# Programming in R

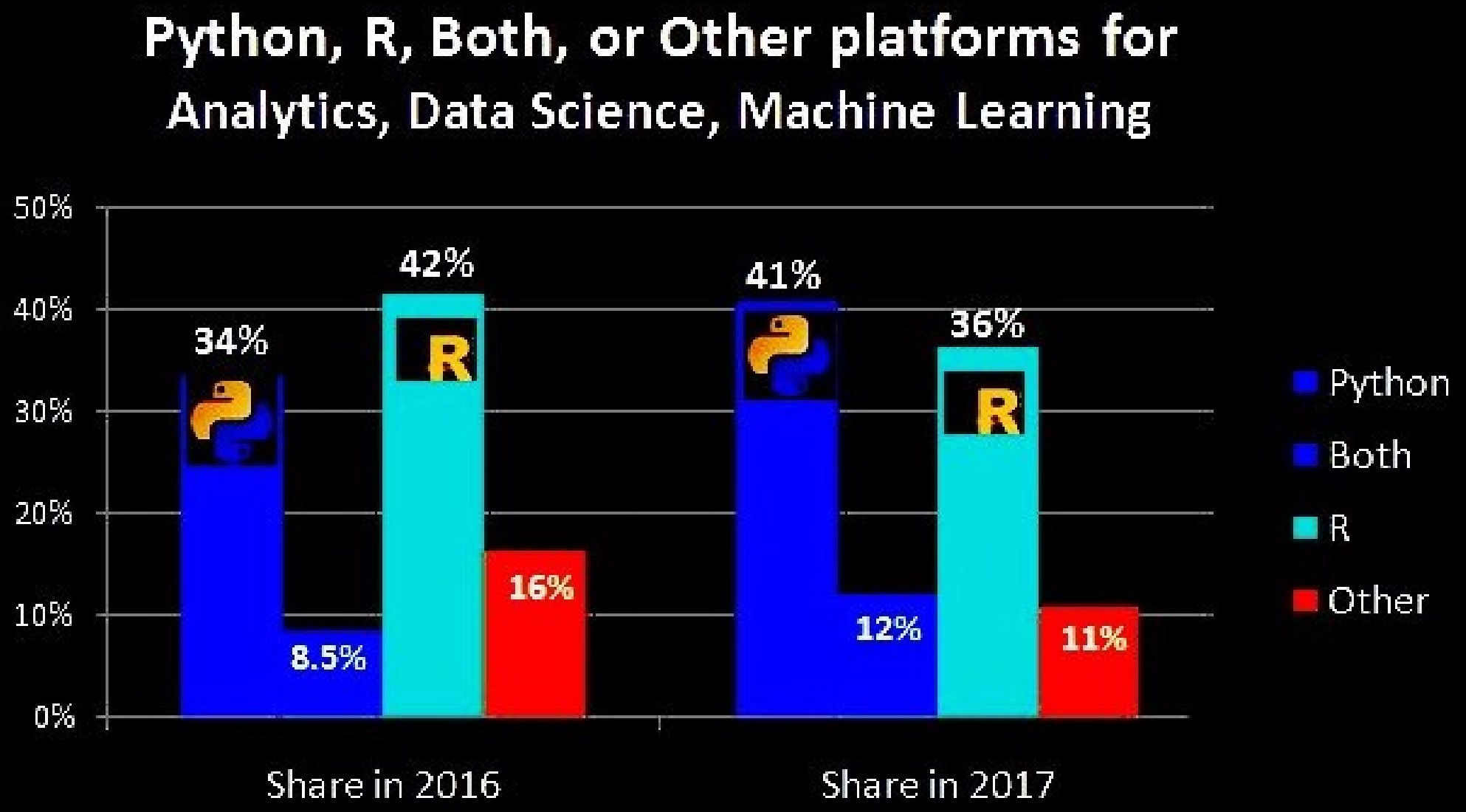
Zem Wang

28 Jun 2022

# Why R?

- The most popular statistical language
- Free and vibrant community
- Efficient and elegant code
- All weather academic writing solution

