 + pressure + humidity + windspeed

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)         545      77.327      7.048    1.8753e-12
    pressure       -0.49765     0.10103    -4.9255     8.481e-07
    humidity        -1.0063    0.049543    -20.312     8.552e-91
    windspeed        2.2114     0.30854     7.1675    7.9072e-13


Number of observations: 19735, Error degrees of freedom: 19731
Root Mean Squared Error: 101
R-squared: 0.0281,  Adjusted R-Squared 0.028
F-statistic vs. constant model: 190, p-value = 1.04e-121
```

We can now assess in two ways whether this model is an improvement compared to the full model we fitted earlier.

First, we can compare quality measures like the AIC or BIC for the models (remember, lower values are better and also recall my dislike of r-squared for model comparison!). You can find quality measures for a fitted model 'm' in Matlab using the command 'm.ModelCriterion'.

```
m1.ModelCriterion
```

```
ans =
     AIC: 2.3820e+05
    AICc: 2.3820e+05
     BIC: 2.3824e+05
    CAIC: 2.3825e+05
```

```
m2.ModelCriterion
```

```
ans =
     AIC: 2.3820e+05
    AICc: 2.3820e+05
     BIC: 2.3823e+05
    CAIC: 2.3824e+05
```

The AIC and certainly the BIC should be somewhat smaller for the reduced model we fitted.

Second, we can use a Likelihood-ratio test, to formally test the null hypothesis that the parameter associated with temperature is equal to zero (notice how this tests something similar to the t-test, but it requires another model to be fitted).

In Matlab, the command is `lratiotest(logL_full, logL_reduced, df)`, where `logL_full` and `logL_reduced` are the log-likelihood for the full and reduced model, respectively (access these for a model `m` with the command `m.LogLikelihood`). 'df' are the degrees of freedom of the test. In practice this is the difference in the number of parameters between the full and reduced model (i.e. here 'df=1'). Perform the Likelihood-ratio test, making sure you output a p-value (check Matlab help pages).

```
[h pvalue] = lratiotest(m1.LogLikelihood,m2.LogLikelihood,1)
```

```
h =
     0

pvalue = 0.9755
```

You should find a large p-value (`p-value=0.9755`), suggesting that we cannot reject the null hypothesis and thus that temperature does not need to be included in our model.

Now repeat this analysis using the Likelihood-ratio test for the variable 'pressure'. You should find a low p-value and reject the null hypothesis.

Another important aspect of the data to check is whether any of the predictors are correlated. If two predictors are highly correlated, then it may not make sense to include them as separate predictors into a model. In addition, high correlations between predictors can lead to issues with model fitting and with performing hypothesis tests on model fits (see next lecture).

Find a function in Matlab that allows you to compute the correlation between the variables included in the model.

```
% e.g. use corrcoef():
corrcoef(pressure,humidity)
```

```
ans = 2x2 double

    1.0000   -0.0920
   -0.0920    1.0000
```

```
corrcoef(pressure,temperature)
```

```
ans = 2x2 double

    1.0000   -0.1433
   -0.1433    1.0000
```

```
corrcoef(pressure,windspeed)
```

```
ans = 2x2 double

    1.0000   -0.2350
   -0.2350    1.0000
```

```
corrcoef(temperature,humidity)
```

```
ans = 2x2 double

    1.0000   -0.5742
   -0.5742    1.0000
```

```
corrcoef(temperature,windspeed)
```

```
ans = 2x2 double

    1.0000    0.1929
    0.1929    1.0000
```

```
corrcoef(humidity,windspeed)
```

```
ans = 2x2 double

    1.0000   -0.1765
   -0.1765    1.0000
```

```

```

Apart from a slightly negative correlation between temperature and humidity, you shouldn't find any issues.

1.4. Fit multiple linear regression model using matrices

Recall the matrix notation for Linear model from lectures. We also saw a convenient way for estimating the parameters of a Linear models that used this matrix notation. Use this approach to recover the parameter estimates of the reduced model you fitted above.

```
% response:
Y = appliances;
% define the matrix X. Don't forget about the intercept.
X = [ones(length(appliances),1) pressure humidity windspeed];
betahat = inv(X'*X)*(X'*Y);
betahat
```

