R for Data Science

https://r4ds.had.co.nz/

Il-Youp Kwak, PhD

Objectives

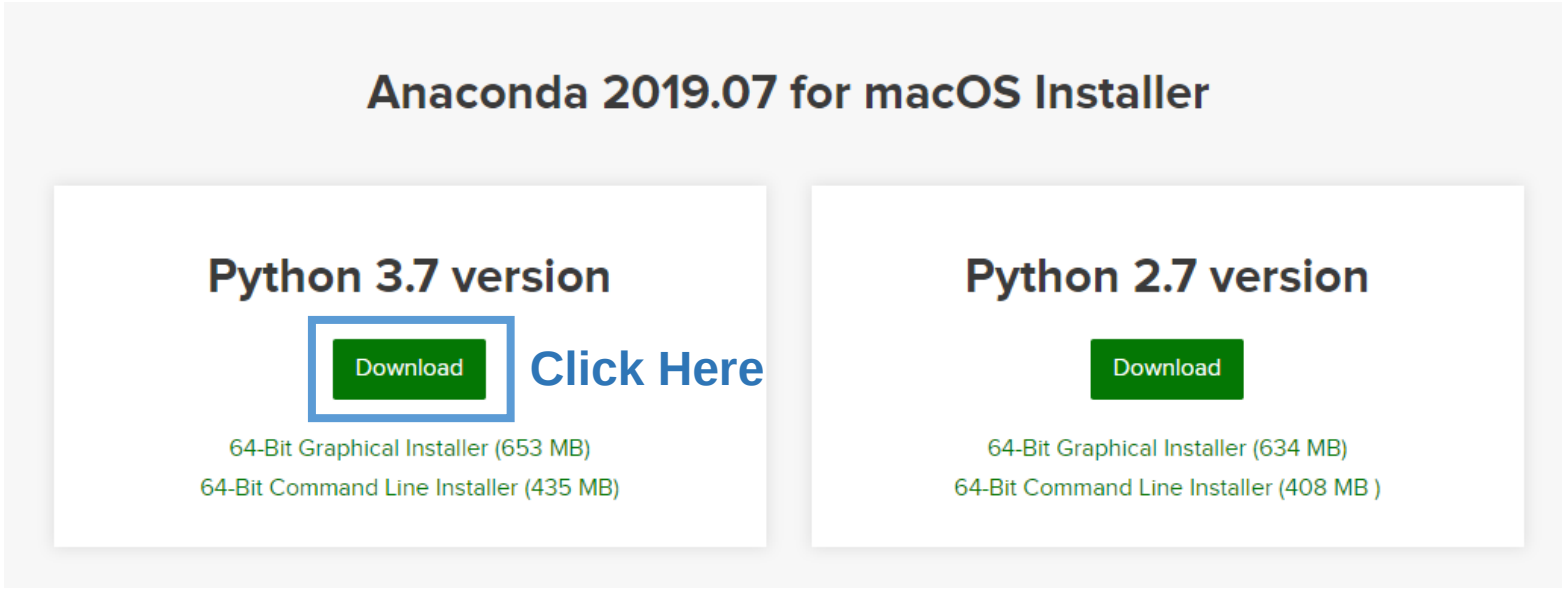
- **Learn some basics in** https://r4ds.had.co.nz/
- **Especially, dplyr and ggplot**

Try Jupyter Notebook with R

- **Easier to learn**
- **Can work on both R and Python**
- **Easy to read .ipynb from github unlike .Rnw**

What is Anaconda?

