CS170A -- HW#4 -- assignment and solution form -- Octave

Your name: Shirley Xuemin He

Your UID: 204468663

Please upload only this notebook to CCLE by the deadline.

Policy for late submission of solutions: We will use Paul Eggert's Late Policy: N days late ⇔ 2^N points deducted} The number of days late is $N = 0$ for the first 24 hrs, $N = 1$ for the next 24 hrs, etc., and if you submit an assignment H hours late, $2^{\lfloor H / 24 \rfloor}$ points are deducted.

Problem 1: Income Tax and Benford's Law

For every presidential candidate (except Donald Trump), income tax returns have been filed prior to an election: http://www.taxhistory.org/www/website.nsf/Web/PresidentialTaxReturns?OpenDocument (http://www.taxhistory.org/www/website.nsf/Web/PresidentialTaxReturns?OpenDocument).

Some of these returns are impressive: Mitt Romney's 2011 tax return was 379 pages long!
Hillary Clinton's income was the highest of the 2016 candidates (except perhaps Trump); see her 2015 income tax return (http://www.taxhistory.org/thp/presreturns.nsf/Returns/FCA79776EFA029088525800D005A016C/$file/HR_Clir
Bernie Sanders' reported income is barely enough for his family to survive in Los Angeles...

1.1 Hillary Clinton's 2014 and 2015 Income Tax returns

In this assignment, take the files `HR_Clinton_2014_tax_return_numbers.txt` and `HR_Clinton_2015_tax_return_numbers.txt` listing numbers in Hillary's tax returns for the last 2 years.

For each of these two files, determine (using the method in the Course Reader) whether the *unique* numbers in this file (please omit duplicates) violate Benford's Law.

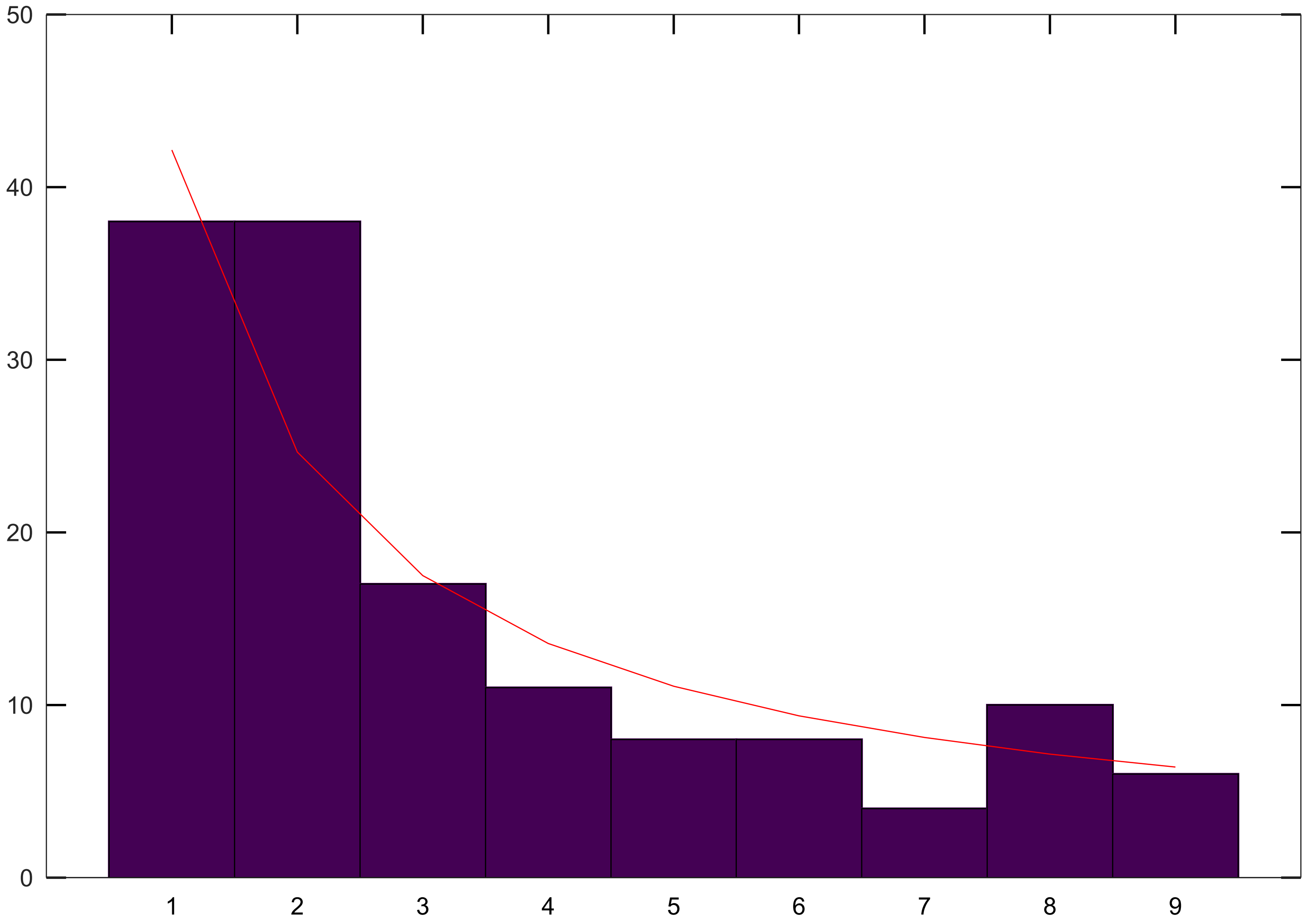
In [23]:

```
% pkg install -forge io
% pkg install -forge statistics
% pkg load statistics
```