```
betahat = 4x1 double

  544.9981
   -0.4976
   -1.0063
    2.2114
```

```
% this is the same as the Matlab finds using 'fitlm()'.
```

(2) Group exercise

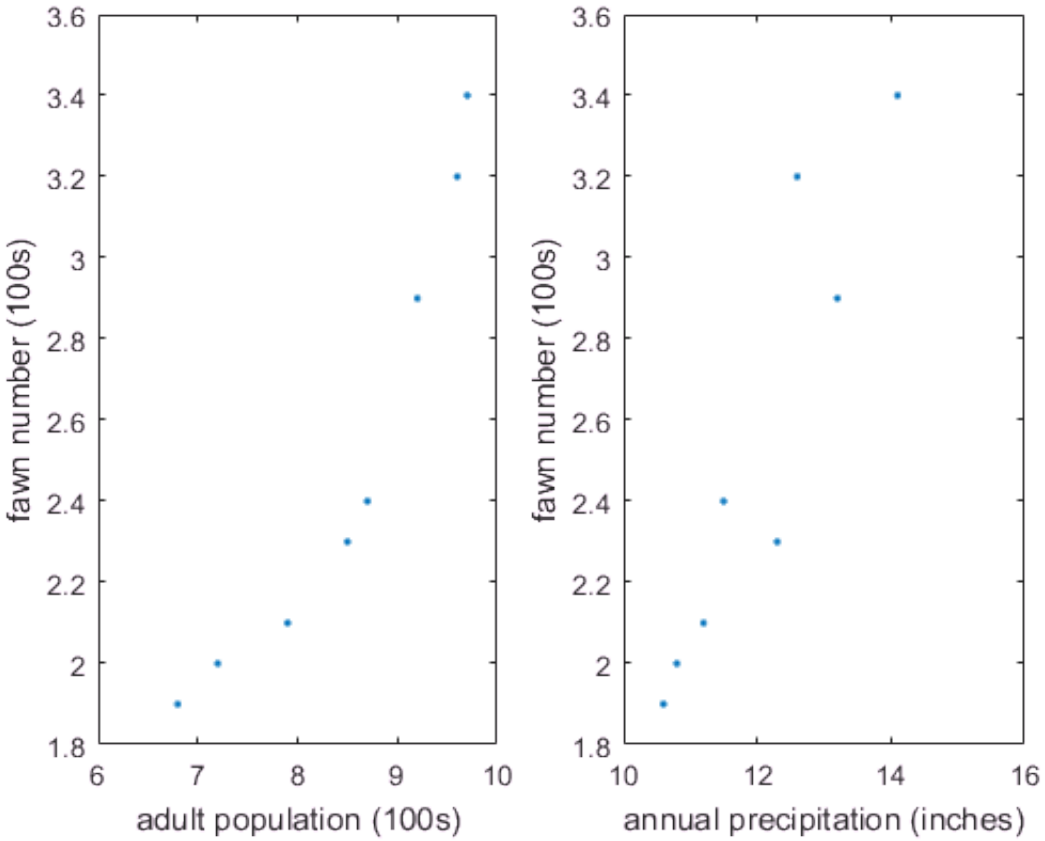
The second part of the lab is a greoup excercise. Each group will be assigned one of the following four data sets. Your task is to perform an exploratory analysis of this data and to use model selection to formulate a Linear Model for the data that only includes relevant parameters.

The datasets are illustrative and have been created for this exercise based on data sets published online (http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/mlr/frames/frame.html.)

Antelope data

The data set 'anteolpe.csv' is based on a data recording the population development of antelopes in the Thunder Basin Grasslands in the USA. The data set includes the fawn count (in hundreds), the adult population (in hundreds) and information on the annual precipitation (inches). Your task is to investigate how the fawn count depends on the other two variables.

```
%% Antelope data:

% In this data, the two predictors are correlated. This means we have to be very
% careful when conducting hypothesis tests on the data (will be covered in lecture 8).
% When the model is fitted with both predictors present, neither t-test for the
% predictors produces a low p-value. However, when models are fitted separately with
% only one of the predictors at a time, we find low p-values in t-tests for both
% predictors. In model selection, we could fit separate models for the two predictors
% and then assess which produces a higher r-squared or AIC. Due to the high
% correlation between predictors, fitting a model with both predictors included is
% not a good idea, unless we only intend to use the model for prediction (see lecture 8).

% Here is one way of looking at the data:

delimiterIn = ',';
A = importdata('antelopes.csv',delimiterIn);

%% plot scatterplots of data
clf
fawns = A.data(:,1); % fawn data
adults = A.data(:,2); % adult data
precipitation = A.data(:,3); % precipitation data

subplot(1,2,1)
plot(adults,fawns,'.')
xlabel('adult population (100s)') % x-axis label
ylabel('fawn number (100s)') % y-axis label

subplot(1,2,2)
plot(precipitation,fawns,'.')
xlabel('annual precipitation (inches)') % x-axis label
ylabel('fawn number (100s)') % y-axis label
```




```
% check correlation between predictors
corrcoef(adults,precipitation)
```

```
ans = 2x2 double

    1.0000    0.9026
    0.9026    1.0000
```

```
% create a table for the data:
data = table(adults,precipitation,fawns,'VariableNames',...
    {'adults','precipitation','fawns'});
% show first few rows of table
data(1:5,:)
```

```
ans =
    adults    precipitation    fawns
    ______    _____________    _____

    9.2       13.2             2.9
    8.7       11.5             2.4
    7.2       10.8               2
    8.5       12.3             2.3
    9.6       12.6             3.2
```

```
% model with both predictors included
m1 = fitlm(data, 'fawns~adults+precipitation')
```

```
m1 =

Linear regression model:
    fawns ~ 1 + adults + precipitation

Estimated Coefficients:
                    Estimate      SE        tStat      pValue
                    ________    _______    _______    ________

    (Intercept)      -2.3155     0.75948    -3.0487    0.028465
    adults           0.29992     0.16241     1.8467     0.12407
    precipitation    0.19158     0.14211     1.3481     0.23548


Number of observations: 8, Error degrees of freedom: 5
Root Mean Squared Error: 0.199
R-squared: 0.913,  Adjusted R-Squared 0.878
F-statistic vs. constant model: 26.2, p-value = 0.00223
```

```
% F-test indicates that icluding predictors is a good idea!

% models including one predictor at a time
m2 = fitlm(data, 'fawns~adults')
```

```
m2 =

Linear regression model:
    fawns ~ 1 + adults

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)    -1.6791      0.63422    -2.6476      0.038152
    adults         0.49753     0.074529     6.6757    0.00054714


Number of observations: 8, Error degrees of freedom: 6
Root Mean Squared Error: 0.212
R-squared: 0.881,  Adjusted R-Squared 0.862
F-statistic vs. constant model: 44.6, p-value = 0.000547
```

```
m3 = fitlm(data, 'fawns~precipitation')
```

```
m3 =

Linear regression model:
    fawns ~ 1 + precipitation

Estimated Coefficients:
                    Estimate       SE        tStat      pValue
                    ________    ________    _______    _________

    (Intercept)      -2.6325     0.87591    -3.0054      0.02384
    precipitation    0.42845    0.072436     5.9149    0.0010394


Number of observations: 8, Error degrees of freedom: 6
Root Mean Squared Error: 0.236
R-squared: 0.854,  Adjusted R-Squared 0.829
F-statistic vs. constant model: 35, p-value = 0.00104
```

