การวิเคราะห์ข้อมูลเบื้องต้น

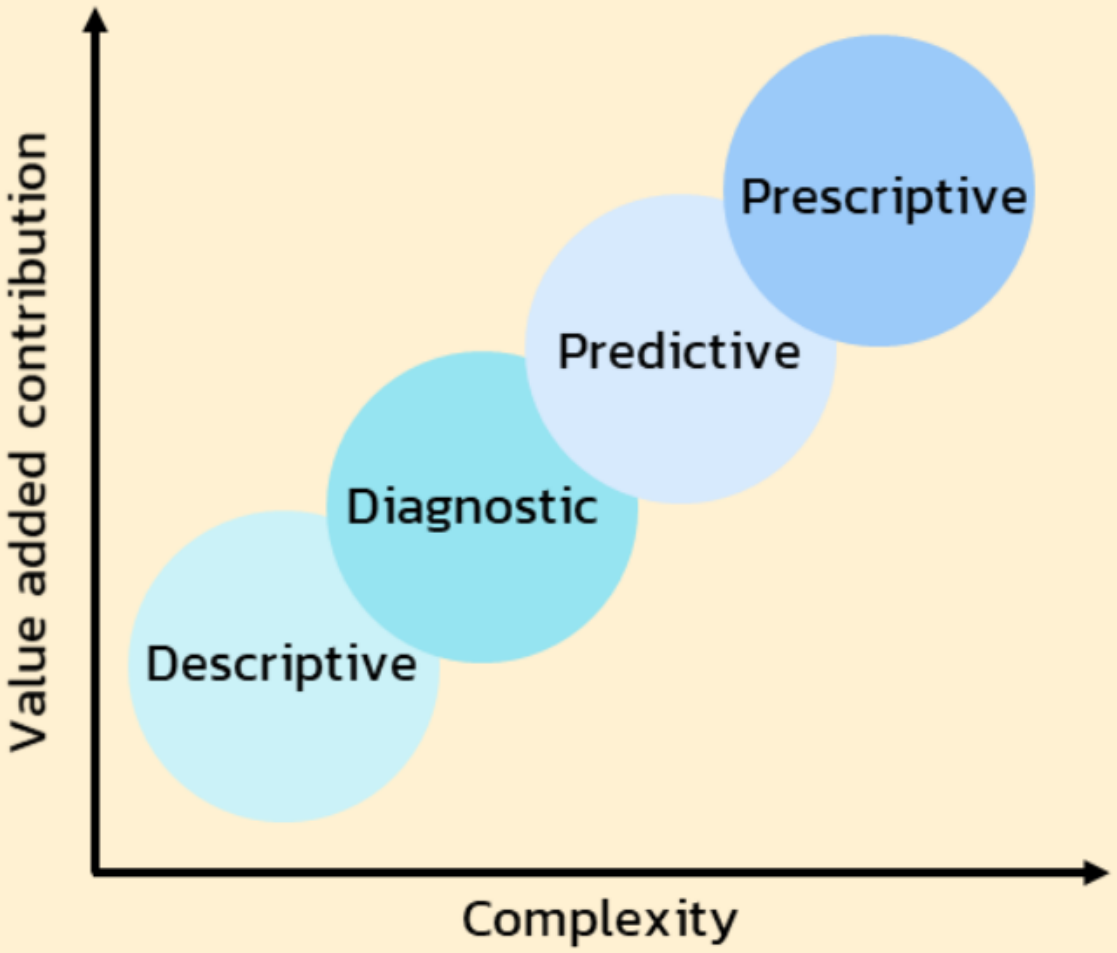
3 ก.ย. 64

เนื้อหา

1. เกริ่นนำ
2. โปรแกรม R
3. การจัดการข้อมูล
4. การทำกราฟ

1 เกริ่นนำ

รูปแบบของการวิเคราะห์ข้อมูล



“Data Analytics”
เป็นศาสตร์ของการวิเคราะห์ข้อมูลต่างๆ จากข้อมูลขนาดใหญ่ (Big Data) เพื่อช่วยในด้านธุรกิจหรือตามวัตถุประสงค์อื่นๆ ที่ต้องการ ซึ่งสามารถแบ่งรูปแบบของการวิเคราะห์ข้อมูลได้ 4 รูปแบบ ดังภาพ

ที่มา: Data Science Academy Thailand

รูปแบบของการวิเคราะห์ข้อมูล

1 การวิเคราะห์ข้อมูลแบบพื้นฐาน (Descriptive Analytics)

DATA SCIENCE ACADEMY THAILAND

"What happened?"