In [26]:

```
Clinton14 = textread('HR_Clinton_2014_tax_return_numbers.txt', '%f');
Clinton14unique = unique(Clinton14);

firstdigit = @(x) floor(x./(10.^floor(log10(x))));

number_of_bins = 9;
nu = number_of_bins-1;

ClintonHistogram = hist(firstdigit(Clinton14unique),1:9);
hist(firstdigit(Clinton14unique),1:9)

BenfordProbabilities = diff(log10(1:10));
N = length(Clinton14unique);
BenfordHistogram = N * BenfordProbabilities;
hold on
plot(1:9, BenfordHistogram, 'r')
title('Histogram of first digits in Clinton 2014 tax return, with Benford expected values')
hold off

ChiSquareStatistic = sum((ClintonHistogram - BenfordHistogram).^2 ./ BenfordHistogram)
ChiSquareProbability = cdf('Chisquare',ChiSquareStatistic, nu)
```

```
ChiSquareStatistic =  12.434
ChiSquareProbability =  0.86712
```

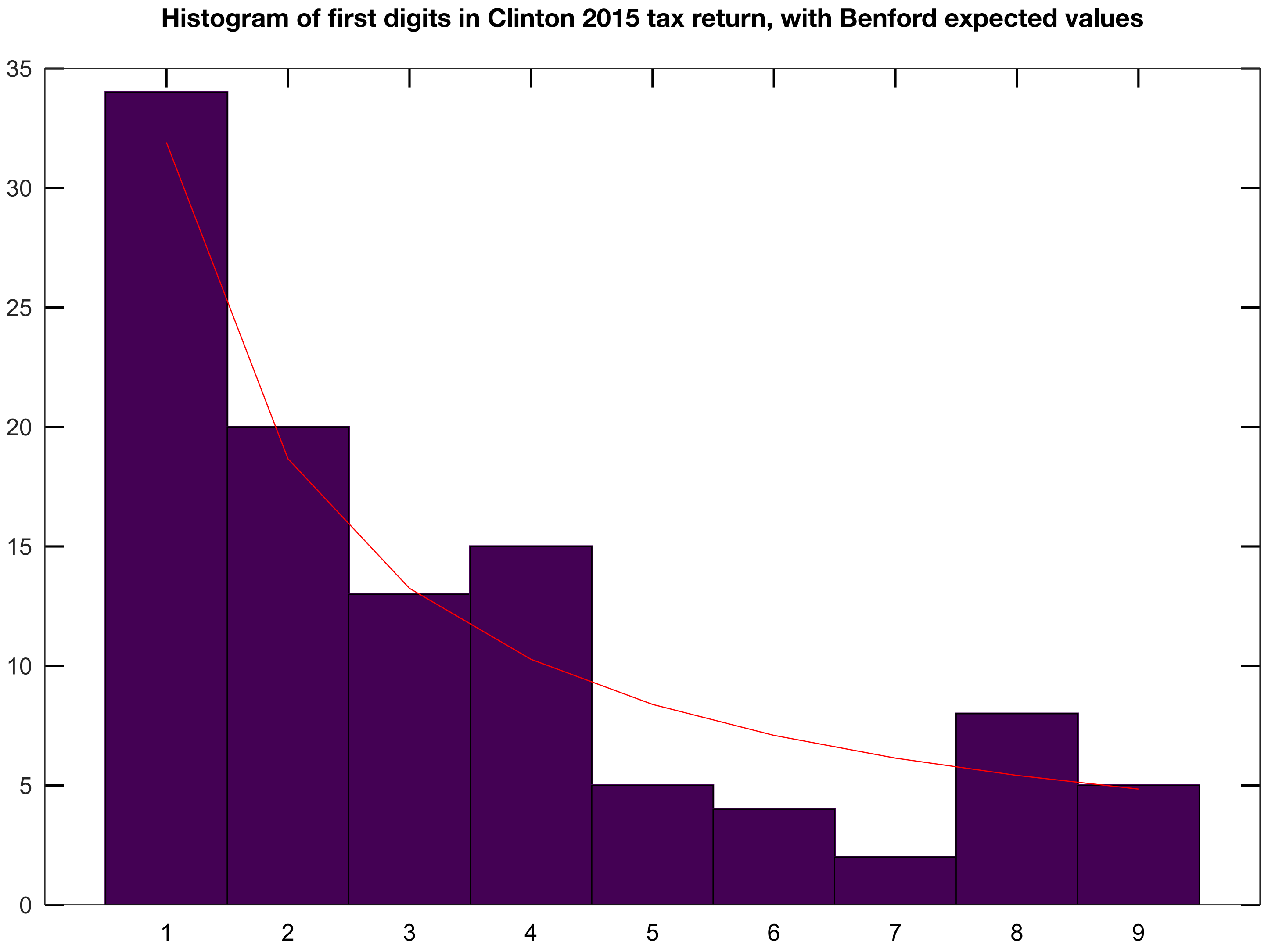


Histogram of first digits in Clinton 2014 tax return, with Benford expected values

