Merge/append data using R/RStudio
(v. 1.0)

Oscar Torres-Reyna
otorres@princeton.edu
**Intro**

**Merge** – adds variables to a dataset. This document will use the `merge` function. Merging two datasets require that both have at least one variable in common (either string or numeric). If string make sure the categories have the same spelling (i.e. country names, etc.). Explore each dataset separately before merging. Make sure to use all possible common variables (for example, if merging two panel datasets you will need country and years).

**Append** – adds cases/observations to a dataset. This document will use the `smartbind` function from the `gtools` package. Appending two datasets require that both have variables with exactly the same name and spelling. If using categorical data make sure the categories on both datasets refer to exactly the same thing (i.e. 1 “Agree”, 2”Disagree”, 3 “DK” on both).
Appending

# Data from 1960 to 1989

```r
mydata6080 = read.csv("http://www.princeton.edu/~otorres/mydata6080.csv",
                      header=TRUE,
                      stringsAsFactors = FALSE)

table(mydata6080$year)
```

# Data from 1990 to 2013

```r
mydata9020 = read.csv("http://www.princeton.edu/~otorres/mydata9020.csv",
                      header=TRUE,
                      stringsAsFactors = FALSE)

table(mydata9020$year)
```

# Appending the two datafiles. Full data from 1960 to 2013

```r
library(gtools)

mydata = smartbind(mydata6080, mydata9020)

table(mydata$year)
```
# Merging join variables from two datafiles
# To an existing file called mydata (created in the previous slides)

```r
str(mydata)
'data.frame': 5130 obs. of 6 variables:
$ country: chr "Estonia" "Thailand" "Hong Kong SAR, China" "Thailand" ... 
$ gdppc : num NA 882 16973 915 16767 ... 
$ unemp : num 0.6 0.9 1.1 1.3 1.4 1.5 1.6 1.6 1.7 ... 
$ export : num NA 9.92e+09 NA 1.08e+10 NA ... 
$ import : num NA 1.66e+10 NA 1.67e+10 NA ...
```

# we are going to add one more variable from a dataset called mydatapol
# Notice the common variables 'country' and 'year' which we are going to use
# to match each row on both files

```r
mydatapol = read.csv("http://www.princeton.edu/~otorres/mydatapol.csv", 
header=TRUE,
stringsAsFactors = FALSE)
```

```r
str(mydatapol)
'data.frame': 3655 obs. of 3 variables:
$ country: chr "Afghanistan" "Albania" "Algeria" "American Samoa" ... 
$ politics: num 1.12 24.05 16.69 NA 89.51 ... 
```

# Merging the two files

```r
mydata <- merge(mydata, mydatapol, by=c("country","year"), all=TRUE)
```

```r
mydata <- mydata[order(mydata$year,mydata$country),]  # Sorting data by year/country
```

(see next slide)
# Merging

The file `mydata` has now one extra variable

```r
str(mydata)

'data.frame':  7170 obs. of  7 variables:
$ country : chr  "Algeria" "Argentina" "Australia" "Austria" ...
$ gdppc   : num  1766 3732 13469 10862 NA ...
$ unemp   : num  NA NA NA NA NA NA NA NA NA NA ...
$ export  : num  2.24e+10 3.98e+09 1.02e+10 1.03e+10 NA ...
$ import  : num  1.61e+10 5.43e+09 1.01e+10 1.14e+10 NA ...
$ politics: num  NA NA NA NA NA NA NA NA NA NA ...

# Common variables do not need to have the same name, for example:
mydata <- merge(mydata1, mydata2,
  by.x=c("country","year"),
  by.y=c("nations","time"), all = TRUE)
```

# If one file has ‘country/year’, but the other only ‘country’,
# you can still merge.
# Values in the data with only ‘country’ will repeat in the ‘country/year’ data

```r
mydata <- merge(mydata1, mydata2, by=c("country"), all=TRUE)
```
Data source

*World Development Indicators (World Bank)*