เป็นการวิเคราะห์ข้อมูลแบบพื้นฐานที่สุด ซึ่งเน้นการอธิบายถึง**สาเหตุ**ของการเกิดเหตุการณ์ต่างๆ ที่ได้เกิดขึ้น หรืออาจกำลังเกิดขึ้น

เช่น รายงานการขาย รายงานผลการดำเนินงาน

ที่มา: Data Science Academy Thailand

รูปแบบของการวิเคราะห์ข้อมูล

2 การวิเคราะห์แบบเชิงวินิจฉัย (Diagnostic Analytics)

DATA SCIENCE ACADEMY THAILAND

"Why something happened?"

เป็นการวิเคราะห์ที่อธิบายถึงสาเหตุของสิ่งที่เกิดขึ้น ปัจจัยต่างๆ และ**ความสัมพันธ์**ของปัจจัยหรือตัวแปรต่างๆที่มีความสัมพันธ์ต่อกัน

เช่น ความสัมพันธ์ระหว่างยอดขายต่อกิจกรรมทางการตลาดแต่ละประเภท ซึ่งการวิเคราะห์รูปแบบนี้อาจช่วยส่งเสริมให้การตัดสินใจเป็นไปในทางที่ถูกต้องมากขึ้น

ที่มา: Data Science Academy Thailand

รูปแบบของการวิเคราะห์ข้อมูล

3 การวิเคราะห์แบบพยากรณ์ (Predictive Analytics)

DATA SCIENCE ACADEMY THAILAND

"What is likely to happen?"

เป็นการวิเคราะห์เพื่อ**พยากรณ์หรือทำนาย** สิ่งที่กำลังจะเกิดขึ้นหรือน่าจะเกิดขึ้น โดยใช้ข้อมูลในอดีตกับแบบจำลองทางคณิตศาตร์ สถิติ หรือเทคโนโลยีการประดิษฐ์ต่างๆ (Artificial intelligence)

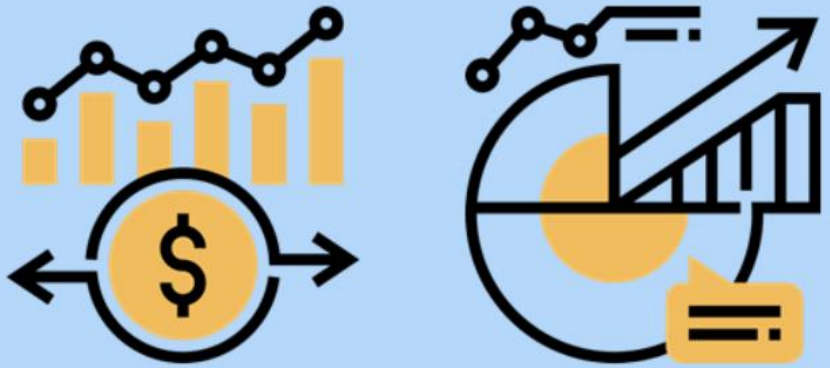
นอกจากนี้การวิเคราะห์แบบพยากรณ์ ยังทำให้เราสามารถวิเคราะห์หาโอกาสและความเสี่ยงต่างๆที่อาจเกิดขึ้นในอนาคตได้ด้วย เช่น การรู้เทรนด์ทางการตลาด การพยากรณ์ยอดขาย การคำนวนความน่าจะเป็นที่บุคคลจะสามารถชำระหนี้ได้ในอนาคต

