# Analyzing Subsets of Data in R

Charlie Gibbons
ARE 212

Spring 2011

Today we are going to use two packages for data analysis: `plyr` and `reshape`. There are two broad sets of functionality that they offer: structural manipulation and analysis according to categorical variables. In many ways, they are the opposites of each other. A simplistic comparison is given in Table 1. Additionally, we are going to use the `ggplot2` package for plotting our results.

Table 1: Comparison of `plyr` and `reshape` packages

| | Analysis levels | Structural manipulation |
|---|---|---|
| `plyr` | Within subpopulations (*i.e.*, subsetting) | Change the class of data (*e.g.*, from a list to a data frame) |
| `reshape` | Across subpopulations (*i.e.*, aggregating) | Change the shape of data (*e.g.*, from "long" to "wide") |

## 1 Calculating returns to eduction

Often times, we will consider how a variable of interest varies across subpopulations in our data. Though we can typically perform this type of analysis using a complex model that includes many interactions, it may be informative to perform the analysis separately for each group. As an example, here we consider how wages respond to changes in educational attainment across state-sex groups using a sample from the 1990 U.S. Census:[1]

```
> X <- read.csv("../Data/IncomeData.csv")
```

We start by limiting our data set to those of working age, above 24 and less than 65. When isolating a single group, `subset` is a good approach:

[1] Operated by the University of Minnesota, IPUMS (`http://usa.ipums.org`) offers easy access to U.S. Census files for the past century.

```
> X <- subset(X, age >= 24 & age < 65)
```

The rest of the analysis will rely on a package called `plyr`:

```
> library(plyr)
```

The `plyr` commands are called `**ply()`, where the first letter is the class of object put into the function (`array`, `data frame`, and `list` are the most common) and the second letter is the class of object returned. Suppose that we wanted to create a separate data set for each state-sex combination. The most effective way to handle all these data sets is to store them as elements of a list. We put our original data frame into the function and want a list out, subset by two factors, state and sex:

```
> state.data <- dlply(X, .(statefip, sex), subset)
```

One of the nicest features of `plyr` is its naming system. It is very easy to identify which data set in this list we are interested in:

```
> names(state.data)
```

Calling `state.data$"New Hampshire.Male"` would give a data set of all men from New Hampshire. This list of data sets could be passed to the `lapply` function that is standard in `R` or to one of the `l*ply` functions.

Typically, subsetting the data like this is a means to an end; we don't want the separate data sets themselves, just the results of analyses performed on them. As a first example, we can summarize the income distribution on a state-sex basis by inputting a data frame, subsetting it by state and sex, then summarizing income within the subset:

```
> inc.summary <- ddply(X, .(statefip, sex), function(x) summary(x$inctot))
```

Here, `x` is just a placeholder representing the data frame that is fed into the function.

Suppose that we wanted to perform a simple regression of income on education for each state. Here, we take our data frame, subset it into data sets, perform the regression, and store the results as a list:

```
> lm.list <- dlply(X, .(statefip, sex), function(x){lm(inctot~educ, data=x)})
```

The regression results are well-labeled and easy to access. We would like all the slope coefficients collected together; we want to turn a list into a data frame by extracting the slope coefficient element from the list:

```
> coef.est <- ldply(lm.list, function(x) coef(x)[2])
```

We should note that these two steps could be combined using:

```
> coef.est <- ddply(X, .(statefip, sex), function(x){
+   lm(inctot~educ, data=x)$coef[2]
+ })
```

# 2 Analyzing the returns to education

It might be easier to analyze our results graphically. We'll use the `ggplot2` package to help us:

```
> library(ggplot2)
```

First, we'll look at the estimates plotted by state, with each point having a different color for men and women:

```
> qplot(educ, statefip, color=sex, data=coef.est,
+       xlab="Returns to education",
+       ylab="State")
```

Men seem to have higher returns to education than women. We can look at a density plot to confirm this:

```
> qplot(educ, color=sex, data=coef.est, geom="density",
+       xlab="Returns to education")
```

We'll look at another plot momentarily.

