post02-arushi-desai

Arushi Desai

11/30/2017

Post 2: Reporting Tools and Data Visualization

Using the WDI Package

What is the WDI Package?

The WDI Package – or the **World Development Indicator Package** – is a package from which one can search, extract, and format data from the World Bank's World Development Indicators. The `WDI` function provides convenient access to the World Bank's WDI download API. For fast searching, the WDI package ships with a local list of available data series. This local list can be updated to the latest version using the `WDIcache` function.

Purpose

I chose to write Post #2 about this package because I think it is important to understand the global economic health, people's health, and more. This package can inform readers about the world's data, and using R, we can present that information visually and easily. I'm excited to dive in and explore WDI's functions and graphical possibilities!

Packages and Loading Data

To start out, we have to load the `WDI` package. Make sure you have it installed. If you don't, do so with `install.packages('WDI')`. We're going to import some other packages that will help us with graphical representation, as well, in the code chunk below. Another great thing about WDI is that data already comes with the package!

```
# load these packages!
library(WDI)
```

```
## Loading required package: RJSONIO
```

```
library(ggplot2)
library(ggvis)
```

```
##
## Attaching package: 'ggvis'
```

```
## The following object is masked from 'package:ggplot2':
##
##     resolution
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following objects are masked from 'package:ggvis':
##
##     add_data, hide_legend
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

Let's Go Through WDI's 4 Main Functions

Fun fact: we can also use regex when using `WDIsearch()`!

- `WDI`: downloads the requested data by using the World Bank's API, parses the resulting JSON file, and formats it in long country-year format. Arguments below.
  - `country`: vector of countries for which data is needed; can use string "all" instead of individual codes, which pulls data for every available country; countried called with ISO-2 character codes, e.g. "US", "CA"
  - `indicator`: character vector indicators codes; more information about this in the `WDIsearch()` function
  - `start`: first year of data
  - `end`: last year of data

- `extra` : TRUE returns extra variables such as region, iso3c code, and incomeLevel
- `cache` : NULL (optional), a list created by `WDIcache()` (see below) to be used with the `extra=TRUE` argument
- example:
  `WDI(country = "all", indicator = "NY.GNS.ICTR.GN.ZS", start = 2005, end = 2011, extra = FALSE, cache = NULL)`
- example: `WDI(country="all", indicator=c("AG.AGR.TRAC.NO","TM.TAX.TCOM.BC.ZS"), start=1990, end=2000)`
- example:
  `WDI(country=c("US","BR"), indicator="NY.GNS.ICTR.GN.ZS", start=1999, end=2000, extra=TRUE, cache=NULL)`

- `WDIcache` : downloads an updated list of available WDI indicators from the World Bank website and returns a data frame for use in the `WDIsearch` function
  - example: `WDIcache()`
- `WDIsearch` : data frame with series code, name, description, and source for the WDI series which match the given criteria. Arguments below.
  - `string` : character string; search for this string using `grep` with ignore.case = TRUE
  - `field` : character string; search this field; admissable fields are 'indicator', 'name', 'description', 'sourceDatabase', and 'sourceOrganization'
  - `short` : TRUE returns only the indicator's code and name; FALSE returns the indicator's code, name, description, and source
  - `cache` : data list generated by the `WDIcache` function; if omitted, `WDIsearch` will search a local list of series
  - example: `WDIsearch(string = "gdp", field = "name", short = TRUE, cache = NULL)`
  - example: `WDIsearch(string='gdp', field='name', cahce=NULL)`
  - example: `WDIsearch(string='AG.AGR.TRAC.NO', field='indicator', cache=NULL)`
- `WDI_data` : a list of two character matrices
  - first character matrix: includes full list of WDI series
  - second character matrix: includes extra country information

Let's Do Some Examples!

We've talked about the functions enough – let's get to applicability!

Searching for Data
We can search for data by using keywords in `WDIsearch`. For example, we want to look for data on Gross Domestic Product, or GDP:

```
head(WDIsearch('gdp'))
```

```
##      indicator
## [1,] "BG.GSR.NFSV.GD.ZS"
## [2,] "BM.KLT.DINV.GD.ZS"
## [3,] "BN.CAB.XOKA.GD.ZS"
## [4,] "BN.CUR.GDPM.ZS"
## [5,] "BN.GSR.FCTY.CD.ZS"
## [6,] "BN.KLT.DINV.CD.ZS"
##      name
## [1,] "Trade in services (% of GDP)"
## [2,] "Foreign direct investment, net outflows (% of GDP)"
## [3,] "Current account balance (% of GDP)"
## [4,] "Current account balance excluding net official capital grants (% of GDP)"
## [5,] "Net income (% of GDP)"
## [6,] "Foreign direct investment (% of GDP)"
```

This will give us the indicator and name of different statistics regarding GDP.
Remember we can use regex to look for specific indicators. Let's look for GDP per capita in constant dollars:

```
WDIsearch('gdp.*capita.*constant')
```

```
##      indicator
## [1,] "GDPPCKD"
## [2,] "NY.GDP.PCAP.KD"
## [3,] "NY.GDP.PCAP.KN"
## [4,] "NY.GDP.PCAP.PP.KD"
##      name
## [1,] "GDP per Capita, constant US$, millions"
## [2,] "GDP per capita (constant 2000 US$)"
## [3,] "GDP per capita (constant LCU)"
## [4,] "GDP per capita, PPP (constant 2005 international $)"
```

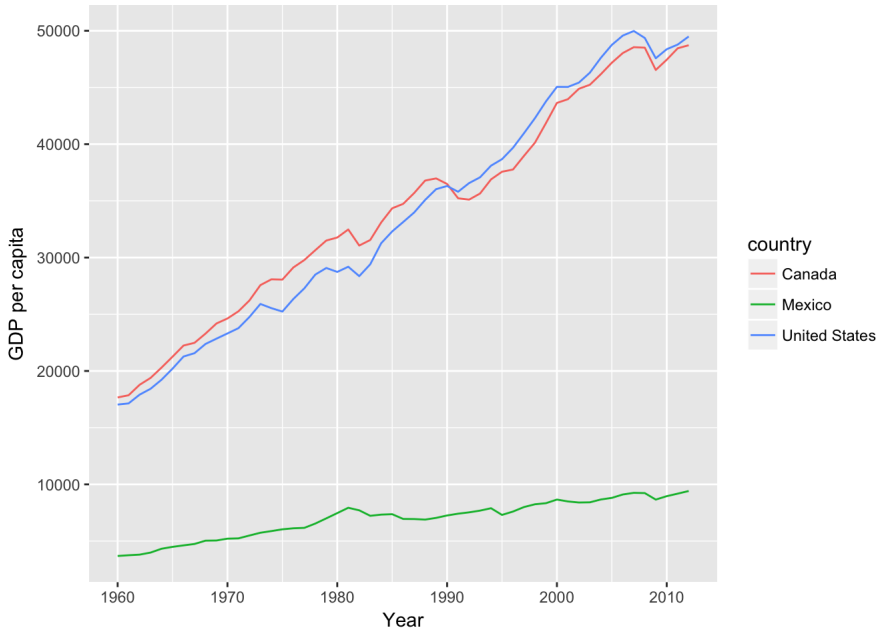
Downloading and Plotting Data Let's download some data for countries in North America and look at the data.

```
NA_dat = WDI(indicator='NY.GDP.PCAP.KD', country=c('MX','CA','US'), start=1960, end=2012)
head(NA_dat)
```

```
##   iso2c country NY.GDP.PCAP.KD year
## 1    CA  Canada       48724.25 2012
## 2    CA  Canada       48456.96 2011
## 3    CA  Canada       47447.48 2010
## 4    CA  Canada       46543.79 2009
## 5    CA  Canada       48510.57 2008
## 6    CA  Canada       48552.70 2007
```

Woohoo! Now let's plot this with a line plot to show the trends of GDP per capita.

```
ggplot(NA_dat, aes(year, NY.GDP.PCAP.KD, color=country)) + geom_line() +
    xlab('Year') + ylab('GDP per capita')
```



As you can see, WDI can be a powerful tool for us to understand trends in the world. It can also be used for different global financial magazines or newspapers, especially in conjunction with a powerful plotting tool like ggplot.

An In-Depth Example

We'll be looking at GDP per capita (constant 2010 US$), life expentancy at birth (total years), and infant mortality rate (per 1,000 live births) indicators.

First, make sure the data is up to date.

```
new_wdi_cache <- WDIcache()
```

Next, let's find the data we want to use for this example.

```
WDIsearch("gdp.*capita.*US\\$", cache = new_wdi_cache) # we want NY.GDP.PCAP.KD
```

```
##      indicator        name
## [1,] "NY.GDP.PCAP.KD" "GDP per capita (constant 2010 US$)"
## [2,] "NY.GDP.PCAP.CD" "GDP per capita (current US$)"
```

```
WDIsearch("life expectancy at birth.*total", cache = new_wdi_cache) # we want SP.DYN.LE00.IN
```

```
##                             indicator
##                        "SP.DYN.LE00.IN"
##                                  name
## "Life expectancy at birth, total (years)"
```

```
WDIsearch("^mortality.*rate.*infant", cache = new_wdi_cache) # we want SP.DYN.IMRT.IN
```

```
##      indicator
## [1,] "SP.DYN.IMRT.MA.IN"
## [2,] "SP.DYN.IMRT.IN"
## [3,] "SP.DYN.IMRT.FE.IN"
##      name
## [1,] "Mortality rate, infant, male (per 1,000 live births)"
## [2,] "Mortality rate, infant (per 1,000 live births)"
## [3,] "Mortality rate, infant, female (per 1,000 live births)"
```

Finally, let's download the data!