```
% comparing r-squared suggests m2 is better.
```

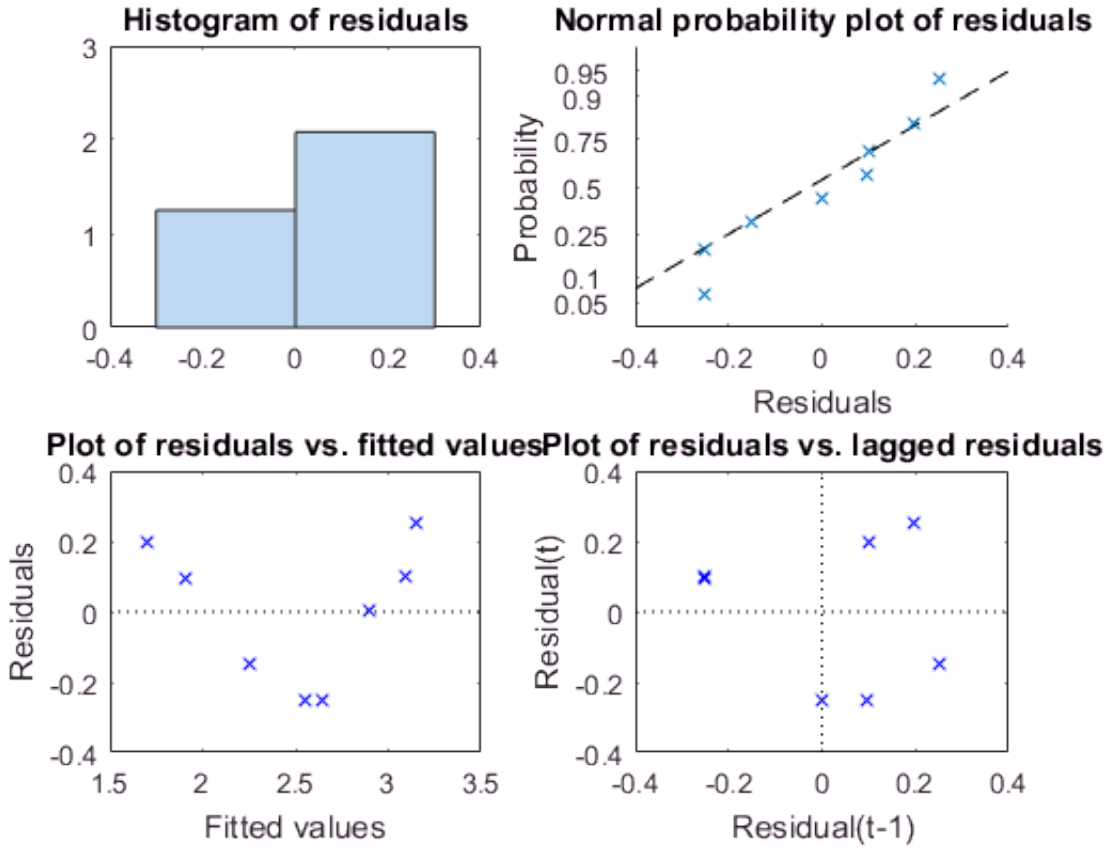
```
% let's look at AICs, to be sure:
m2.ModelCriterion
```

```
ans =
     AIC: -0.4086
    AICc: 1.9914
     BIC: -0.2498
    CAIC: 1.7502
```

```
m3.ModelCriterion
```

```
ans =
     AIC: 1.2714
    AICc: 3.6714
     BIC: 1.4303
    CAIC: 3.4303
```

```
% ... AIC is also lower for m2.

% Check residual plots for m2:
% distribution of residuals
clf
subplot(2,2,1)
plotResiduals(m2)
% Q-Q plot to check normality
subplot(2,2,2)
plotResiduals(m2,'probability')
% residuals versus fitted values (check for homoscedasticity, or worse, trends in
% the mean of residuals!)
subplot(2,2,3)
plotResiduals(m2,'fitted')
% auto-correlation (via lagged residuals)
subplot(2,2,4)
plotResiduals(m2,'lagged') % want no trend in this!
```



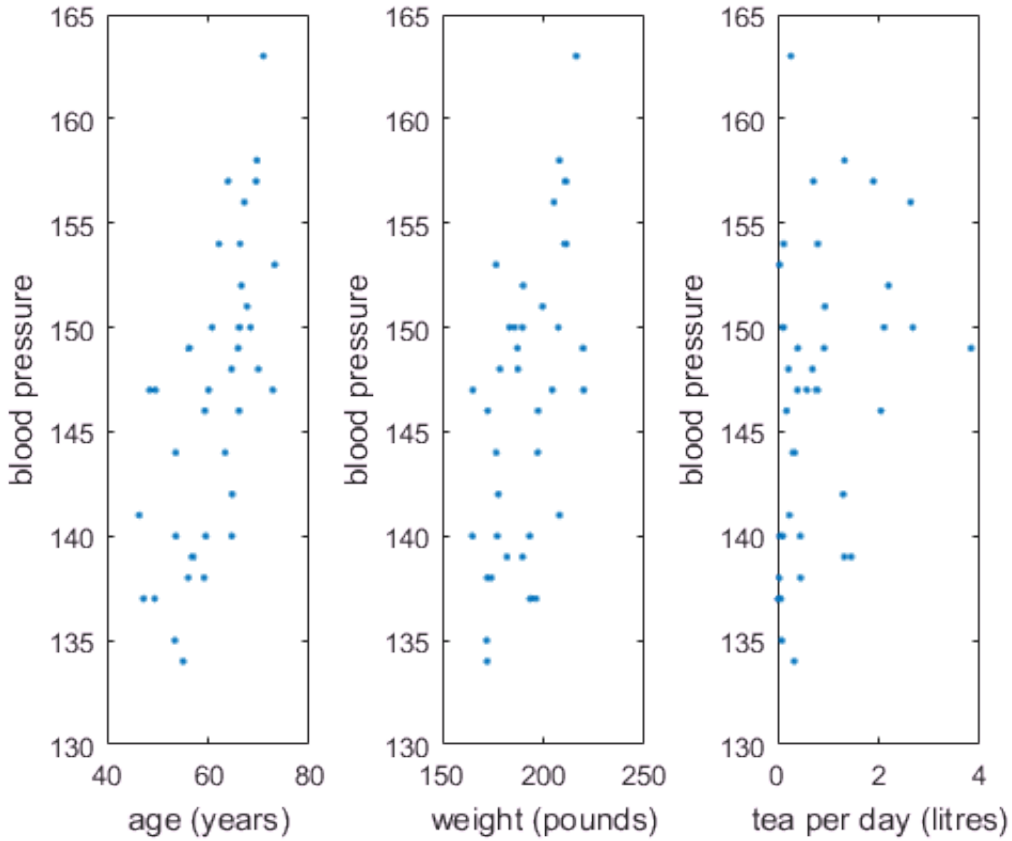

