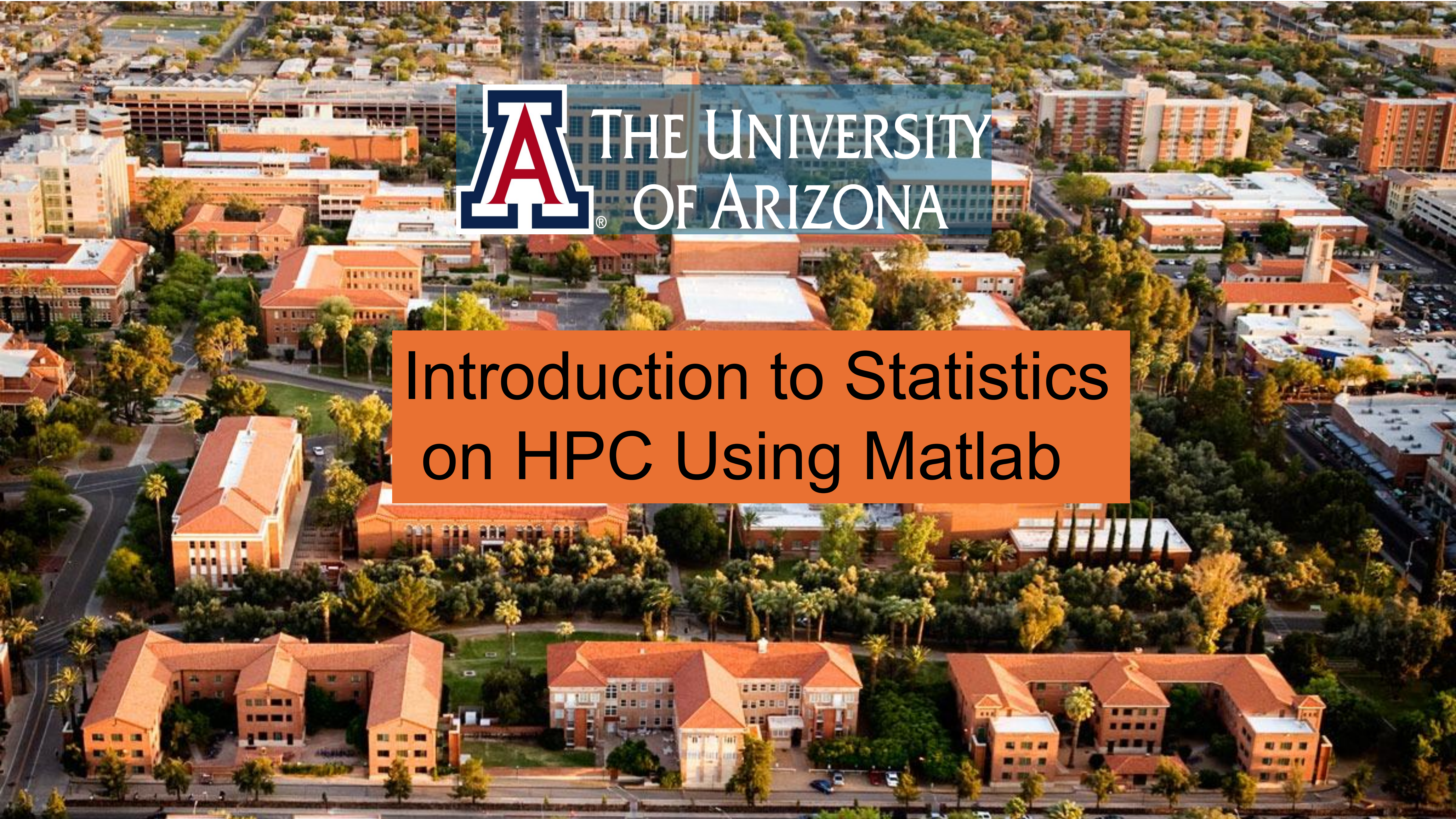


The University of Arizona

Introduction to Statistics on HPC Using Matlab

- Another Introduction to HPC workshop
- Leverages Matlab Statistics Toolbox
- Examples shown on HPC

Statistics and AI

Statistics plays a crucial role in AI by providing the foundation for data analysis, modeling, and decision-making.

Within AI, statistics serves as the backbone for various tasks such as predictive modeling, pattern recognition, and data interpretation.

By leveraging statistical techniques, AI systems can analyze complex data sets, identify patterns, and make informed decisions based on empirical evidence.

David Blackwell

Nvidia has named their new GPU after David Blackwell,
an American mathematician and statistician.

Blackwell provided pioneering work in Bayesian statistics.
Bayesian statistics is a powerful framework that uses probability theory to quantify uncertainty and update beliefs based on new evidence.

In AI applications, Bayesian methods play a crucial role in tasks such as machine learning, pattern recognition, and probabilistic reasoning.

MATLAB®

The Language of Technical Computing

The MATH WORKS Inc.

- Campus license
- HPC license

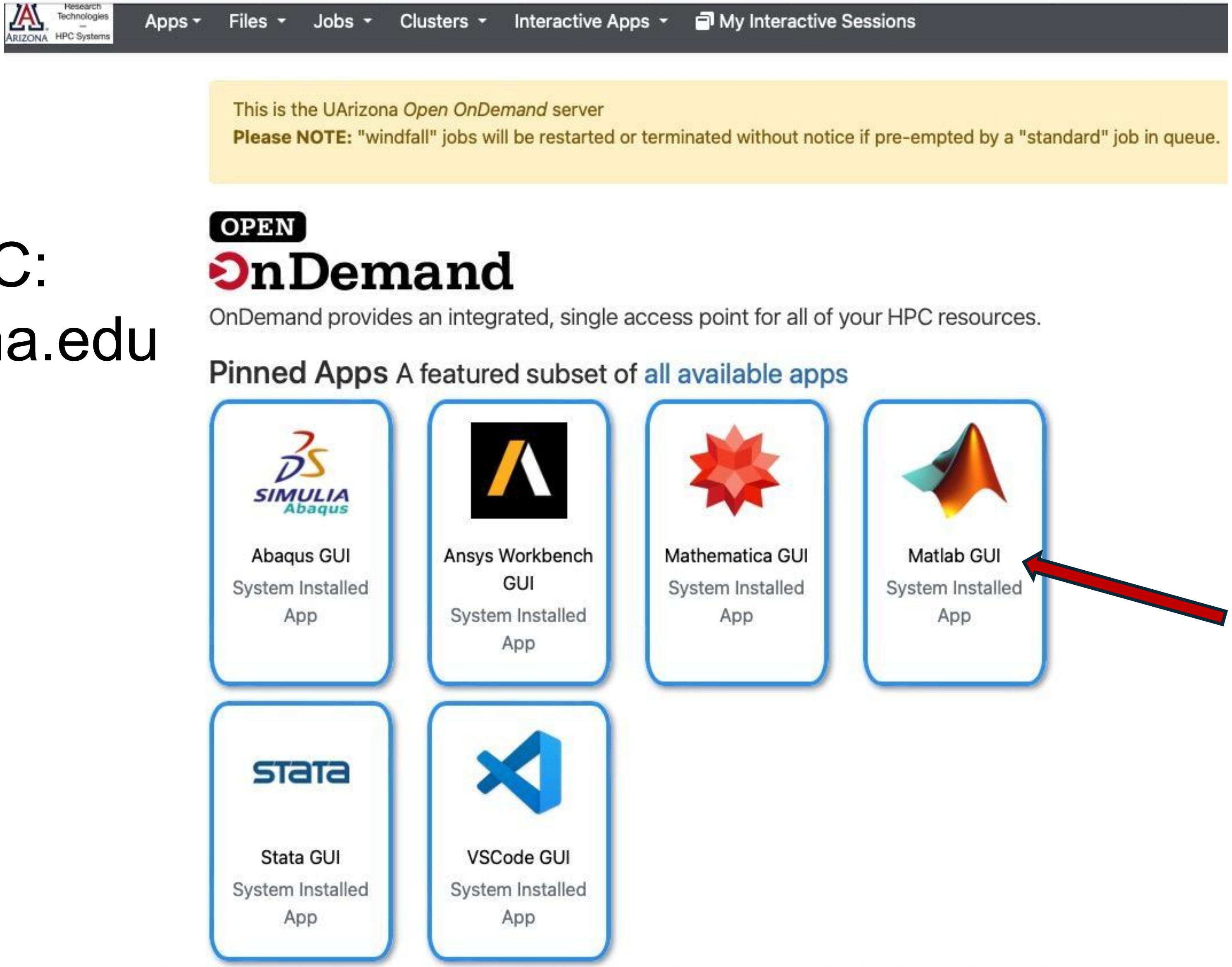
Why Use Matlab on HPC?

1. Use for large datasets
2. Use for multiple concurrent jobs
3. Collaboration workspaces
4. It looks very similar to laptop

Statistics and Machine Learning Toolbox

- Use descriptive statistics, visualizations and clustering.
- Fit probably distributions to data.
- Generate random numbers for:
    - Monte Carlo simulations
    - Regression and Classification to draw inferences
    - Principal Component Analysis, Regularization,
    - Dimensionality Reduction and Feature Selection

Interactive HPC: ood.hpc.arizona.edu



Click on Matlab tile

THE UNIVERSITY OF ARIZONA

MATLAB

Home / My Interactive Sessions / Matlab GUI

Interactive Apps

Desktops

- Interactive Desktop

GUIs

- Abaqus GUI
- Ansys Workbench GUI
- Mathematica GUI
- Matlab GUI
- Stata GUI
- VSCode GUI

Servers

- Jupyter Notebook
- RStudio Server

Matlab GUI

This app will launch the Matlab GUI on a UAz cluster. You will be able to interact with the Matlab GUI through a Turbo VNC based session.

Cluster

ElGato Cluster

☐ I would like to receive an email when the session starts

Matlab GUI version

Default

This defines the version of the application you want to use

Run Time

4

Enter the number of wall clock hours the job is allowed to run.

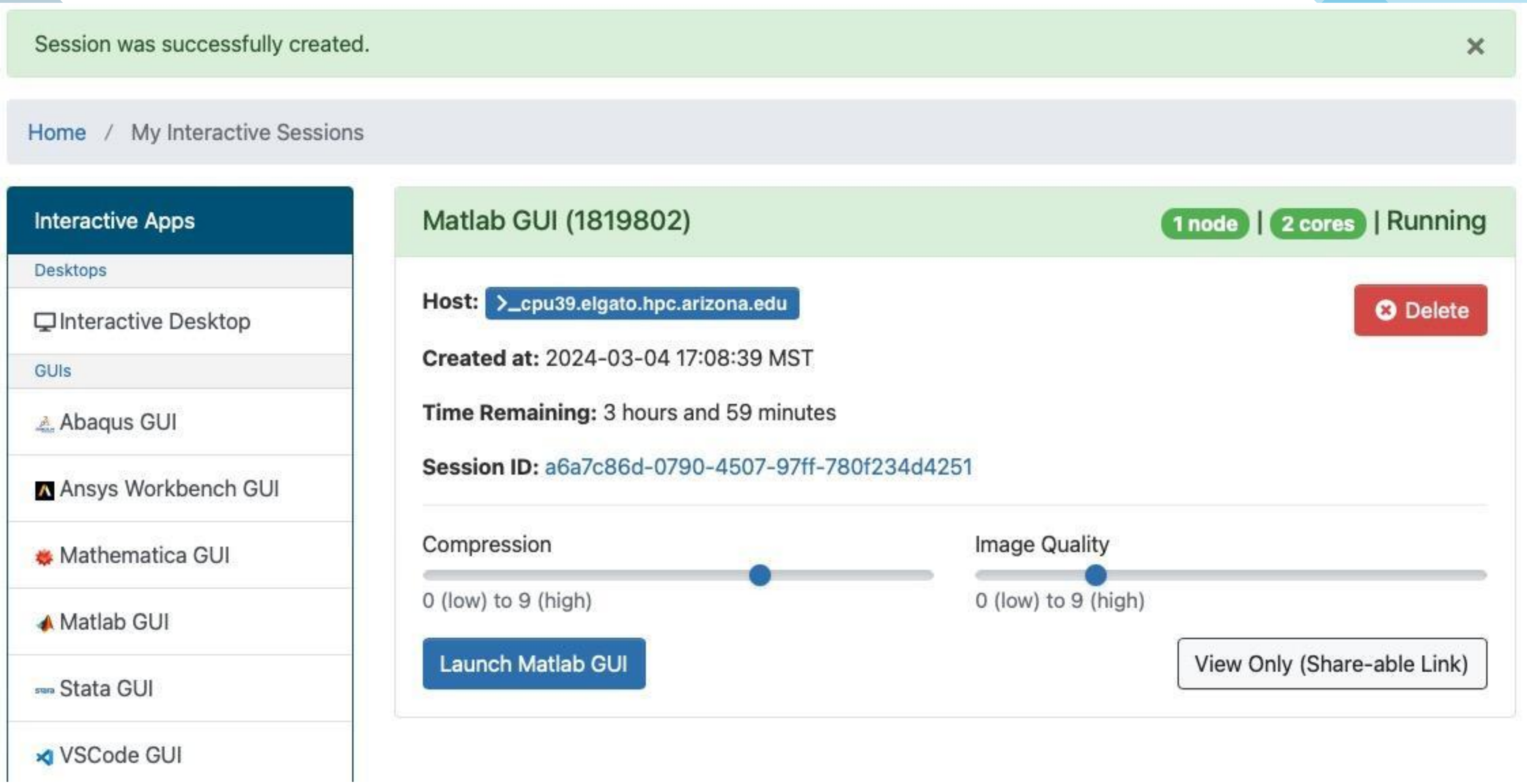
Core count on a single node

2

Enter the number of cores on a single node that the job is allowed to use.