In [25]:

```
Clinton15 = textread('HR_Clinton_2015_tax_return_numbers.txt', '%f');
Clinton15unique = unique(Clinton15);

firstdigit = @(x) floor(x./(10.^floor(log10(x))));

number_of_bins = 9;
nu = number_of_bins-1;

ClintonHistogram = hist(firstdigit(Clinton15unique),1:9);
hist(firstdigit(Clinton15unique),1:9)

BenfordProbabilities = diff(log10(1:10));
N = length(Clinton15unique);
BenfordHistogram = N * BenfordProbabilities;
hold on
plot(1:9, BenfordHistogram, 'r')
title('Histogram of first digits in Clinton 2015 tax return, with Benford expect
ed values')
hold off

ChiSquareStatistic = sum((ClintonHistogram - BenfordHistogram).^2 ./ BenfordHist
ogram)
ChiSquareProbability = cdf('Chisquare',ChiSquareStatistic, nu)
```

```
ChiSquareStatistic =  9.1634
ChiSquareProbability =  0.67130
```



The numbers in both files roughly follow the Benford's Law.

1.2 Donald Trump's 1995 Income Tax return

3 pages of Donald Trump's 1995 Income Tax form (https://assets.documentcloud.org/documents/3117920/Pages-From-Donald-Trump-s-1995-Income-Tax-Returns.pdf) have been published by the New York Times (http://www.nytimes.com).

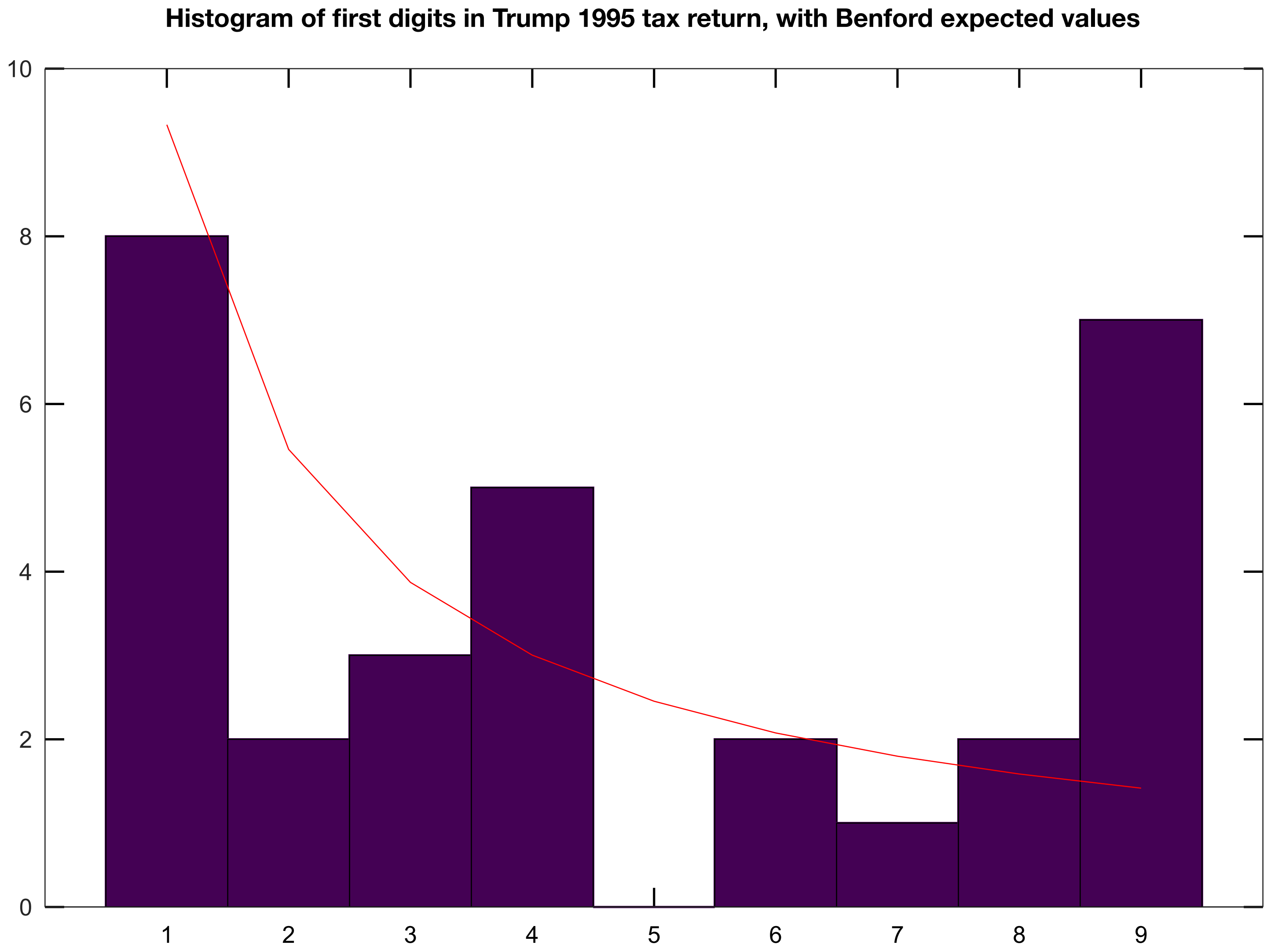
The file `DJ_Trump_1995_tax_return_numbers.txt` lists numbers in this (partial) income tax return.