# Python vs R


Python, R, Both, or Other platforms for
Analytics, Data Science, Machine Learning
50%
40%
30%
20%
10%
0%
34%
8.5%
42%
16%
41%
12%
36%
11%
Share in 2016
Share in 2017
Python
Both
R
Other

# Python is beautiful...

```
1  import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
```

# Python is a disciplined language...

```
if type(value) == unicode: # all string returned should be unicode
    # if value contains non-ascii character (Chinese character),
    # set the actual column width to half the rwidth value.
    # because 1 Chinese character takes the space of 2 ASCII
    # characters.
    try: value.encode('ascii')
    except UnicodeEncodeError:
        actual = rwidth / 2

    # if the value string is longer then the required width,
    # split it into multiple lines.
    if len(value) > actual:
        for ind, line in enumerate(split(value, actual)):
            if ind > 0: formatted += '\n{0:<{1}} : '.format(' ', lwidth)
            formatted += line
    else:
        formatted += value
```

# but if you program out of the box...

```
func = lambda *vals: None
send_email = lambda receivers, subject, content: func(
    ['def'
        for sender in [['zem.wang@anu.edu.au',... ]]

        for msg in [(lambda mime_text: tail(
                        mime_text.__setitem__('subject', subject),
                        mime_text.__setitem__('from', head(*sender)),
                        mime_text.__setitem__('to', ';'.join(receivers)),
                        mime_text
                    ))(MIMEText(content, 'html', 'gbk'))]

        for smtp in [SMTP_SSL(host="smtp.office365.com", port=465)]
    ],
    smtp.login(*sender),
    smtp.sendmail(head(*sender), receivers, str(msg))
)
```

# and not friendly to lambdas...

```
def get_securities(portfolio_list):
    securities = pandas.concat(map(
                    lambda portfolio: (
                        (lambda table: (
                            table.assign(portfolio='.'.join(portfolio))
                        ))(wind.pos(*portfolio))
                    ), portfolio_list
                ), ignore_index=True)
    return securities
```