```
wdi_dat <- WDI(indicator = c("NY.GDP.PCAP.KD", "SP.DYN.LE00.IN", "SP.DYN.IMRT.IN"), start = 1960, end = 2015, extr
a = TRUE)
names(wdi_dat)
```

```
##  [1] "iso2c"          "country"        "year"           "NY.GDP.PCAP.KD"
##  [5] "SP.DYN.LE00.IN" "SP.DYN.IMRT.IN" "iso3c"          "region"
##  [9] "capital"        "longitude"      "latitude"       "income"
## [13] "lending"
```

Let's clean up this data, and remove all entries that are aggregated regional values.

```
wdi_dat <- subset(wdi_dat, region != "Aggregates") # this also removes NAs
```

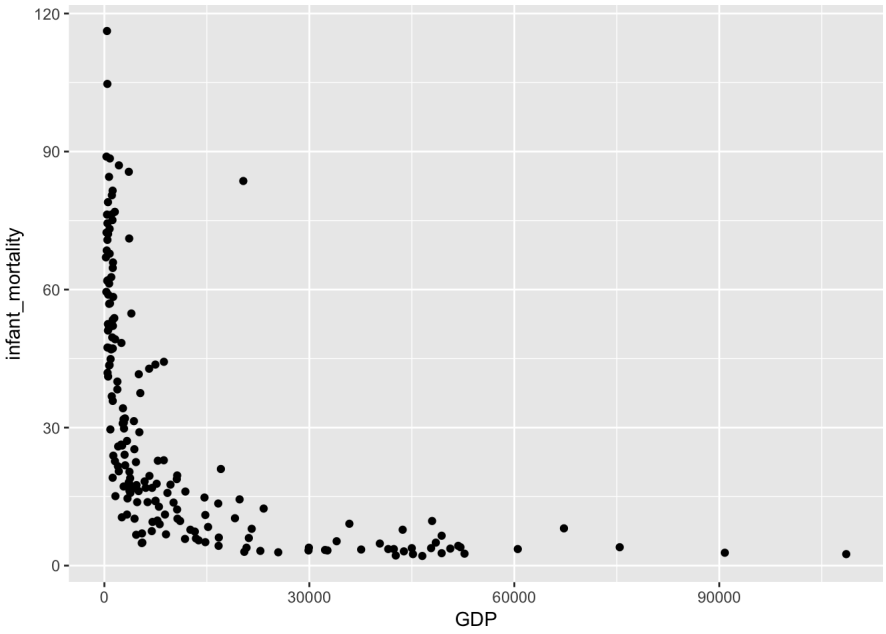
Now, rename the indicators so we can understand them better.

```
names(wdi_dat)[which(names(wdi_dat) == "NY.GDP.PCAP.KD")] <- "GDP"
names(wdi_dat)[which(names(wdi_dat) == "SP.DYN.LE00.IN")] <- "life_expectancy"
names(wdi_dat)[which(names(wdi_dat) == "SP.DYN.IMRT.IN")] <- "infant_mortality"
```

Let's plot some stuff!!
First, we'll look at the relationship between GDP and infant mortality.

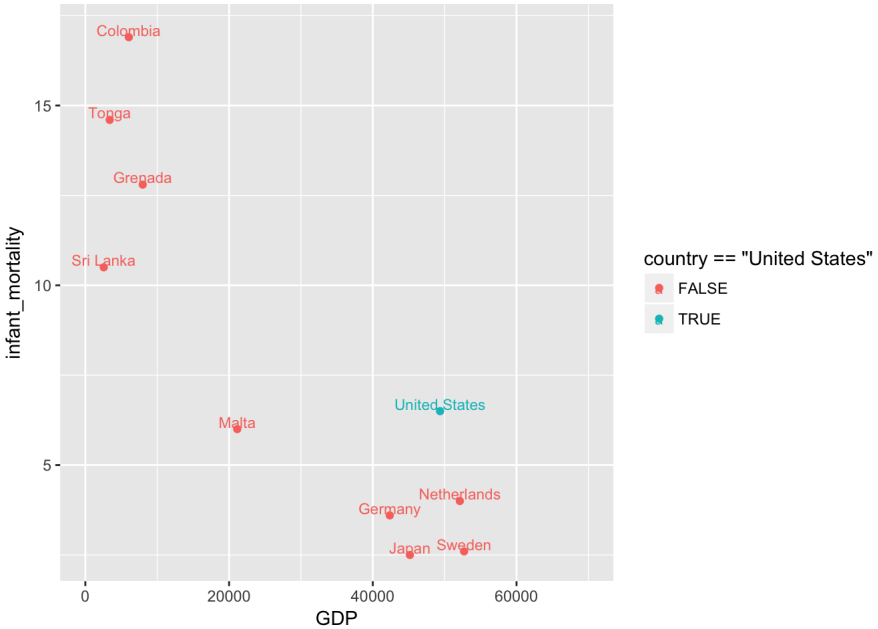
```
ggplot(subset(wdi_dat, year == 2008), aes(x = GDP, y = infant_mortality)) + geom_point()
```