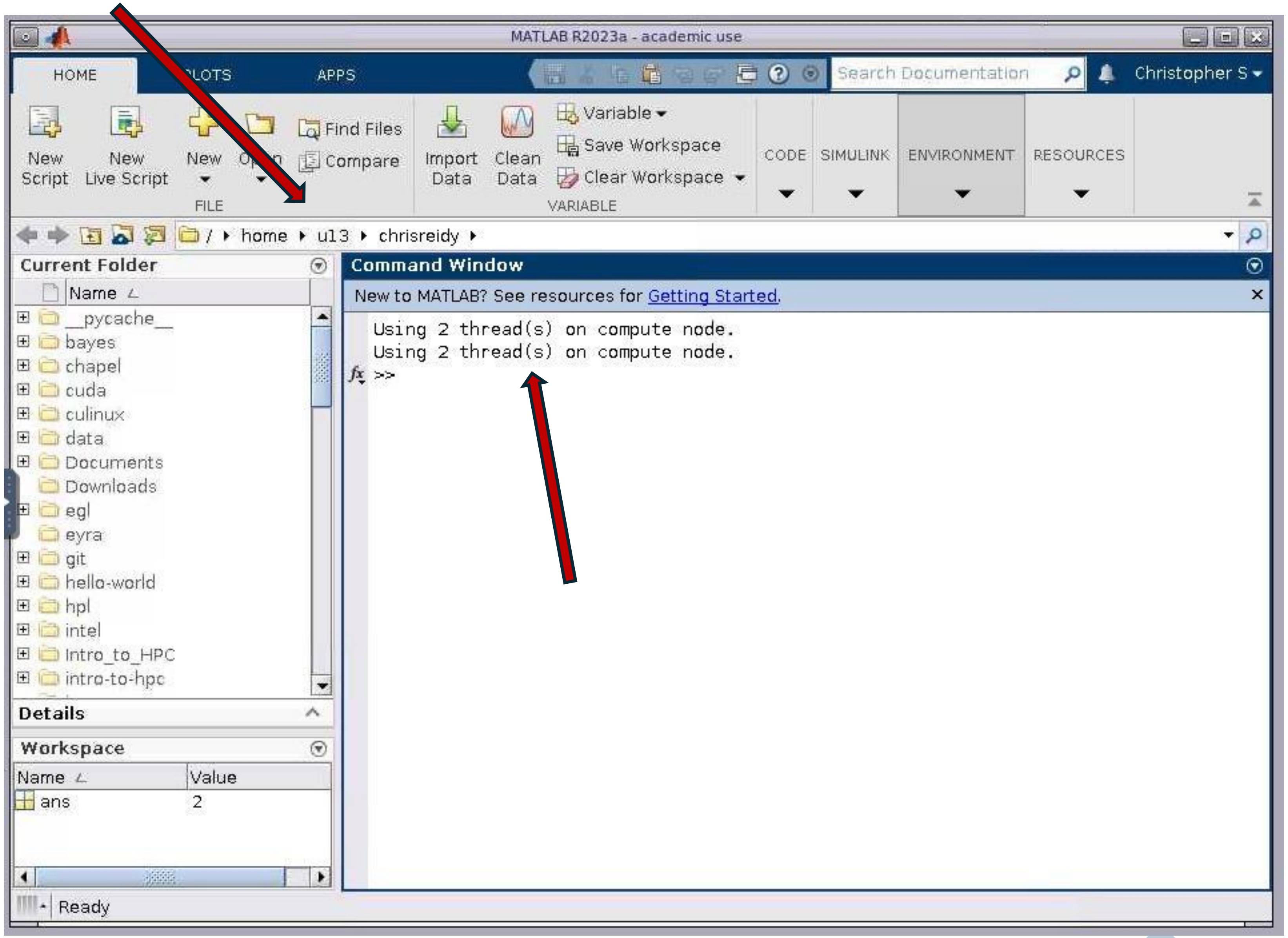
THE UNIVERSITY OF ARIZONA

MATLAB



THE UNIVERSITY OF ARIZONA

MATLAB


MATLAB R2023a - academic use
HOME
PLOTS
APPS
Search Documentation
Christopher S
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Clean Data
Variable
Save Workspace
Clear Workspace
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
/ home u13 chrisreidy
Current Folder
Name
__pycache__
bayes
chapel
cuda
culinux
data
Documents
Downloads
egl
eyra
git
hello-world
hpl
intel
Intro_to_HPC
intro-to-hpc
Details
Workspace
Name
Value
ans
2
Ready
Command Window
New to MATLAB? See resources for Getting Started.
Using 2 thread(s) on compute node.
Using 2 thread(s) on compute node.
>>

Statistics and Machine Learning Toolbox

Analyze and model data using statistics and machine learning

- ❑ Descriptive Statistics
- ❑ Cluster Analysis ANOVA
- ❑ Regression and Classification
- ❑ Dimensionality Reduction
- ❑ Probability Distributions
- ❑ Hypothesis Tests
- ❑ Industrial Statistics
- ❑ Analysis of Big Data
- ❑ Code Generation

Descriptive Statistics
Visualizing Multivariate Data

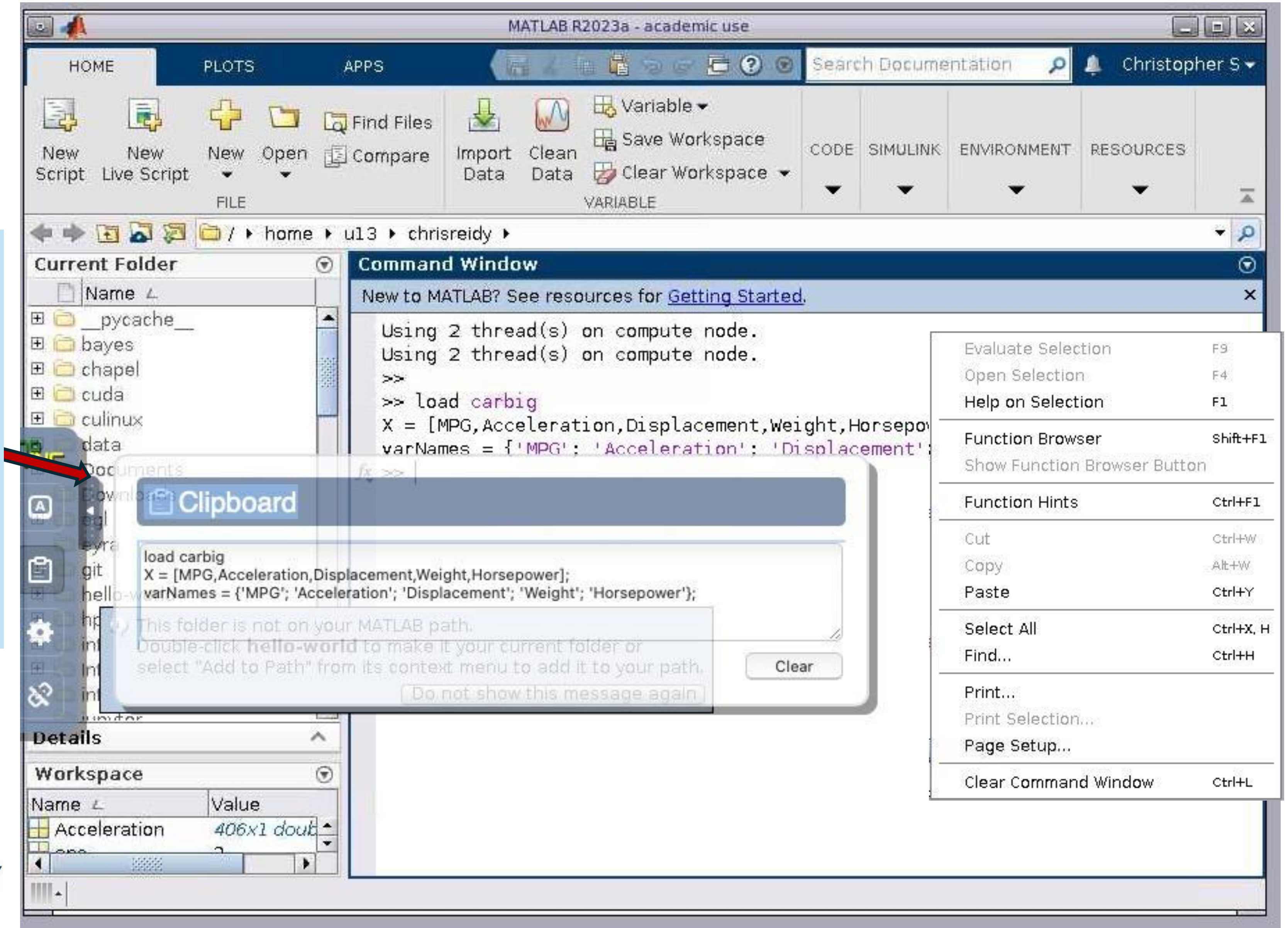
Efficiently summarize and validate large datasets using descriptive statistics and frequency distributions, including measures of central tendency and dispersion, stem plots, bar/pie charts and histograms.

Of the available examples we choose Visualizing Multivariate Data using three different statistical plots.

Descriptive Statistics
Visualizing Multivariate Data

To Cut and Paste:

1. Open Arrow
2. Select Clipboard
3. Paste with cmd-V
4. Right click
5. Paste



THE UNIVERSITY OF ARIZONA

MATLAB

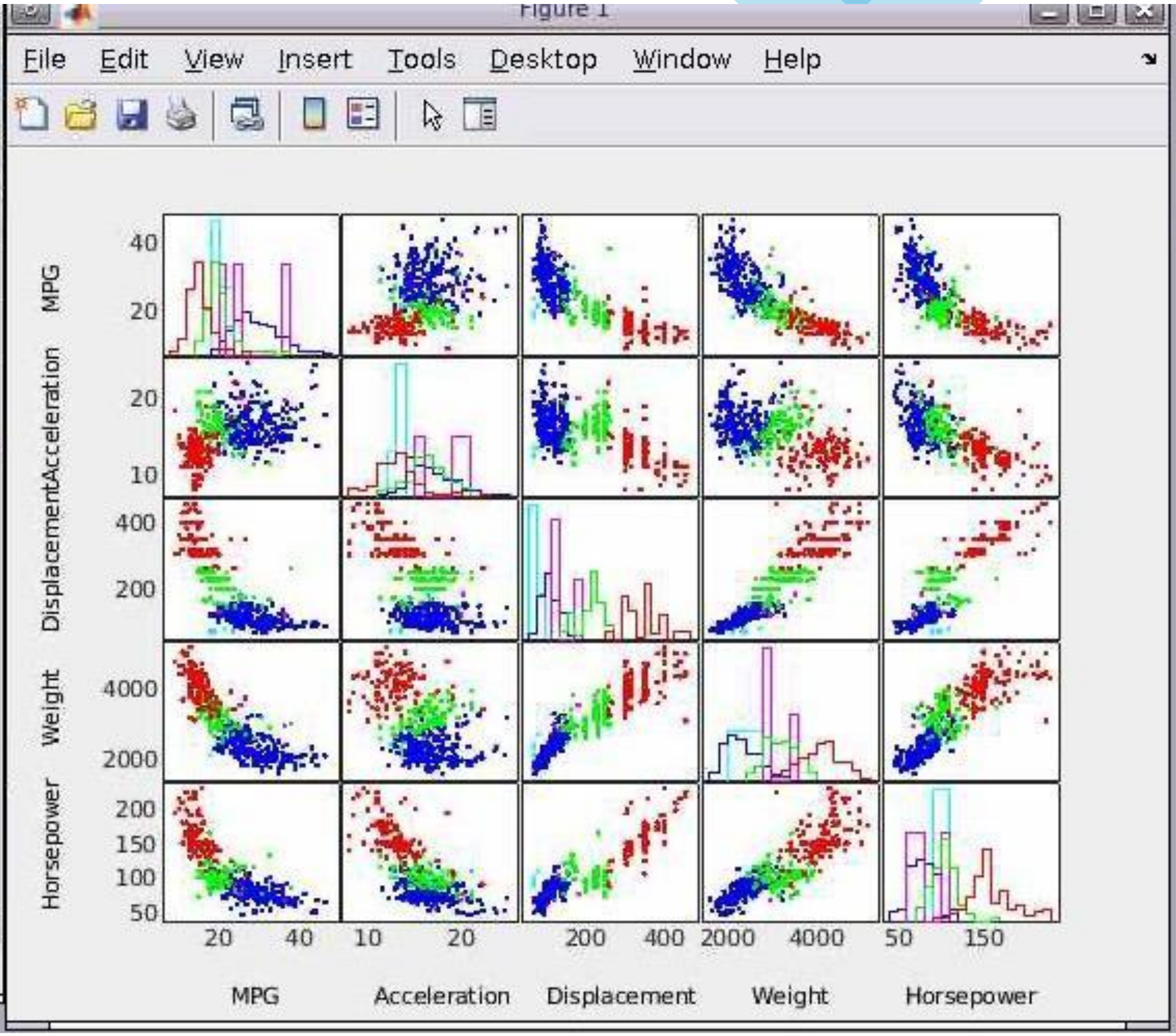
Descriptive Statistics
Visualizing Multivariate Data

Command Window

New to MATLAB? See resources for Getting Started.

```
Using 2 thread(s) on compute node.
Using 2 thread(s) on compute node.
>>
>> load carbig
X = [MPG,Acceleration,Displacement,Weight,Horsepower];
varNames = {'MPG'; 'Acceleration'; 'Displacement'; 'Weight'; 'Horsepower'};
>> figure
gplotmatrix(X,[],Cylinders,['c' 'b' 'm' 'g' 'r'],[],[],false);
text([.08 .24 .43 .66 .83], repmat(-.1,1,5), varNames, 'FontSize',8);
text(repmat(-.12,1,5), [.86 .62 .41 .25 .02], varNames, 'FontSize',8, 'Rotation
Warning: MATLAB has disabled some advanced graphics rendering features by
switching to software OpenGL. For more information, click here.
>>
```

Scatter Plot Matrices

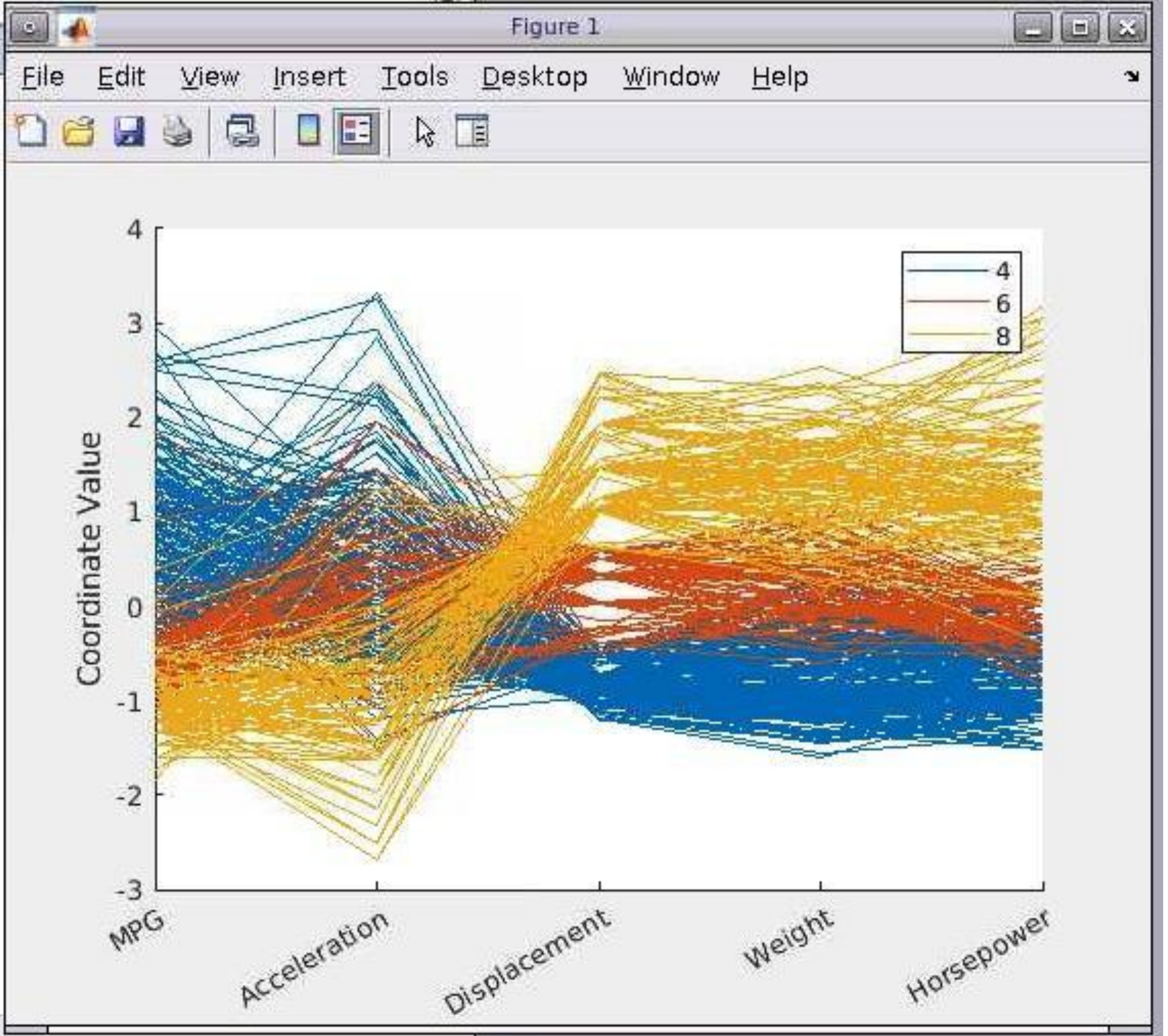


Descriptive Statistics
Visualizing Multivariate Data

Command Window

New to MATLAB? See resources for Getting Started.

```
>> Cyl468 = ismember(Cylinders,[4 6 8]);
parallelcoords(X(Cyl468,:), 'group',Cylinders(Cyl468), ...
               'standardize','on', 'labels',varNames)
fx >>
```



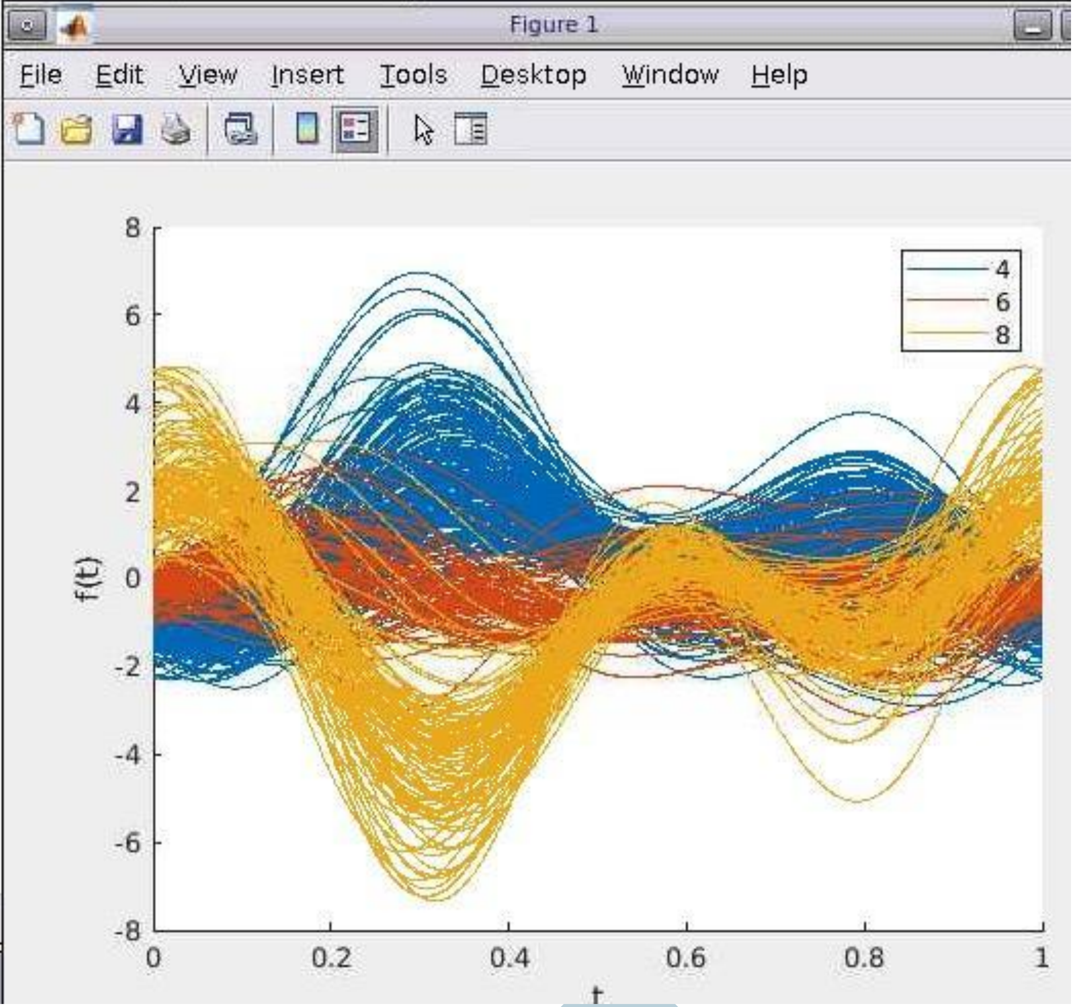
Parallel Coordinates Plot

Descriptive Statistics
Visualizing Multivariate Data

Command Window

New to MATLAB? See resources for Getting Started.

```
>> andrewsplot(X(Cyl468,:), 'group',Cylinders(Cyl468), 'standardize','on')
>>
>> t1 = 1/3;
[1/sqrt(2) sin(2*pi*t1) cos(2*pi*t1) sin(4*pi*t1) cos(4*pi*t1)]

ans =

    0.7071    0.8660   -0.5000   -0.8660   -0.5000

>>
```