Determine whether the unique numbers in this file violate Benford's Law.

In [24]:

```
Trump95 = textread('DJ_Trump_1995_tax_return_numbers.txt', '%f');
Trump95unique = unique(Trump95);

firstdigit = @(x) floor(x./(10.^floor(log10(x))));

number_of_bins = 9;
nu = number_of_bins-1;

TrumpHistogram = hist(firstdigit(Trump95unique),1:9);
hist(firstdigit(Trump95unique),1:9)

BenfordProbabilities = diff(log10(1:10));
N = length(Trump95unique);
BenfordHistogram = N * BenfordProbabilities;
hold on
plot(1:9, BenfordHistogram, 'r')
title('Histogram of first digits in Trump 1995 tax return, with Benford expected values')
hold off

ChiSquareStatistic = sum((TrumpHistogram - BenfordHistogram).^2 ./ BenfordHistogram)
ChiSquareProbability = cdf('Chisquare',ChiSquareStatistic, nu)
```

```
ChiSquareStatistic =  28.786
ChiSquareProbability =  0.99965
```



The numbers in this file violate the Benford's Law.

Problem 2: Earthquakes

For this assignment we want you to test whether earthquakes in a given region occur with uniform frequency throughout the year. Some geologists claim earthquakes are completely unpredictable, occurring randomly.

Specifically, we want you to compare the histograms of months in which the earthquakes compare with an expected uniform histogram.

To do this, given a dataset of 20,000 earthquakes, we want you to create a histogram with 12 bins giving the number of earthquakes that occurred within each of the 12 months of the year. This is the **observed histogram** O. Since there are 20,000 earthquakes, the total of all bins in O will be 20,000.

Next, create an expected histogram E of 12 bins whose bins have values

```
E = ([ 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 ] / 365) * 20000
```

giving a uniform expected number of earthquakes in each month. (Simply ignore leap years for this assignment.)
We want you to use the χ^2 test to determine whether the earthquakes occur uniformly or not. In Matlab you can use the `cdf()` function to compute probabilities of χ^2 values. In Octave, you can install and load the `cdf()` function with commands like:

```
%%%  Issue the following commands in Octave:

pkg -forge install io  %%% install the latest 'io' package ('statistics' package seems to need it)

pkg -forge install statistics

pkg  load  statistics

help cdf

ChiSquareProbability = cdf( 'chisquare', ChiSquareStatistic, nu )
```

Each dataset is a .csv file obtained from a server at http://www.iris.edu/seismon (http://www.iris.edu/seismon). Earthquake times are reported in UTC (Coordinated Universal Time), which is essentially the same thing as GMT (Greenwich Mean Time). However, all we will need is the month in which each earthquake occurred.

2.1 Do large earthquakes in the area around New Zealand happen uniformly around the year?

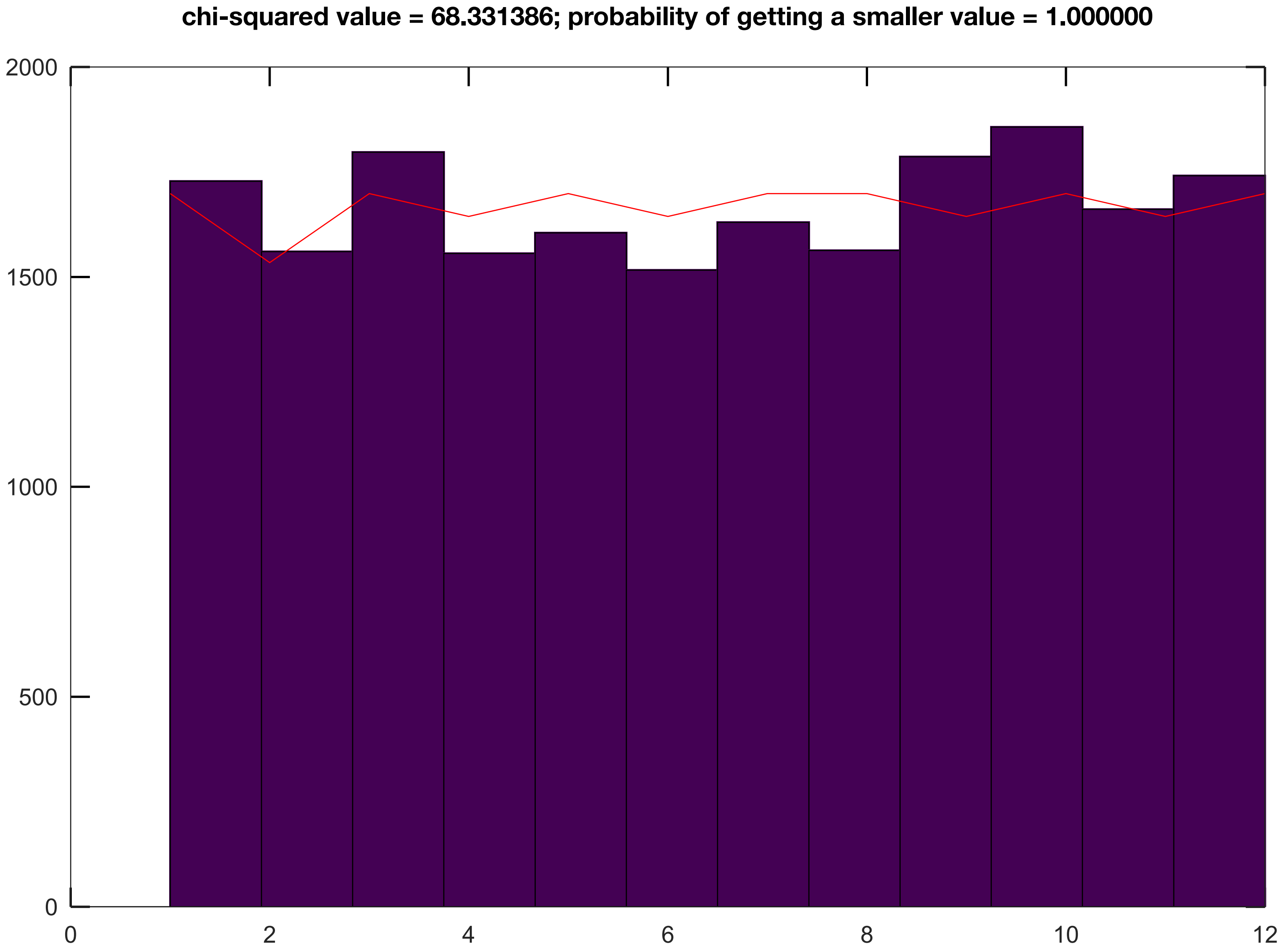
On Sunday November 13 there was a 7.8 earthquake in New Zealand.

