MIE237 February 2-3 Labs

Neil Montgomery

February 1, 2016

Summary of you will do in this lab

You'll do some questions for tests of independence, and a little bit of regression (much more later).

1. Using the real data file formats I've provided, analyze the scenarios presented in 10.86, 10.87, and 10.89. (I work out 10.88 for you.)
2. Do 11.1(a), to start learning how to do regression analyses. (I've reproduced 11.5 that we used in class.) You can do the basic analysis for any of the textbook questions on pages 398 to 400.

The usual advice

I've told you where to get the textbook data. I've fixed a few files, you might want to get those updates from the repository. The PDF of this lab doesn't show all the code, but the `.Rmd` source file of the lab does. Data analysis consists of some graphical and/or numerical exploration, the analysis itself, a verification of assumptions, and a conclusion/interpretation.

Test of independence worked example

We'll look at 10.88 from the book. The relevant file is `Ex10.88.txt`.

Comments on textbook data files

The files given for section 10.12 questions are not actual datasets, in the sense that they are not rows of records with columns of variables. It is possible to use the `chisq.test` function in R but it is a bit artificial.

More importantly this presents a problem when trying to make nice plots—`ggplot2` expects datasets and not toy examples—and even things like adding marginal totals to tables. So what I've done is create files that look like actual datasets for the questions to do in this lab.

(And the files for exercises 10.86 and 10.87 are hopelessly mangled so I've also provided a fixed versions here.)

10.88

I'll show you how to work with the summary table first.

```
# I've used read.delim rather than import because we need R to know that the
# first column is actually row names and not a variable.
men_table <- read.delim("Ex10.88.txt", row.names = 1)
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote =
## quote, : incomplete final line found by readTableHeader on 'Ex10.88.txt'
```

There's no good way to make plots from a summary table, so we'll put that off for a moment. Here's a nice view of the table and the actual χ^2 results.

```
kable(men_table)
```

	X0.1	X2.3	Over3
Elementary	14	37	32
Secondary	19	42	17
College	12	17	10

```
(men_table_chisq <- chisq.test(men_table))
```

```
##
##  Pearson's Chi-squared test
##
## data:  men_table
## X-squared = 7.4644, df = 4, p-value = 0.1133
```

We can access the results directly and print nice things in the text like: *the observed value of the test statistic is* 7.4643933 *and the p-value is* 0.1132897. We can also access the expected cell counts like this:

```
kable(men_table_chisq$expected)
```

	X0.1	X2.3	Over3
Elementary	18.675	39.84	24.485
Secondary	17.550	37.44	23.010
College	8.775	18.72	11.505