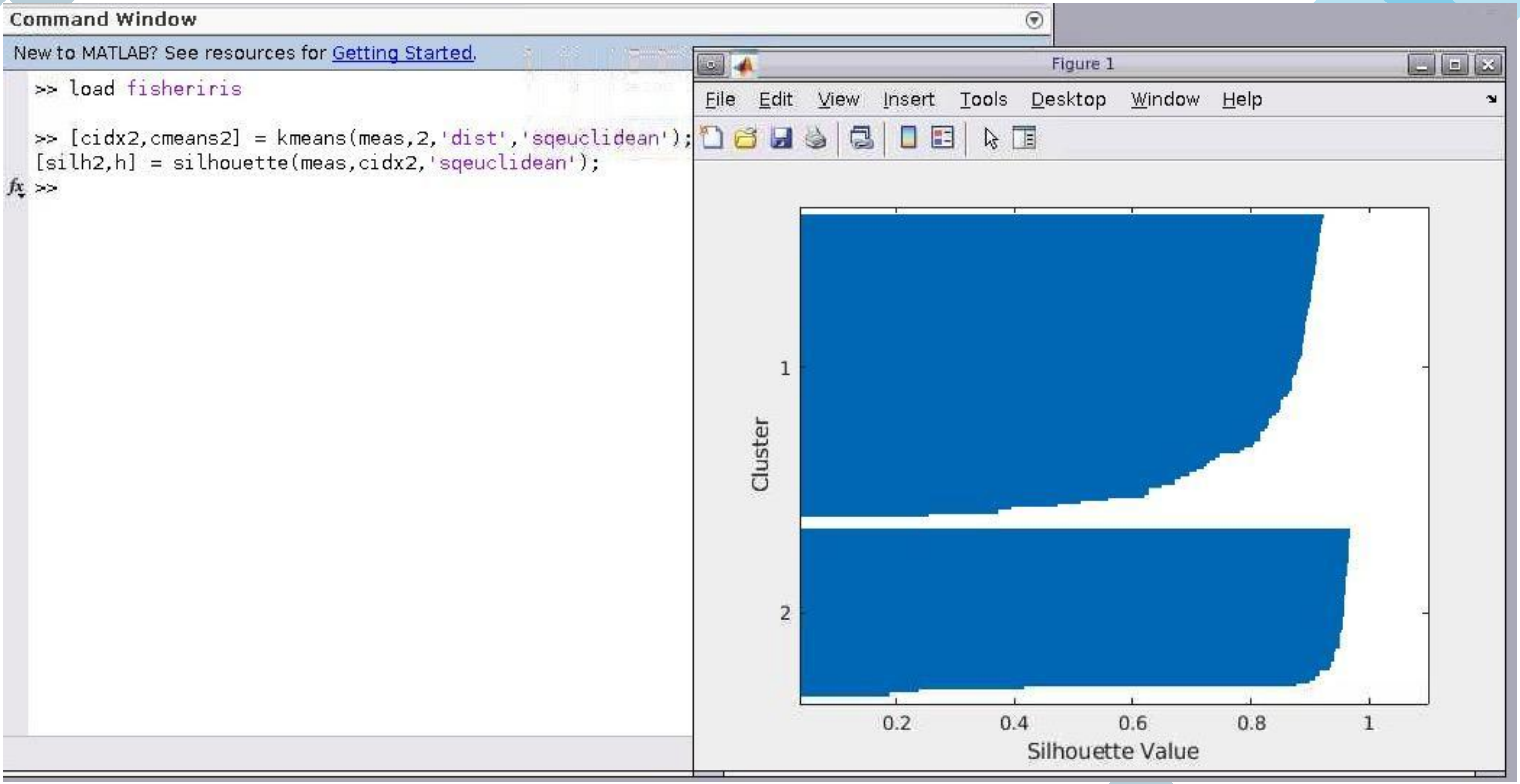
Andrews Plot

Cluster Analysis
K-Means Clustering

Data often naturally fall into groups (or clusters) of observations, where the characteristics of objects in the same cluster are similar.

This K-means example is extensive, and we just demonstrate the first features.

Cluster Analysis
K-Means Clustering

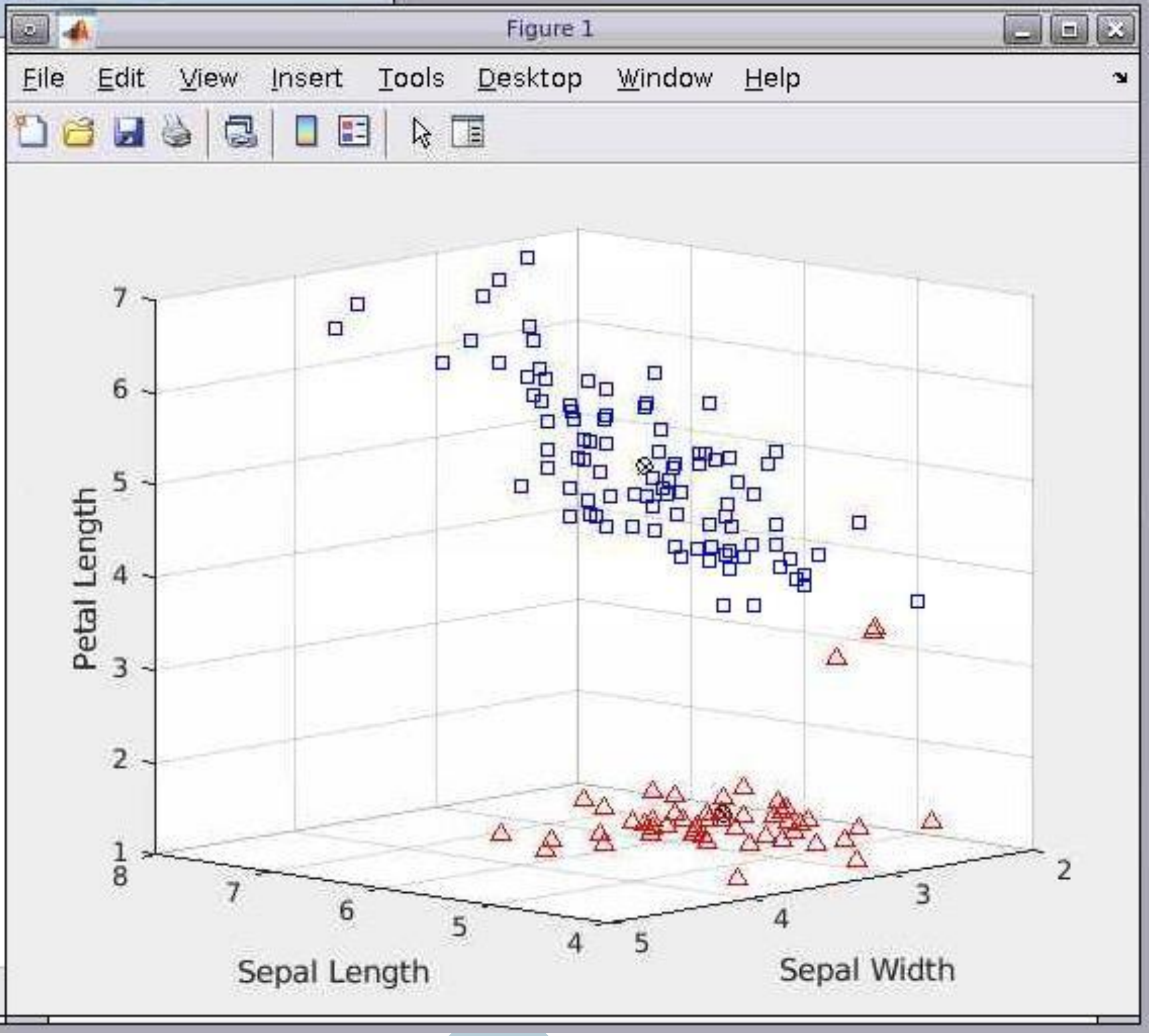


Cluster Analysis
K-Means Clustering

Command Window

New to MATLAB? See resources for Getting Started.

```
>> load fisheriris

>> [cidx2,cmeans2] = kmeans(meas,2,'dist','sqeuclidean');
[silh2,h] = silhouette(meas,cidx2,'sqeuclidean');
>> ptsymb = {'bs','r^','md','go','c+'};
for i = 1:2
    clust = find(cidx2==i);
    plot3(meas(clust,1),meas(clust,2),meas(clust,3),ptsymb{i});
    hold on
end
plot3(cmeans2(:,1),cmeans2(:,2),cmeans2(:,3),'ko');
plot3(cmeans2(:,1),cmeans2(:,2),cmeans2(:,3),'kx');
hold off
xlabel('Sepal Length');
ylabel('Sepal Width');
zlabel('Petal Length');
view(-137,10);
grid on
>>
```



ANOVA
Analysis of Variance

In an ANOVA model, each grouping variable represents a fixed factor.
The levels of that factor are a fixed set of values.
The goal is to determine whether different factor levels lead to significantly different response values or outcomes.

This example starts the ANOVA process.
It continues with F-Statistics and Variance Components using mean squares and confidence bounds

ANOVA
Analysis of Variance

Part 1

Command Window

New to MATLAB? See resources for Getting Started.

```
>> clear
>> load mileage
>> factory  = repmat(1:3,6,1);
>> carmod = [ones(3,3); 2*ones(3,3)];
>> mileage = mileage(:);
factory = factory(:);
carmod = carmod(:);
[mileage factory carmod]

ans =

   33.3000    1.0000    1.0000
   33.4000    1.0000    1.0000
   32.9000    1.0000    1.0000
   32.6000    1.0000    2.0000
   32.5000    1.0000    2.0000
   33.0000    1.0000    2.0000
   34.5000    2.0000    1.0000
   34.8000    2.0000    1.0000
   33.8000    2.0000    1.0000
   33.4000    2.0000    2.0000
   33.7000    2.0000    2.0000
   33.9000    2.0000    2.0000
   37.4000    3.0000    1.0000
   36.8000    3.0000    1.0000
   37.6000    3.0000    1.0000
```

ANOVA
Analysis of Variance

Part 2

Command Window

New to MATLAB? See resources for Getting Started.

```
ans =

   33.3000    1.0000    1.0000
   33.4000    1.0000    1.0000
   32.9000    1.0000    1.0000
   32.6000    1.0000    2.0000
   32.5000    1.0000    2.0000
   33.0000    1.0000    2.0000
   34.5000    2.0000    1.0000
   34.8000    2.0000    1.0000
   33.8000    2.0000    1.0000
   33.4000    2.0000    2.0000
   33.7000    2.0000    2.0000
   33.9000    2.0000    2.0000
   37.4000    3.0000    1.0000
   36.8000    3.0000    1.0000
   37.6000    3.0000    1.0000
   36.6000    3.0000    2.0000
   37.0000    3.0000    2.0000
   36.7000    3.0000    2.0000

>> [pvals,tbl,stats] = anovan(mileage, {factory carmod}, ...
'model',2, 'random',1,'varnames',{'Factory' 'Car Model'});
>>
```

Figure 1: N-Way ANOVA

File Edit View Insert Tools Desktop Window Help

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Factory	53.3511	2	26.6756	1333.78	0.0007
Car Model	1.445	1	1.445	72.25	0.0136
Factory:Car Model	0.04	2	0.02	0.18	0.8411
Error	1.3667	12	0.1139		
Total	56.2028	17			

Constrained (Type III) sums of squares.

Statistics and Machine Learning Toolbox
Regression
Bayesian Analysis for a Logistic Regression Model

Use the Regression Learner app or programmatically train and assess models such as linear regression, Gaussian processes, support vector machines, neural networks, and ensembles.

This example starts on Bayesian inferences for a logistic regression model. Try the whole example on your own.

Statistics and Machine Learning Toolbox
Regression
Bayesian Analysis for a Logistic Regression Model

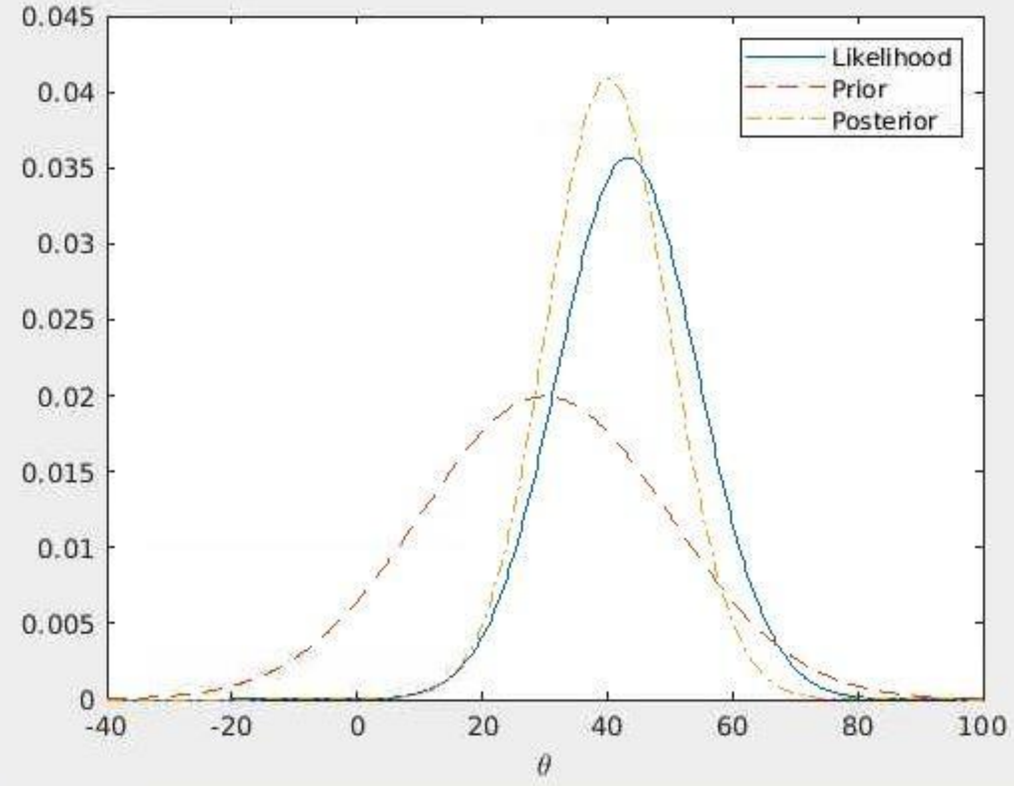
Command Window

New to MATLAB? See resources for Getting Started.

```
>> rng(0,'twister');
>> n = 20;
sigma = 50;
x = normrnd(10,sigma,n,1);
mu = 30;
tau = 20;
theta = linspace(-40, 100, 500);
y1 = normpdf(mean(x),theta,sigma/sqrt(n));
y2 = normpdf(theta,mu,tau);
postMean = tau^2*mean(x)/(tau^2+sigma^2/n) + sigma^2*mu/n/(tau^2+sigma^2/n);
postSD = sqrt(tau^2*sigma^2/n/(tau^2+sigma^2/n));
y3 = normpdf(theta, postMean,postSD);
plot(theta,y1,'-', theta,y2,'--', theta,y3,'-.')
legend('Likelihood','Prior','Posterior')
xlabel('\theta')
>>
```