```
% Plot of residuals versus fitted values looks a bit odd - there is a trend
% suggesting a linear model fit is not idea (compare to scatterplot - any
% ideas? See e.g. lecture 8).
```

Blood pressure data

This data ('blood_pressure.csv') looks at how peoples' systolic blood pressure depends on their age, weight and tea consumption (litres per day). Investigate whether age, weight or tea consumption have an effect on blood pressure. Please do not adjust your lifestyle based on this data, it is for illustration only ;-) .

```
%% Blood pressure data:

% This is a nice example where some predictors should and other should not
% be included in a statistical model.

delimiterIn = ',';
A = importdata('blood_pressure.csv',delimiterIn);

%% plot scatterplots of data
clf
pressure = A.data(:,1); % blood pressure data
age = A.data(:,2); % age data
weight = A.data(:,3); % weight data
```

```
tea = A.data(:,4); % litres of tea per day

subplot(1,3,1)
plot(age,pressure,'.')
xlabel('age (years)') % x-axis label
ylabel('blood pressure') % y-axis label

subplot(1,3,2)
plot(weight,pressure,'.')
xlabel('weight (pounds)') % x-axis label
ylabel('blood pressure') % y-axis label

subplot(1,3,3)
plot(tea,pressure,'.')
xlabel('tea per day (litres)') % x-axis label
ylabel('blood pressure') % y-axis label
```




```
% It looks like there could be a trend for age and weight. Tea consumption is
% inconclusive.

% check correlation between predictors
corrcoef(age, weight)
```

```
ans = 2x2 double

    1.0000   -0.2246
   -0.2246    1.0000
```

```
corrcoef(age, tea)
```

```
ans = 2x2 double

    1.0000    0.3338
    0.3338    1.0000
```

```
corrcoef(weight, tea)
```

```
ans = 2x2 double

    1.0000    0.0769
    0.0769    1.0000
```

```
% there is no indication of any strong correlations.

% create a table for the data:
data = table(age,weight,tea,pressure,'VariableNames',...
    {'age','weight','tea','bloodpressure'});
% show first few rows of table
data(1:5,:)
```

```
ans =
     age      weight      tea       bloodpressure
    ______    ______    ________    _____________

    65.928    187.21      3.8248    149
    56.327    219.52     0.92495    149
    49.489    193.52    0.014065    137
    67.687    199.64     0.93967    151
    73.157    176.71    0.046183    153
```

```
% start with a full model that includes all predictors:
m1 = fitlm(data, 'bloodpressure~age+weight+tea')
```

```
m1 =

Linear regression model:
    bloodpressure ~ 1 + age + weight + tea

Estimated Coefficients:
                   Estimate       SE        tStat        pValue
                   ________    ________    ________    __________

    (Intercept)     38.525       4.3667      8.8224    1.5802e-10
    age            0.79041     0.039222      20.152    3.4049e-21
    weight          0.3097      0.01627      19.035    2.2336e-20
    tea            -0.0934      0.31287    -0.29853       0.76702


Number of observations: 40, Error degrees of freedom: 36
Root Mean Squared Error: 1.63
R-squared: 0.95,  Adjusted R-Squared 0.946
F-statistic vs. constant model: 227, p-value = 1.95e-23
```

```
% F-test suggests that the model is better than a constant model.
% T-tests suggest effects of age and weight are non-zero.

% Use a Likelihood-ratio test to investigate if tea should be included
% in the model:
% fit a reduced model:
```

```
m2 = fitlm(data, 'bloodpressure~age+weight')
```

```
m2 =

Linear regression model:
    bloodpressure ~ 1 + age + weight

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     38.861       4.1669    9.3261    2.9665e-11
    age            0.78618      0.03612    21.766    1.0966e-22
    weight          0.3089     0.015847    19.492    4.7232e-21


Number of observations: 40, Error degrees of freedom: 37
Root Mean Squared Error: 1.61
R-squared: 0.95,  Adjusted R-Squared 0.947
F-statistic vs. constant model: 349, p-value = 9.59e-25
```

```
% Likelihood-ratio test:
[h pvalue] = lratiotest(m1.LogLikelihood,m2.LogLikelihood,1)
```

```
h =
     0

pvalue = 0.7532
```

```
% ... pvalue is large, so we cannot reject the null hypothesis that the parameter
% associated with tea consumption is zero.