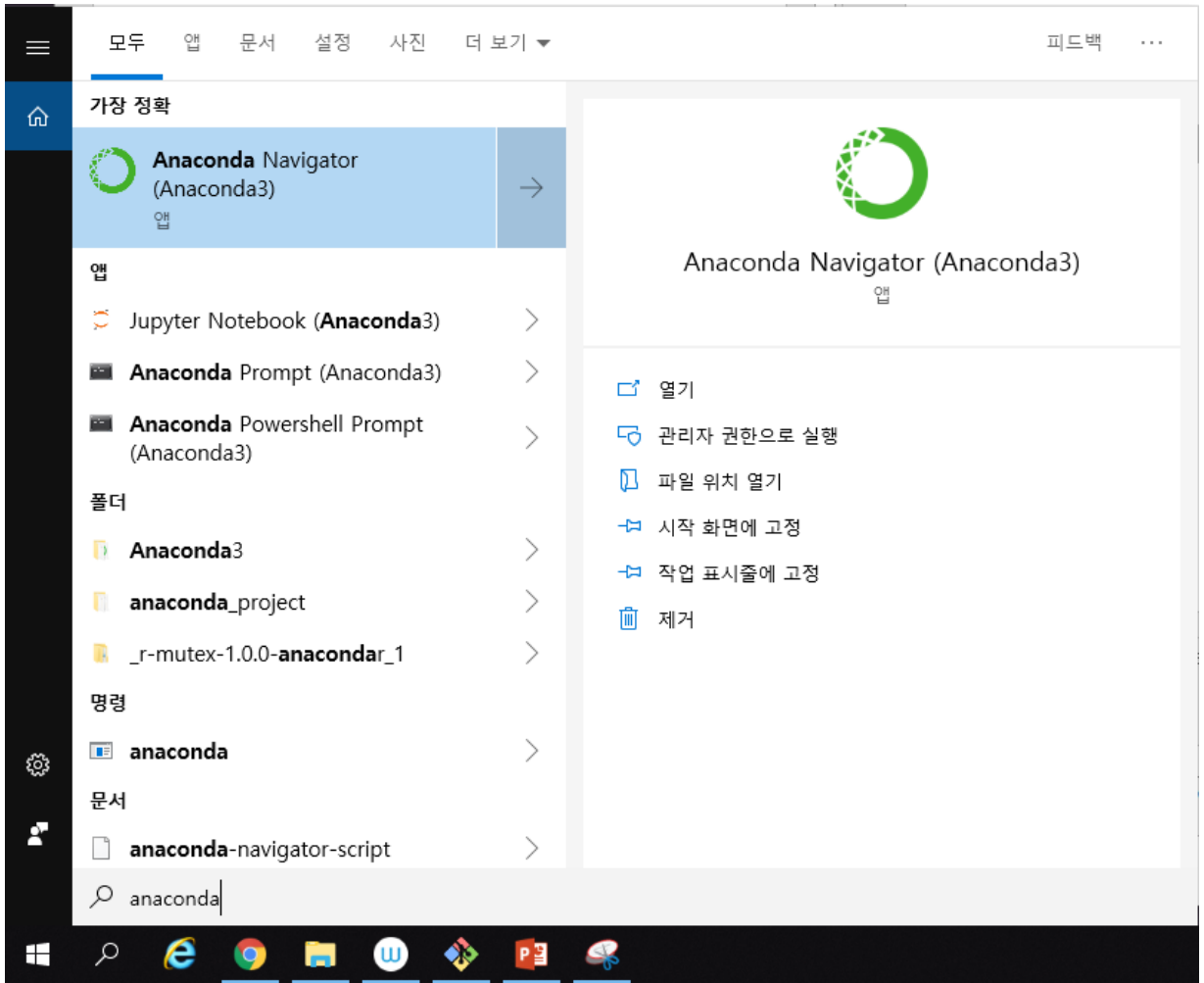
- **World's Most Popular Python/R Data Science Platform**
- **Downloadable:** https://www.anaconda.com/distribution/



