Poli 5D Social Science Data Analytics

R: Regression & Function

Shane Xinyang Xuan
ShaneXuan.com

March 8, 2017

Contact Information

Shane Xinyang Xuan

xxuan@ucsd.edu

The teaching staff is a team!

Professor Roberts	M	1600-1800 (SSB 299)
Jason Bigenho	Th	1000-1200 (Econ 116)
Shane Xuan	M	1100-1150 (SSB 332)
	Th	1200-1250 (SSB 332)

Supplemental Materials

UCLA STATA starter kit

`http://www.ats.ucla.edu/stat/stata/sk/`

Princeton data analysis

`http://dss.princeton.edu/training/`

Announcement

- Problem Set 4 due on 3/15 at noon!

Announcement

- Problem Set 4 due on 3/15 at noon!
- Use my office hours if you need help

Announcement

- Problem Set 4 due on 3/15 at noon!
- Use my office hours if you need help
- Teaching evaluation is out

Announcement

- Problem Set 4 due on 3/15 at noon!
- Use my office hours if you need help
- Teaching evaluation is out
  - https://academicaffairs.ucsd.edu/Modules/Evals/

Announcement

- Problem Set 4 due on 3/15 at noon!
- Use my office hours if you need help
- Teaching evaluation is out
  - https://academicaffairs.ucsd.edu/Modules/Evals/
  - Your opinion is really important to my department so that we can gauge students' interests

Announcement

- Problem Set 4 due on 3/15 at noon!
- Use my office hours if you need help
- Teaching evaluation is out
  - `https://academicaffairs.ucsd.edu/Modules/Evals/`
  - Your opinion is really important to my department so that we can gauge students' interests
  - Your opinion is really important to me because it helps me improve my teaching

Announcement

- Problem Set 4 due on 3/15 at noon!
- Use my office hours if you need help
- Teaching evaluation is out
  - https://academicaffairs.ucsd.edu/Modules/Evals/
  - Your opinion is really important to my department so that we can gauge students' interests
  - Your opinion is really important to me because it helps me improve my teaching
  - Your opinion is really important to future students because the teaching staff are constantly making adjustments to the course so that students can get the most out of the it

Road map

Some quick notes before we start today's section:

- Make sure that you pass around the attendance sheet

Road map

Some quick notes before we start today's section:

- Make sure that you pass around the attendance sheet
- We will talk about regression in R

Road map

Some quick notes before we start today's section:

- Make sure that you pass around the attendance sheet
- We will talk about regression in R
- We will talk about function in R

Linea regression in R

- Syntax: `lm(y~x, data)`

Linea regression in R

- Syntax: `lm(y~x, data)`
  - Note that you can do a multiple regression using `lm(y~x1+x2+..., data)`

Linea regression in R

- Syntax: `lm(y~x, data)`
  - Note that you can do a multiple regression using `lm(y~x1+x2+..., data)`
- How to plot a regression line on a scatterplot?

Linea regression in R

- Syntax: `lm(y~x, data)`
  - Note that you can do a multiple regression using `lm(y~x1+x2+..., data)`
- How to plot a regression line on a scatterplot?
  - Step 1: Generate scatter plot using `plot(x,y)`

Linea regression in R

- Syntax: `lm(y~x, data)`
  - Note that you can do a multiple regression using `lm(y~x1+x2+..., data)`
- How to plot a regression line on a scatterplot?
  - Step 1: Generate scatter plot using `plot(x,y)`
  - Step 2: Fit regression using `lm(y~x, data)`

Linea regression in R

- Syntax: `lm(y~x, data)`
  - Note that you can do a multiple regression using `lm(y~x1+x2+..., data)`
- How to plot a regression line on a scatterplot?
  - Step 1: Generate scatter plot using `plot(x,y)`
  - Step 2: Fit regression using `lm(y~x, data)`
  - Recommendation: You can save the fit into an object called `m1`

Linea regression in R

- Syntax: `lm(y~x, data)`
  - Note that you can do a multiple regression using `lm(y~x1+x2+..., data)`
- How to plot a regression line on a scatterplot?
  - Step 1: Generate scatter plot using `plot(x,y)`
  - Step 2: Fit regression using `lm(y~x, data)`
  - Recommendation: You can save the fit into an object called `m1`
  - Step 3: Add the regression line using `abline(m1)`

