Graphical Displays of Statistical Data Using Matlab's Stat Pack

Joshua Linton

May 4, 2011

Purpose

The goal of this document is to explain how to graphically display data using the Matlab's stat pack. It is written for AP Statistics students who are already familiar with the basics of using Matlab (storing values, creating matrices, ect..) but are interested in learning how to make graphs using a computer instead of by hand. Since the data analyzed by AP Statistics students generally involves only examining one variable at a time, this guide will only discuss the basics of group and multivariate analysis, and will not discuss creating Data Set arrays, a topic which requires a small book in and of its self. Despite it brevity, this guide should still provide a solid grasp of how to create the most common types of graphs, formatted in a mathematically truthful and correct manner.

1 Categorical Variables

Categorical variables organize observations into categories that cannot be quantified. Examples include hair color, gender, sex, and zip code. All of the possible values a categorical variable can take on, like male and female, are called levels. Note that categorical variables may be words or numbers, and never have units.

1.1 Inputting Categorical Variables

To input a categorical variable type

```
CatVar=nominal({'value1'; 'value2'; 'value3';...})
```

The command `nominal` tells Matlab that the variable `CatVar` contains categorical data. To import data from a spread sheet use the command

```
uiopen('Path of File',1)
```

This will save the spread sheet as a variable named `textdata`, which can then be converted to categorical variable using the `nominal()` command.
Question: In 2001 the Gallop Poll asked a random sample of adults about their opinions of working parents. Of the people sampled, 3 though it is best if "both work", 9 though it is best if "one works", 2 thought it was best if "neither work", and 1 had "no opinion". Save this data to a variable called "workpref"
Solution:

```
workpref=nominal({'both work'; 'both work'; 'both work';
 'one works'; 'one works';  'one works';  'one works';
  'one works'; 'one works';  'one works';  'one works';
  'one works'; 'neither work'; 'neither work'; 'no opinion'})
```

1.2 Describing Categorical Variables

To create a frequency table, a list of the percents and counts of each level within a categorical variable, type `tabulate(CatVar)`. To view the number of values of a categorical variable takes on type `length(CatVar)`. To view the levels of a categorical variable type `getlabels(CatVar)`.
Question Create a frequency tabe for the variable `workpref`. Use a command to display `workpref`'s levels.
Solution

```
tabulate(workpref), getlabels(workpref)
```

1.3 Displaying categorical variables

Often, it is useful to create a graphical display of a categorical variable. If the categorical variable describes how a "whole" is divided up into several categories, then a pie cart may be an appropriate display. Alternatively, a bar chart can show the counts of each level. To compare the distributions of two or more categorical variables each of which divide up a "whole", use a segmented bar chart. If the categories do not show how a "whole" is split up, use a grouped or stacked bar chart.

1.3.1 Pie Charts

The basic syntax for creating a pie chart is

```
pie([NumLevel 1, NumLevel2],{'LabelLevel1','LabelLevel2'})
```

To make a pie chart, create a frequency table for your categorical variable, and then type in the counts of each level in the first part of the pie function, and the labels of each slice in second. A more direct approach is to simply type `pie(levelcounts(CatVar), getlabels(CatVar))` In this command, `levelcounts()` generates a matrix with the counts of each of the levels, and `getlabels()` makes a matrix containing all the names of the levels. If you do not type in labels for the pie chart then each piece will be labeled with its percent of the whole.
Question: Make a pie chart for `workpref`
Solution:

```
  pie(levelcounts(workpref), getlabels(workpref))
```

1.3.2 Bar Charts

To make a bar chart first create a frequency table, and then type

```
 bar([NumLevel 1, NumLevel2])
```

To label the categories type

```
{set(gca,'XTickLabel',{'lLevel1Label','lLevel2Label'})
```

To avoid making a frequency table and labeling the levels by hand use the command

```
 bar(levelcounts(CatVar)),set(gca,'XTickLabel',getlabels(CatVar))
```

Question: Make and label a bar chart for `workpref`
Solution:

```
   bar(levelcounts(workpref)),set(gca,'XTickLabel',
  getlabels(workpref))
```

1.3.3 Stacked & Grouped Bar Charts

To make a grouped bar chart create a matrix, `m`, in which each column contains a single group and the rows within a given column contain the counts of each level. Then type `bar(m','grouped')`. To make a stacked bar chart instead type `bar(m','stacked')`. To avoid figuring out the counts of each level manually use the levelcounts command in the same manner as above. To create a segmented bar chart, where the high of each bar is 100% it is necessary to change the matrix so that it contains the frequencies instead of the counts of each categorical variable. To do this divide the count of each level by the total number of observations and multiply by 100. Then type `bar(m','stacked')`. Label the categories using the command `set(gca,'XTickLavel,'Label1', 'Label2')`.
Question: The Gallop Poll conducted the same survery in 1991, and found 14 thought it is best if "both work", 27 thought it is best if "one works", 4 thought it was best if "neither work", and 1 had "no opinion". To see if poeples' opinions changed from 1991 to 2001 make a grouped bar chart and a segmented bar chart for `workpref` and `workpref1991`. Label the bars in the second graph witht the years.
Solution:

```
 workpref1991=[14;4;1;27]
    %Eneter counts in alphabetical order
  m=[workpref1991,levelcounts(workpref)]
  bar(m','grouped')
  m=[workpref1991./sum(workpref1991)*100,levelcounts(workpref)
  ./sum(levelcounts(workpref))*100]
  bar(m','stacked')  set(gca,'XTickLabel',{'1991','2001'})
```

2 Editing Graphs

After a graph is created, it is easy to add a title, label the axis, or even change the color of various parts of the graph. To do this first create the graph and then type `propertyeditor`.