The file `NewZealandQuakes.csv` is a list of the 20,000 largest earthquakes since 1970 in the Tonga/Kermadec Islands region near New Zealand. This region, where the Australian continent and Pacific Ocean meet, is very highly active earthquake-wise. It is an amazing geological formation, a wall that is over 400 miles deep.

Does this distribution of month values of earthquakes differ significantly from a uniform distribution?

In [12]:

```
E = ([ 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 ] / 365.0) * 20000;

NewZealandQuakes = dlmread("NewZealandQuakes.csv",",");
months = NewZealandQuakes(2:end,2); % The month column

nbins = 12;
%nu = number_of_bins-1;
[ObservedHistogram,BinCenters] = hist(months,linspace(1,12,nbins));
%N = length(months);
%p = 1/nbins;
df = nbins-1;
UniformHistogram = E;
ChiSqValue = sum((ObservedHistogram - UniformHistogram) .^2 ./ UniformHistogram)
;
ChiSqProb = cdf('Chisquare', ChiSqValue, df);

hist(months, nbins);
hold on
plot(BinCenters, UniformHistogram, 'r')
title(sprintf('chi-squared value = %f; probability of getting a smaller value =
%f', ChiSqValue,ChiSqProb));
hold off
```



The distribution is very close to a uniform distribution.

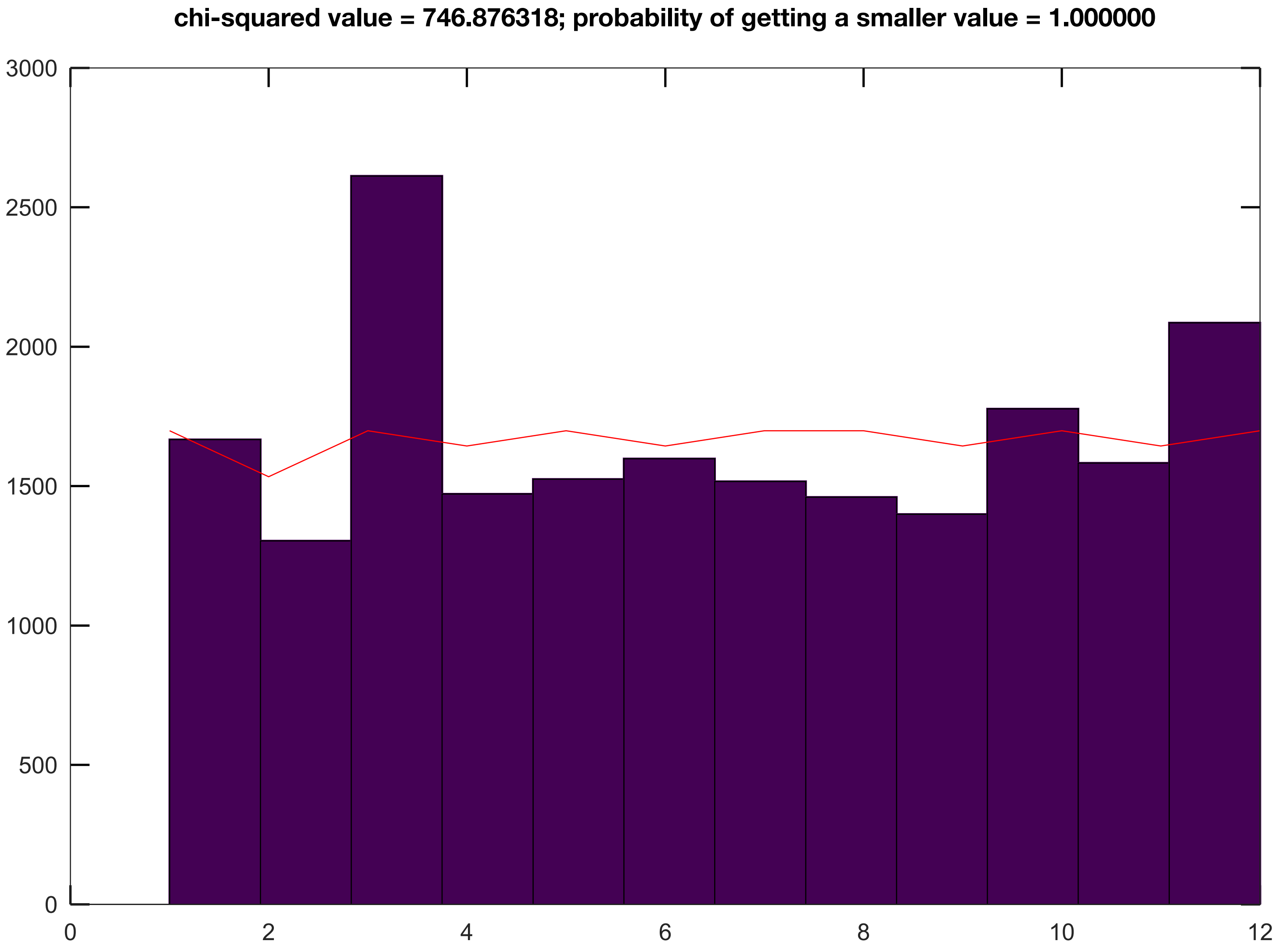
2.2 Do large earthquakes in the area around Japan happen uniformly around the year?