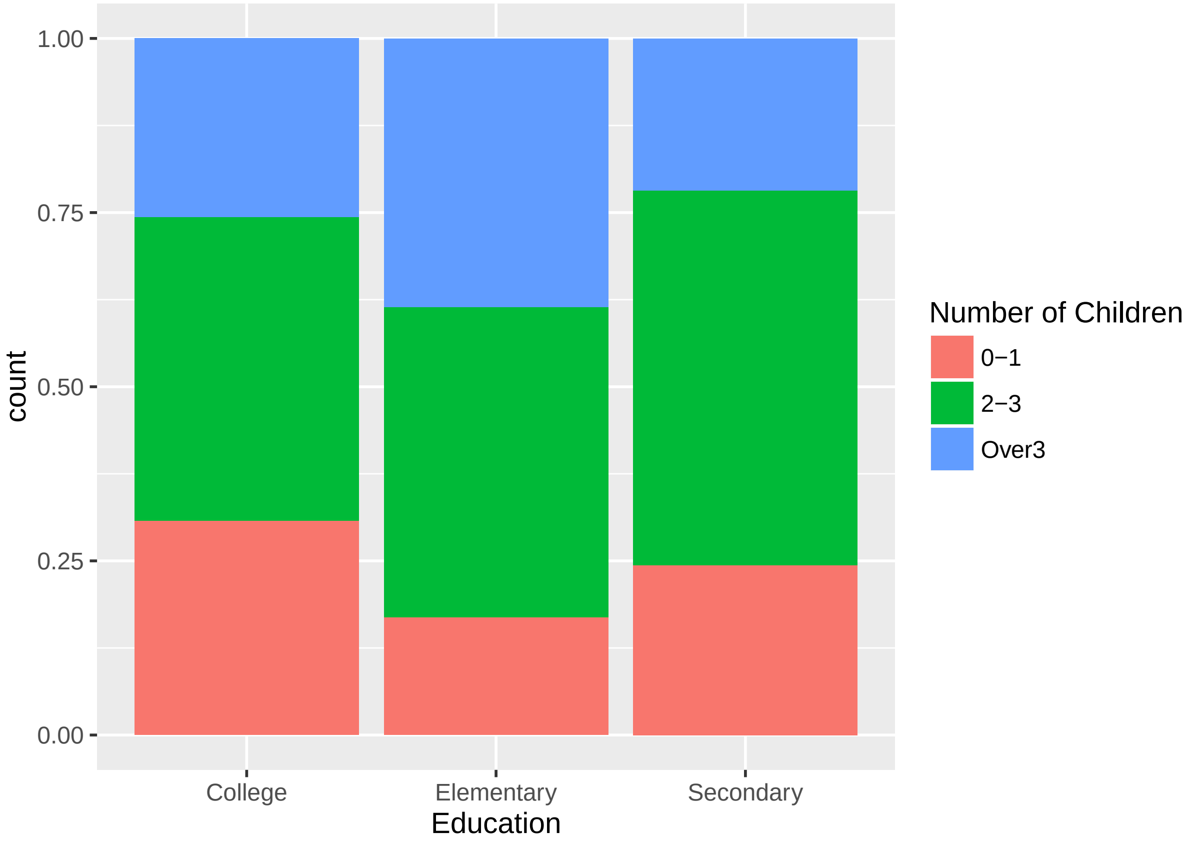
All of them well over 5, so the χ^2 approximation is good enough.

But that's about it with the file as in table form already. But at least with this you can check your hand calculations when you try exercises. Let's move on to the proper way, with actual data.

```
men <- import("Ex10.88.csv")
```

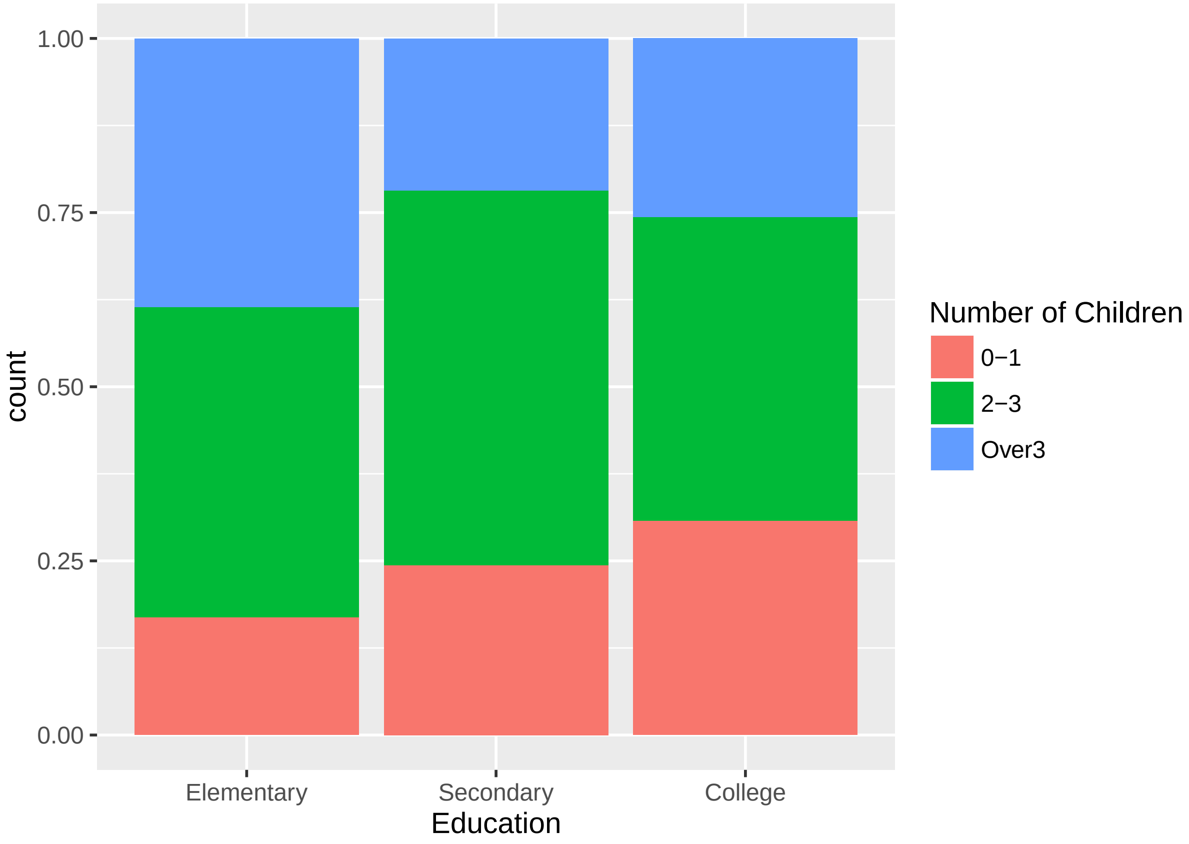
From now on much of the R code is in the lab source file and not printed in this document.