This data set is in "long" form—there are two rows, one for each sex, for each state. Suppose that we wanted it in "wide" form with two columns for each state. The easiest way to accomplish this is using a package based upon `plyr` called `reshape`:

```
> library(reshape)
```

We need to take our `coef.est` data set and *melt* it according to the categorical variables that it contains, state and sex:[2]

```
> coef.melt <- melt(coef.est)
```

Now, we *cast* the `coef.melt` data set by defining the variable(s) that we want defining the rows, inserting a tilde, then identifying the variable that should define the columns:

[2]The `melt` function identifies these automatically, though we could specify them if we chose.

```
> coef.wide <- cast(coef.melt, statefip ~ sex)
```

We could have multiple variables defining the rows; `cast(coef.melt, statefip + sex ~ .)` would return us to our original configuration. Note that the `.` notation means that no variable should go there. Adding margins to the cast data set shows row and column averages:

```
> coef.wide <- cast(coef.melt, statefip ~ sex,
+                   margins=c("grand_row", "grand_col"))
```

Though not particularly useful here (since we only have two categorical variables), we could break our data apart into a list with a separate slot for each sex:

```
> cast(coef.melt, statefip ~ . | sex)
```

Let's use our wide version of our results to look at an additional plot. Men seem to have higher returns than women across all states, but does this hold within each state? We'll look at the distribution of the difference between the male returns and the female returns:

```
> qplot(Male - Female, data=coef.wide,
+       xlab="Difference between male and female returns")
```

We see that only two states have higher returns for women than men. Which are they?

```
> coef.wide$statefip[which(coef.wide$Female > coef.wide$Male)]
```

```
[1] Maine        North Dakota
52 Levels: Alabama Alaska Arizona Arkansas California Colorado ... (all)
```

Thus far, we have been simply changing the arrangement of our data. We can also use these functions to aggregate our data. We may be concerned that an errant call may aggregate our data unwittingly, but there's a warning for that:

```
> ag.ex <- cast(coef.melt, statefip ~ .)
```

As a simple example of aggregation, we can create a simple average of the returns to education for men and women by casting our data set without dividing the male and female results, instead averaging them together:

```
> coef.avg <- cast(coef.melt, statefip ~ ., mean)
```

As it stands, this list has a jumble of numbers sorted in alphabetical order by state. Instead, it may be useful to sort this result in decreasing order of the estimated returns to schooling:

```
> coef.avg <- coef.avg[order(coef.avg[,2], decreasing=TRUE),]
```

It is much easier to look for patterns with the estimates arranged this way.

# 3 Demeaning within state-sex interaction

Returning to our original data set, let's try something (the logic behind this process will be explained below). We take our original data frame, subset it by state and sex, then transform the education and income data by subtracting off their within-group means. The `scale` function subtracts off the mean and divides by the standard deviation of a set of data; we don't want to divide by the standard deviation, only center the data:

```
> X.scaled <- ddply(X, .(statefip, sex), transform,
+                   inctot = scale(inctot, scale=FALSE),
+                   educ   = scale(educ,   scale=FALSE))
```

Now, performing a simple regression using the scaled data, we have:

```
> summary(lm(inctot ~ educ, data=X.scaled))

Call:
lm(formula = inctot ~ educ, data = X.scaled)

Residuals:
   Min     1Q Median     3Q    Max
-69301 -10998  -2570   6813 364258

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.57e-14   4.47e+01       0        1
educ         3.52e+03   2.00e+01     176   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 22200 on 246046 degrees of freedom
Multiple R-squared: 0.112,        Adjusted R-squared: 0.112
F-statistic: 3.1e+04 on 1 and 246046 DF,  p-value: <2e-16
```

The estimated returns to education is the same as running a regression using the uncentered data set with a full set of state-sex dummies:[3]

```
> interact.reg <- lm(inctot ~ educ + statefip*sex, data=X)
```

[3] As an exercise, you should try to show this.

