Basic Data Analysis

Stephen Turnbull

Business Administration and Public Policy

Lecture 9: June 6, 2013

Abstract

Regression and R.

Regression

- Correlation shows the *statistical strength* of a relationship: how far two variables are from being independent. At a correlation of 1 (or -1), two variables are *perfectly correlated*.

- From a policy standpoint, although correlation between policy and result is necessary (if the result is independent of the policy, there's no point in conducting policy), if the *functional strength* of the relationship is weak, then the policy will be ineffective.

- With imperfect correlation, the relation of changes in one variable to changes in another is uncertain.

- A *regression model* specifies *a combined functional and statistical model, allowing simultaneous estimation of both functional parameters and statistical ones.*

The regression model

- We identify a *dependent (random) variable* Y, and one or more *independent (random) variables* X_1, ..., X_n.

- *Endogenous* is a near synonym for *dependent. Explanatory* is a synonym for *independent*, and *exogenous* is a near synonym.

- We assume a *functional relationship* among the variables, $Y = f(X)$, and the *statistical model* that $\epsilon = Y - f(X)$ is a random variable with mean zero ($f(X)$ is an unbiased predictor of Y), and *known* distribution across observations.

- In a data set, this becomes $\epsilon^t = Y^t - f(X^t)$. That is, each observation contains a measurement of Y and of each independent variable X_i. ϵ^t is unobservable, and f is unknown. The problem is to determine f.

The basic linear regression model

- We want to simplify the problem.
- First, we simplify the statistical model by assuming that in the data set, ϵ^1, ..., ϵ^T (T observations on all variables) are *i.i.d.* with mean 0 and variance σ^2.
- Next, we simplify the functional model by assuming that the unknown characteristics are *linear*. That is, the model is that there are coefficients a_T, ..., a_n and $f(X) = \sum_{i=1}^n a_i X_i$.
- We can rewrite the model now as

$$Y^t = a_1 X_1^t + \cdots + a_n X_n^t + \epsilon^t, \quad t = 1, \ldots, T.$$

Linear regression

- We often include the *degenerate* or *trivial* random variable $X_1^t \equiv 1$. Then a_1 is the Y-*intercept* of the equation.
  - Statistical packages handle the intercept in different ways. Some require you to specify it explicitly, using a predefined variable (often C or 1). Some provide an option to the regression command to add an intercept term, others provide an option to suppress the intercept term.

- Use of t for "time" is obvious, but it might be that t identifies individuals in a sample, or any other way of collecting observations (*e.g.*, one for each of the prefectures of Japan).

Estimating the parameters

- Our parameters are a_1, ..., a_n, and σ^2.
  - Don't forget σ^2!
  - ϵ is *not* a parameter! It's an unobservable r.v.
- The means of all ϵ^t are *known* to be 0.
- Several strategies for estimation: pick the a_is to
  - Minimize $\sum_{t=1}^{T}(e^t)^2$ where $e^t = Y^t - \sum_{i=1}^{n} a_i X_i^t$ (the *least squares* strategy). This strategy automatically results in $\sum_{t=1}^{T} e_t = 0$.
  - Constrain $\frac{1}{n}\sum_{t=1}^{T} e^t = 0$ and *maximize likelihood* of the configuration of e^ts.

The least-squares formula

- In this model (i.i.d. with symmetric distributions for the ϵ^t), all the plausible strategies lead to the same computation.

- In the *bivariate model with intercept* $Y_t = a + bX_t + \epsilon_t$ (note change of notation! parameters have different letters and the observation index is now a subscript), the formulæ are

$$\begin{aligned}
\hat{b} &= \frac{\sum_{t=1}^{T} x_t y_t}{\sum_{t=1}^{T} (x_t)^2} \\
\hat{a} &= \frac{\sum_{t=1}^{T} Y_t}{T} - \hat{b}\frac{\sum_{t=1}^{T} X_t}{T} \\
\hat{\sigma}^2 &= \frac{\sum_{t=1}^{T} e_t^2}{T-2}
\end{aligned}$$

where $x_t = X_t - \frac{1}{n}\sum_{t=1}^{T} X_t$, $x_t = Y_t - \frac{1}{n}\sum_{t=1}^{T} Y_t$, and $e_t = Y_t - \hat{Y}_t$.

