Intro to Social Science Data Analysis

Lecture 2: Types of Data

Christopher Gandrud

September 5, 2012

1. Review
2. Why do we care about data?
3. General Types of Data
4. Data Frames in R
5. First Assignment

Recap

Last week we:

- **Installed** R, RStudio, & Dropbox
- Learned how to **Compile Notebooks**
- Discussed how R is an **object-oriented** programming language.
  - Basic object modes
  - Basic object types
  - Commands, Functions, & Arguments
- How to **install add-on packages**

Recap

Last week we:

- **Installed** R, RStudio, & Dropbox
- Learned how to **Compile Notebooks**
- Discussed how R is an **object-oriented** programming language.
  - Basic object modes
  - Basic object types
  - Commands, Functions, & Arguments
- How to **install add-on packages**

Recap

Last week we:

- **Installed** R, RStudio, & Dropbox
- Learned how to **Compile Notebooks**
- Discussed how R is an **object-oriented** programming language.
  - Basic object modes
  - Basic object types
  - Commands, Functions, & Arguments
- How to **install add-on packages**

Recap

Last week we:

- **Installed** R, RStudio, & Dropbox
- Learned how to **Compile Notebooks**
- Discussed how R is an **object-oriented** programming language.
  - Basic object modes
  - Basic object types
  - Commands, Functions, & Arguments
- How to **install add-on packages**

Recap

Last week we:

- **Installed** R, RStudio, & Dropbox
- Learned how to **Compile Notebooks**
- Discussed how R is an **object-oriented** programming language.
  - Basic object modes
  - Basic object types
  - Commands, Functions, & Arguments
- How to **install add-on packages**

Recap

Last week we:

- **Installed** R, RStudio, & Dropbox
- Learned how to **Compile Notebooks**
- Discussed how R is an **object-oriented** programming language.
  - Basic object modes
  - Basic object types
  - Commands, Functions, & Arguments
- How to **install add-on packages**

Recap

Last week we:

- **Installed** R, RStudio, & Dropbox
- Learned how to **Compile Notebooks**
- Discussed how R is an **object-oriented** programming language.
  - Basic object modes
  - Basic object types
  - Commands, Functions, & Arguments
- How to **install add-on packages**

Quick Quiz

With a partner: describe the following code/output in as much detail as possible.

```
# A quick quiz
Population <- c(14.3, 6.3, 66.7)

Countries <- c("Cambodia", "Laos", "Thailand")

NewData <- cbind(Countries, Population)

sum(Population)

## [1] 87.3
```

Today: how do we handle data in R?

Why do we care about data?

Why do we care about data?

We want to answer questions.

Process of Investigation

We seek to answer questions with a **process of investigation** (Diez et al. 2011, 1):

1. Identify a question or problem.
2. *Think of possible answers (hypotheses).*
3. Collect relevant data on the topic.
4. Analyse the data.
5. Form a conclusion.

Process of Investigation

We seek to answer questions with a **process of investigation** (Diez et al. 2011, 1):

1. Identify a question or problem.
2. *Think of possible answers (hypotheses).*
3. Collect relevant data on the topic.
4. Analyse the data.
5. Form a conclusion.

Process of Investigation

We seek to answer questions with a **process of investigation** (Diez et al. 2011, 1):

1. Identify a question or problem.
2. *Think of possible answers (hypotheses).*
3. Collect relevant data on the topic.
4. Analyse the data.
5. Form a conclusion.

Process of Investigation

We seek to answer questions with a **process of investigation** (Diez et al. 2011, 1):

1. Identify a question or problem.
2. *Think of possible answers (hypotheses).*
3. Collect relevant data on the topic.
4. Analyse the data.
5. Form a conclusion.

Process of Investigation

We seek to answer questions with a **process of investigation** (Diez et al. 2011, 1):

1. Identify a question or problem.
2. *Think of possible answers (hypotheses).*
3. Collect relevant data on the topic.
4. Analyse the data.
5. Form a conclusion.

For Example

Research question: Are some authoritarian regimes more likely to go to war than others?

Hypothesis: Weeks (2012) hypothesised that military regimes are more likely to start wars than civilian authoritarian regimes and democracies.