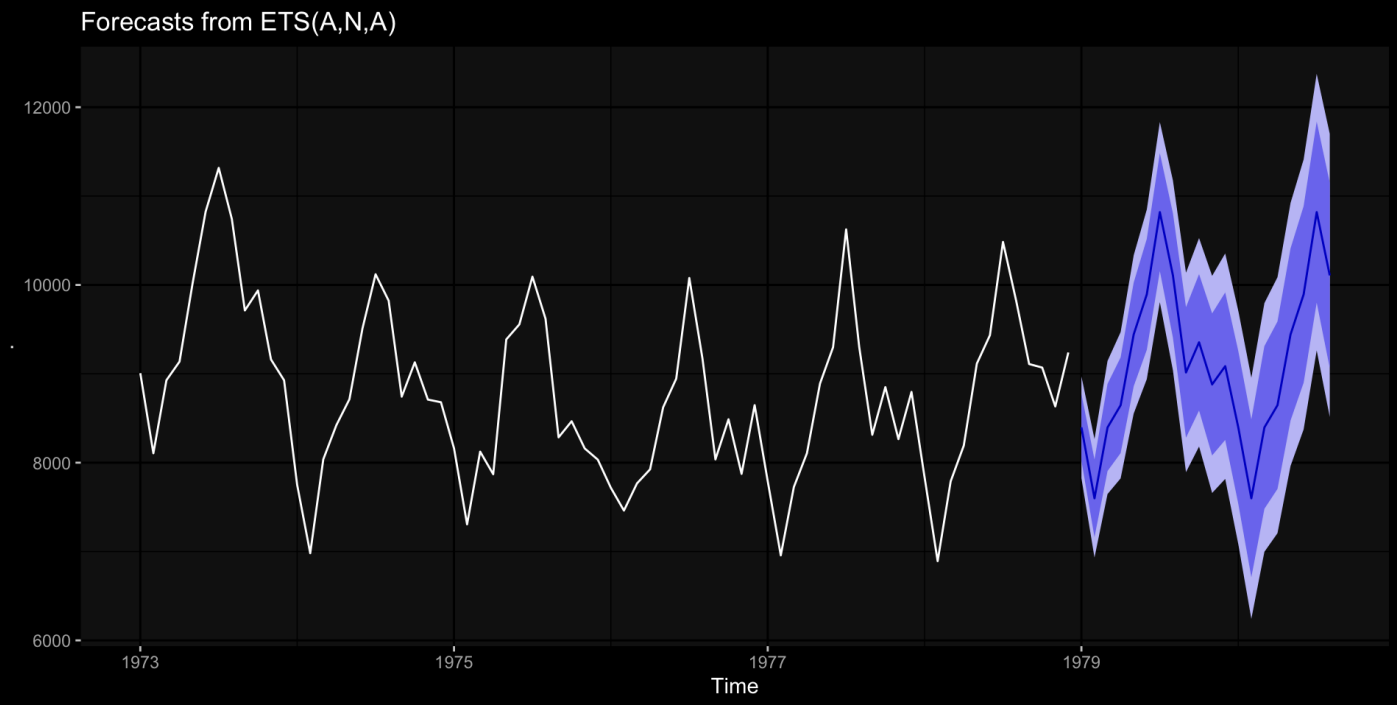
# The same functionality with R

R supports functional programming very well, and with pipes