```
## Warning: Removed 25 rows containing missing values (geom_point).
```



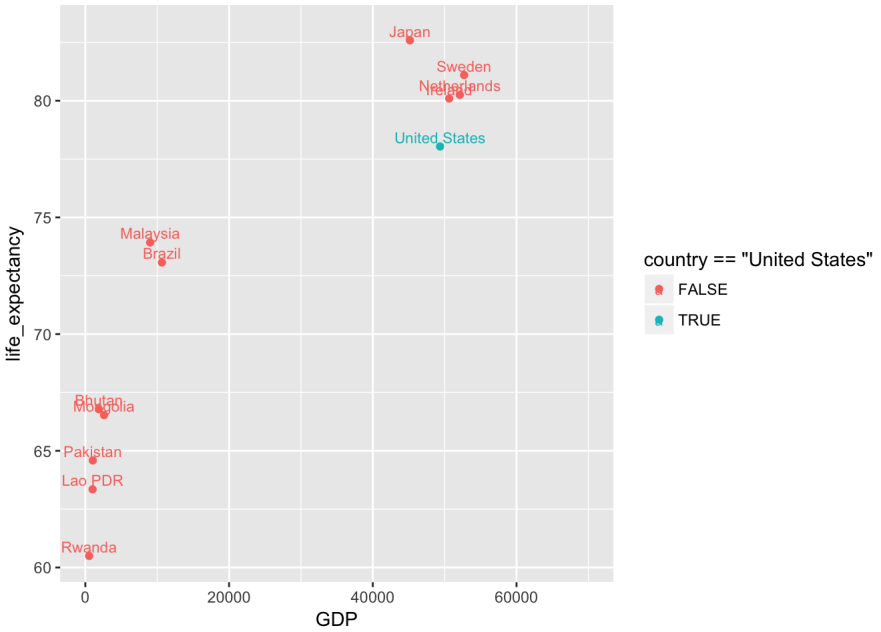
Now, let's compare the US GDP and infant mortality to that of other countries, both developed and less developed.

```
infmort_countries <- subset(wdi_dat, country %in% c("United States", "Tonga", "Colombia", "Grenada", "Sri Lanka",
"Malta", "Germany", "Japan", "Sweden", "Netherlands"))

ggplot(subset(infmort_countries, year == "2008"), aes(x = GDP, y = infant_mortality, color = country == "United St
ates")) +
  geom_point() +
  geom_text(aes(label = country), size=3, nudge_y = 0.2) +
  scale_x_continuous(limits = c(0, 70000))
```



Let's also compare the US GDP and life expentancy to that of some specific countries.

```
lifexp_countries <- subset(wdi_dat, country %in% c("United States", "Rwanda", "Mongolia", "Pakistan", "Lao PDR", "
Bhutan", "Malaysia", "Brazil", "Ireland", "Japan", "Sweden", "Netherlands"))

ggplot(subset(lifexp_countries, year == "2008"), aes(x = GDP, y = life_expectancy, color = country == "United Stat
es")) +
  geom_point() +
  geom_text(aes(label = country), size=3, nudge_y = 0.4) +
  scale_x_continuous(limits = c(0, 70000))
```



Let's compare GDP and infant mortality over multiple years.

```
g <- ggplot(subset(infmort_countries, year > "1999" & year <= "2015"), aes(x = GDP, y = infant_mortality, color =
country == "United States", tooltip = country)) +
  geom_point() +
  facet_wrap(~ year) +
  theme(legend.position="none")

g
```

Conclusion

And that's it! You are now an expert in WDI. Use this data and its features for understanding the world around you and being a more informed global citizen.

Resources

Find more information and examples from the resources where I drew my information from:

- R Documentation for WDI
- CRAN Documentation for WDI
- R Documentation for `WDI` function
- More Introductory Information about WDI

- RPubs, Helpful Graphical Examples
- Help with Plotting
- More Help with Plotting