

R TUTORIAL, #2: GRAPHICS

The (>) symbol indicates something that you will type in.

A bullet (•) indicates what the R program should output (and other comments).

PRESIDENTS' AGES DATA SET

The following is a list of the ages of the U.S. presidents when they became president. (The ages are in years, rounded down.)

```
57 61 57 57 58 57 61 54 68 51 49 64 50 48 65 52 56 46 54 49 51 47 55 55
54 42 51 56 55 51 54 51 60 62 43 55 56 61 52 69 64 46 54
```

(President Obama was 47, but he is not included in the table presented in the class notes. The notes were written before 2009.)

ENTERING THE DATA SET

- > Type: ages = scan()
- > Press RETURN or ENTER.
 - You should see '1:' in your console window.
- > Copy and paste the 43 numbers in the PRESIDENTS' AGES DATA SET above.
 - Notice that, unlike for the 'c' command, commas (,) are NOT used with 'scan'.
 - Copying can be done by using CTRL-C.
 - Pasting can be done by using CTRL-V.

STEMPLOTS

- We will do a stemplot of the 43 ages:
 - > Type: stem(ages)
- What if there are decimal parts in the data?
 - > Type: stem(c(1.5, 2.6, 3.7))

SCATTERPLOTS

- We will do a scatterplot:
 - > Type: `mid1 <- c(73, 64, 89)`
 - > Type: `mid2 <- c(77, 51, 93)`
 - For example, Student #1 got a 73 on Midterm 1 and a 77 on Midterm 2.
 - > Type: `plot(mid1, mid2)`
 - > Type: `plot(mid1, mid2, xlim=c(0,100), ylim=c(0,100))`
 - The `xlim` and `ylim` “arguments” allow you to control the dimensions of the graph.

TIME SERIES GRAPHS

- Here is U.S. population census data (population in millions of people):
 - > Type: `year <- c(1920, 1940, 1960, 1980, 2000)`
 - > Type: `population <- c(106, 132, 179, 227, 281)`
 - > Type: `barplot (population, names.arg=year)`
 - You should see a time series barplot. “arg” is short for “argument” in the same way that x is the argument of $f(x)$. Population is a function of time here.
 - Again, you can use the ‘`main=`’ argument to provide a title for the plot, ‘`xlab=`’ and ‘`ylab=`’ to provide labels for the axes, etc.
 - > Type: `plot(year, population, ylim=c(0,300))`
 - You should see a scatterplot.
 - The ‘`ylim`’ argument helps, because, otherwise, the “population” axis won’t start at 0.
 - > Type: `plot(year, population, ylim=c(0,300), type = "l")`
 - Note 1: We want the letter ‘l’, not the number ‘1’.
 - Note 2: Sometimes, quotation marks (“ ”) cannot be copy-pasted into your console, especially if they are not in “text” format.
 - Here, the points are connected by line segments to form a “polygon.”

PARETO CHARTS and COLORS

- Here is qualitative data on the political parties of the U.S. Presidents through George W. Bush:

```
> Type: party <- c("R", "D", "Other", "W")
```

```
> Type: freq <- c(18, 15, 6, 4)
```

- Here is a Pareto chart:

```
> Type: barplot (freq, names.arg=party)
```

- We will color the bars differently using the ‘col’ argument:

```
> Type: barplot (freq, names.arg=party, col=rainbow(4))
```

- We use ‘4’ because there are 4 bars.

- We will now shade the bars differently using ‘gray’:

```
> Type: barplot (freq, names.arg=party, col=gray(c(0.2, 0.4, 0.6, 0.8)))
```

- Lower numbers correspond to darker shades of gray.

- We will now color the bars red using ‘rgb’:

```
> Type: barplot (freq, names.arg=party, col=rgb(1,0,0))
```

- ‘rgb’ stands for “red-green-blue.”
- To color the bars green, you use (0,1,0).
- To color the bars blue, you use (0,0,1).
- You can also do color mixtures such as (.5,.5,0).

- To color the bars with different colors, you can assign ‘col’ a vector of colors:

```
> Type: barplot(freq, names.arg=party, col=c(rgb(1,0,0), rgb(0,1,0), rgb(0,0,1),  
rgb(0,.5,.5)))
```

PIE CHARTS

- We will do a pie chart:

```
> pie(freq, party)
```

- Many statisticians do not like pie charts, because people are not good at visually judging the relative areas of the pie pieces.