making code really readable.

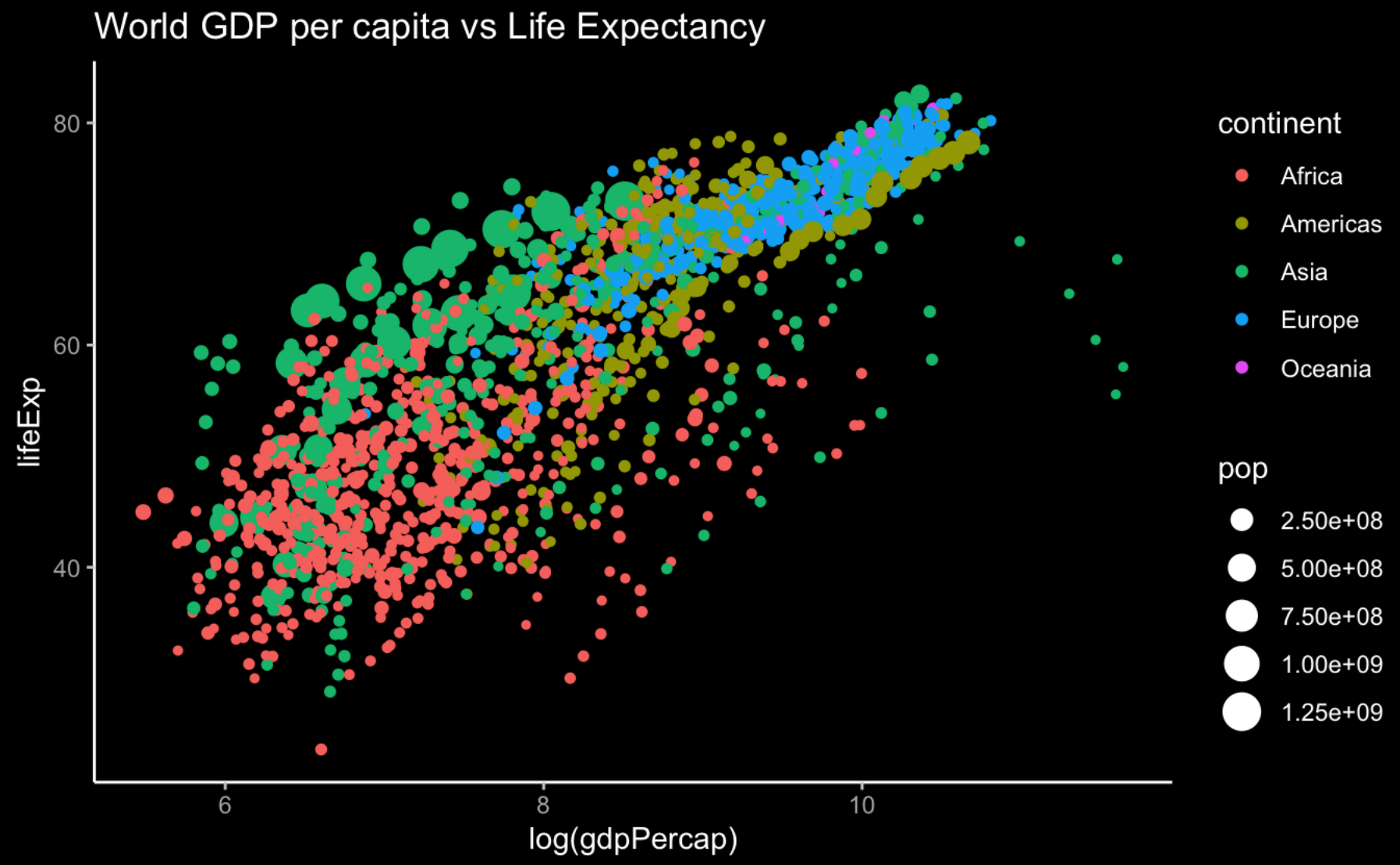
```
portfolio_list %>%
  get_position() %>%
  map(., cbind)
```