Data Gathering: What data does she need to investigate this hypothesis?

For Example

Research question: Are some authoritarian regimes more likely to go to war than others?

Hypothesis: Weeks (2012) hypothesised that military regimes are more likely to start wars than civilian authoritarian regimes and democracies.

Data Gathering: Country-year data on regime type (military regime, civilian authoritarian regime, democracy), whether a war was started, & other factors (military power, level of economic development, etc.)

For Example

Research question: Are some authoritarian regimes more likely to go to war than others?

Hypothesis: Weeks (2012) hypothesised that military regimes are more likely to start wars than civilian authoritarian regimes and democracies.

Data Gathering: Country-year data on regime type (military regime, civilian authoritarian regime, democracy), whether a war was started, & other factors (military power, level of economic development, etc.)

Analyse Data & Form Conclusions: We'll talk about this more in parts 2 & 3 of the course.

Variables

Regime type, conflict, economic development etc. are **concepts**.

Concepts can be operationalised as **variables**.

For example, economic development is often operationalised as the variable Gross Domestic Product per Capita (GDP/Capita).

In a data set **variables** are usually the **columns**.

country	year	reg_4state	ocbu	ocbu_lag	se_ptrade_i_ocbu	se_high_equity_ocbu	se_eu_ocbu	se_basel_
Afghanistan	1987	1	0	NA	NA	NA	0.0000000	0.000000
Afghanistan	1988	1	0	0	NA	0.0000000	0.0000000	0.000000
Afghanistan	1989	1	0	0	NA	0.0000000	0.0000000	0.000000
Afghanistan	1990	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1991	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1992	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1993	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1994	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1995	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1996	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1997	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1998	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1999	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	2000	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	2001	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	2002	1	0	0	0.36137640	NA	0.0000000	0.000000
Afghanistan	2003	1	0	0	0.38224009	NA	0.0000000	0.000000
Afghanistan	2004	1	0	0	0.34683919	NA	0.0000000	0.000000
Afghanistan	2005	1	0	0	0.27944830	NA	0.0000000	0.000000
Afghanistan	2006	1	0	0	0.30699331	NA	0.0000000	0.000000
Albania	1987	1	0	NA	NA	NA	0.0000000	0.000000
Albania	1988	1	0	0	0.00000000	0.0000000	0.0000000	0.000000
Albania	1989	1	0	0	0.00254820	0.0000000	0.0000000	0.000000
Albania	1990	1	0	0	0.00143640	0.0000000	0.0000000	0.000000
Albania	1991	1	0	0	0.00480600	0.0000000	0.0000000	0.000000

Observations

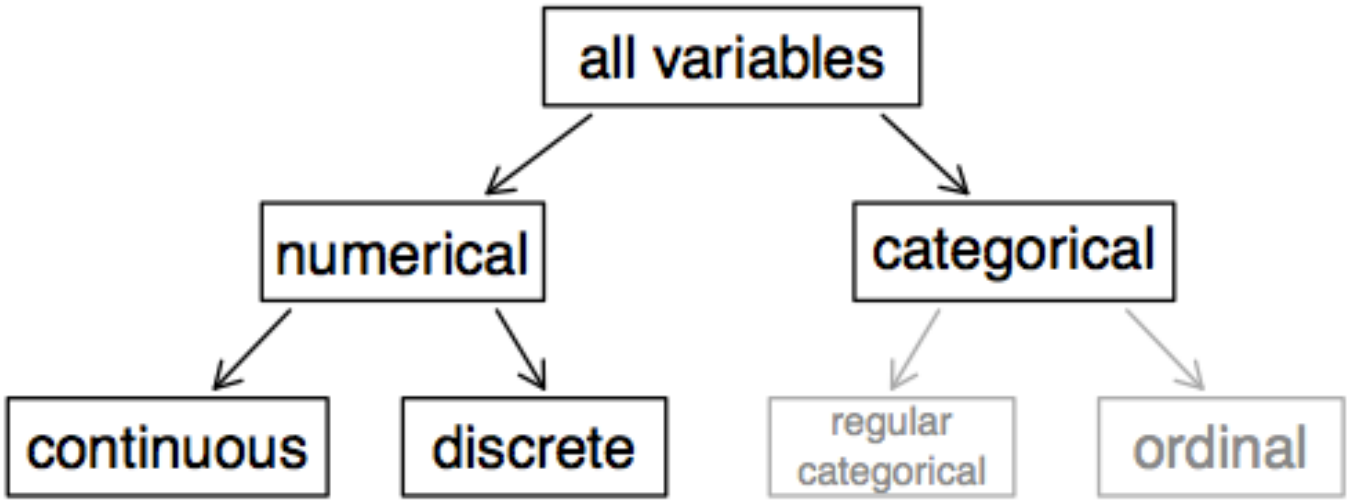
Each time we measure our variables we create an **observation**,