% AICs suggest the same (AIC for m2 is lower)
m1.ModelCriterion
```

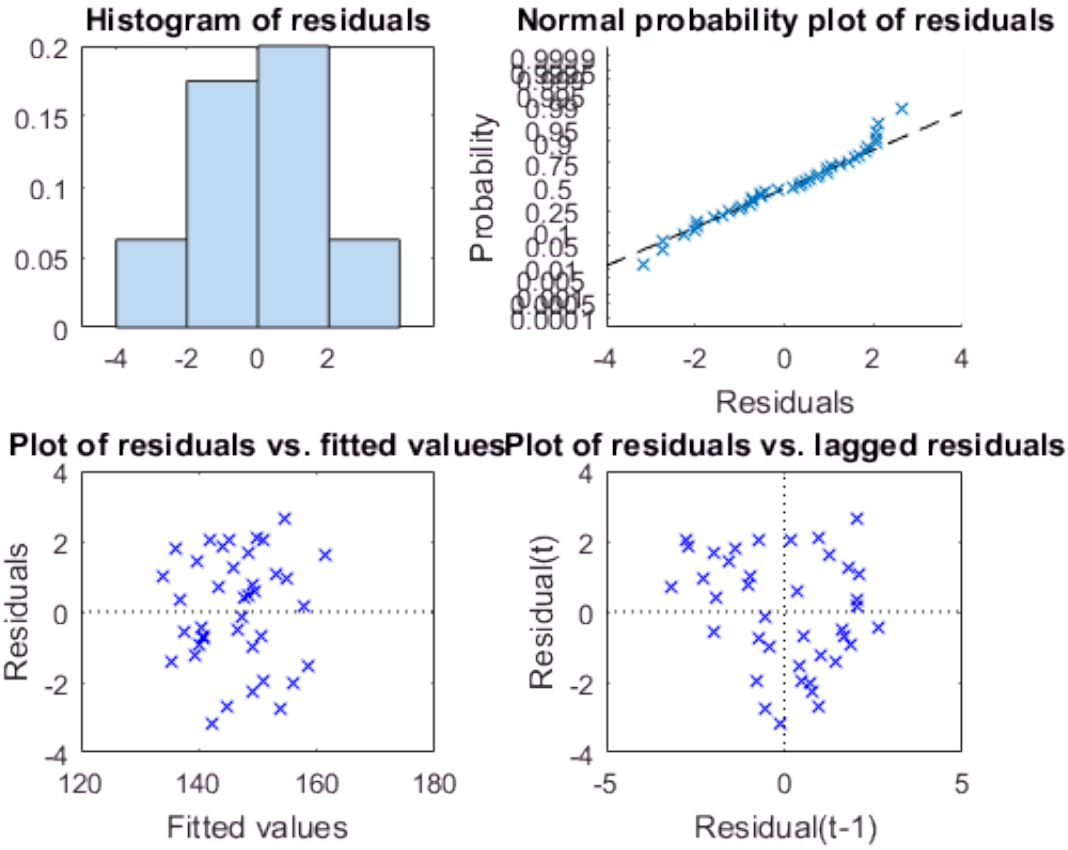
```
ans =
     AIC: 156.4603
    AICc: 157.6032
     BIC: 163.2158
    CAIC: 167.2158
```

```
m2.ModelCriterion
```

```
ans =
     AIC: 154.5592
    AICc: 155.2259
     BIC: 159.6258
    CAIC: 162.6258
```

```
% ... could now proceed in a similar way to double-check that age and weight
% should both be included in the model, but seeing that they are not correlated
% and both t-tests produce low p-values this would be overkill.

% Check residual plots for model m2:
% distribution of residuals:
clf
subplot(2,2,1)
plotResiduals(m2)
% Q-Q plot to check normality
```

```
subplot(2,2,2)
plotResiduals(m2,'probability')
% residuals versus fitted values (check for homoscedasticity, or worse, trends in
% the mean of residuals!)
subplot(2,2,3)
plotResiduals(m2,'fitted')
% auto-correlation (via lagged residuals)
subplot(2,2,4)
plotResiduals(m2,'lagged') % want no trend in this!
```



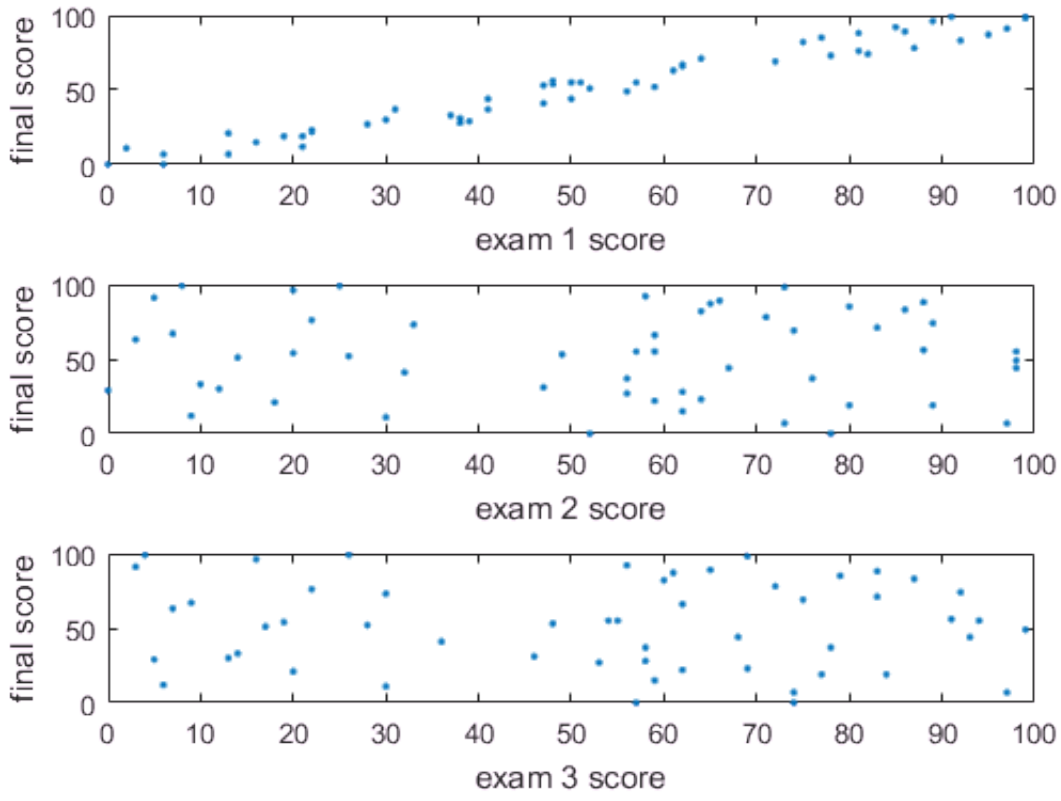

```
% ... these look fine...
```

Exam scores data

The data set on exam scores ('exam_scores.csv') is a hypothetical record of students' performance on a university module. It includes the results from three exams and the final exam. Your task is to investigate if any of the three exams predict students' performance in the final exam.