The file `JapanQuakes.csv` is a list of the 20,000 largest earthquakes since 1970 in the region around Japan. A magnitude 9.0 earthquake hit this area in 2011.

Does the distribution of month values of earthquakes in this region differ significantly from a uniform distribution?

In [13]:

```
E = ([ 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 ] / 365.0) * 20000;

JapanQuakes = dlmread("JapanQuakes.csv",",");
months = JapanQuakes(2:end,2); % The month column

nbins = 12;
%nu = number_of_bins-1;
[ObservedHistogram,BinCenters] = hist(months,linspace(1,12,nbins));
%N = length(months);
%p = 1/nbins;
df = nbins-1;
UniformHistogram = E;
ChiSqValue = sum((ObservedHistogram - UniformHistogram) .^2 ./ UniformHistogram)
;
ChiSqProb = cdf('Chisquare', ChiSqValue, df);

hist(months, nbins);
hold on
plot(BinCenters, UniformHistogram, 'r')
title(sprintf('chi-squared value = %f; probability of getting a smaller value =
%f', ChiSqValue,ChiSqProb));
hold off
```



The distribution is not very close to a uniform distribution, the number of earthquakes peaks in March.

Problem 3: Using Newton's Method on Matrices

The Course Reader explains that Newton's method for computing inverses b^{-1} works by defining $f(x) = (1/x - b)$, giving a Newton iteration $x_{n+1} \ = \ g(x_n)$, where:

$$g(x) \ = \ x \ - \ f(x)/f'(x) \ = \ x \ - \ \frac{1/x - b}{-1/x^2} \ = \ x \ (2 \ - \ bx) \, .$$

A similar iteration obtained from $f(X) \ = \ (X^{-1} \ - \ B)$ can be used on matrices:

$$g(X) \ = \ X \ - \ (\nabla f(X))^{-1} \, f(X) \ = \ X \ - \ (-X^2) \, (X^{-1} - B) \ = \ X \ (2 \, I \ - \ B \, X) \, .$$

Use the iteration to obtain B^{-1} for each of the Hilbert matrices B:

- `B = hilb(4)`
- `B = hilb(8)`
- `B = invhilb(4)`
- `B = invhilb(8)`

For `B = hilb(...)`, you can start with initial value $X = I$, the identity matrix.

For `B = invhilb(...)`, you will need to find a good starting value yourself.

For each such matrix B, determine how many iterations are needed for convergence, and measure the relative error $||X - B^{-1}|| \, / \, ||B^{-1}||$.

The function `invhilb()` can be used to obtain the exact value of the inverse of `hilb()`.

In [27]:

```
B = hilb(4);
Binv = invhilb(4);

X = eye(4);  % starting value for the iteration
I = eye(4);
Xnext = X;
count = 0;
RelativeError = norm(X - Binv)/norm(Binv);

while (RelativeError > 2^(-18))
  X = Xnext;
  Xnext = X * (2*I - B*X);
  count = count + 1;
  RelativeError = norm(Xnext - Binv)/norm(Binv);
endwhile

printf('%d iterations are needed for convergence', count);
RelativeError = norm(Xnext - Binv)/norm(Binv)
```

```
17 iterations are needed for convergence
RelativeError =    3.1266e-06
```

In [28]:

```
B = invhilb(4);
Binv = hilb(4);

X=B'/(norm(B,1)*norm(B,Inf));  % starting value for the iteration
I = eye(4);
Xnext = X;
count = 0;
RelativeError = norm(X - Binv)/norm(Binv);

while (RelativeError > 2^(-18))
  X = Xnext;
  Xnext = X * (2*I - B*X);
  count = count + 1;
  RelativeError = norm(Xnext - Binv)/norm(Binv);
endwhile

printf('%d iterations are needed for convergence', count);
RelativeError = norm(Xnext - Binv)/norm(Binv)
```

```
33 iterations are needed for convergence
RelativeError =    1.1604e-09
```

