ISTA 116 Lab: Week 6

Colin Dawson

Last Revised September 26, 2011

1 Multivariate Categorical Data

How can we examine relationships between more than 2 variables? **Three-way contingency tables**, or n-way contingency tables.

The `student.expenses` dataset has data on what students spent money on.

```
> library(UsingR)
> data(student.expenses)
> ?student.expenses #what is in the dataset?
> attach(student.expenses)
> #a 2-way joint frequency table
> (JointFreqTable2Way <- table(cell.phone, car))
> #a 3-way joint frequency table
> (JointFreqTable3Way <- table(cell.phone, car, cable.modem))
```

How do we condition on multiple variables at once?

```
> #conditioning on car
> (CondPropTable2Way <- prop.table(JointFreqTable2Way, margin=2))
> #conditioning on car and cable.modem
> (CondPropTable3Way <- prop.table(JointFreqTable3Way, margin=c(2,3)))
```

2 Grouped Numeric Data

What happens if we have a bivariate data set with one categorical and one numeric variable?

- Can't really look at a contingency table any more, especially if the numeric variable is continuous.
- Instead we can *condition* on the categorical variable, and look at the *distributions* of the numeric variable.

The `reaction.time` dataset in the `UsingR` library has data on people's reaction times while driving, based on whether they were using a cell phone at the time.

2.1 Conditional Summary Statistics

```
> data(reaction.time)
> attach(reaction.time)
> ##Compute the means for each group
> mean(time[control=="T"]) #With a cell phone
> mean(time[control=="C"]) #Without a cell phone
> median(time[control=="T"])
> median(time[control=="C"])
```

The `tapply()` function is a compact way to get summaries for each group at once:

```
> tapply(X = time,  #What am I summarizing?
           INDEX = control, #What am I conditioning on?
           FUN = mean  #What function am I computing on each group?
           )
> tapply(time,
           INDEX = control,
           FUN = fivenum
           )
```

2.2 Plotting Multiple Conditional Distributions

As with univariate numeric data, seeing the entire distribution is often more valuable than just seeing a few summary numbers.

- Two old graphics, one new one
  - Box Plots
  - Density Curves
  - Quantile-Quantile Plots (New!)

2.2.1 Side-by-side Box Plots

```
> ## Side-by-side Box Plots
> withcell <- time[control == "T"]
> nocell <- time[control == "C"]
> boxplot(withcell,nocell,
            names = c("With Cell","No Cell"),
            ylab = "Reaction Times",
            col = "forestgreen"
            )
> ## Another Way (No need to define new variables)
> boxplot(time ~ control,  #Plot time, conditioned on control
            data = reaction.time, #Like a with() statement
            names = c("With Cell","No Cell"),
            ylab = "Reaction Times",
            col = "forestgreen"
            )
```

Exercise: The `Modarres05.csv` data set contains information about levels of a toxic chemical (TcCB) at each of two different sites. Create side-by-side box plots of the `contamination` variable, separated by site. What can you see from the plot? What might we do to see the two distributions more clearly?

2.2.2 Overlaid Density Plots

```
> ## Overlaid density plots
> ## (No histograms this time)
> cellDensity <- density(withcell) #Use the new variables
> nocellDensity <- density(nocell)
> plot(cellDensity, #Use plot() for the first one
         type = "l",
```

```
          xlab = "Reaction Time",
          ylab = "Density",
          main = "",
          lwd = 2,
          col = "darkblue")
> lines(nocellDensity, #Use lines() for the second one
           lwd = 2,
           col = "forestgreen"
           )
> legend(1.55,5,  #Coordinates for the upper left corner
            legend = c("Cell Users","Controls"),
            lty = 1,
            lwd = 2,
            col = c("darkblue","forestgreen")
            )
> detach(reaction.time)
```

Exercise: Create a useful plot of overlaid density curves for the toxic waste data.

2.2.3 QQ Plots

Box plots and density curves are the same principle with two (or more) samples as for one. *Quantile-Quantile* plots, however, are specifically designed for comparing two distributions.

- From each data set, compute some set of quantiles (e.g. 0.00, 0.10, 0.20, 0.30, ..., 1.00)
- Then, for each quantile, plot the value for one distribution on the x-axis, and the value for the other on the y-axis.
  - So, for example, if the 0.10 quantile has a value of 5 for the first distribution, and a value of 8 for the second, draw a point on the plot at (5,8). Do the same for the other quantiles.
- When the data sets are the same size (with n points), you can just compute quantiles in $\frac{1}{n-1}$ increments, so the points on the plot correspond to actual data points.
  - E.g., with 5 data points, $x_{(1)}$ (the lowest) is the 0.00 quantile, $x_{(2)}$ (the

next lowest) is the 0.25 quantile (Q_1), $x_{(3)}$ is the 0.5 quantile (the median), $x_{(4)}$ 0.75 quantile (or Q_3), and $x_{(5)}$ is the 1.00 quantile.

```
> #We'll create two short made-up variables to see this in action
> x1 <- c(1,10,4,6)
> x2 <- c(15,9,12,5)
> #What should I see?
> qqplot(x1,x2)
> #What if I plot a variable against itself?
> qqplot(x1,x1)
> #What if one variable is shifted by a constant?
> qqplot(x1, x1+2)
```

It is often a good idea to use the same axes if the two variables are being measured on the same scale. This is just a matter of setting `xlim=` and `ylim=`.

```
> lower <- min(x1,x2) #Find the lowest in either variable
> upper <- max(x1,x2) #Find the highest in either variable
> lims <- c(lower,upper) #Create a single limit vector
> qqplot(x1, x2, xlim = lims, ylim = lims)
> lower <- min(x1,x1+2)
> upper <- max(x1,x1+2)
> lims <- c(lower,upper)
> #Can you see the shift now?
> qqplot(x1, x1+2, xlim = lims, ylim = lims)
```

Exercise: Create a QQ-Plot of the cleanup and reference contaminations so that we can easily spot differences. What do you see?

3 Bivariate Numeric Data

In the previous section, we can think of the data as containing two independent numeric variables, but if we want to think about the variables we have for a particular observation, what were they?

What if we truly have two numeric variables for *each observation* (that is, we have numeric *pairs*)?

3.1 Scatterplots

Now, a single point has two "coordinates". So... why not plot it using those coordinates?

- A *scatterplot* displays each data point in two dimensions, with the value of one variable on the x-axis, and the value of the other on the y-axis.
- In `R`, we can just use the `plot()` function.
- How is this different from a QQ-Plot?

The `blood` data set (in the `UsingR` library) contains blood pressure readings for 15 individuals. `Machine` contains readings by an automated machine; `Expert` contains readings by an expert.

```
> #I already made UsingR visible
> data(blood)
> attach(blood)
> #Plot Machine "as a function of" Expert
> #That is, for each expert reading (on the x-axis)
> #Plot the corresponding machine reading on the y-axis
>
> #What do you expect to see?
>
> plot(Machine ~ Expert)
>
> # Does one variable cause the other?
```

Exercise: The `emissions` data set (in `UsingR`) examines CO_2 emissions and Gross Domestic Product (GDP), both total and per capita, for various countries. Plot CO_2 as a function of total GDP, and then as a function of GDP per capita. What relationships do you notice? Is one stronger than the other?

4 HW 3 Questions?