

Lab 4: **R**markdown, package **dplyr**, package **stargazer**, and regression plots

Ngoc Phan

September 28, 2018

Intro to RMarkdown

- Create Markdown Document
- Knit to HTML/PDF/Word
- Headers
- **Bold**
- *Italic*
- Bullet points
- Embedded link
- R code chunks: Labels, options

Manipulating/cleaning data with dplyr

As I showed you in the last lab, cleaning data with base R can be tricky and confusing. However, the beauty of R is that you can import packages that make these tasks much more straightforward.

```
library(readstata13)
happy <- read.dta13("happy_planet.dta")
colnames(happy)
```

```
#install.packages("dplyr")
library(dplyr)
```

In the following subsections, we are going to contrast how we do certain tasks in base R and the way we do it in **dplyr**.

Select: Keeping and dropping variables

```
#Base R
happy1 <- happy[, c("country", "lifesat010", "hdi", "gdppercapitapp")]
colnames(happy1)

#Dplyr
happy2 <- select(happy, country, lifesat010, hdi)
happy2 <- select(happy, country:population)

#Drop variables
happy3 <- select(happy, -region)
```

Filter: Returning rows with matching conditions

```
#Base R
happy4 <- happy[happy$Africa == 1, ]

#Dplyr
happy4 <- filter(happy, Africa == 1)
happy5 <- filter(happy, population > 5)
```

Dropping missing values

```
#summary(happy)
#is.na(happy$hdi)

#filter(happy, is.na(hdi))
#filter(happy, !is.na(hdi))

happy <- filter(happy, !is.na(hdi))
```

Arrange: Sorting data

```
#Create a dataset of Western countries, keeping only four variables
west <- filter(happy, West == 1)
west <- select(west, country, lifesat010, hdi, population)

#Base R
#order(west$lifesat010)

west1 <- west[order(west$lifesat010), ]

#head(west)
west2 <- west[c(6, 4, 2), ]
#west2

#Dplyr
#arrange(west, lifesat010)
#arrange(west, -lifesat010)
#arrange(west, desc(lifesat010))

#arrange(west, hdi, population)
```

Mutate: Creating new variables

```
#Base R
west$pop <- west$population*1000000
#west

#Dplyr
```

```
west <- select(west, -pop)
west <- mutate(west, pop = population*1000000)
```

Nice regression output with Stargazer

```
m1 <- lm(lifesat010 ~ hdi + lifeexpyears, data = happy)
summary(m1)
```

```
#install.packages("stargazer")
library(stargazer)
stargazer(m1, title = "Regression of Life Satisfaction on HDI")
```

```
stargazer(m1, title = "Regression of Life Satisfaction on HDI",
  dep.var.labels = "Life Satisfaction",
  covariate.labels = c("Human Development Index", "GDP per capita PPP"))
```

Plot regression line

```
m2 <- lm(lifesat010 ~ hdi, data = happy)
plot(happy$hdi, happy$lifesat010, main = "Regression Line",
  xlab = "HDI", ylab = "Life Satisfaction")
abline(m2)
```

```
##?abline
#m1$coefficients

plot(happy$hdi, happy$lifesat010, main = "Regression Line",
  xlab = "HDI", ylab = "Life Satisfaction")
abline(m1$coefficients["(Intercept)"], m1$coefficients["hdi"])
```

Plot coefficients and confidence intervals

```
happy <- mutate(happy, hdi100 = hdi*100)
m3 <- lm(lifesat010 ~ hdi100 + gdppercapitappp + lifeexpyears, data = happy)
summary(m3)
```

```
#install.packages("arm")
library(arm)
coefplot(m3)
```

```
coefplot(m3, main = "Impact on Life Satisfaction",
  varnames = c("Intercept", "HDI", "GDP per capita PPP", "Life Expectancy"))
```

```
#summary(m3)$coefficients[, 1]
#summary(m3)$coefficients[, 2]
#summary(m3)$coefficients

coefplot(summary(m3)$coefficients[, 1],
```

```
summary(m3)$coefficients[, 2],  
main = "Impact on Life Satisfaction")
```

```
coefplot(summary(m3)$coefficients[c("hdi100", "gdppercapitapp"), 1],  
summary(m3)$coefficients[c("hdi100", "gdppercapitapp"), 2],  
varnames = c("HDI", "GDP per capita PPP"),  
main = "Impact on Life Satisfaction")
```