In [29]:

```
B = invhilb(8);
Binv = hilb(8);

X=B'/(norm(B,1)*norm(B,Inf));  % starting value for the iteration
I = eye(8);
Xnext = X;
count = 0;
RelativeError = norm(X - Binv)/norm(Binv);

while (RelativeError > 2^(-18))
  X = Xnext;
  Xnext = X * (2*I - B*X);
  count = count + 1;
  RelativeError = norm(Xnext - Binv)/norm(Binv);
endwhile

printf('%d iterations are needed for convergence', count);
RelativeError = norm(Xnext - Binv)/norm(Binv)
```

```
73 iterations are needed for convergence
RelativeError =    2.9980e-08
```

Problem 4: Regularized Optimization in Ridge Regression

In the traditional least squares model $\mathbf{y} = X\mathbf{c}$, the vector of coefficients $\mathbf{c}$ is often chosen to minimize the least squared error

$$\epsilon(\mathbf{c}) \; = \; ||\mathbf{y} - X\mathbf{c}||^2.$$

We can add a *constraint* that requires $||\mathbf{c}||^2$ to be small. This is often implemented by adding a **regularization term** $R(\mathbf{c})$ to the error that is very large for values of $\mathbf{c}$ that violate the constraints. In this case the iteration seeks to minimize the **regularized problem**

$$\epsilon(\mathbf{c}) + R(\mathbf{c}) \; = \; ||\mathbf{y} - X\mathbf{c}||^2 + R(\mathbf{c}).$$

4.0 Get the data

The dataset for this problem is called `prostate.csv`. Read in this file. The columns are lcavol,lweight,age,lbph,svi,lcp,gleason,pgg45,lpsa.

The regression problem is to predict lpsa from the other 8 variables.

In [30]:

```
prostate = dlmread("prostate.csv",",");
prostate = prostate(2:end,:);

y = prostate(:,end);  % lpsa

X = prostate(:,1:(end-1));  % lcavol lweight age lbph svi lcp gleason pgg45
```

4.1 Tikhonov regularization

Tikhonov regularization $R(\mathbf{c}) = ||T\,\mathbf{c}||^2$ uses a covariance matrix T that is chosen to scale the $\mathbf{c}$ values properly. In this case we want to solve the Tikhonov regularized least squares problem, minimizing

$$L(\mathbf{c}, T) \;=\; \epsilon(\mathbf{c}) \;+\; R(\mathbf{c}) \;=\; ||\mathbf{y} \;-\; X\mathbf{c}||^2 \;+\; ||T\,\mathbf{c}||^2 \;=\; \left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} \;-\; \begin{pmatrix} X \\ T \end{pmatrix} \mathbf{c} \right\|^2 .$$

To minimize the expression on the right, we need to find the least squares solution of:

$$\begin{pmatrix} X \\ T \end{pmatrix} \mathbf{c} \;=\; \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} .$$

What are the **normal equations** for this equation that we can use to obtain the least squares solution?

My answer (to be filled in with Markdown and Equations):

$$\begin{pmatrix} X & T \end{pmatrix} \begin{pmatrix} X \\ T \end{pmatrix} \mathbf{c} \;=\; \begin{pmatrix} X & T \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} .$$

From this equation give a formula for the **least squares solution $\mathbf{c}$**:

My answer (to be filled in with Markdown and Equations):

$$\mathbf{c} \;=\; \left(\begin{pmatrix} X & T \end{pmatrix} \begin{pmatrix} X \\ T \end{pmatrix} \right)^{-1} \begin{pmatrix} X & T \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} .$$

4.2 Ridge Regression

Ridge Regression is a popular form of this least squares problem when $T'\,T \;=\; \lambda\,I$ for some $\lambda \geq 0$ (so that $T'\,T$ has a 'ridge' down the diagonal). As λ decreases to 0, the problem reduces to the ordinary least squares, while as λ grows large, $||\mathbf{c}||$ becomes increasingly emphasized and minimization reduces it to zero.

Assume that the singular values of X are $\sigma_1\,,\,\ldots\,,\,\sigma_p$. What are the singular values of $(X'\,X \;+\; T'\,T)$?

<u>My answer (to be filled in with Markdown and Equations):</u>

We are assuming that X has SVD USV', where S is a diagonal matrix whose i-th singular value is σ_i, so $(X'X)$ has SVD $VS'SV'$, with i-th singular value σ_i^2.

Therefore, since $T'T = \lambda I$, and $VDV' = D$ for any constant diagonal matrix $D = \lambda I$, the i-th singular value of $(X'X + T'T)$ is

$$......someformulainvolving\sigma_i......$$

Also give the singular values of the **hat matrix**

$$H_\lambda \;=\; X\,(X'\,X \;+\; \lambda\,I)^{-1}\,X'.$$

<u>My answer (to be filled in with Markdown and Equations):</u>

Again, we are assuming that X has SVD USV', where S has i-th singular value σ_i, and $(X'X) = VS'SV'$ has i-th singular value σ_i^2.

