ISTA 116 Lab: Week 2

Colin Dawson

Last Revised August 29, 2011

1 Finish up R fundamentals

2 Warmup R Exercise

- Load the UsingR package: `library("UsingR")`
- Load the survey data set: `data(survey)`
- Compute the mean height of “Female” and “Male” respectively. (NOTE: use mean(x,na.rm=TRUE) to ignore missing data)

3 Univariate Data

- *Uni-* for “one”: data varies on only one characteristic
- Three types:
  - Categorical (aka “nominal”: values defined by name)
  - Discrete numeric
  - Continuous numeric
- What are the differences?

4 Summarizing Categorical Data

- Since we don't have numbers, we can only count how often each category occurs.
- `table()` in `R`
  - Usage: `table(aCategoricalDataVector)`
- If we have a data frame, we need to get the variable out.

```
> table(survey$Sex)
```

4.1 The `attach()` and `with()` functions

Occasionally, if we want to do several things in one data frame, it is annoying to have to keep typing its name. Two options to save typing:

`attach()`	Make all variables in a data frame directly accessible until `detach()`ed
`with()`	Serves as a temporary `attach()` that holds only over the expressions inside it

```
> names(survey)
 [1] "Sex"    "Wr.Hnd" "NW.Hnd" "W.Hnd"  "Fold"   "Pulse"  "Clap"
 [8] "Exer"   "Smoke"  "Height" "M.I"    "Age"
> table(Sex)
Error in table(Sex) : object 'Sex' not found
> attach(survey)
> table(Sex)
Sex
Female   Male
   118    118
> detach(survey)
> table(Sex)
Error in table(Sex) : object 'Sex' not found
```

```
> with(survey, table(Sex))
Sex
Female   Male
   118    118
```

- **Caution**: When using `attach()`, be sure to `detach()` later, to avoid clutter and confusion
- **Caution 2**: Only use these to read, not to assign. Must still use dollar-sign notation on left-hand side of assignments.

4.2 Getting Information From Tables

- (Univariate) tables are really just vectors with an added `names` attribute.
- `names(tableName)` gives a character vector of the category names (just like with a data frame)
- `names(tableName) <- someNewNames` lets you change the labels

Exercise: Make a table of the Smoke types in the `survey` dataset. Change the label of `Occas/Regul` to full words instead of abbreviations.

- Access individual table entries using a subset expression containing the category name(s)

```
> smokeTable = with(survey, table(Smoke))
> smokeTable
Smoke
Heavy Never Occas Regul
   11   189    19    17
```

- Subsetting by name works for any vector or data frame with a `names` attribute

```
> names(smokeTable)
[1] "Heavy" "Never" "Occas" "Regul"
```

- Convert counts to proportions with `prop.table()`

```
> prop.table(smokeTable)
Smoke
     Heavy      Never      Occas      Regul 
0.04661017 0.80084746 0.08050847 0.07203390 
```

- Show fewer decimal places with `round()`

```
> round(prop.table(smokeTable), digits = 2)
Smoke
Heavy Never Occas Regul 
 0.05  0.80  0.08  0.07 
```

Exercise: Create a table from the `Exer` (exercise) column of the survey. Convert your table to a table containing percentages.

5 Visualizing Categorical Data

Tables are well and good for small numbers of categories, but sometimes we want a picture.

For categorical data, three common options:

- Bar Plot
- Pie Chart
- Dot Chart

5.1 Bar Plots

- A barplot displays category frequencies as bar heights
- Created in `R` with the `barplot()` function
- Note: `barplot()` requires a table!

```
> barplot(smokeTable)
```

Common Bar Plot Options

`main=`	Specify an overall title for the plot
`xlab=` and `ylab=`	Label the entire x- and y-axes
`col=`	Color of bars. Can be one value, or one for each bar.
`ylim=`	Set min and max values for the y axis
`names.arg=`	Vector of bar labels

```
> total <- sum(smokeTable)
> barplot(smokeTable,
          main = "Smoking Frequency",
          xlab = "Frequency",
          ylab = "Count",
          col = c("red", "orange", "yellow", "green"),
          ylim = c(0,total)
          )
```

- You might want to add extra bars. Do this by editing the table.

```
> smokeTable2 <- c(smokeTable , Total = total)
> barplot(smokeTable2,
          main = "Smoking Frequency",
          xlab = "Frequency",
          ylab = "Count",
          col = c("red", "orange", "yellow", "green", "blue"),
          ylim = c(0,total),
          names.arg = c("Heavy","Never","Occas","Regul","All")
          )
```

Exercise: Create a barplot for the `Exer` variable.

5.2 Pie Charts

Another option for displaying categories is the pie chart.

- This can make sense when your primary interest is to see percentages of the whole.
- However, not easy to compare slices to each other.
- More easily cluttered than a barplot with many categories.
- Use the `pie()` function in `R` (also takes a table).

```
> pie(smokeTable)
```

Common Pie Chart Options	
`main=`	Specify an overall title for the plot
`col=`	Colors of slices.
`labels=`	Vector of slice labels (same as `names.arg` for barplots)
`radius=`	How big should the pie itself be within the plot?

```
> pie(smokeTable,
        main = "Smoking Frequency",
        col = c("red","grey","green","blue"),
        )
```

Exercise: Create a pie chart for the `Exer` variable.

5.3 Dot Charts

The “Cleveland Dot Plot” is yet another way to display counts.

- Basically a bar plot on its side, and without the bars
- Need not start at zero (unlike a bar plot)
- When might this be a good choice? When is a bar plot better?

```
> dotchart(smokeTable)
```

Common Dot Chart Options

`main=`	Specify an overall title for the plot
`xlab=` and `ylab=`	Axis labels
`labels=`	A vector of category labels
`xlim=`	Specify the range on the x-axis
`color=`	Colors for the points followed by colors for the
`pch=`	“Point character”: what symbol(s) should be used for the “dots”? Takes numeric values. category labels

```
> dotchart(smokeTable2,
             main = "Ye Olde Smoke Dotte Plotte",
             xlab = "Count",
             ylab = "Frequency",
             labels = c("Total","Regul","Occas","Never","Heavy"),
             color = c( "blue","orange","grey","green","red"),
             pch = c(16,15,14,13,12)
             )
```

Exercise: Three guesses what to do.

6 Stem and Leaf Plots

```
> stem(survey$Height)

  The decimal point is 1 digit(s) to the right of the |

  15 | 0224
  15 | 555566777777899
  16 | 00000000333333334444
  16 | 555555555555555555677777777888888888888899999
  17 | 000000000000000000111112222222233333333334
  17 | 555555555566777778888999999
  18 | 00000000000000000233333333344
  18 | 55555555777888899
  19 | 00011123
  19 | 56
  20 | 0
```

7 Strip Charts

```
> stripchart(survey$Pulse,method="stack")
```

8 Histograms

```
> hist(survey$Height)
```

9 Last Minute HW1 Issues?