Figure 1

File Edit View Insert Tools Desktop Window Help

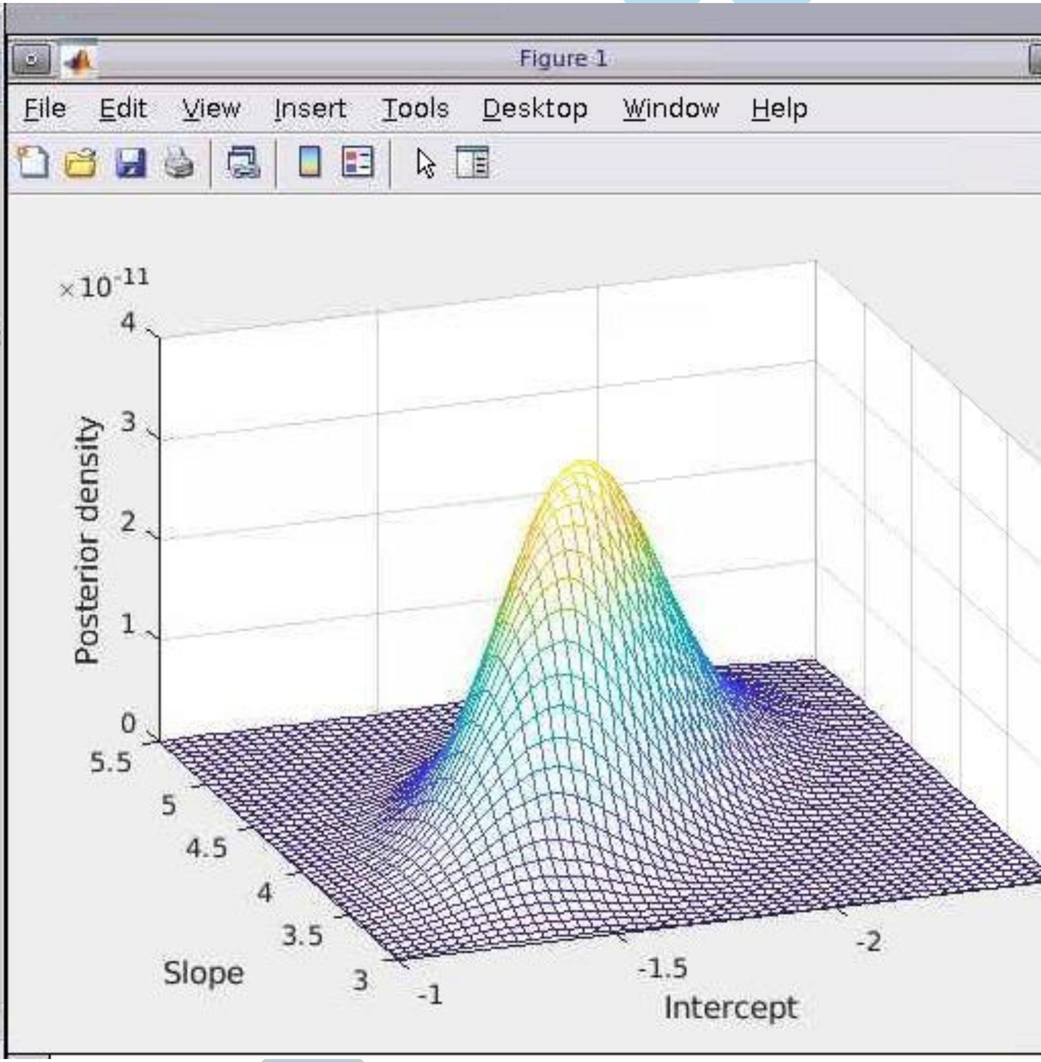


Statistics and Machine Learning Toolbox
Regression
Bayesian Analysis for a Logistic Regression Model

Command Window

New to MATLAB? See resources for Getting Started.

```
>> % A set of car weights
weight = [2100 2300 2500 2700 2900 3100 3300 3500 3700 3900 4100 4300]';
weight = (weight-2800)/1000;     % recenter and rescale
% The number of cars tested at each weight
total = [48 42 31 34 31 21 23 23 21 16 17 21]';
% The number of cars that have poor mpg performances at each weight
poor = [1 2 0 3 8 8 14 17 19 15 17 21]';logitp = @(b,x) exp(b(1)+b(2).*x)./(1+exp(b(1)+b
>> prior1 = @(b1) normpdf(b1,0,20);    % prior for intercept
prior2 = @(b2) normpdf(b2,0,20);    % prior for slope
>> post = @(b) prod(binopdf(poor,total,logitp(b,weight))) ...  % likelihood
         * prior1(b(1)) * prior2(b(2));                        % priors
>> b1 = linspace(-2.5, -1, 50);
b2 = linspace(3, 5.5, 50);
simpost = zeros(50,50);
for i = 1:length(b1)
    for j = 1:length(b2)
        simpost(i,j) = post([b1(i), b2(j)]);
    end;
end;
mesh(b2,b1,simpost)
xlabel('Slope')
ylabel('Intercept')
zlabel('Posterior density')
view(-110,30)
>>
```



THE UNIVERSITY OF ARIZONA

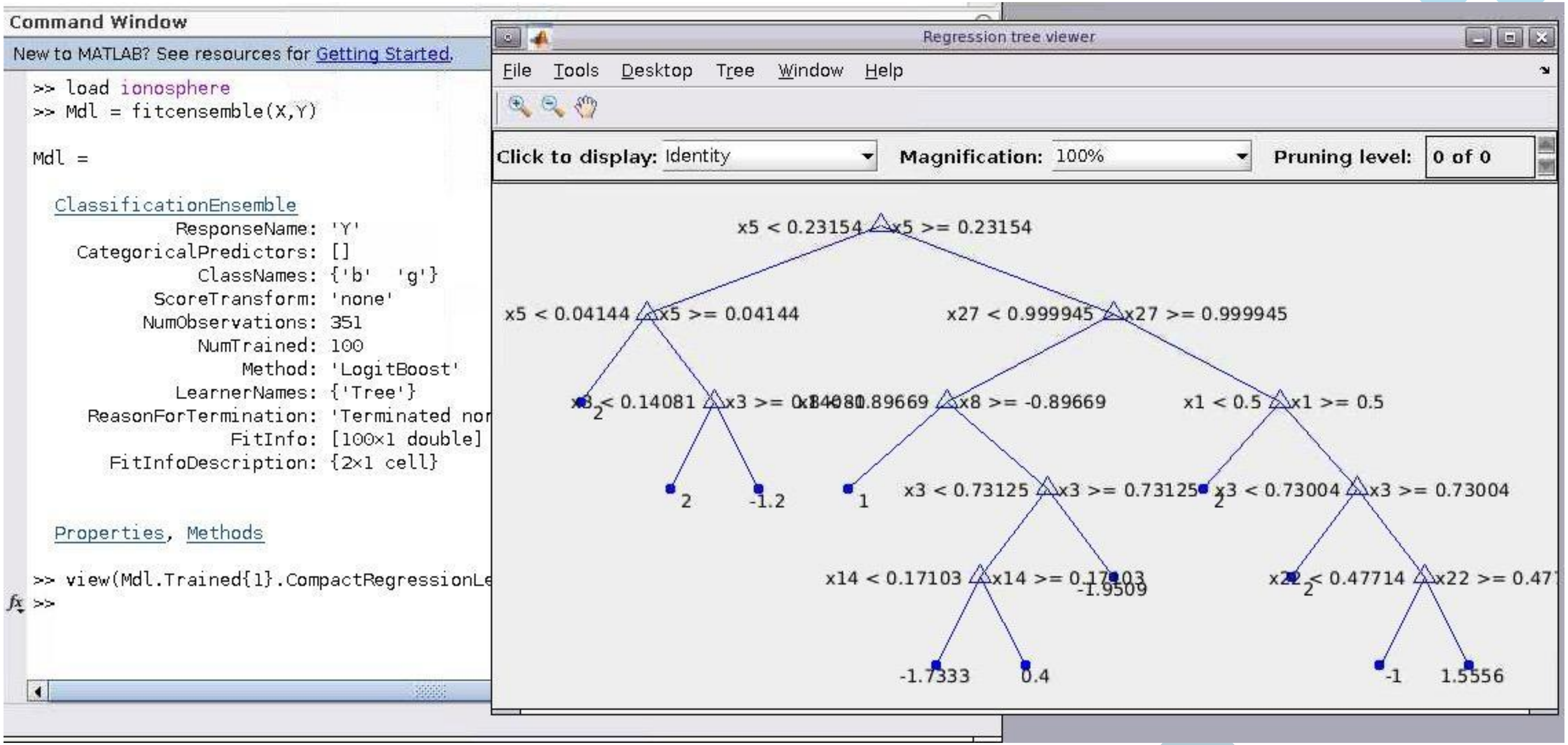
MATLAB

Statistics and Machine Learning Toolbox
Classification
Train Classification Ensemble

Use the Classification Learner app or programmatically train
and validate models such as logistic regression,
support vector machines, boosted trees, and shallow neural networks.

This example shows how to create a classification tree ensemble.

Statistics and Machine Learning Toolbox
Classification
Train Classification Ensemble

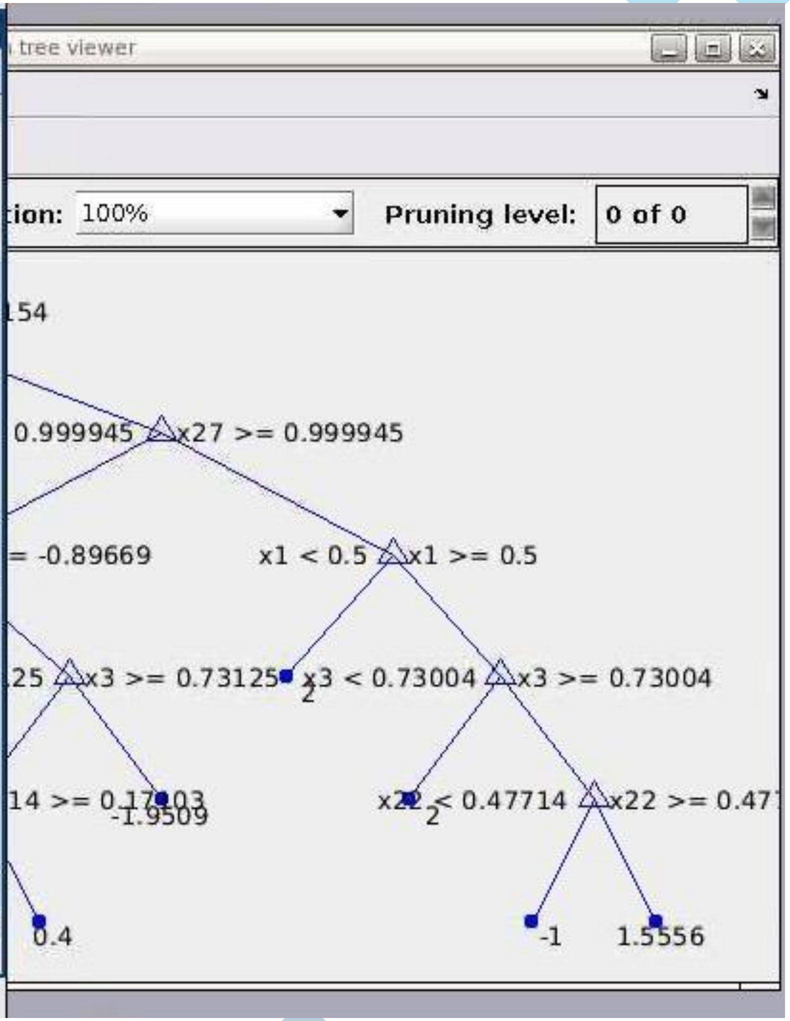


Statistics and Machine Learning Toolbox
Classification
Train Classification Ensemble