Using the centered data is much faster because the interacted model is using a data set that has an intercept, the education variable, and 99 dummy variables. The simple procedure doesn't give us the values for the state-sex dummies, but these are typically of little interest.

# A R code

```
###################################################
### chunk number 1:
###################################################
#line 37 "Subsetting.Rnw"
X <- read.csv("../Data/IncomeData.csv")


###################################################
### chunk number 2:
###################################################
#line 42 "Subsetting.Rnw"
X <- subset(X, age >= 24 & age < 65)


###################################################
### chunk number 3:
###################################################
#line 46 "Subsetting.Rnw"
library(plyr)


###################################################
### chunk number 4:
###################################################
#line 50 "Subsetting.Rnw"
state.data <- dlply(X, .(statefip, sex), subset)


###################################################
### chunk number 5:
###################################################
#line 54 "Subsetting.Rnw"
names(state.data)


###################################################
### chunk number 6:
###################################################
#line 60 "Subsetting.Rnw"
inc.summary <- ddply(X, .(statefip, sex), function(x) summary(x$inctot))


###################################################
```

```
### chunk number 7:
####################################################
#line 66 "Subsetting.Rnw"
lm.list <- dlply(X, .(statefip, sex), function(x){lm(inctot~educ, data=x)})


####################################################
### chunk number 8:
####################################################
#line 70 "Subsetting.Rnw"
coef.est <- ldply(lm.list, function(x) coef(x)[2])


####################################################
### chunk number 9:
####################################################
#line 74 "Subsetting.Rnw"
coef.est <- ddply(X, .(statefip, sex), function(x){
  lm(inctot~educ, data=x)$coef[2]
})


####################################################
### chunk number 10:
####################################################
#line 83 "Subsetting.Rnw"
library(ggplot2)


####################################################
### chunk number 11:
####################################################
#line 87 "Subsetting.Rnw"
theme_set(theme_grey(base_family="CM"))


####################################################
### chunk number 12:
####################################################
#line 90 "Subsetting.Rnw"
qplot(educ, statefip, color=sex, data=coef.est,
      xlab="Returns to education",
      ylab="State")
```

```
#####################################################
### chunk number 13:
#####################################################
#line 95 "Subsetting.Rnw"
ggsave("./Plots/Returns-state.pdf")
embedCM("./Plots/Returns-state.pdf")


#####################################################
### chunk number 14:
#####################################################
#line 106 "Subsetting.Rnw"
qplot(educ, color=sex, data=coef.est, geom="density",
      xlab="Returns to education")


#####################################################
### chunk number 15:
#####################################################
#line 110 "Subsetting.Rnw"
ggsave("./Plots/Returns-density.pdf")
embedCM("./Plots/Returns-density.pdf")


#####################################################
### chunk number 16:
#####################################################
#line 123 "Subsetting.Rnw"
library(reshape)


#####################################################
### chunk number 17:
#####################################################
#line 127 "Subsetting.Rnw"
coef.melt <- melt(coef.est)


#####################################################
### chunk number 18:
#####################################################
#line 131 "Subsetting.Rnw"
coef.wide <- cast(coef.melt, statefip ~ sex)
```

```
###################################################
### chunk number 19:
###################################################
#line 135 "Subsetting.Rnw"
coef.wide <- cast(coef.melt, statefip ~ sex,
                  margins=c("grand_row", "grand_col"))


###################################################
### chunk number 20:
###################################################
#line 141 "Subsetting.Rnw"
cast(coef.melt, statefip ~ . | sex)


###################################################
### chunk number 21:
###################################################
#line 146 "Subsetting.Rnw"
qplot(Male - Female, data=coef.wide,
      xlab="Difference between male and female returns")


###################################################
### chunk number 22:
###################################################
#line 150 "Subsetting.Rnw"
ggsave("./Plots/Returns-diff.pdf")
embedCM("./Plots/Returns-diff.pdf")


###################################################
### chunk number 23:
###################################################
#line 161 "Subsetting.Rnw"
coef.wide$statefip[which(coef.wide$Female > coef.wide$Male)]


###################################################
### chunk number 24:
###################################################
#line 166 "Subsetting.Rnw"
ag.ex <- cast(coef.melt, statefip ~ .)
```

```
#####################################################
### chunk number 25:
#####################################################
#line 171 "Subsetting.Rnw"
coef.avg <- cast(coef.melt, statefip ~ ., mean)


#####################################################
### chunk number 26:
#####################################################
#line 175 "Subsetting.Rnw"
coef.avg <- coef.avg[order(coef.avg[,2], decreasing=TRUE),]


#####################################################
### chunk number 27:
#####################################################
#line 183 "Subsetting.Rnw"
X.scaled <- ddply(X, .(statefip, sex), transform,
                  inctot = scale(inctot, scale=FALSE),
                  educ   = scale(educ,   scale=FALSE))


#####################################################
### chunk number 28:
#####################################################
#line 189 "Subsetting.Rnw"
options(useFancyQuotes = "TeX")


#####################################################
### chunk number 29:
#####################################################
#line 192 "Subsetting.Rnw"
summary(lm(inctot ~ educ, data=X.scaled))


#####################################################
### chunk number 30:
#####################################################
#line 196 "Subsetting.Rnw"
interact.reg <- lm(inctot ~ educ + statefip*sex, data=X)
```

