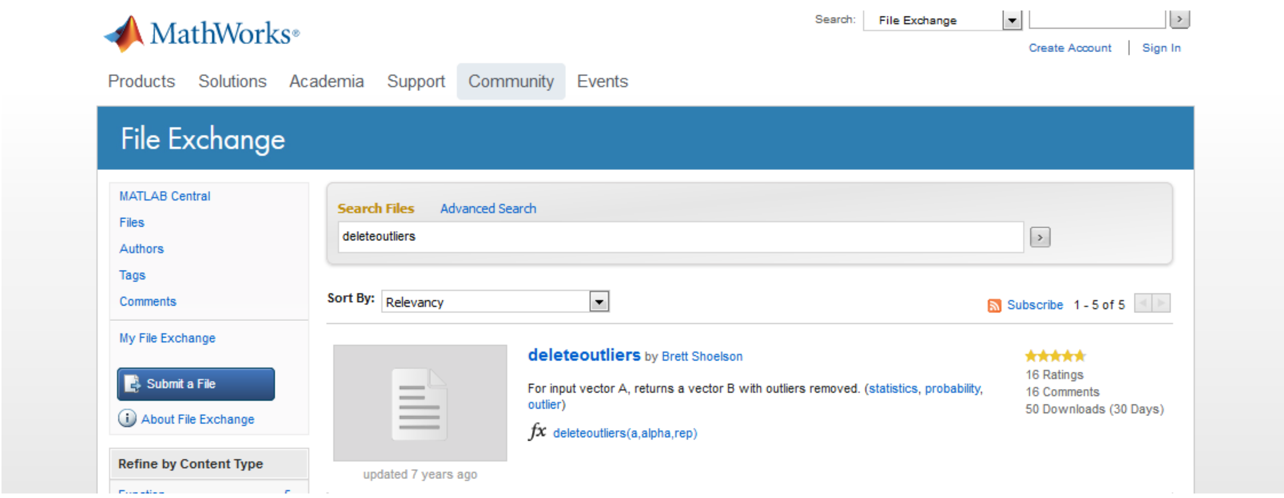
Engineering Maths EMAT 30007, Extra lab

Ksenia Shalonova

1 Functions `deleteoutliers` and `taucv`

This function can be used for detecting outliers in the normal distribution based on Grubbs test.



Download file 'outliers.txt'.

```
>> x=load(file);
>> x=sort(x(:));
>> [y,idx,outl]=deleteoutliers(x,0.1); % delete outliers
>> if isempty(idx),
>> disp('No outliers')
>> else
  >> fprintf('There are = %d\n',length(idx))
 >> end
```

Function `taucv` computes the critical value for the detection of outliers using Pope test. It depends on significance level (Type I error), degree of freedom and the number of observations.

2 Polynomial regression modeling using Gram Schmidt orthonormalization - ADVANCED - NOT FOR THE EXAM:

(a) Download file 'regr.txt'.

```
>> xy=load(file); % input two columns
>> x=xy(:,1); % x-values
>> y=xy(:,2); % y-values
>> n=length(x); % number of paired observations
>> fprintf('Number of paired observations n=%d.\n',n);
```

(b) Implementation of the Gram-Schmidt algorithm

```
>> sg=0.01; % significance level
>> kmax=20; % max degree for the polynomial is set to 20
>> k=[0:kmax];
>> X=repmat(x,1,kmax+1).^repmat(k,n,1); % form matrix of functions 1, x^2 ...
>> X(:,1)=X(:,1)/norm(X(:,1)); % normalise first column
>> for ki=2:kmax+1, % orthonormalise columns
  >> Sum=0; %
  >> for kk=1:ki-1,
    >> Sum=Sum+(X(:,ki)'*X(:,kk))*X(:,kk);
```

```
  >> end
  >> X(:,ki)=X(:,ki)-Sum;
  >> X(:,ki)=X(:,ki)/norm(X(:,ki));
>> end
>> [b,bint]=regress(y,X,sg); % apply regression model
>> disp('Confidence intervals');
>> fprintf('significance level sg=%5.2f\n',sg)
>> disp(' k     CI_low          CI_upper');
>> fprintf('%2.0f  %12.7f  %12.7f\n',[k;bint']);
>> np=find(prod(bint,2)>0); % take into consideration the degree
>> fprintf('Possible degrees for polynomial  %d',k(np));
>> fprintf('.\n');
```

The following MATLAB output shows the maximum degree for the polynomial that is equal to 2. Notice that we have chosen only confidence intervals that do not contain 0.

```
significance level sg= 0.01
 k     CI_low          CI_upper
 0  -101.2164094   -74.4210433
 1    51.1407839    77.9361500
 2  -103.8897883   -77.0944222
 3   -11.9491766    14.8461895
 4    -7.8240843    18.9712818
 5   -16.4258633    10.3695028
 6    -5.6586070    21.1367592
 7   -19.2181320     7.5772341
 8   -12.5856517    14.2097144
 9    -4.1026787    22.6926874
10   -13.8967925    12.8985736
11   -12.7305531    14.0648130
12   -12.0239363    14.7714298
13    -7.5936973    19.2016697
14   -16.5967343    10.1986364
15    -9.2194476    17.5768482
16   -14.9777503    11.8228937
17   -13.2594593    13.8646300
18   -10.2265980    18.5920362
19    -7.4718797    19.6531404
20   -15.4011520    13.4220377
Possible degrees for polynomial:  0  1  2.
```

(c) Now we can fit the polynomial with the highest degree equal to 2.

```
>> mmax=max(k(np)); % maximal degree
>> p=polyfit(x,y,mmax); % polynomial coefficients
>> yt=polyval(p,x); % predicted
>> figure;
>> plot(x,y,'k.',x,yt,'k-');
>> set(get(gcf,'CurrentAxes'),...
>>   'FontName','Times New Roman Cyr','FontSize',10)
>> title('\bfPolynomial regression')
>> xlabel('\itx')
>> ylabel('\ity')
```