Command Window

New to MATLAB? See resources for Getting Started.

```
  ClassificationEnsemble
            ResponseName: 'Y'
   CategoricalPredictors: []
              ClassNames: {'b'  'g'}
          ScoreTransform: 'none'
         NumObservations: 351
              NumTrained: 100
                  Method: 'LogitBoost'
            LearnerNames: {'Tree'}
    ReasonForTermination: 'Terminated normally after completing the requested number
                 FitInfo: [100×1 double]
      FitInfoDescription: {2×1 cell}


  Properties, Methods

>> view(Mdl.Trained{1}.CompactRegressionLearner,'Mode','graph');
>> label = predict(Mdl,mean(X))

label =

  1×1 cell array

    {'g'}

>>
```



THE UNIVERSITY OF ARIZONA

MATLAB

Statistics and Machine Learning Toolbox
Dimensionality Reduction and Feature Extraction

This toolbox has many tools including:

- Sequential feature selection
- Partial least squares regression
- Principal components regression
- Principal components analysis
- Discriminant Analysis Classifier
- Random forest predictors
- Feature extraction workflow
- Extract mixed signals
- Visualize high-dimensional data
- Factor analysis
- Nonnegative matrix factorization
- Multidimensional scaling

Statistics and Machine Learning Toolbox
Dimensionality Reduction and Feature Extraction
Classical Multidimension Scaling

Part 1

Command Window

New to MATLAB? See resources for Getting Started.

```
>> rng default;  % For reproducibility
X = [normrnd(0,1,10,3),normrnd(0,.1,10,1)];
D = pdist(X,'euclidean');
>> [Y,eigvals] = cmdscale(D);
>> format short g
[eigvals eigvals/max(abs(eigvals))]

ans =

        35.41            1
       11.158      0.31511
       1.6894      0.04771
       0.1436    0.0040553
   5.4637e-15    1.543e-16
   3.2157e-15   9.0813e-17
   2.2212e-15   6.2727e-17
   1.3179e-15   3.7219e-17
  -2.3377e-15  -6.6019e-17
    -3.47e-15  -9.7995e-17
```

THE UNIVERSITY OF ARIZONA

MATLAB

Statistics and Machine Learning Toolbox
Dimensionality Reduction and Feature Extraction
Classical Multidimension Scaling

Part 2

```
>> maxerr4 = max(abs(D - pdist(Y)))   % Exact reconstruction

maxerr4 =

   1.7764e-15

>> maxerr3 = max(abs(D - pdist(Y(:,1:3))))  % Good reconstruction in 3D

maxerr3 =

     0.043142

>>
>> maxerr2 = max(abs(D - pdist(Y(:,1:2))))  % Poor reconstruction in 2D

maxerr2 =

      0.98315

>> max(max(D))

ans =

       5.8974

>>
```

THE UNIVERSITY OF ARIZONA

MATLAB

Probability Distributions
Compare Multiple Distribution Fits

Fit continuous and discrete distributions,
use statistical plots to evaluate goodness-of-fit,
and compute probability density functions.

This example shows how to fit multiple probability distribution objects to the same set of sample data.

Probability Distributions
Compare Multiple Distribution Fits

Part 1

Command Window

New to MATLAB? See resources for Getting Started.

```
>> load carsmall
>> Origin = categorical(cellstr(Origin));
MPG2 = MPG(Origin~='Italy');
Origin2 = Origin(Origin~='Italy');
Origin2 = removecats(Origin2,'Italy');
>> [WeiByOrig,Country] = fitdist(MPG2,'weibull','by',Origin2);
[NormByOrig,Country] = fitdist(MPG2,'normal','by',Origin2);
[LogByOrig,Country] = fitdist(MPG2,'logistic','by',Origin2);
[KerByOrig,Country] = fitdist(MPG2,'kernel','by',Origin2);
>> WeiByOrig

WeiByOrig =

  1×5 cell array

  Columns 1 through 2

    {1×1 prob.WeibullDistribution}    {1×1 prob.WeibullDistribution}

  Columns 3 through 4

    {1×1 prob.WeibullDistribution}    {1×1 prob.WeibullDistribution}

  Column 5

    {1×1 prob.WeibullDistribution}
```

Probability Distributions
Compare Multiple Distribution Fits

Part 2

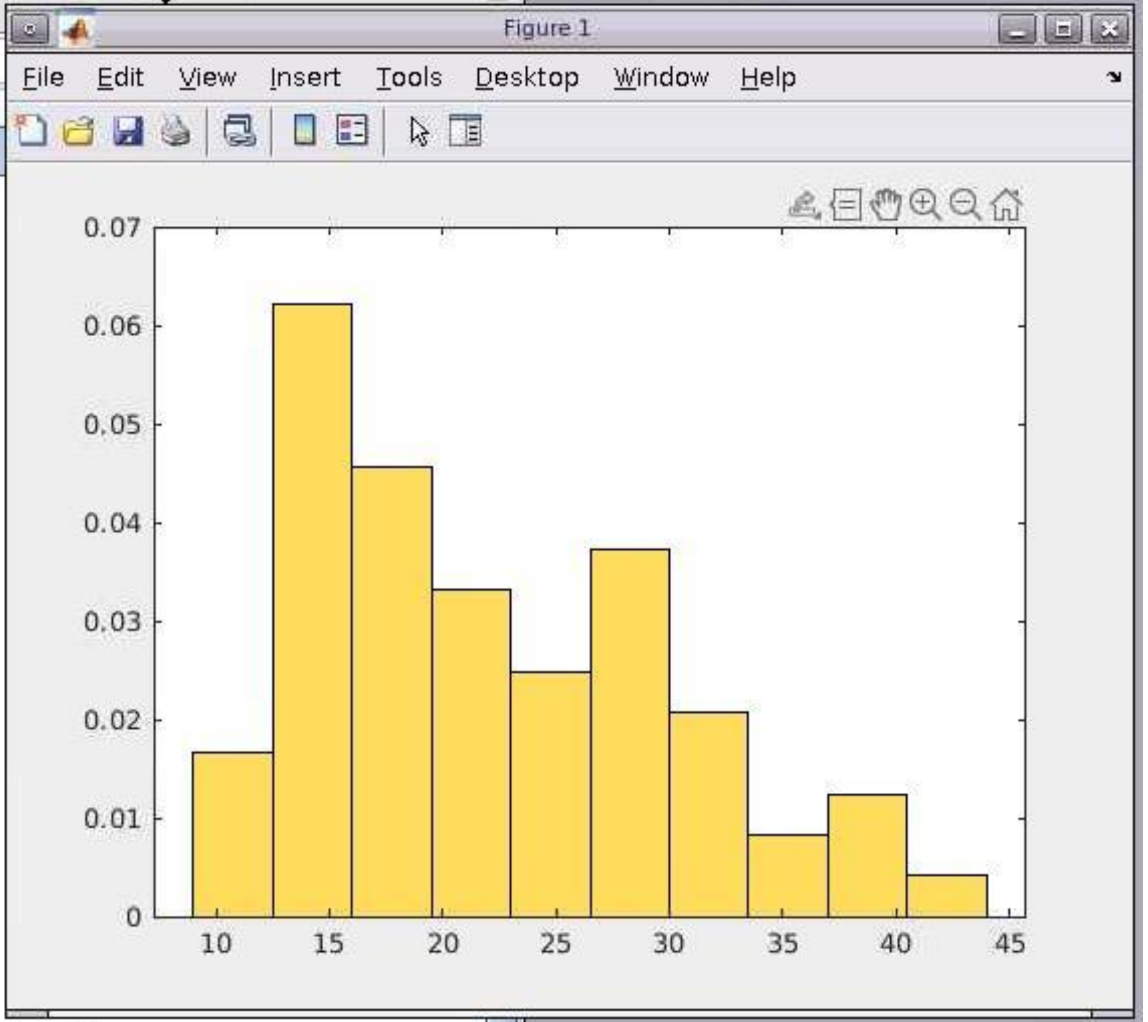
VARIABLE

chrisreidy ▸

Command Window

New to MATLAB? See resources for Getting Started.

```
>> Country

Country =

  5×1 cell array

    {'France' }
    {'Germany'}
    {'Japan'  }
    {'Sweden' }
    {'USA'    }

>> WeiUSA = WeiByOrig{5};
NormUSA = NormByOrig{5};
LogUSA = LogByOrig{5};
KerUSA = KerByOrig{5};

x = 0:1:50;
pdf_Wei = pdf(WeiUSA,x);
pdf_Norm = pdf(NormUSA,x);
pdf_Log = pdf(LogUSA,x);
pdf_Ker = pdf(KerUSA,x);
>> data = MPG(Origin2=='USA');
figure
histogram(data,10,'Normalization','pdf','FaceColor',[1,0.8,0]);
fx >>
```



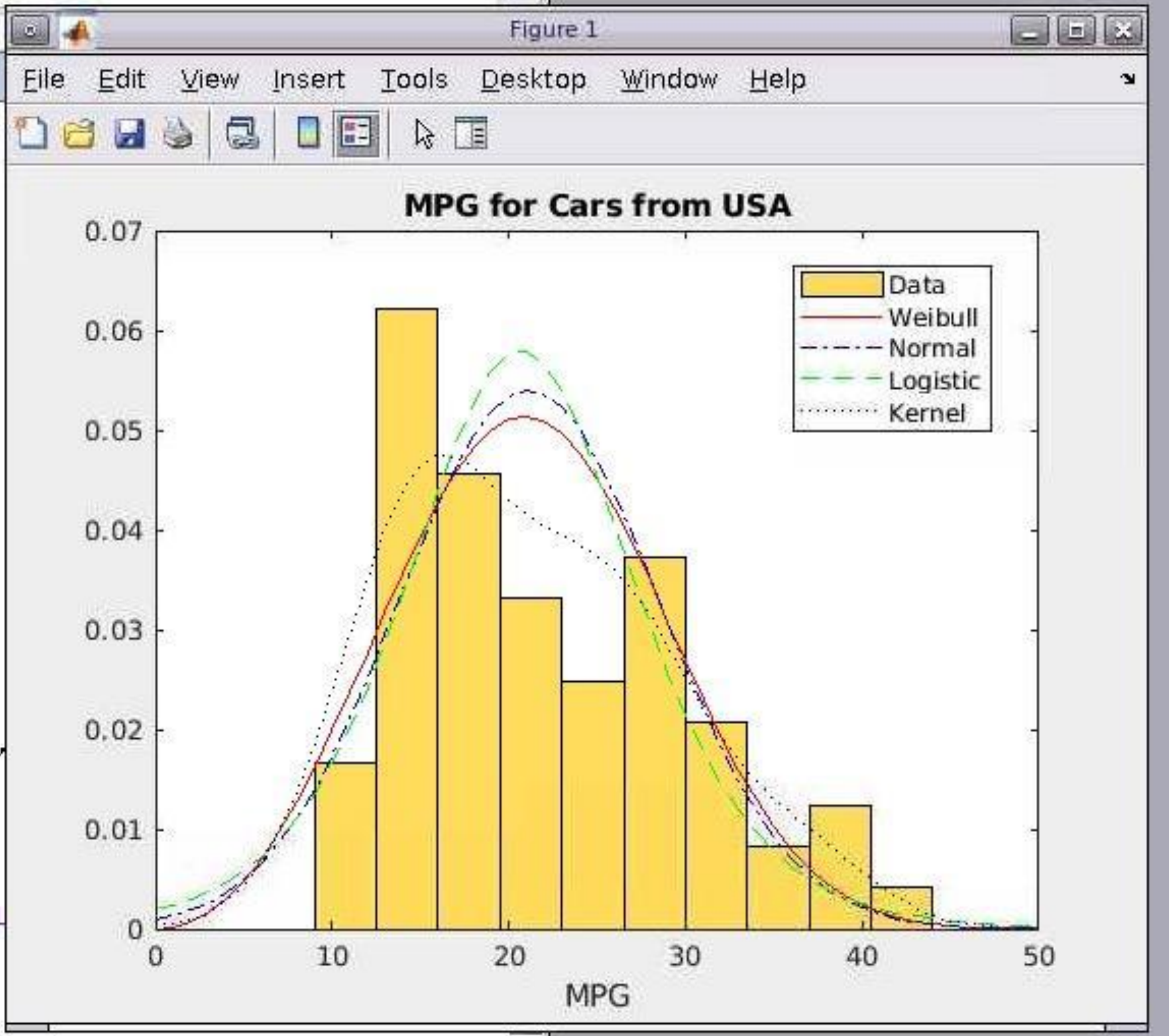
Probability Distributions
Compare Multiple Distribution Fits