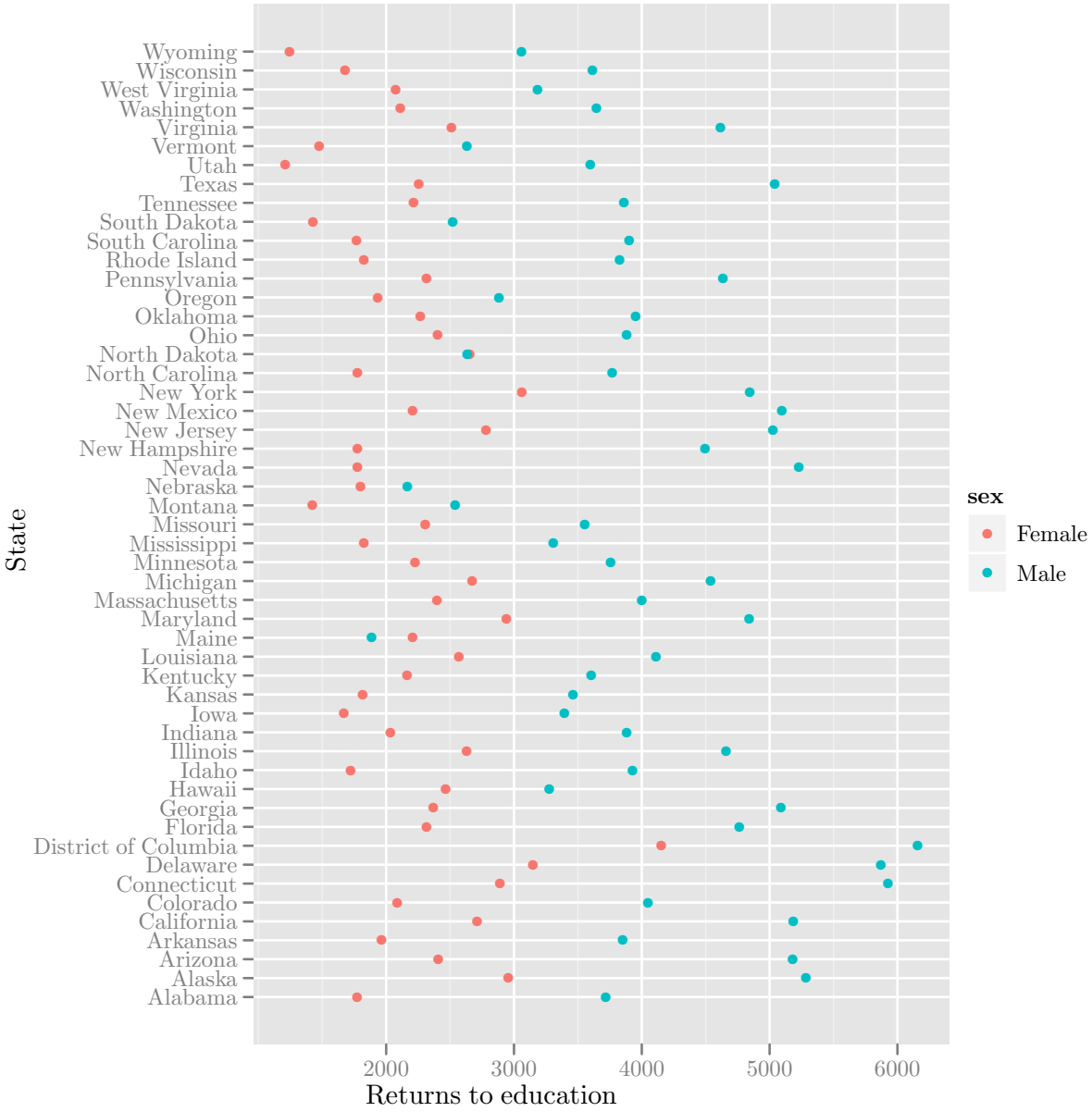


Figure 1: Returns to education by state and sex