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`
  - Fitted values: `fitted(m1)`

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`
  - Fitted values: `fitted(m1)`
  - Residuals: `resid(m1)`

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`
  - Fitted values: `fitted(m1)`
  - Residuals: `resid(m1)`
  - Get $\hat{\beta}_{0,}$, $\hat{\beta}_1$, ...: `coef(m1)`

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`
  - Fitted values: `fitted(m1)`
  - Residuals: `resid(m1)`
  - Get $\hat{\beta}_{0,}$, $\hat{\beta}_1$, ...: `coef(m1)`
- Residuals can be useful while subsetting our data

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`
  - Fitted values: `fitted(m1)`
  - Residuals: `resid(m1)`
  - Get $\hat{\beta}_{0,}$, $\hat{\beta}_1$, ...: `coef(m1)`
- Residuals can be useful while subsetting our data
- Explain the following code to me:

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`
  - Fitted values: `fitted(m1)`
  - Residuals: `resid(m1)`
  - Get $\hat{\beta}_{0,}$, $\hat{\beta}_1$, ...: `coef(m1)`
- Residuals can be useful while subsetting our data
- Explain the following code to me:

```
data$country[resid(m1)==max(resid(m1))]
```

Extract information from the fit in R

- Extracting information from the fit (for example, `m1`)
  - Get a summary: `summary(m1)`
  - Fitted values: `fitted(m1)`
  - Residuals: `resid(m1)`
  - Get $\hat{\beta}_{0,}$, $\hat{\beta}_1$, ...: `coef(m1)`
- Residuals can be useful while subsetting our data
- Explain the following code to me:

```
data$country[resid(m1)==max(resid(m1))]
```

  - Within the `data.frame` object `data`, find the country with maximum residuals

Save results in R

We can save the plots generated by R using the following code:

Save results in R

We can save the plots generated by R using the following code:

```
png("title.png")
... # your code here
dev.off()
```

Save results in R

We can save the plots generated by R using the following code:

```
png("title.png")
... # your code here
dev.off()
```

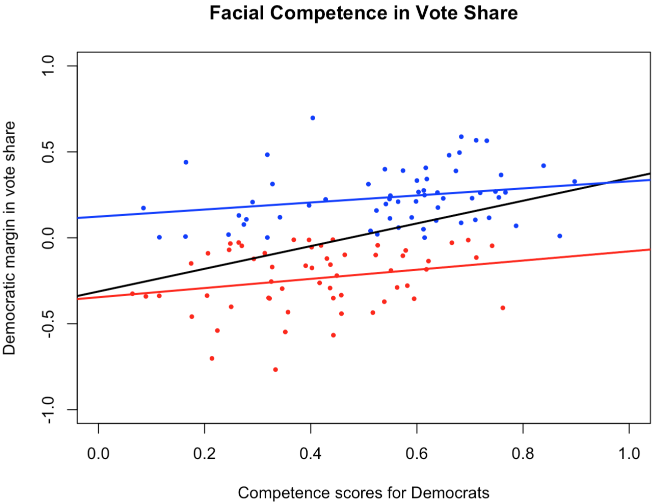
Note that the plots are saved in your working directory

Regression in R: Exercise

- We will look at the face data experiment (Imai 4.2)

Regression in R: Exercise

- We will look at the face data experiment (Imai 4.2)



Regression in R: Exercise

Read data

```
setwd("~/Dropbox/Poli5D/dataforlecture")
face <- read.csv("face.csv")
```

Regression in R: Exercise

Read data

```
setwd("~/Dropbox/Poli5D/dataforlecture")
face <- read.csv("face.csv")
```

Define vote share for Democrats and Republicans

```
face$d.share <- face$d.votes/(face$d.votes + face$r.votes)
face$r.share <- face$r.votes/(face$d.votes + face$r.votes)
face$diff.share <- face$d.share-face$r.share
```

Regression in R: Exercise

Read data

```
setwd("~/Dropbox/Poli5D/dataforlecture")
face <- read.csv("face.csv")
```