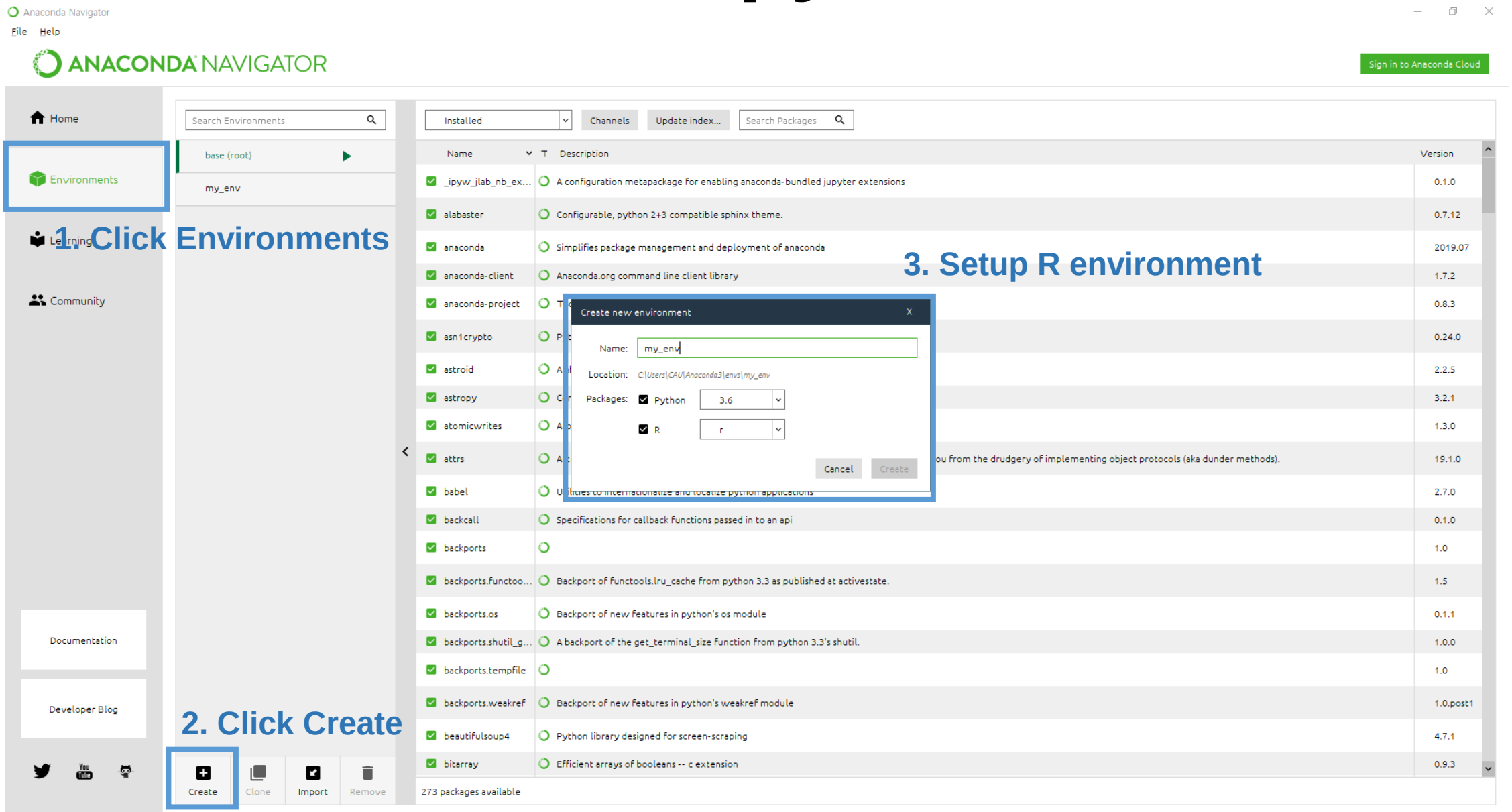
https://wonderbout.tistory.com/22

Run R from Jupyter Notebook

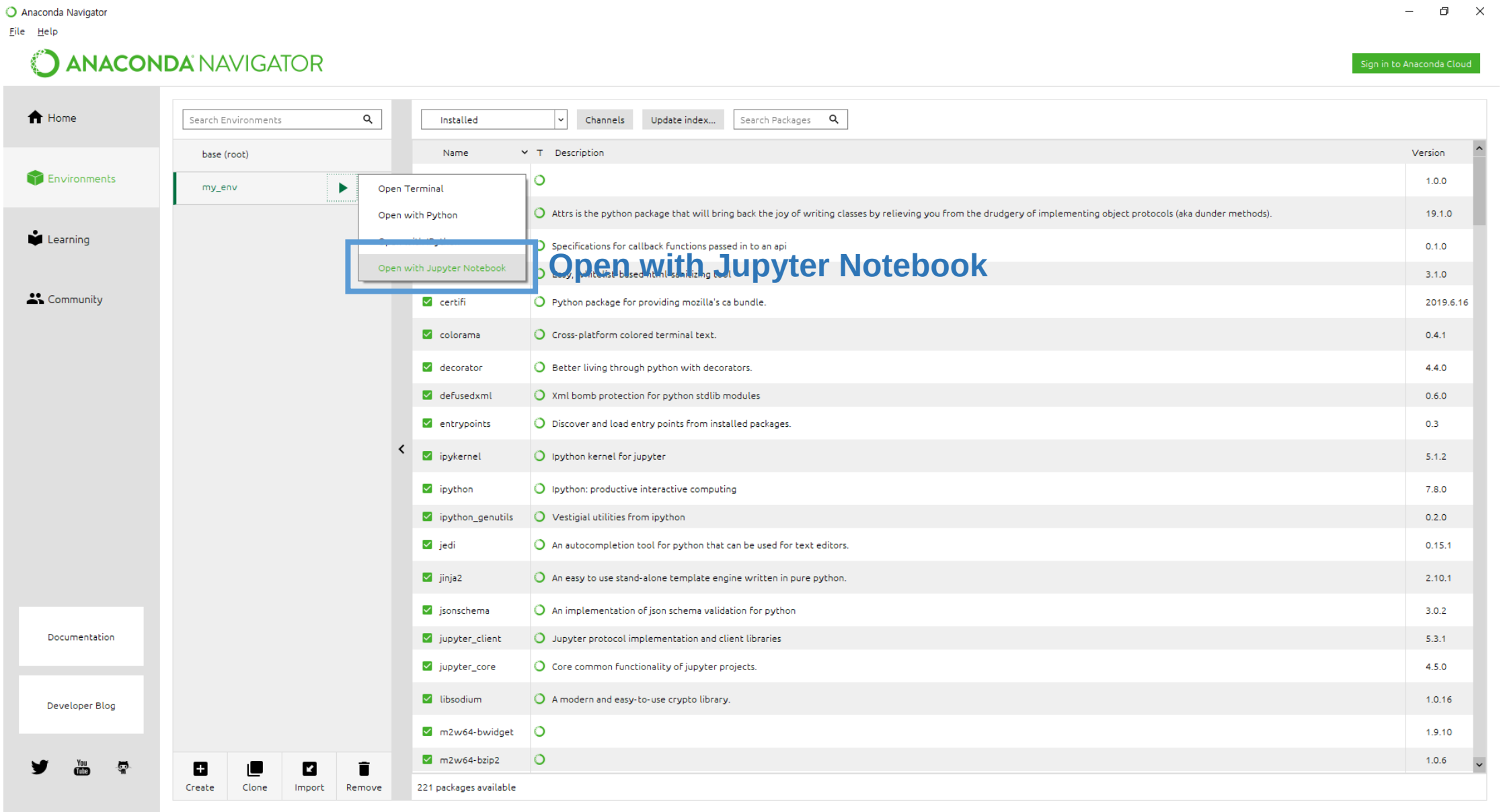
- **Run Anaconda Navigator**



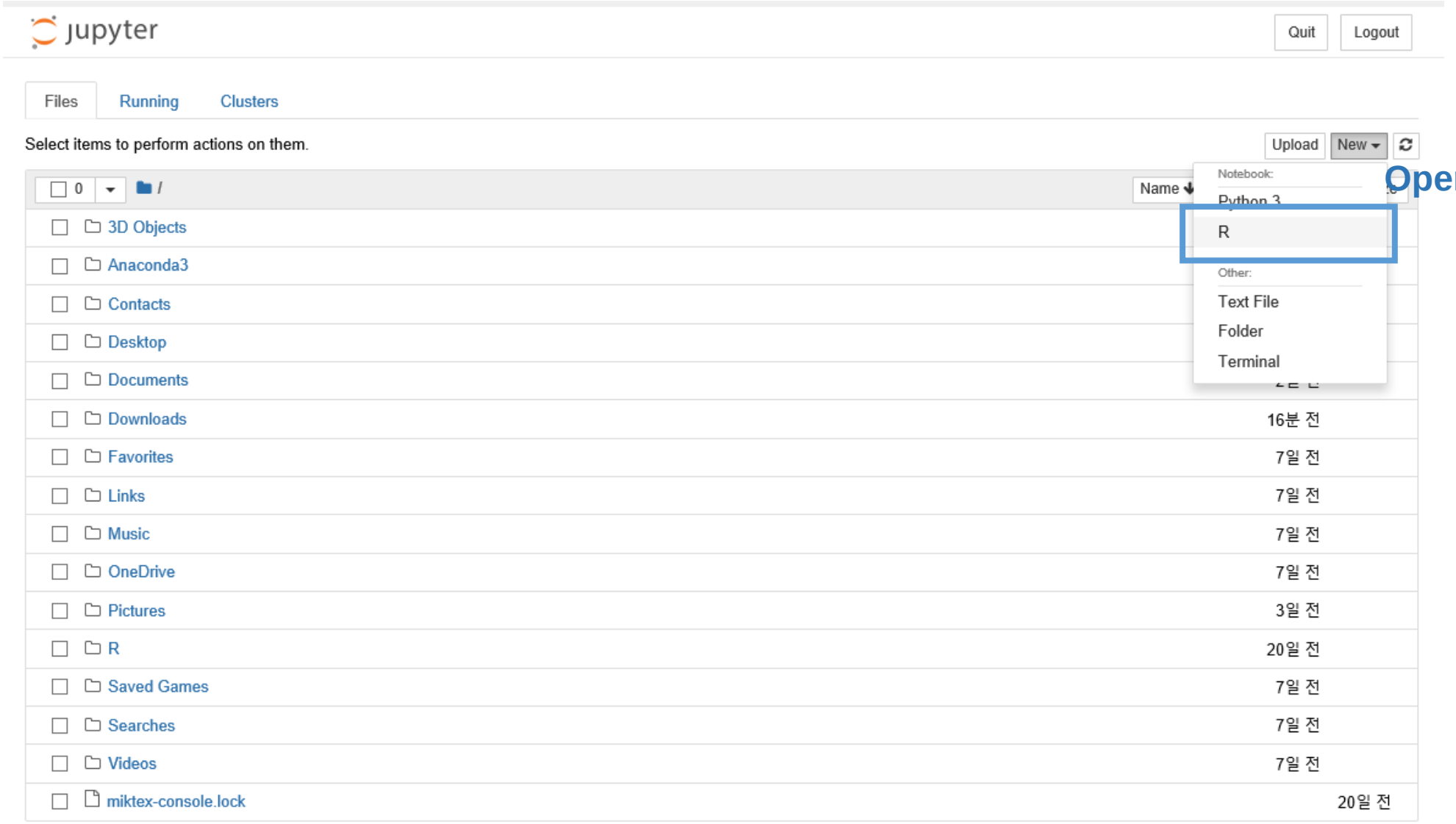
Run R from Jupyter Notebook



Run R from Jupyter Notebook



Run R from Jupyter Notebook