```
%% Exam score data:

% In this example only one variable predicts the response. The final model does
% not even include an intercept.

delimiterIn = ',';
A = importdata('exam_scores.csv',delimiterIn);
```

```
%% plot scatterplots of data
exam1 = A.data(:,1);
exam2 = A.data(:,2);
exam3 = A.data(:,3);
final = A.data(:,4);
clf
subplot(3,1,1)
plot(exam1,final,'.')
xlabel('exam 1 score') % x-axis label
ylabel('final score') % y-axis label

subplot(3,1,2)
plot(exam2,final,'.')
xlabel('exam 2 score') % x-axis label
ylabel('final score') % y-axis label

subplot(3,1,3)
plot(exam3,final,'.')
xlabel('exam 3 score') % x-axis label
ylabel('final score') % y-axis label
```




```
% From this it looks like only exam1 has an effect.

% create a table for the data:
data = table(exam1,exam2,exam3,final,'VariableNames',...
    {'exam1','exam2','exam3','final'});
data(1:5,:)
```

```
ans =
```

```
    exam1    exam2    exam3    final
    _____    _____    _____    _____

    85       58       56       92
    28       56       53       27
    51       57       55       55
    89       20       16       96
    81       88       83       88
```

```
% start with a model that only includes exam1 as a predictor:
m1 = fitlm(data, 'final~exam1')
```

```
m1 =

Linear regression model:
    final ~ 1 + exam1

Estimated Coefficients:
                   Estimate       SE        tStat        pValue
                   ________    ________    ________    __________

    (Intercept)    -0.3879       1.7004    -0.22813       0.82046
    exam1          0.99984     0.028632      34.921    2.7964e-37


Number of observations: 53, Error degrees of freedom: 51
Root Mean Squared Error: 5.92
R-squared: 0.96,  Adjusted R-Squared 0.959
F-statistic vs. constant model: 1.22e+03, p-value = 2.8e-37
```

```
% F-test suggests that the model is better than a constant model.
% T-test for intercept suggests that an intercept may not be needed
% (see also scatterplot above).

% consider a reduced model without an intercept, use Likelihood-ratio
% test to see if this is better than m1
m0 = fitlm(data, 'final~-1+exam1')
```

```
m0 =

Linear regression model:
    final ~ exam1

Estimated Coefficients:
              Estimate       SE       tStat       pValue
              ________    ________    ______    __________

    exam1     0.9941      0.013572    73.246    3.8221e-54


Number of observations: 53, Error degrees of freedom: 52
Root Mean Squared Error: 5.87
```

```
% Likelihood-ratio test:
[h pvalue] = lratiotest(m1.LogLikelihood,m0.LogLikelihood,1)
```

```
h =
     0

pvalue = 0.8162
```

```
% ... pvalue is large, so we cannot reject the null hypothesis that the
% parameter associated with the intercept is zero.

% for completeness, look at AICs of models including exam2 and exam3
% as predictors:
m2a = fitlm(data, 'final~-1+exam1+exam2');
m2b = fitlm(data, 'final~-1+exam1+exam3');
m0.ModelCriterion
```

```
ans =
     AIC: 338.9654
    AICc: 339.0438
     BIC: 340.9357
    CAIC: 341.9357
```

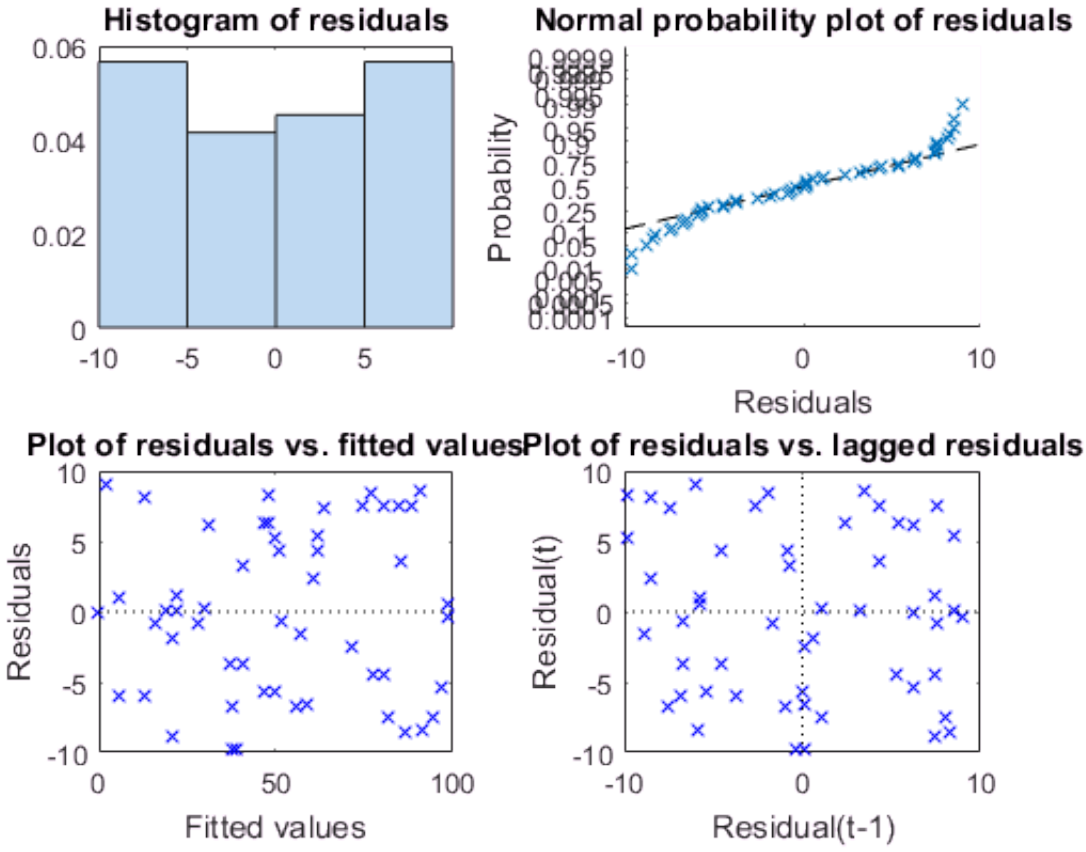
```
m2a.ModelCriterion
```