Part 3

Command Window

New to MATLAB? See resources for Getting Started.

```
    {'France' }
    {'Germany'}
    {'Japan'  }
    {'Sweden' }
    {'USA'    }

>> WeiUSA = WeiByOrig{5};
NormUSA = NormByOrig{5};
LogUSA = LogByOrig{5};
KerUSA = KerByOrig{5};

x = 0:1:50;
pdf_Wei = pdf(WeiUSA,x);
pdf_Norm = pdf(NormUSA,x);
pdf_Log = pdf(LogUSA,x);
pdf_Ker = pdf(KerUSA,x);
>> data = MPG(Origin2=='USA');
figure
histogram(data,10,'Normalization','pdf','FaceColor',[1,
>> line(x,pdf_Wei,'LineStyle','-','Color','r')
line(x,pdf_Norm,'LineStyle','-.','Color','b')
line(x,pdf_Log,'LineStyle','--','Color','g')
line(x,pdf_Ker,'LineStyle',':','Color','k')
legend('Data','Weibull','Normal','Logistic','Kernel','L
title('MPG for Cars from USA')
xlabel('MPG')
fx >>
```



THE UNIVERSITY OF ARIZONA

MATLAB

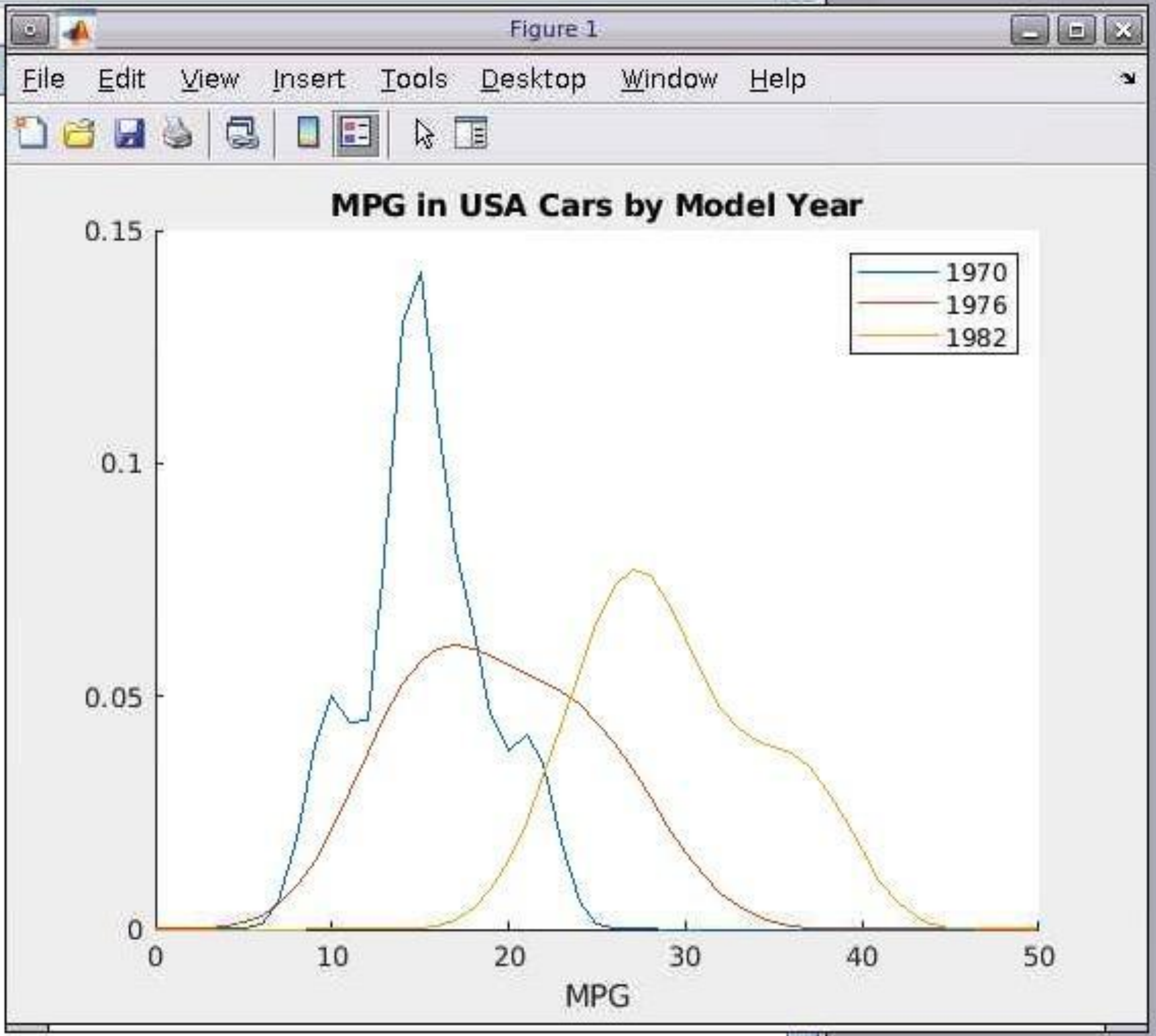
Probability Distributions
Compare Multiple Distribution Fits

Part 4

Command Window

New to MATLAB? See resources for Getting Started.

```
Names =

  14×1 cell array

    {'France↵70' }
    {'France↵76' }
    {'Germany↵70'}
    {'Germany↵76'}
    {'Germany↵82'}
    {'Italy↵76'  }
    {'Japan↵70'  }
    {'Japan↵76'  }
    {'Japan↵82'  }
    {'Sweden↵70' }
    {'Sweden↵76' }
    {'USA↵70'    }
    {'USA↵76'    }
    {'USA↵82'    }

>> figure
hold on
for i = 12 : 14
    plot(x,pdf(KerByYearOrig{i},x))
end
legend('1970','1976','1982')
title('MPG in USA Cars by Model Year')
xlabel('MPG')
hold off
fx >>
```



THE UNIVERSITY OF ARIZONA

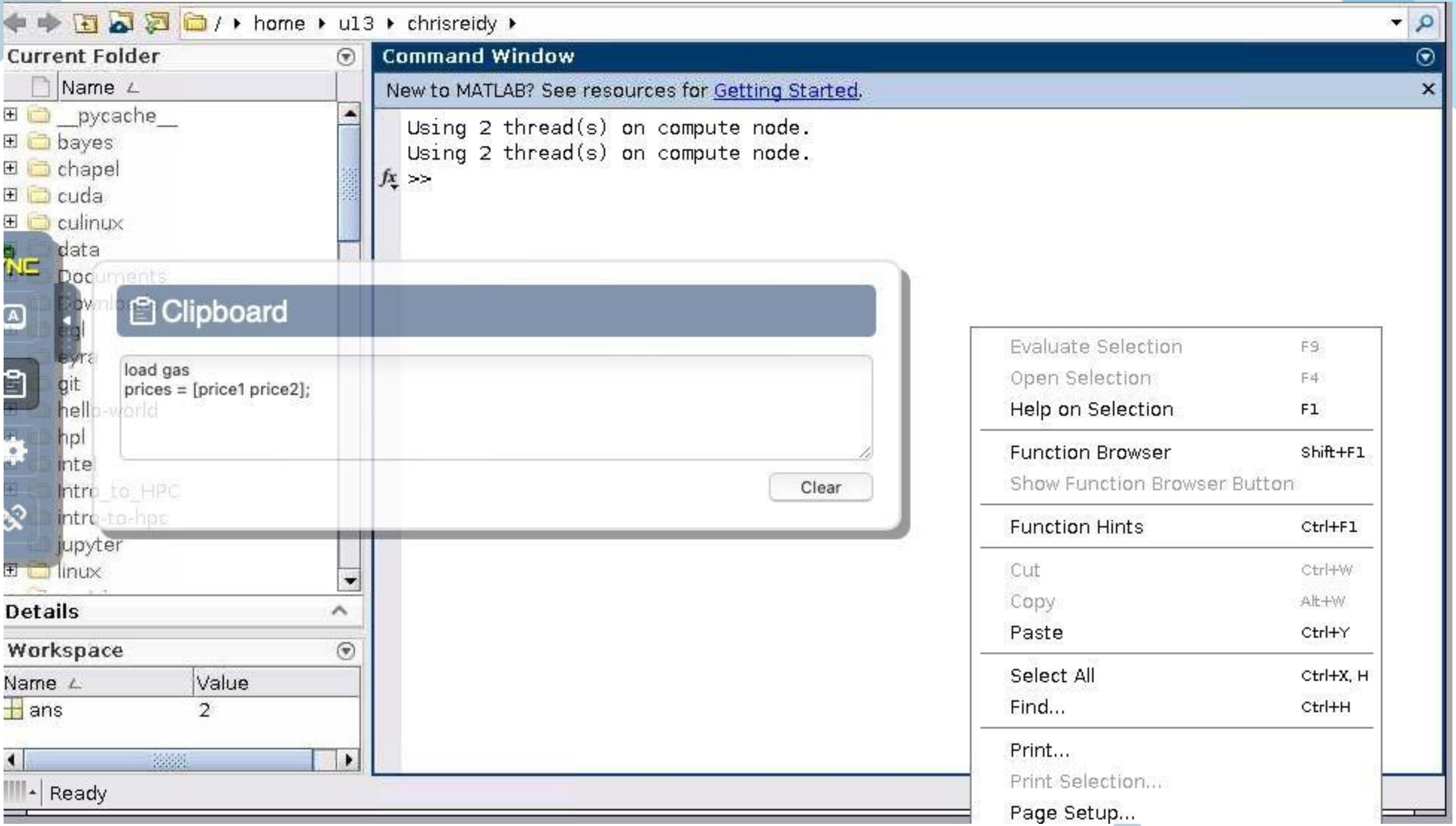
MATLAB

Hypothesis Testing

Draw inferences about a population
based on statistical evidence from a sample.
Perform parametric and nonparametric tests.