Open with R

What is ggplot ?

- **is a package for data visualization**
- **Use grammar of graphics**

Alternatives: R base functions

Aesthetic mapping

- **ggplot(aes(x = , y= , color= , size= , alpha= , shape=))**

Just try it

https://r4ds.had.co.nz/data-visualisation.html#aesthetic-mappings

Facets

- **Split your plot into facets (facet_wrap, facet_grid)**

Just try it

https://r4ds.had.co.nz/data-visualisation.html#facets

Geometric objects

- **geom stand for geometrical objects**

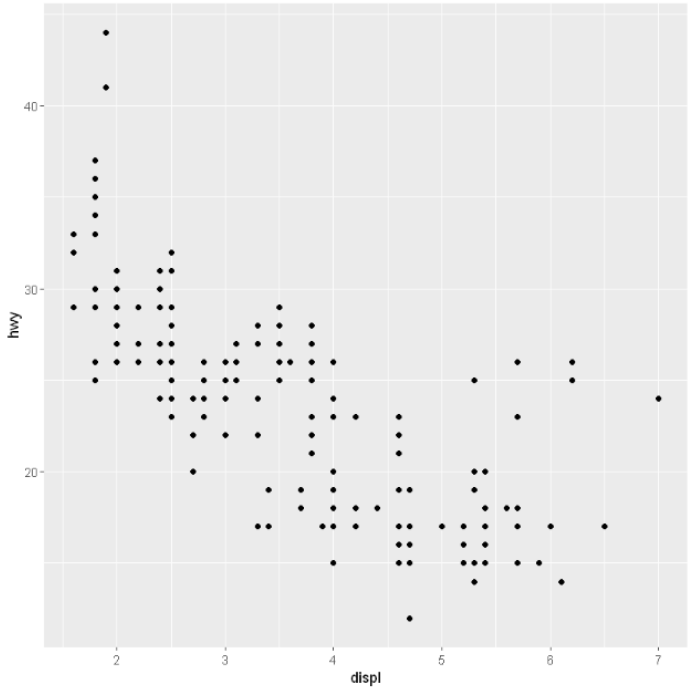
- **geom_point, geom_smooth, geom_bar, geom_violin, geom_abline, etc**