```
ans =
     AIC: 340.9465
    AICc: 341.1865
     BIC: 344.8871
    CAIC: 346.8871
```

```
m2b.ModelCriterion
```

```
ans =
     AIC: 340.9341
    AICc: 341.1741
     BIC: 344.8747
    CAIC: 346.8747
```

```
% ...as expected, there is no improvement in AIC when additional predictors
% are included. Likelihood-ratio test is not needed, as both m2a and m2b
% have more parameters that m0.

% Check residual plots for model m0:
% distribution of residuals:
clf
subplot(2,2,1)
plotResiduals(m0)
% Q-Q plot to check normality
subplot(2,2,2)
plotResiduals(m0,'probability')
% residuals versus fitted values (check for homoscedasticity, or worse,
% trends in the mean of residuals!)
subplot(2,2,3)
plotResiduals(m0,'fitted')
% auto-correlation (via lagged residuals)
subplot(2,2,4)
plotResiduals(m0,'lagged') % want no trend in this!
```



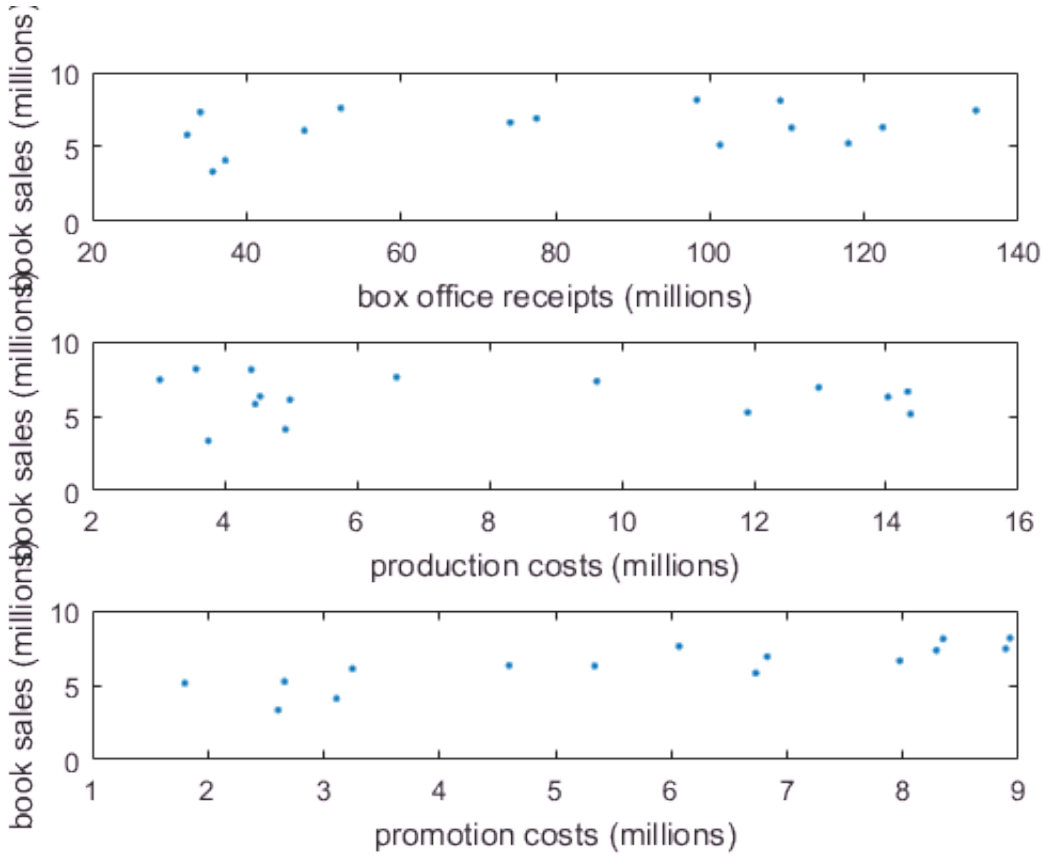

```
% ... the errors don't seem to follow a normal distribution. Ideally a different
% model should be fitted (see lecture 10 for ideas).
```

Hollywood movie data

The file 'hollywood_movies.csv' investigates revenues from book sales after the release of hollywood movies. Three possible predictors are included in the data: first year box office receipts, production costs and promotional costs. Investiate which out of these three variables predicts book sales.

```
%% Hollywood movie data

% In this data, two variables are predictive, but one has a very weak effect.
% Such weak effects might not show up if only few data points are available.

delimiterIn = ',';
A = importdata('hollywood_movies.csv',delimiterIn);

%% plot scatterplots of data
clf
boxoffice = A.data(:,1);
production = A.data(:,2);
promotion = A.data(:,3);
books = A.data(:,4);
```

```
subplot(3,1,1)
plot(boxoffice,books,'.')
xlabel('box office receipts (millions)') % x-axis label
ylabel('book sales (millions)') % y-axis label

subplot(3,1,2)
plot(production,books,'.')
xlabel('production costs (millions)') % x-axis label
ylabel('book sales (millions)') % y-axis label

subplot(3,1,3)
plot(promotion,books,'.')
xlabel('promotion costs (millions)') % x-axis label
ylabel('book sales (millions)') % y-axis label
```




```
% promotion seems to have an effect - for the other variables it's difficult
% to tell.