Define vote share for Democrats and Republicans

```
face$d.share <- face$d.votes/(face$d.votes + face$r.votes)
face$r.share <- face$r.votes/(face$d.votes + face$r.votes)
face$diff.share <- face$d.share-face$r.share
```

Regression

```
fit <- lm(diff.share~d.comp, data=face)
fit.D <- lm(diff.share~d.comp, data=face[face$w.party=="D",]
fit.R <- lm(diff.share~d.comp, data=face[face$w.party=="R",]
```

Regression in R: Exercise

Read data

```
setwd("~/Dropbox/Poli5D/dataforlecture")
face <- read.csv("face.csv")
```

Define vote share for Democrats and Republicans

```
face$d.share <- face$d.votes/(face$d.votes + face$r.votes)
face$r.share <- face$r.votes/(face$d.votes + face$r.votes)
face$diff.share <- face$d.share-face$r.share
```

Regression

```
fit <- lm(diff.share~d.comp, data=face)
fit.D <- lm(diff.share~d.comp, data=face[face$w.party=="D",]
fit.R <- lm(diff.share~d.comp, data=face[face$w.party=="R",]
```

Generate the plot

```
plot(face$d.comp, face$diff.share, pch=19, cex=0.5,
  col=ifelse(face$w.party=="R", "red", "blue"))
abline(fit.D, col="blue")
abline(fit.R, col="red")
abline(fit, lwd=2)
```

Function in R

- Make sure that you read Imai 1.3.4

Function in R

- Make sure that you read Imai 1.3.4
- We might want to create our own functions with `function()`

Function in R

- Make sure that you read Imai 1.3.4
- We might want to create our own functions with `function()`
- Syntax:

```
myfunction <- function(x,y,z,...){
  ... # your code here to get output
  return(output)
}
```

Function in R: Example

- Example:

```
ourExample <- function(n){
  sq <- n*n
  return(sq)
}
```

Function in R: Example

- Example:

```
ourExample <- function(n){
  sq <- n*n
  return(sq)
}
```

- What's going to be the output of `ourExample(2)`?

Function in R: Example

- Example:

  ```
  ourExample <- function(n){
    sq <- n*n
    return(sq)
  }
  ```

- What's going to be the output of `ourExample(2)`?

  ```
  > ourExample(2)
  [1] 4
  ```

Function in R: Exercise

- I have the formula[1]

$$A = P \times \left(1 + \frac{R}{100}\right)^n$$

Write R code to calculate A, where n changes from 1 to 15, if $P_0 = 5000$ and $R_0 = 11.5\%$.

[1] Exercise adapted from Adams & Stephens's *Introduction to R*, available at `http://wwwf.imperial.ac.uk/~das01/RCourse/`.

Function in R: Exercise

- I have the formula[1]

$$A = P \times \left(1 + \frac{R}{100}\right)^n$$

  Write R code to calculate A, where n changes from 1 to 15, if $P_0 = 5000$ and $R_0 = 11.5\%$.

- Result

```
> difference
[1] 81.19778
```

[1] Exercise adapted from Adams & Stephens's *Introduction to R*, available at `http://wwwf.imperial.ac.uk/~das01/RCourse/`.

Function in R: Exercise

Set up

```
P0 <- 5000
R0 <- 0.115
n0 <- 1
n1 <- 15
```

Function in R: Exercise

Set up

```
P0 <- 5000
R0 <- 0.115
n0 <- 1
n1 <- 15
```

Write the function

```
calculateA <- function(P, R, n){
  withinParenthesis <- 1+R/100
  compound <- withinParenthesis^n
  output <- P * compound
  return(output)
}
```

Function in R: Exercise

Set up

```
P0 <- 5000
R0 <- 0.115
n0 <- 1
n1 <- 15
```

Write the function

```
calculateA <- function(P, R, n){
  withinParenthesis <- 1+R/100
  compound <- withinParenthesis^n
  output <- P * compound
  return(output)
}
```

Results

```
A.year0 <- calculateA(P0, R0, n0)
A.year1 <- calculateA(P0, R0, n1)
difference <- A.year1 - A.year0
difference
```