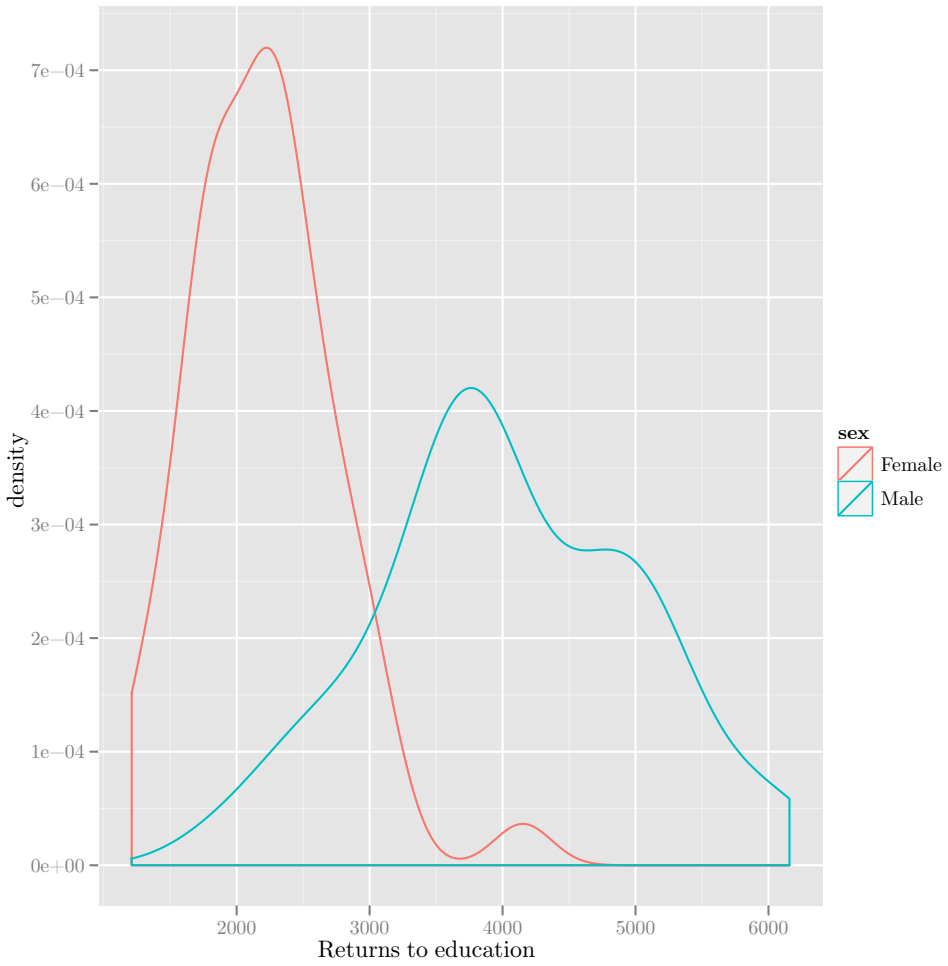


Figure 2: Densities of returns to education by sex