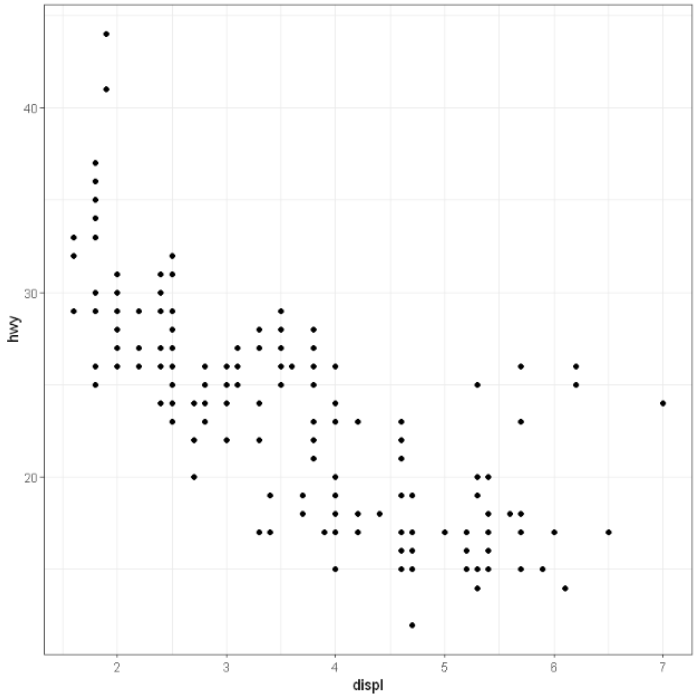
Just try it

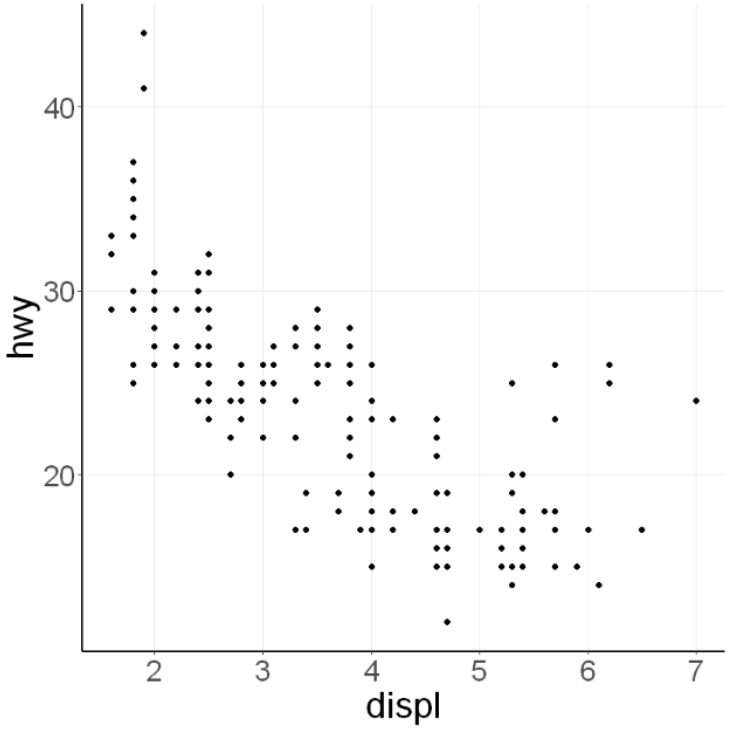
https://r4ds.had.co.nz/data-visualisation.html#geometric-objects

Theme

- What will you choose?