Therefore, the i-th singular value of the hat matrix $H_\lambda = X(X'X + \lambda I)^{-1}X'$ is

$$......someformulainvolving\sigma_i......$$

4.3 Degrees of Freedom

Define **degrees of freedom** $\mathrm{df}(\lambda) \;=\; \mathrm{trace}(\,H_\lambda\,)$, where H_λ is the hat matrix.

Give a formula for $\mathrm{df}(\lambda)$ in terms of λ and the singular values of X:

<u>My answer (to be filled in with Markdown and Equations):</u>

Assuming the i-th singular value of the hat matrix $H_\lambda = X(X'X + \lambda I)^{-1}X'$ is as in the previous answer,

$$df(\lambda) = ...someformulainvolving\sigma_i values......$$

Also give a formula for $\mathrm{df}(0)$:

My answer (to be filled in with Markdown and Equations):

$$df(0) = \ldots\ldots some formula involving \sigma_i \ldots\ldots$$

4.4 Actually computing regression coefficients for various values of λ

Using the prostate dataset, let $\mathbf{y}$ be the vector `lcavol`, and let X be the matrix with the other variables.

Also normalize X and $\mathbf{y}$ (i.e., replace them by their z-scores), so that all columns have the same scale.

Plot the value of the ridge regression coefficients $\mathbf{c}$ for all values of λ from 0 to $2\,||X'\,X||$.

In [ ]:

```
% COMPLETE THIS CODE SO THAT IT PLOTS THE COEFFICIENTS c FOR VARIOUS VALUES OF l
ambda
% AND INCLUDE THE PLOT IN THE OUTPUT YOU SUBMIT.

max_lambda = round( 2 * norm(X' * X) )
N = 100
lambda_values = linspace( 0, max_lambda, N )
c_values = zeros(p,N)
colors = { 'r' 'g' 'b' 'c' 'm' 'y' 'k' 'b' }
% varnames = { 'lcavol' 'lweight' 'age' 'lbph' 'svi' 'lcp' 'gleason' 'pgg45' }

for j=1:N
   c_values(:,j) = ...
end

plot( lambda_values, c(1,:), colors(i))
hold on
for i=2:p
   plot( ..., colors(i) )
end
xlabel('lambda')
ylabel('coefficient value')
title('Ridge regression coefficients for various values of lambda')
% legend( ... varnames ... )   %  add a legend showing variable names
hold off
```

4.5 Plotting the Generalized Cross-Validation function

An 'optimal' value of λ will yield coefficients $\mathbf{c}$ that balances the residual sum of squares:

$$RSS = ||\mathbf{y} - \widehat{\mathbf{y}}||^2 = ||\mathbf{y} - H_\lambda \, \mathbf{y}||^2$$

Note $\widehat{\mathbf{y}} = H_\lambda \, \mathbf{y}$ depends on λ and the regularization error $||\mathbf{c}||^2$.

Define the **Generalized Cross-Validation** (GCV) measure

$$\mathrm{GCV}(\lambda) = \frac{n}{(n - \mathrm{df}(\lambda))^2} \, || \, \mathbf{y} - \widehat{\mathbf{y}} \, ||^2 .$$

Ridge Regression is often implemented by finding a value of λ that minimizes $\mathrm{GCV}(\lambda)$.

For the values of λ that you used above, compute $\mathrm{GCV}(\lambda)$.

In [ ]:

```
% COMPLETE THIS CODE SO THAT IT PLOTS GCV(lambda) FOR VARIOUS VALUES OF lambda
% AND INCLUDE THE PLOT IN THE OUTPUT YOU SUBMIT.

max_lambda = round( 2 * norm(X' * X) )
N = 100
lambda_values = linspace( 0, max_lambda, N )
GCV = zeros(N,1)

for j=1:N
   GCV(j) = ... some function of lambda_values(j) ....
end

minimal_lambda_position = find( GCV = min(GCV) )
minimal_lambda = lambda_values( minimal_lambda_position )

plot( lambda_values, GCV, 'b' )
xlabel('lambda')
ylabel('GCV(lambda)')
title( sprintf('GCV(lambda) is minimal at lambda = %7.3f', minimal_lambda) )
hold on
plot( [minimal_lambda], [min(GCV)], 'r.' )
hold off
```

4.6 Actually Optimizing Generalized Cross-Validation

First, develop a function that computes $\mathrm{GCV}(\lambda)$.

Then, use an optimization routine to find a value λ in the interval $[\,0,\ 2\,||X'\,X||\,]$ for which $\mathrm{GCV}(\lambda)$ is minimal.

In [ ]:

```
%  In Octave, execute the following to download/install, and load, the 'optim' package:

pkg -forge install optim

pkg  load optim


%  optimization functions in Octave / Matlab

help fzero    % find a root of a univariate function

help fminbnd  % minimization of a univariate function

help fminsearch % minimization of a multivariate function

help fminunc % minimization of a multivariate function (with gradients)

help lsqlin   % linear least squares

help quadprog % quadratic programming

help lsqnonlin % nonlinear least squares

help fsolve   % solve set of nonlinear equations (vector-valued function)
```

In [ ]:

```
% COMPLETE THIS CODE SO THAT IT FINDS A VALUE OF lambda THAT MINIMIZES GCV(lambda).

max_lambda = round( 2 * norm(X' * X) )

GCV_function = @(lambda) ...

minimal_lambda =  fminbnd( ... )
```