# Using R to make forecast

```r
library(forecast)

# forecast Accidental Deaths in the US 1973-1978
USAccDeaths %>%
  ets() %>%
  forecast(h=20) %>%
  autoplot()
```



# Data Visualization



# Stata Plot

```
#delimit ;
graph twoway
  (scatter lifeexp lgdp [aweight=pop]
   if continent == "Asia", mcolor(red))
  (scatter lifeexp lgdp [aweight=pop]
   if continent == "Europe", mcolor(blue))
  (scatter lifeexp lgdp [aweight=pop]
   if continent == "Africa", mcolor(yellow))
  (scatter lifeexp lgdp [aweight=pop]
   if continent == "Americas", mcolor(green))
   ,
   legend(label(1 "Asia") label(2 "Europe")
          label(3 "Africa") label(4 "Americas"))
   title("GDP per capita vs. life expectancy");
#delimit cr
```

- Plot object is an global variable – bad practice
- Plotting is not data visualizing
- Inseparable between data mapping and graph styling

# Do this in R – ggplot

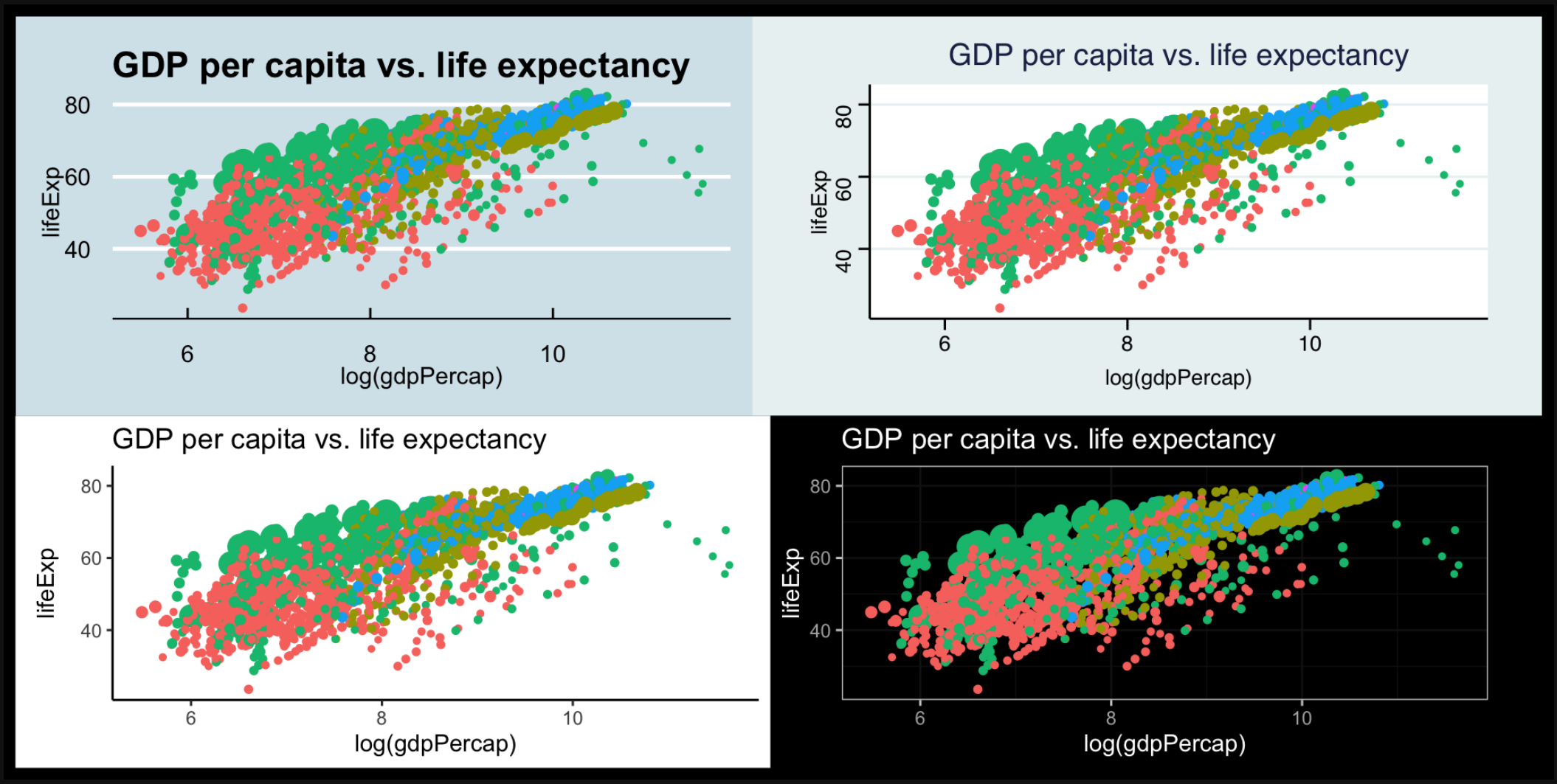
```
p = ggplot(gapminder, aes(
  x = log(gdpPercap),
  y = lifeExp,
  size = pop,
  color = continent)) +
  geom_point()

p = p + ggtitle("GDP per capita vs. life expectancy")
p = p + dark_theme_gray()
```

- The plot is an object, open to extension and modification
- Grammar of graphics: mapping between data and visual elements
- Separation between data visualization and graph styling