Here is a plot of the data that is nice for visualizing independence.



Hmmm. . . the `Education` variable levels are in alphabetical order rather than the order as in the text. No worries. . .

```
men$Education <- factor(men$Education,
                        levels=c("Elementary", "Secondary", "College"))
men %>%
  ggplot(aes(x=Education)) +
  geom_bar(aes(fill = `Number of Children`), position = "fill")
```



Here is the table of counts along with marginal totals.

	0-1	2-3	Over3	Sum
Elementary	14	37	32	83
Secondary	19	42	17	78
College	12	17	10	39
Sum	45	96	59	200

Here are the χ^2 test results, which are of course the same as before.

```
chisq.test(men$Education, men$`Number of Children`)
```

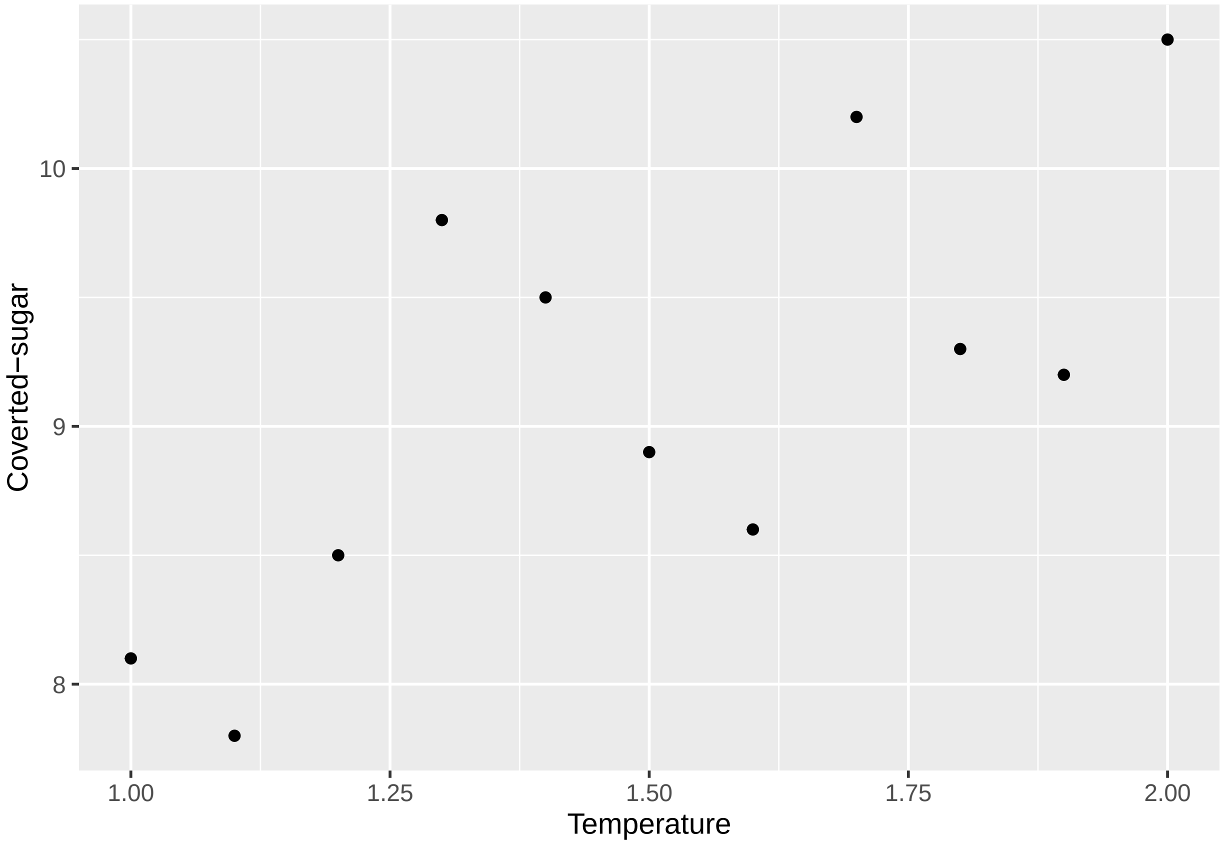
```
##
##  Pearson's Chi-squared test
##
## data:  men$Education and men$`Number of Children`
## X-squared = 7.4644, df = 4, p-value = 0.1133
```