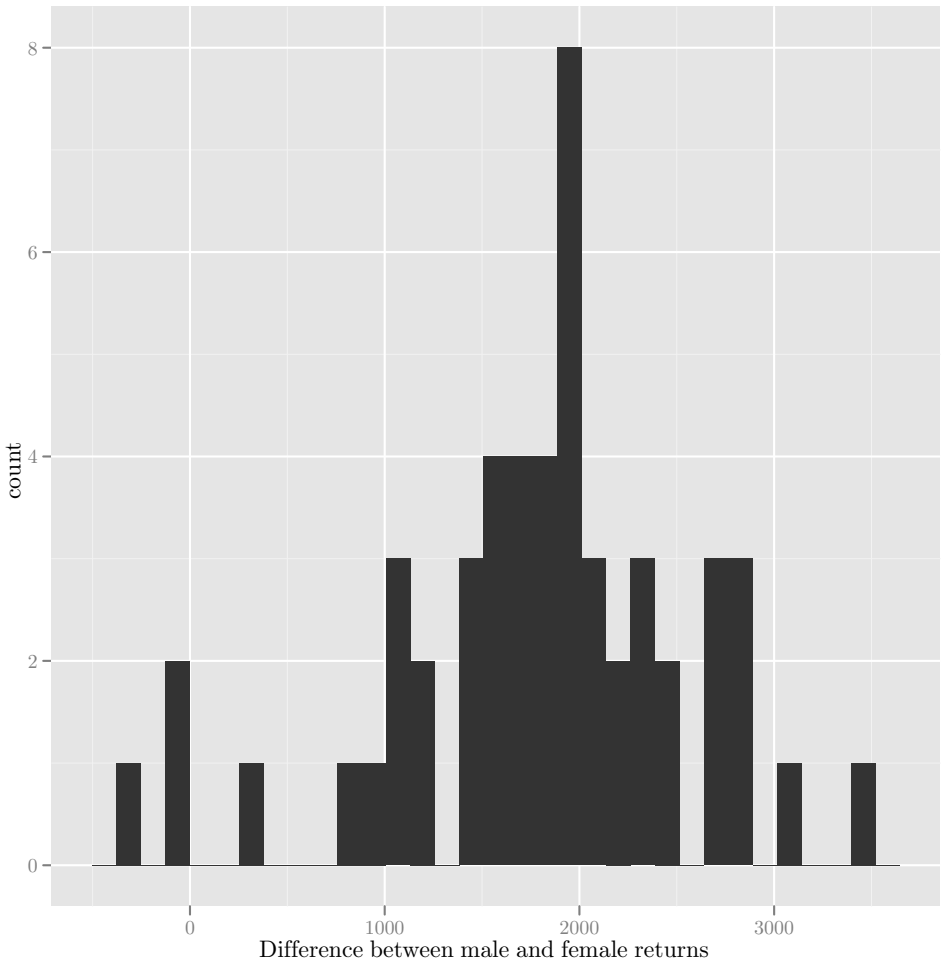


Figure 3: Difference in returns to education between men and women