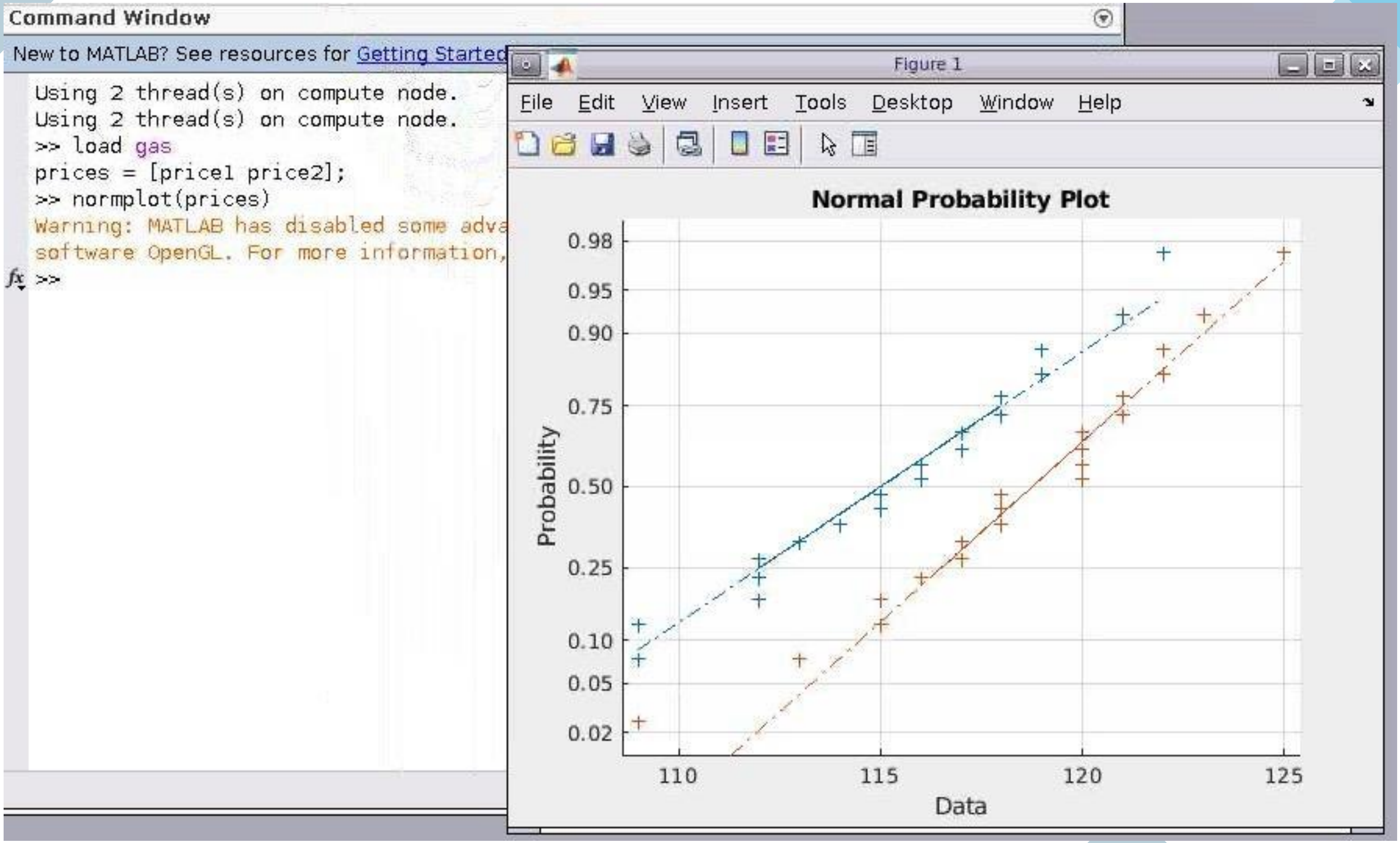
This example shows how to use hypothesis testing
to analyze gas prices measured across the state of Massachusetts
during two separate months.

Hypothesis Testing



Hypothesis Testing

Part 1



Hypothesis Testing

Part 2

Command Window

New to MATLAB? See resources for Getting Started.

```
Using 2 thread(s) on compute node.
Using 2 thread(s) on compute node.
>> load gas
prices = [price1 price2];
>> normplot(prices)
Warning: MATLAB has disabled some advanced graphics rendering features by switching to
software OpenGL. For more information, click here.
>> lillietest(price1)

ans =

     0

>> lillietest(price2)

ans =

     0

>> sample_means = mean(prices)

sample_means =

  115.1500  118.5000

>>
```

THE UNIVERSITY OF ARIZONA

MATLAB

Hypothesis Testing

Part 3

Command Window

New to MATLAB? See resources for Getting Started.

```
>> [h,pvalue,ci] = ztest(price1/100,1.15,0.04)

h =

     0


pvalue =

    0.8668


ci =

    1.1340
    1.1690

>> [h,sig,ci] = ttest2(price1,price2)

h =

     1


sig =

    0.0083
```

THE UNIVERSITY OF ARIZONA

MATLAB

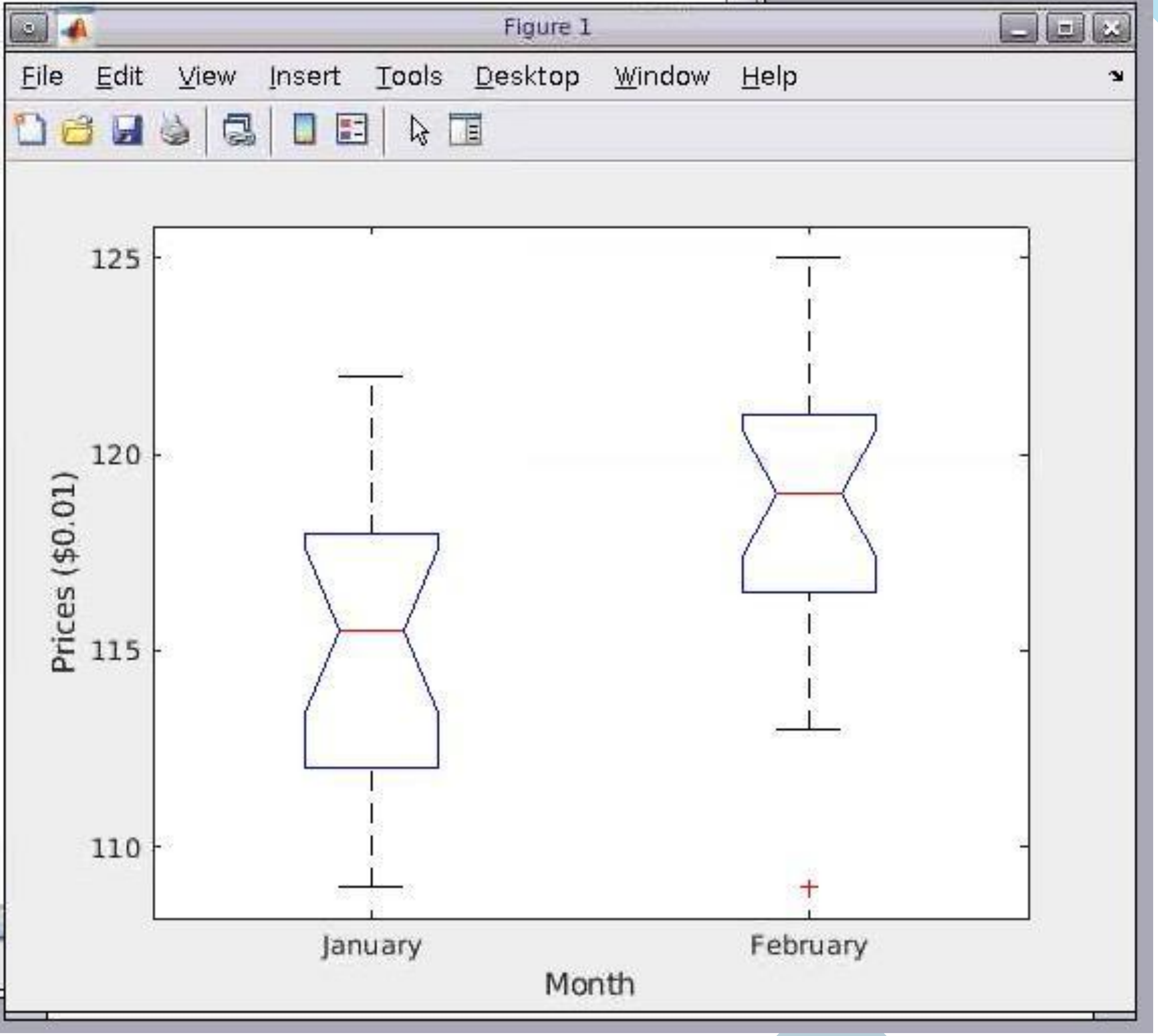
Hypothesis Testing

Part 4

Command Window

New to MATLAB? See resources for Getting Started.

```
    1.1340
    1.1690

>> [h,sig,ci] = ttest2(price1,price2)

h =

     1


sig =

    0.0083


ci =

   -5.7845
   -0.9155

>> boxplot(prices,1)
h = gca;
h.XTick = [1 2];
h.XTickLabel = {'January','February'};
xlabel('Month')
ylabel('Prices ($0.01)')
>>
```



Industrial Statistics

Analyzing Survival or Reliability Data
Survivor Functions for Two Groups
Hazard and Survivor Functions for Different Groups
Cox Proportional Hazards Model Object
Cox Proportional Hazards for Censored Data
Cox Proportional Hazards with Time-Dependent Covariates
Control Charts

Industrial Statistics
Control Charts

Control charts are created with the controlchart function.
Any of the following chart types may be specified:

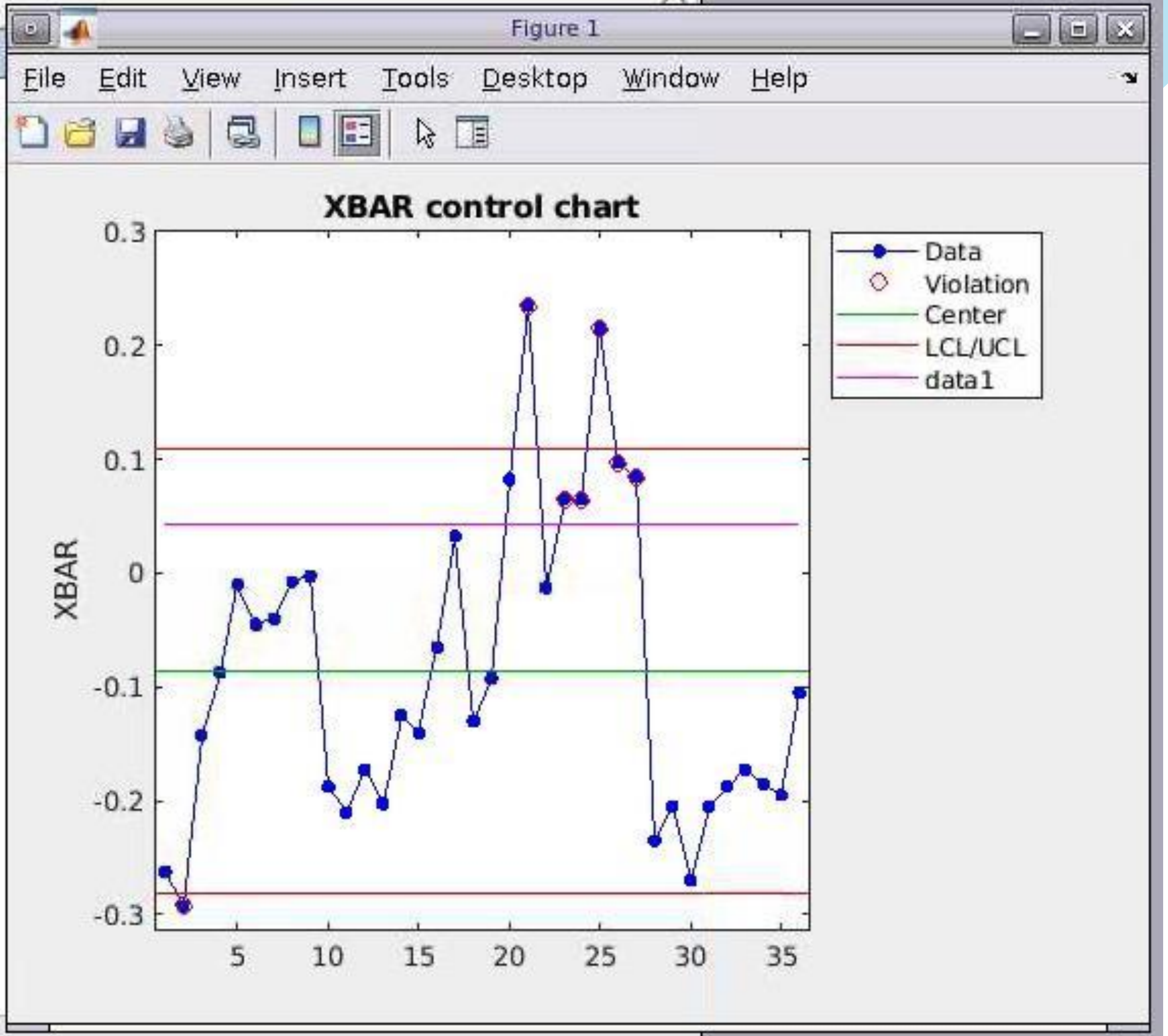
- Xbar or mean
- Standard deviation
- Range
- Exponentially weighted moving average
- Individual observation
- Moving range of individual observations
- Moving average of individual observations
- Proportion defective
- Number of defectives
- Defects per unit
- Count of defects

Industrial Statistics
Control Charts

Command Window

New to MATLAB? See resources for Getting Started.

```
>> load parts
>> st = controlchart(runout,'rules','we2');
>> x = st.mean;
cl = st.mu;
se = st.sigma./sqrt(st.n);
hold on
plot(cl+2*se,'m')
>> st = controlchart(runout,'rules','we2');
>> x = st.mean;
>> cl = st.mu;
>> se = st.sigma./sqrt(st.n);
>> hold on
>> plot(cl+2*se,'m')
>> R = controlrules('we2',x,cl,se);
I = find(R)

I =

    21
    23
    24
    25
    26
    27

fx >>
```



THE UNIVERSITY
OF ARIZONA