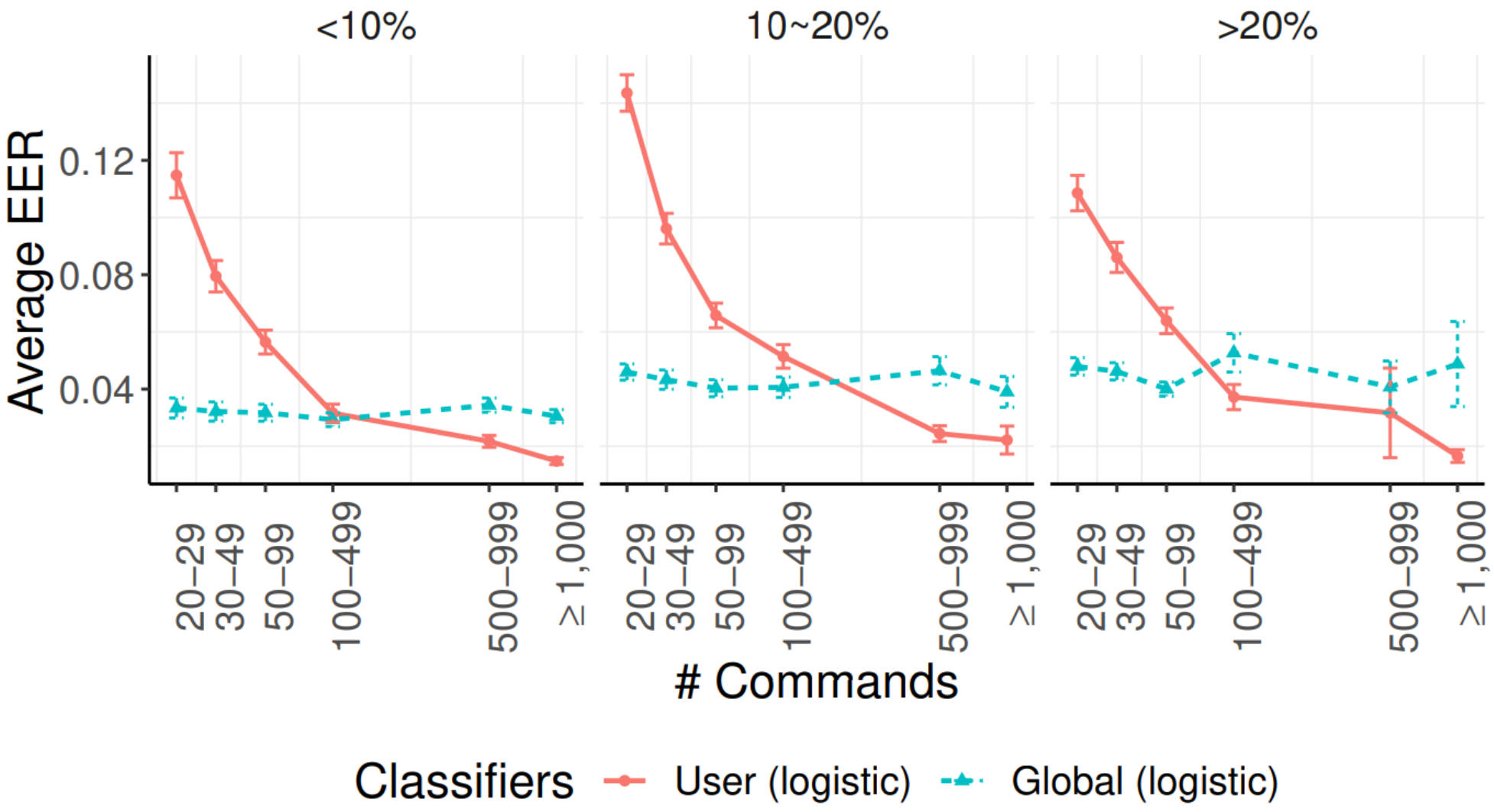


Theme

- **Theme that I like**

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) + theme_bw() +
   theme(axis.line = element_line(size = .8, colour = "black"),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         text = element_text(size = 25)
         )
```

Example


<10%
10~20%
>20%
Average EER
0.12
0.08
0.04
20–29
30–49
50–99
100–499
500–999
≥ 1,000
Commands
Classifiers
User (logistic)
Global (logistic)

Use postscript() or tiff() for quality