Comments on the formula

- Note the denominator in the formula for $\hat{\sigma}^2$! This is an application of "degrees of freedom." In order to compute e_t, we first must compute $\hat{a}$ and $\hat{b}$, losing 2 degrees of freedom. To get an unbiased estimate of σ^2, we must inflate the sample standard deviation by the factor $\frac{n}{n-2} > 1$.
- The generalization to n variables, with or without intercept, is "a simple matter of linear algebra." We will leave it to the computer.

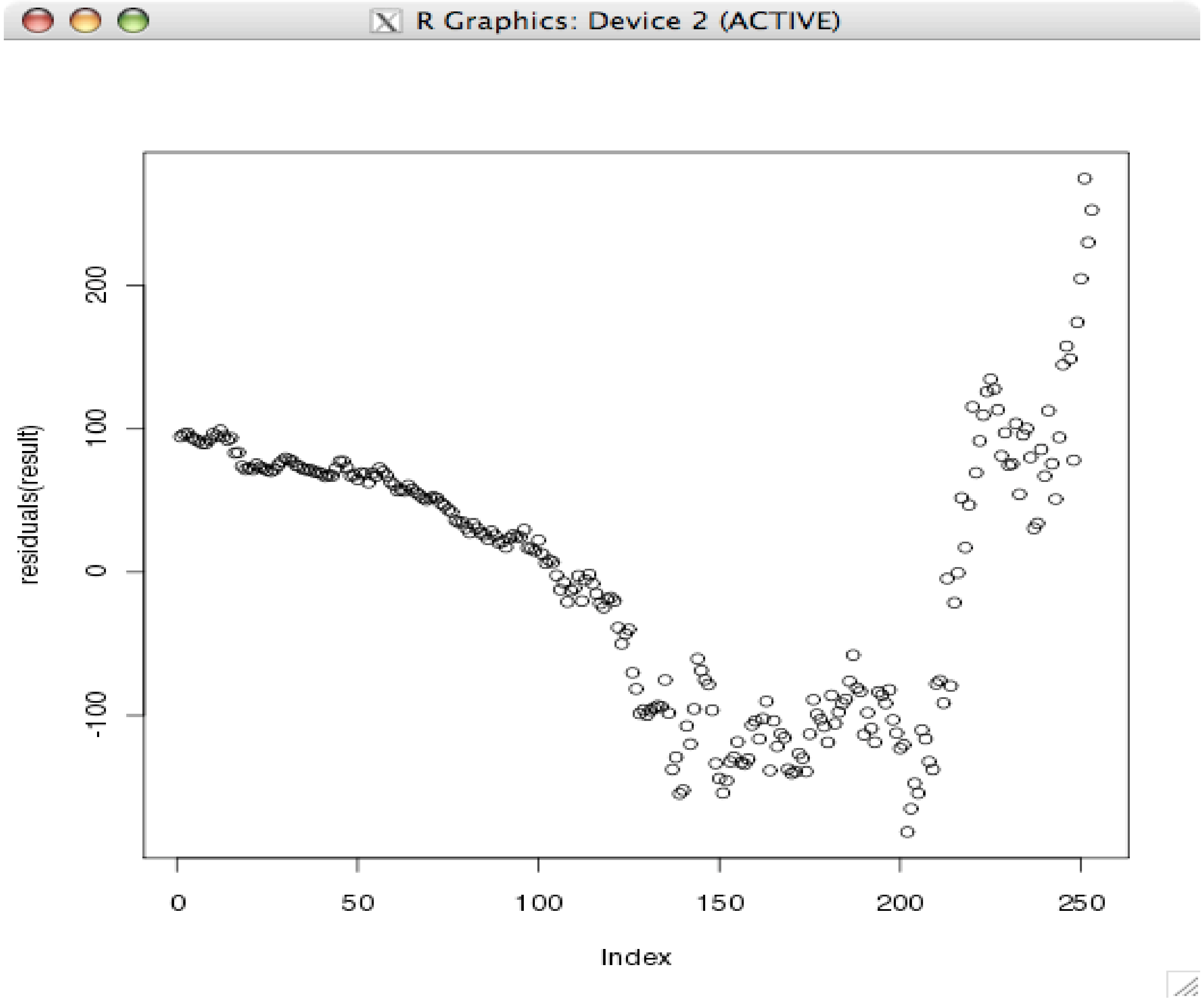
An example session: Regression results

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.042e+02  8.056e+00  -12.93   <2e-16 ***
GDP          6.995e-01  1.321e-03  529.34   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 91.73 on 251 degrees of freedom
Multiple R-squared: 0.9991,     Adjusted R-squared: 0.9991
F-statistic: 2.802e+05 on 1 and 251 DF,  p-value: < 2.2e-16

> plot(residuals(result))
```

A simple graph



That doesn't look very random!

Statistical software: general

- Modern statistical software is generally designed to use an ASCII character set to encode statistical terms. Thus "χ^2" becomes something like "`chi2`", and "$\Phi(z)$" becomes "`normal_cdf(z)`".
- Linear equations are typically reduced to lists of data variables, with the computed coefficients labelled with the variable name instead of special symbols.
- Variables generally have multiletter (and number) names, rather than being a single character as is typical in algebra.
- The biggest hurdle for most statisticians is learning to get the data in and out, and selecting subsets of data to work with. The actual statistical commands are usually easy to remember, and to look up if you forget.

The `R` statistical software package

- `R` is a free software implementation of the statistical programming language `Splus` developed and distributed by Bell Labs.
  - You can download it from `http://www.r-project.org/` for Windows and Mac, and some Unix systems. Most Linux and free BSD distributions have prebuilt packages.
  - `R` is not the easiest package to use. `Splus` and `SPSS` are probably much easier with GUIs, while `TSP` and `Shazam` are well-tuned to economics and many business applications. (I use it because it's free software, and offers some extra flexibility I sometimes need. Sorry.)