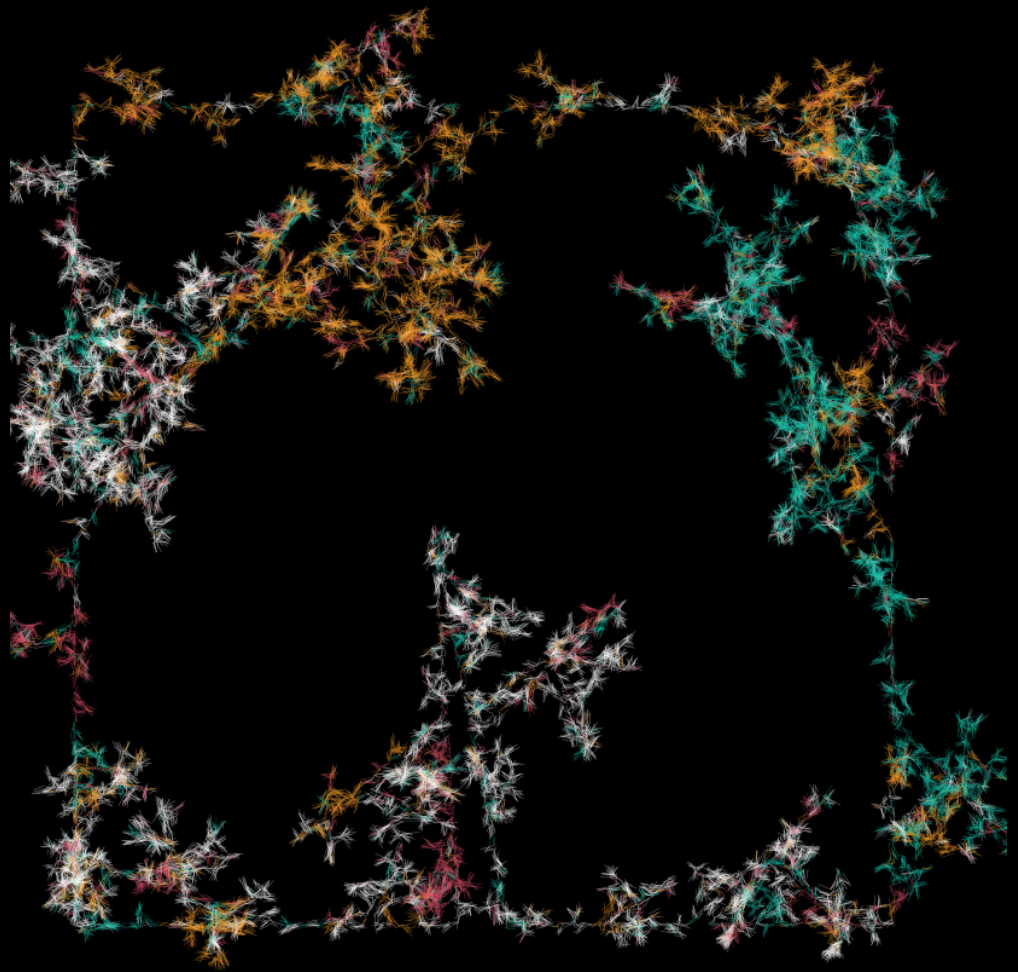
# Composition of plots

```
library(ggthemes)
library(patchwork)

p1 = p + theme_economist() + theme(legend.position = "none")
p2 = p + theme_stata()     + theme(legend.position = "none")
p3 = p + theme_classic()   + theme(legend.position = "none")
p4 = p + dark_theme_bw()   + theme(legend.position = "none")

p_all = (p1 + p2) / (p3 + p4)
```

# GGPlot with themes: showcase



# GGArt - Create art with ggplot!



# Data manipulation (wide to long)

GDP per capita of all countries in the world

| year | Austria | Australia | Brazil | India |
|---|---|---|---|---|
| 1952 | 6137.076 | 10039.60 | 2108.944 | 546.5657 |
| 1957 | 8842.598 | 10949.65 | 2487.366 | 590.0620 |
| 1962 | 10750.721 | 12217.23 | 3336.586 | 658.3472 |
| 1967 | 12834.602 | 14526.12 | 3429.864 | 700.7706 |
| 1972 | 16661.626 | 16788.63 | 4985.711 | 724.0325 |
| 1977 | 19749.422 | 18334.20 | 6660.119 | 813.3373 |
| 1982 | 21597.084 | 19477.01 | 7030.836 | 855.7235 |

# Data manipulation (to long)

| country | year | gdpPercap | lifeExp |
|---|---|---|---|
| Afghanistan | 1952 | 779.4453 | 28.801 |
| Afghanistan | 1957 | 820.8530 | 30.332 |
| Afghanistan | 1962 | 853.1007 | 31.997 |
| Afghanistan | 1967 | 836.1971 | 34.020 |
| Australia | 1952 | 10039.5956 | 69.120 |
| Australia | 1957 | 10949.6496 | 70.330 |
| Australia | 1962 | 12217.2269 | 70.930 |
| Australia | 1967 | 14526.1246 | 71.100 |

# Do it in Stata (the painful way)

```
import delimited "gdp.csv", clear
rename * gdp*
rename gdpyear year
reshape long gdp, i (year) j(country) string
save "gdp.dta"

import delimited "life.csv", clear
rename * lifexp*
rename lifexpyear year
reshape long lifexp, i (year) j(country) string
save "life.dta"

merge 1:1 country year using gdp
gen lgdp = ln(gdp)
regress lifexp lgdp, vce(robust)
```

- Stata expects variable names all begin with the same stubname
- Operate on global variables; original dataset vanishes after merging
- Non-Intuitive code especially for complex problems