Observations are usually the **rows** of the data set.

country	year	reg_4state	ocbu	ocbu_lag	se_ptrade_i_ocbu	se_high_equity_ocbu	se_eu_ocbu	se_basel_
Afghanistan	1987	1	0	NA	NA	NA	0.0000000	0.000000
Afghanistan	1988	1	0	0	NA	0.0000000	0.0000000	0.000000
Afghanistan	1989	1	0	0	NA	0.0000000	0.0000000	0.000000
Afghanistan	1990	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1991	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1992	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1993	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1994	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1995	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1996	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1997	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1998	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	1999	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	2000	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	2001	1	0	0	NA	NA	0.0000000	0.000000
Afghanistan	2002	1	0	0	0.36137640	NA	0.0000000	0.000000
Afghanistan	2003	1	0	0	0.38224009	NA	0.0000000	0.000000
Afghanistan	2004	1	0	0	0.34683919	NA	0.0000000	0.000000
Afghanistan	2005	1	0	0	0.27944830	NA	0.0000000	0.000000
Afghanistan	2006	1	0	0	0.30699331	NA	0.0000000	0.000000
Albania	1987	1	0	NA	NA	NA	0.0000000	0.000000
Albania	1988	1	0	0	0.00000000	0.0000000	0.0000000	0.000000
Albania	1989	1	0	0	0.00254820	0.0000000	0.0000000	0.000000
Albania	1990	1	0	0	0.00143640	0.0000000	0.0000000	0.000000
Albania	1991	1	0	0	0.00480600	0.0000000	0.0000000	0.000000

Levels of Measurement

Variables can be at different **levels of measurement**.



Source: Diaz et al. (2011, 5)

Levels of Measurment Examples

		Also Known As	Example
Numerical	Continuous		GDP/Capita
	Discrete		People in a city
Categorical	Ordinal		Satisfaction with democracy (5-point scale)
	Binary	Dummy (0/1)	Gender
	Nominal	Regular Categorical	Country names

Levels of Measurement

Levels of measurement are important because they **determine** what kinds of **statistical analyses** we can do.

We'll talk more about this beginning Week 5.

Now: We need to keep in mind what level of measurement our data is in.

Tip: Try to have your data as close to **Continuous** as possible.

Data Frames 1

The main type of object we will use in R to store data are called **data frames**.

So far we have worked with matrices.

Matrices have rows and columns.

```r
# A quick matrix example
Population <- c(14.3, 6.3, 66.7)

Countries <- c("Cambodia", "Laos", "Thailand")

NewData <- cbind(Countries, Population)

NewData
```

```
##      Countries  Population
## [1,] "Cambodia" "14.3"
## [2,] "Laos"     "6.3"
## [3,] "Thailand" "66.7"
```

Matrices vs. Data Frames

Matrices can only have data with **one mode**.

Data frames can have **multiple modes**.

To create a data frame from multiple vectors use the `data.frame` command.

```r
# A quick data.frame example

NewData <- data.frame(Countries, Population)

NewData

##   Countries Population
## 1  Cambodia       14.3
## 2      Laos        6.3
## 3  Thailand       66.7
```

Multi-mode data frames

```
# Check variables' class
class(NewData$Countries)

## [1] "factor"


class(NewData$Population)

## [1] "numeric"
```

New Things

- What is the dollar sign ($)?
- What is a `factor`?

New Things

- What is the dollar sign ($)?
- What is a `factor`?

Factors 1

Factors are an R term for **categorical** variables.