  - `R` does provide GUI for the Mac (at least Mac OS X 10.5 "Leopard") and Windows; I'll let you know how those work when I've tried them.

A session with the `R` statistical software package

Today we will use `R` to

- Load data from text and `.xls` files
- Print out data sets
- Do some simple regressions and look at the output summaries

Starting `R` and getting help

- To start `R` click on the icon, or type `R` on the command line.
- `R` help and manuals are all online, distributed with `R`. Type `help()` for information on the help system, or `help.start()` to bring up a list of resources such as manuals in your web browser (Firefox, Safari, Opera, or IE).
  - Following the trail `An Introduction to R > An Introductory Session > A sample session` and working through the examples is *strongly* recommended.
  - **Note:** `R` must be running for browser help to work!
- Type `demo()` to get a demonstration of how `R` works on some more or less real problems.

Loading data into `R`

- This is something that can be more annoying in `R` than in more GUI packages like SPSS.
- Start `R` (see last slide).
- Use `read.table` to read text tables or spreadsheets (including `.xls` and `.csv`).
- For the `.csv` files we use, use the form `db <- read.table("datafile.csv",sep=",",header=TRUE)`.

Hints on `read.table`

- If the form `db <- read.table("datafile.csv",sep=",",header=TRUE)` doesn't work, try reading `help(read.table)`. (Yes, I know it will make your head hurt. Do it anyway, you're a graduate student in training.)

- For some files, `sep` may be a semicolon or tab. Use a text editor (Notepad, Emacs, maybe Word) to look at the file.

- For some files, there may be no variable names, so use `header=FALSE` (or leave out the `header` option).

- The `data` function looks simpler, but that is because it is designed for use with data *pre-packaged for R*. This isn't worth the trouble for us.

- If that doesn't help (for most people, it's more pain than it's worth), ask an expert. *Try classmates first*, that's how they become experts!

An example session: Starting