# The more complex the problem, the more obscure the code...

```
foreach var in choice own_happiness family_s_happiness health romantic_life
  social_life control_over_your_life life_s_level_of_spirituality ///
  life_s_level_of_fun social_status life_s_non_boringness physical_comfort
  gen dm`var'=.
  forvalues i=2/11 {
    summarize `var' if question_number==`i', meanonly
    replace dm`var'=`var'-r(mean) if question_number==`i'
  }
}
noisily simex (choice1 = qnum1 qnum2 qnum3 qnum4 qnum5 qnum6 qnum7 qnum8 qn
(w1:_Mown_happiness) (w2:_Mfamily_s_happiness) (w3:_Mhealth) (w4:_Mromantic
(w5:_Msocial_life) (w6:_Mcontrol_over_your_life) (w7:_Mlife_s_level_of_spir
(w8:_Mlife_s_level_of_fun) (w9:_Msocial_status) (w10:_Mlife_s_non_boringnes
(w11:_Mphysical_comfort) (w12:_Msense_of_purpose), seed(339487731) suuinit(
gen binary_choice=.
replace binary_choice=0  if inlist(choice1,1,2,3)
replace binary_choice=1 if inlist(choice1,4,5,6)
esttab q4 q7 using column4and7.rtf, replace b(%9.2f) se(%9.3f) drop(qnum* _
```

# Let's do it in R

```r
library(tidyverse)

gdp <-
  read_csv("data/gdp.csv") %>%
  pivot_longer(
    cols = -1,
    names_to = "country",
    values_to = "gdpPercap"
  )

...

data <- full_join(gdp, life, by = c("country", "year"))

fit <- lm(lifeExp ~ log(gdpPercap), data)
```

# Functional programming

```
c("data/gdp.csv", "data/life.csv") %>%
  map(read_csv) %>%
  map2(c("gdpPercap", "lifeExp"),
       ~pivot_longer(
         data = .x,
         cols = -1,
         names_to = "country",
         values_to = .y)) %>%
  reduce(~full_join(.x, .y, by = c("country", "year"))) %>%
  lm(lifeExp ~ log(gdpPercap), data = .)
```

# Summarize regression results

```
1 library(modelsummary)
2
3 fit1 = lm(lifeExp ~ log(gdpPercap), data = gapminder)
4 fit2 = lm(lifeExp ~ log(gdpPercap) + log(pop), data = gapminder)
5 fit3 = lm(lifeExp ~ log(gdpPercap) + log(pop) + factor(continent), data = g
6
7 tbl = modelsummary(list(fit1, fit2, fit3), stars = T, coef_omit = "factor",
```

# Professional academic output

| | Model 1 | Model 2 | Model 3 |
|---|---|---|---|
| (Intercept) | −9.101*** | −28.771*** | −12.015*** |
| | (1.228) | (2.076) | (2.266) |
| log(gdpPercap) | 8.405*** | 8.344*** | 6.587*** |
| | (0.149) | (0.143) | (0.182) |
| log(pop) | | 1.279*** | 0.866*** |
| | | (0.111) | (0.111) |
| Num.Obs. | 1704 | 1704 | 1704 |
| R2 | 0.652 | 0.677 | 0.714 |

+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

# Auto-generated LaTex table!

```latex
\begin{table}
\centering
\begin{tabular}[t]{lccc}
\toprule
  & Model 1 & Model 2 & Model 3\\
\midrule
(Intercept) & \num{-9.101}*** & \num{-28.771}*** & \num{-12.015}***\\
 & (\num{1.228}) & (\num{2.076}) & (\num{2.266})\\
log(gdpPercap) & \num{8.405}*** & \num{8.344}*** & \num{6.587}***\\
 & (\num{0.149}) & (\num{0.143}) & (\num{0.182})\\
log(pop) &  & \num{1.279}*** & \num{0.866}***\\
 &  & (\num{0.111}) & (\num{0.111})\\
\midrule
Num.Obs. & \num{1704} & \num{1704} & \num{1704}\\
R2 & \num{0.652} & \num{0.677} & \num{0.714}\\
\bottomrule
\multicolumn{4}{l}{\rule{0pt}{1em}+ p $<$ 0.1, * p $<$ 0.05, ** p $<$ 0.01,
\end{tabular}
```