% check correlation between predictors
corrcoef(boxoffice, production)
```

```
ans = 2x2 double

    1.0000    0.1586
    0.1586    1.0000
```

```
corrcoef(boxoffice, promotion)
```

```
ans = 2x2 double
```

```
    1.0000    0.1438
    0.1438    1.0000
```

```
corrcoef(production, promotion)
```

```
ans = 2x2 double

    1.0000   -0.1653
   -0.1653    1.0000
```

```
% there is no indication of any strong correlations.

% create a table for the data:
data = table(boxoffice,production,promotion,books,'VariableNames',...
    {'boxoffice','production','promotion','books'});
data(1:5,:)
```

```
ans =
    boxoffice    production    promotion    books
    _________    __________    _________    ______

    47.507       4.9925         3.249      6.0961
    110.59       14.024        5.3395      6.2804
    37.277       4.9226        3.1117      4.1037
    109.13       4.4074        8.3472      8.1003
    122.38        4.542        4.6012      6.3111
```

```
% start with a full model that includes promotion:
m1 = fitlm(data, 'books~promotion')
```

```
m1 =

Linear regression model:
    books ~ 1 + promotion

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     3.6711      0.52411    7.0043    9.2812e-06
    promotion      0.46103     0.084546     5.453    0.00011065


Number of observations: 15, Error degrees of freedom: 13
Root Mean Squared Error: 0.802
R-squared: 0.696,  Adjusted R-Squared 0.672
F-statistic vs. constant model: 29.7, p-value = 0.000111
```

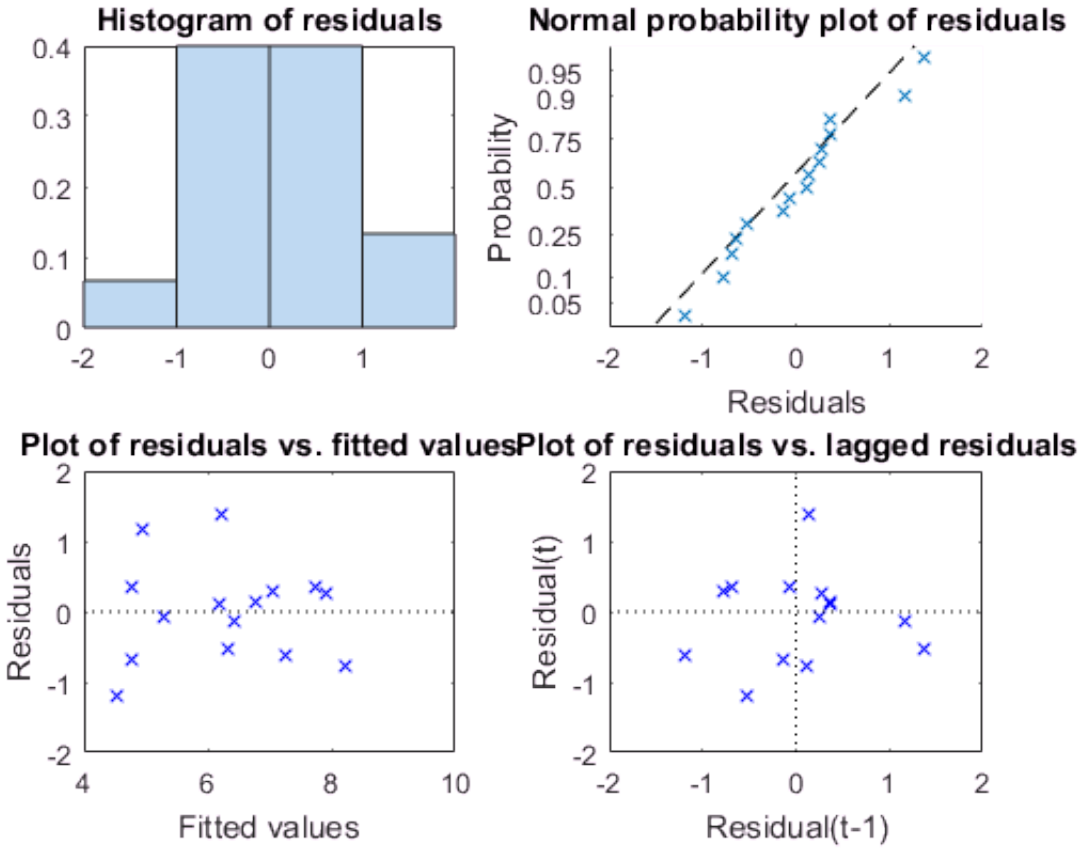
```
% F-test suggests that the model is better than a constant model.
% T-tests suggest that hypothesis of zero effect of promotion can
% be rejected.

% Fit models that separately add production and boxoffice to m1 and use
% Likelihood-ratio tests to see if these additional predictors should
% be included:
m2 = fitlm(data, 'books~promotion+boxoffice');
m3 = fitlm(data, 'books~promotion+production');
[h pvalue] = lratiotest(m2.LogLikelihood,m1.LogLikelihood,1)
```

```
h =
     0

pvalue = 0.0861
```

```
% ... there is a weak indication that we cannot reject the null hypothesis
% that the parameter associated with boxoffice is zero.
[h pvalue] = lratiotest(m3.LogLikelihood,m1.LogLikelihood,1)
```

```
h =
     0

pvalue = 0.5832
```

```
% ... we cannot reject the null hypothesis that production has zero effect.

% Check residual plots for model m2:
% distribution of residuals:
clf
subplot(2,2,1)
plotResiduals(m2)
% Q-Q plot to check normality
subplot(2,2,2)
plotResiduals(m2,'probability')
% residuals versus fitted values (check for homoscedasticity, or worse,
% trends in the mean of residuals!)
subplot(2,2,3)
plotResiduals(m2,'fitted')
% auto-correlation (via lagged residuals)
subplot(2,2,4)
plotResiduals(m2,'lagged') % want no trend in this!
```




```
% ... these look fine...
```