```
chibi:DataAnalysis steve$ R

R version 2.11.0 (2010-04-22)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> help.start()
starting httpd help server ... done
If the browser launched by '/usr/bin/open' is already running, it is
    *not* restarted, and you must switch to its window.
Otherwise, be patient ...
```

An example session: Load and examine

```
> help(read.table)
> usgdp <- read.table("data/US-GDP-1947.1-2010.1.csv",sep=",",header=TRUE)
> usgdp[0:2]
    Year Quarter
1   1947       1
2   1947       2
3   1947       3
4   1947       4
5   1948       1
    [about 240 lines deleted]
249 2009       1
250 2009       2
251 2009       3
252 2009       4
253 2010       1
```

An example session: Examine parts

```
> usgdp$GDP[1:5]
[1] 237.2 240.4 244.5 254.3 260.3
> usgdp$.Goods[1:5]
[1]  95.6  98.3 100.4 103.5 105.1
> attach(usgdp)
> Year[1:5]
[1] 1947 1947 1947 1947 1948
> GDP[1:5]
[1] 237.2 240.4 244.5 254.3 260.3
```

An example session: A simple regression

```
> result <- lm(Consumption ~ GDP)
> summary(result)

Call:
lm(formula = Consumption ~ GDP)

Residuals:
    Min      1Q  Median      3Q     Max
-181.46  -93.33   24.45   71.70  274.71
```

A simple example

Due June 20, 11:45.

Consider the equation for free-fall: $s = s_0 + v_0 t + \frac{1}{2} g t^2$, where s is height in meters at any time, s_0 is the height in meters at the initial time, v_0 is velocity in meters per second at the initial time, g is acceleration due to gravity in meters per second per second, and t is time in seconds since the initial time. For the rest of this problem, set $s_0 = v_0 = 0$ and $g = 9.8$.

1. Generate the height s and the squared time t^2 for each second t from 0 to 100.

   Enter this data into your statistical program as three variables.

   (a) Generate the *correlation matrix* for the three variables.

   (b) Calculate the linear regression for height on time with an intercept (constant variable).

   (c) Calculate the linear regression for height on time and time-squared with an intercept (constant variable).

2. Generate the time t and time-squared t^2 for each height in meters from 0 to -100.

   Enter this data into your statistical program as three variables.

   (a) Generate the *correlation matrix* for the three variables.

   (b) Calculate the linear regression for height on time with an intercept (constant variable).

   (c) Calculate the linear regression for height on time and time-squared with an intercept (constant variable).

3. Compare the results of question 1 with those of question 2. That is, compare part 1a to 2a and so on.

Get the data

Due June 20, 11:45.

1. Get the data set `Section1All_csv.csv` from the home page.

   This data set has several sections with different kinds of data. *After reading and thinking about the rest of the problems*, pick one section; using data across sections is a bad idea.

2. Input the data into your statistical package, and print out the data of the section (only!—no fair printing everything and editing the output) you have picked.

   There are two basic ways to accomplish this: create a new data set with exactly the rows and columns you need, or input the whole thing and use the package to pick out "your" variables. Also, many packages prefer that variables be columns and rows be observations, but this sheet has the opposite orientation.

Correlation matrix

3. Generate a correlation matrix for all the variables in your section.
4. Think of some way in which *some* of the variables in your section are related. Refer to scientific theory where possible.

Define and estimate a model

5. Define a regression model for *the variables you picked*.
   (a) Explain why you picked the dependent variable.
   (b) Write down your regression model.
   (c) Estimate the regression model using your statistical package.

Add an unrelated variable

6. Add a random, and therefore unrelated, variable to the model.
   (a) Use Excel or your statistical package to generate a series of random numbers, enough to make a new variable for your data set.
   (b) Add it to the data set, and print out the data set (*i.e.*, your model variables plus the random variable).
   (c) Add the random variable to your model of problem 5 as an explanatory variable, and estimate the new regression model.
   (d) Define and execute a hypothesis test that the new variable is in fact statistically unrelated to the model.