Component selector

In R the $ is called the **component selector**.

It allows us to **select a specific column** of a data set.

```
# Select the Countries variable from NewData
NewData$Countries

## [1] Cambodia Laos     Thailand
## Levels: Cambodia Laos Thailand
```

Factors 2

We now have the `Countries` variable and can see its `Levels`

Giving this variable `Levels` doesn't really make sense.

One solustion is to use the `stringsAsFactors = FALSE` option with `data.frame`.

```r
# Create data.frame with no factors
NewData <- data.frame(Countries,
                      Population,
                      options(
                        stringsAsFactors = FALSE))

NewData$Countries

## [1] "Cambodia" "Laos"     "Thailand"


# Show NewData variable names
names(NewData)

## [1] "Countries"         "Population"
## [3] "stringsAsFactors"
```

Subsetting 1

How do we get rid of the `StringsAsFactors` variable?

Use subscripts to **subset** the data!

These are the square braces: `[]`.

All cells in an object have an **address**: `[row, column]`.

We want to keep the first and second columns:

```
# Subset NewData, columns 1 & 2
NewData <- NewData[, 1:2]

# Show variable names
names(NewData)

## [1] "Countries"  "Population"
```

We'll play around with subscripts a lot more in the seminar.

Subsetting Preview

Preview: We can subset not just by location but also by **observation value**.

For example, to subset the data to include only countries with populations greater than 7 million:

```
# Create new object for countries with > 7m pop.
MoreThan7 <- NewData[NewData$Population > 7, ]

# Show contents of MoreThan7
MoreThan7

##   Countries Population
## 1  Cambodia       14.3
## 3  Thailand       66.7
```

Before the seminar it might be a good idea to look at the help file for the `subset` command.

```
?subset
```

Factor Level Assignment 1

What if we have a variable without factor levels, but want to assign them?

Factor Level Assignment 2

Level	Code
Coastal	1
Not Coastal	0

```
# Create variable
Coastal <- c(1, 0, 1)

# Combine with Countries
CoastalDF <- data.frame(Countries,
                        Coastal,
                        options(
                          stringsAsFactors = FALSE))

# Remove stringsAsFactors variable
CoastalDF <- CoastalDF[, 1:2]
```

```r
# Merge with NewData
MergedData <- merge(x = NewData,
                    y = CoastalDF,
                    by = "Countries")

# Show variable names
names(MergedData)

## [1] "Countries"  "Population" "Coastal"


# Show the Coastal variables class
class(MergedData$Coastal)

## [1] "numeric"
```

Factor Level Assignment 3

Use the `factor` command to add the factor levels

```
# Turn Coastal into a factor & specify levels
MergedData$Coastal <- factor(MergedData$Coastal,
                             labels = c(
                               "Not Coastal",
                               "Coastal"))

# Show levels
MergedData$Coastal

## [1] Coastal     Not Coastal Coastal
## Levels: Not Coastal Coastal
```

Merging 1

What was this?

```
# Merge with NewData
MergedData <- merge(x = NewData,
                    y = CoastalDF,
                    by = "Countries")
```

Merging 2

We saw how you can use the `cbind` command to attach columns together.

This is usually **not** a good way to two data sets together.

For `cbind` the observations have to be in the **same order** in both objects.

This is **very uncommon**.

Merging 2

The `merge` command matches each observation.

You tell it what the **observation ID variable** is with the `by` argument.

For example `by = "Countries"`

We'll see more of this next week.

Due: Monday 24 September

Create a new data frame with country-level data from at least **three** different sources.

Create a folder in your Dropbox `Public` folder and **email me the link**.

The folder needs to include:

1. The new data frame saved as a `.csv` file.
2. A text file **describing the variables and their sources**.
3. A notebook `.html` file detailing how you created the data frame and saved it as a `.csv`.

References I

Diaz, David M., Christopher D. Barr, and Mine Çetinkaya-Rundel. OpenIntro Statistics. 1st ed.
`http://www.openintro.org/stat/downloads.php`.

Weeks, Jessica L. 2012. Strongmen and Straw Men: Authoritarian Regimes and the Initiation of International Conflict. American Political Science Review 106(2): 326347.