2.0.4 Labels

To add a label such as a title click on the white space around the graph and type the label into the appropriate box.

2.0.5 Axes

Click on the X, Y, or Z axis pane to label one of the axis or change the scale of the tick marks.

2.0.6 Colors

Click on any part of the graph to edit its color.

2.0.7 Legend

To add a legend, click on the legend button which is located in the window's toolbox.

2.0.8 Saving Graphs

To save a graph click on file in the upper left hand corner and then on Export Setup. Set the Width and Height in the Size menu, and the Resolution in the Rendering menu, and then click export. In the export dialogue box specify the format to save the graph and then click save and ok. Note: To save the graph as vector art so it can be printed at any size without a loss of resolution, save it as an .eps file.

3 Quantitative Variables

Quantitative variables are variables that are recorded as numbers. Examples include height, age and weight. Quantitative variables always have units.

3.1 Inputting Quantitative Variables

To input a quantitative variable type

```
QuantVar=[value1; value2; value3;]
```

Question: Store the following horsepowers to the variable hp. 155,142, 125, 150, 68, 95, 97, 75, 103, 125, 115, 133, 105, 85, 110, 120, 130, 129, 138, 135, 88, 109, 65, 80, 80, 71, 68, 90, 115, 115, 90, 70, 65, 69, 78, 97, 110, 71.
Solution: `hp=[155;142; 125; 150; 68; 95; 97; 75; 103; 125; 115; 133; 105; 85; 110; 120; 130; 129; 138; 135; 88; 109; 65; 80; 80; 71; 68; 90; 115; 115; 90; 70; 65; 69; 78; 97; 110; 71]`

3.2 Describing Quantitative Variables

To numerically describe quantitative variables use the follow commands:

Command	Description	Forumla
min(QuantVar)	Smallest value of QuantVar	
median(QuantVar)	Medium value of QuantVar	
max(QuantVar)	Largest value of QuantVar	
prctile(QuantVar,p)	Pth percentile of QuantVar	
range(QuantVar)	Difference between the smallest and largest value of QuantVar	
iqr(QuantVar)	Interquartile range of QuantVar	
mean(QuantVar)	Average value of QuantVar	$\frac{\sum_{i=1}^{n} x_i}{n}$
std(QuantVar)	Standard deviation of QuantVar	$\sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$

Question: Find the mean and standard deviation of `hp`
Solution: `mean(hp), std(hp)`

3.3 Displaying Quantitative Variables

There are several different ways to view the distribution of a quantitative variable. The most common way is to create a histogram, which groups like values into bins and uses bars to show how many values fit in each bin. In order to see the individual values, try a stem-and-leaf plot. To visualize the min, max, median, IQR, and outliers use a box plot. To see if the distribution is normal, create a normal probability plot. To see if two quantitative variables are correlated, try a scatter plot.

3.3.1 Histogram

To make a histogram type `hist(QuantVar)`. To specify the number of bins type `hist(QuantVat,NBins)`. Edit the axis using the property editor.

3.3.2 Stem-and-Leaf Plot

Matlab does not have a built in function to create stem-and-leaf charts, as such, it is necessary to download the program StemLeafPlot.m from `http://www.mathworks.com/matlabcentral/fileexchange/30217-stem-and-leaf-plot`. To use this program type `StemLeafPlot(QuantVar)`. Note: This program makes a stem-and-leaf plot in the command window, not as a separate figure.

3.3.3 Boxplot

To make a box plot type `boxplot(QuantVar)`. Data which are more than 1.5 times the IQR away from the median will be represented at outliers.

3.3.4 Normal Probability Plot

The command `normplot(QuantVar)` creates a normal probability plot of `QuantVar`. The straighter the fit between the data points is, the more normal their distribution.

Question: Make a histogram, stem-and-leaf plot, box plot, and normal probability plot of `hp` to determine if the values are normally disctibuted. Which of these methods is best at determining if values are normally distributed?
Solution: `hist(hp), StemLeafPlot(hp),boxplot(hp),normplot(hp)`

3.3.5 Scatterplot

To make a scatter plot type **plot(xvar,yvar,'.')**. To add a regression line first generate a least squared best fit line. Type `regstats(yvar,xvar, 'linear'` Then select the regression statistics to generate (probably coefficients, residuals, and the R-square Statistic) Then plot both the points and the regression line with the command `plot(xvar,year,'.',polyval(beta ,linspace(min(xvar),max(xvar))))`. To make a residual plot to ensure that the residuals don't display any clear platterns type `scatter(xvar,r)`
Question: The ages and prices of several used Toyota Corollas are given bellow. Find a LSR line through the points and the correlation coefficient. Make a residual plot to see the distribution of the errors.

age	price1	price2
1	12995	10950
2	10495	
3	10995	10998
5	8700	6995
9	3200	2250
13	1750	

Solution:

```
 age=[1 1 2 3 3 5 5 9 9 13]
  price=[12995 10950 10495 10995 10998 8700 6995 3200 2250 1750]
  regstats(price,age)
% Select coeficients, residuals, and R-squared Statistic
  beta, rsquare
scatter(age, r)
```

4 copyright

Copyright Joshua Linton 2011. This document may be used for all educational purposes.

5 sources

Matlab User's Guide
Bock, David E., Paul F. Velleman, and Veaux Richard D. De. Stats Modeling the World. Boston: Addison-Wesley, 2004. Print.