# RMarkdown (Quarto)

# RMarkdown overview

```
51  # Introduction
52
53  Academic workflow, certainly in political science, is at a crossroads. The *American
54
55  There are workflow implications to the Lacour controversy as well. Political science,
56
57  These recent events underscore the DART push and cast a shadow over our workflow. How
58
59  I think there is reason for optimism. We only struggle with it now because we have to
60
61  # Getting Started with YAML
62
63  The lion's share of a R Markdown document will be raw text, though the front matter m
64
65  ```{r eval=FALSE}
66  ---
67  output:
68    pdf_document:
69      citation_package: natbib
70      keep_tex: true
71      fig_caption: true
72      latex_engine: pdflatex
73      template: ~/Dropbox/miscelanea/svm-r-markdown-templates/svm-latex-ms.tex
74  title: "A Pandoc Markdown Article Starter and Template"
75  thanks: "Replication files are available on the author's Github account..."
76  author:
77  - name: Steven V. Miller
78    affiliation: Clemson University
79  - name: Mary Margaret Albright
80    affiliation: Pendelton State University
81  - name: Rembrandt Q. Einstein
82    affiliation: Springfield University
83  abstract: "This document provides an introduction to R Markdown, argues for its..."
```

A Pandoc Markdown Article Starter and Template *

**Steven V. Miller** *Clemson University*

This document provides an introduction to R Markdown, argues for its benefits, and presents a sample manuscript template intended for an academic audience. I include basic syntax to R Markdown and a minimal working example of how the analysis itself can be conducted within R with the `knitr` package.

*Keywords*: pandoc, r markdown, knitr

**Introduction**

Academic workflow, certainly in political science, is at a crossroads. The *American Journal of Political Science* (*AJPS*) announced a (my words) "show your work" initiative in which authors who are tentatively accepted for publication at the journal must hand over the raw code and data that produced the results shown in the manuscript. The editorial team at *AJPS* then reproduces the code from the manuscript. Pending successful replication, the manuscript moves toward publication. The *AJPS* might be at the fore of this movement, and it could be the most aggressive among political science journals, but other journals in our field have signed the joint Data Access & Research Transparency (DART) initiative. This, at a bare minimum, requires uploading code from quantitatively-oriented published articles to in-house directories hosted by the journal or to services like Dataverse.

There are workflow implications to the Lacour controversy as well. Political science, for the foreseeable future, will struggle with the extent of the data fraud perpetrated by Michael Lacour in an article co-authored with Donald P. Green in *Science*, the general scientific journal of record in the United States. A failure to reproduce LaCour's results with different samples uncovered a comprehensive effort by LaCour to "fake" data that provided results to what we felt or believed to be true (i.e. "truthiness"). However, fake data can have real consequences for both the researcher and those who want to learn from it and use it for various purposes. Even research done honestly may suffer the same fate if researchers are not diligent in their workflow.

These recent events underscore the DART push and cast a shadow over our workflow. However, good workflow has always been an issue in our discipline. Cloud storage services like Dropbox are still relatively new among political scientists. Without cloud storage, previous workflow left open the possibility that work between a home computer and an office computer was lost as a function of a corrupted thumb drive, an overheated power supply, or, among other things, the wave of viruses that would particularly affect Microsoft users every summer. Social sciences, un-

# Academic authorizing

- Content-focusing structured document
- LaTeX should have died, but hasn’t
- Intuitive and effortless structured documents with citation and cross-referencing capabilities
- The answer is R+Markdown

# RMarkdown structured document

```
# Heading level 1
## Heading level 2
### Heading level 3

You can add emphasis by making text **bold** or *italic*.

Embed LaTeX equations:

$E = ma^2$

Create a list:

- Item 1
- Item 2
- Item 3

Use `@` for citation @bond1995 or cross-referencing @fig-showcase.
```

# All weather solution for academics

# I have a dream...

> I have a dream that one day all students and researchers will forget what “formatting a paper” even means. I have a dream that one day journals and grad schools no longer have style guides. I have a dream that one day no missing $ is inserted, and \hbox is never overfull. [And this is not talking about MS Word.] —— Yihui Xie