```
postscript(file = "test.eps", width=7, height=7)
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) + theme_bw() +
   theme(axis.line = element_line(size = .8, colour = "black"),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         text = element_text(size = 25)
         )
dev.off()
```

ggplot Practice

What is dplyr ?

- is a package for data manipulation
- Functions are coded in C++
- Fast and efficient

Alternatives: data.table package, R base functions

filter()

- **Return a subset of the rows**

Just try it

https://r4ds.had.co.nz/transform.html#filter-rows-with-filter

arrange()

- **Reorders the rows according to single or multiple variables**

Just try it

https://r4ds.had.co.nz/transform.html#arrange-rows-with-arrange

select()

- **Return a subset of the columns**

Just try it

https://r4ds.had.co.nz/transform.html#select

mutate()

- **Add columns from existing data**

Just try it

https://r4ds.had.co.nz/transform.html#add-new-variables-with-mutate

summarize()

- **Produce summary statistic for each group (using group_by())**

Just try it

https://r4ds.had.co.nz/transform.html#grouped-summaries-with-summarise

Improve readability using pipe operators

`x %>% f(y)` turns into `f(x, y)`, and `x %>% f(y) %>% g(z)` turns into `g(f(x, y)`

```r
by_dest <- group_by(flights, dest)
delay <- summarise(by_dest,
  count = n(),
  dist = mean(distance, na.rm = TRUE),
  delay = mean(arr_delay, na.rm = TRUE)
)
delay <- filter(delay, count > 20, dest != "HNL")
```

```r
delays <- flights %>%
  group_by(dest) %>%
  summarise(
    count = n(),
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  ) %>%
  filter(count > 20, dest != "HNL")
```

Improve readability using pipe operators

```
bop(
  scoop(
    hop(foo_foo, through = forest),
    up = field_mice
  ),
  on = head
)
```

```
foo_foo %>%
  hop(through = forest) %>%
  scoop(up = field_mice) %>%
  bop(on = head)
```

dplyr Practice

https://r4ds.had.co.nz/transform.html

What is tidyr ?

- **Represent the same data multiple ways**

```
table1
#> # A tibble: 6 x 4
#>   country      year  cases population
#>   <chr>       <int>  <int>      <int>
#> 1 Afghanistan  1999    745   19987071
#> 2 Afghanistan  2000   2666   20595360
#> 3 Brazil       1999  37737  172006362
#> 4 Brazil       2000  80488  174504898
#> 5 China        1999 212258 1272915272
#> 6 China        2000 213766 1280428583
```

```
table2
#> # A tibble: 12 x 4
#>   country      year type           count
#>   <chr>       <int> <chr>          <int>
#> 1 Afghanistan  1999 cases            745
#> 2 Afghanistan  1999 population   19987071
#> 3 Afghanistan  2000 cases           2666
#> 4 Afghanistan  2000 population   20595360
#> 5 Brazil       1999 cases           37737
#> 6 Brazil       1999 population  172006362
#> # … with 6 more rows
```

```
table3
#> # A tibble: 6 x 3
#>   country      year rate
#> * <chr>       <int> <chr>
#> 1 Afghanistan  1999 745/19987071
#> 2 Afghanistan  2000 2666/20595360
#> 3 Brazil       1999 37737/172006362
#> 4 Brazil       2000 80488/174504898
#> 5 China        1999 212258/1272915272
#> 6 China        2000 213766/1280428583
```

```
table4a  # cases
#> # A tibble: 3 x 3
#>   country     `1999` `2000`
#> * <chr>        <int>  <int>
#> 1 Afghanistan    745   2666
#> 2 Brazil       37737  80488
#> 3 China       212258 213766
```

```
table4b  # population
#> # A tibble: 3 x 3
#>   country         `1999`     `2000`
#> * <chr>            <int>      <int>
#> 1 Afghanistan   19987071   20595360
#> 2 Brazil       172006362  174504898
#> 3 China       1272915272 1280428583
```

gather()

- gather those columns into a new pair of variables

Just try it

https://r4ds.had.co.nz/tidy-data.html#gathering

spread()

- opposite of gathering

Just try it

https://r4ds.had.co.nz/tidy-data.html#spreading

Thank you!
Q & A