Simple regression basics worked example (11.5)

A regression analysis starts with data import (and in real life—data cleaning), plotting the data, analysis, and verification of model assumptions. We'll just do plotting and a little analysis this week.

We re-do the sugar/temperature example from class (11.5). Import and plot the data.

```
sugar <- import("Ex11.05.txt")
sugar %>%
  ggplot(aes(x=Temperature, y=`Coverted-sugar`)) + geom_point()
```



Boy that spelling mistake is annoying. Here's how you would fix it. This code isn't run, so the spelling mistake will persist in this document. In real life I would have done this first.

```
# NOT RUN
names(sugar)[2] <- "Converted-sugar"
```

Let's go ahead with the regression analysis. The function is `lm`. Here are a few ways to use it. I usually use the second, so I commented out the first.

```
# lm(`Coverted-sugar` ~ Temperature, data = sugar)

sugar %>%
  lm(`Coverted-sugar` ~ Temperature, data = .)
```

```
##
## Call:
## lm(formula = `Coverted-sugar` ~ Temperature, data = .)
##
## Coefficients:
## (Intercept)  Temperature
##       6.414        1.809
```

It doesn't print much of use. The two common commands used to print useful summaries are **summary** and **anova** (the latter of which we'll get to this week in detail in class.)

```
sugar %>%
  lm(`Coverted-sugar` ~ Temperature, data = .) -> sugar_lm

(sugar_lm_summary <- summary(sugar_lm))
```

```
##
## Call:
## lm(formula = `Coverted-sugar` ~ Temperature, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7082 -0.4868 -0.1227  0.5109  1.0346
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.4136     0.9246   6.936 6.79e-05 ***
## Temperature   1.8091     0.6032   2.999    0.015 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6326 on 9 degrees of freedom
## Multiple R-squared:  0.4999, Adjusted R-squared:  0.4443
## F-statistic: 8.996 on 1 and 9 DF,  p-value: 0.01497
```

```
anova(sugar_lm)
```

```
## Analysis of Variance Table
##
## Response: Coverted-sugar
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Temperature  1 3.6001  3.6001  8.9959 0.01497 *
## Residuals    9 3.6017  0.4002
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `sugar_lm` and `sugar_lm_summary` objects have many components that we will define and access, some honestly more easy to get to than others, such as:

```
sugar_lm$coefficients
```

```
## (Intercept) Temperature
##    6.413636    1.809091
```

```
# The p-value
sugar_lm_summary$coefficients[2,4]
```

```
## [1] 0.0149729
```