ที่มา: Data Science Academy Thailand

รูปแบบของการวิเคราะห์ข้อมูล

4 การวิเคราะห์แบบให้คำแนะนำ (Prescriptive Analytics)

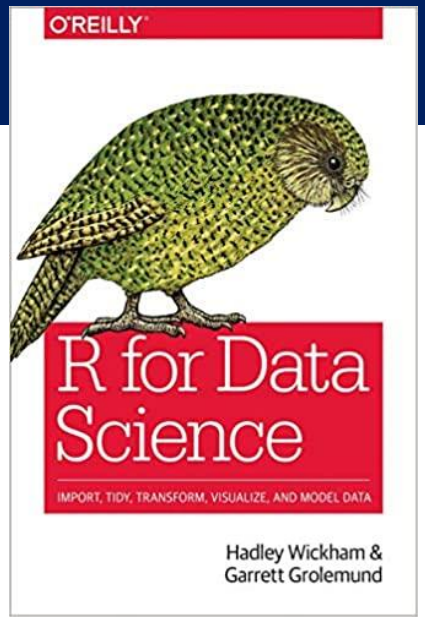
"What action to take?"



เป็นการวิเคราะห์ที่มีความซับซ้อนและยากที่สุด เพราะไม่เพียงพยากรณ์หรือทำนายว่าอะไรจะเกิดขึ้น ข้อดี ข้อเสีย สาเหตุ และระยะเวลาของสิ่งที่จะเกิดขึ้น แต่ยัง**ให้คำแนะนำ**ทางเลือกต่างๆ รวมถึงผลที่จะตามมาของแต่ละทางเลือกด้วย

ที่มา: Data Science Academy Thailand

ขั้นตอนวิเคราะห์ข้อมูล



นำเข้า Import

จัดรูปแบบ Tidy

แปลง Transform

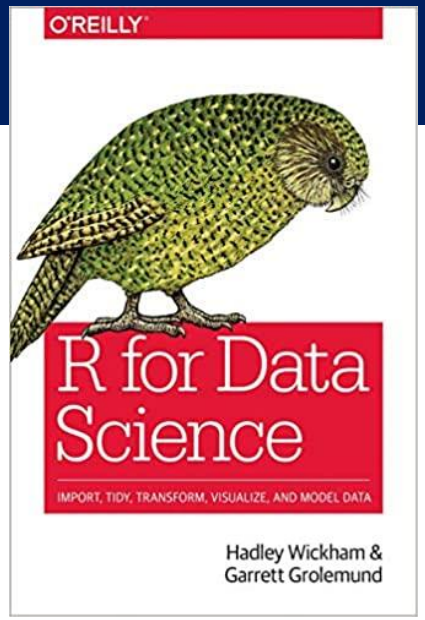
ทำกราฟ Visualize

วิเคราะห์ Model

ทำความเข้าใจ Explore

นำเสนอ Communicate

ขั้นตอนวิเคราะห์ข้อมูล



นำเข้า
Import

จัดรูปแบบ
Tidy

แปลง
Transform

ทำกราฟ
Visualize

วิเคราะห์
Model

ทำความเข้าใจ Explore

นำเสนอ
Communicate

2 โปรแกรม R

โปรแกรม R

- R เป็นโปรแกรม/ภาษาคอมพิวเตอร์ สำหรับการวิเคราะห์ทางสถิติ
- นอกจากนี้ยังนิยมใช้ใน
  - การจัดการข้อมูล (Data Wrangling)
  - แสดงผลข้อมูลด้วยภาพ (Data Visualization)
  - วิทยาการข้อมูล (Data Science)
  - การเรียนรู้ของเครื่อง (Machine Learning)
- R เป็น open-source หมายความว่า เราสามารถใช้ได้ ฟรี
- R ได้รับการพัฒนาอย่างต่อเนื่องจากโปรแกรมเมอร์ทั่วโลก

การใช้โปรแกรม R

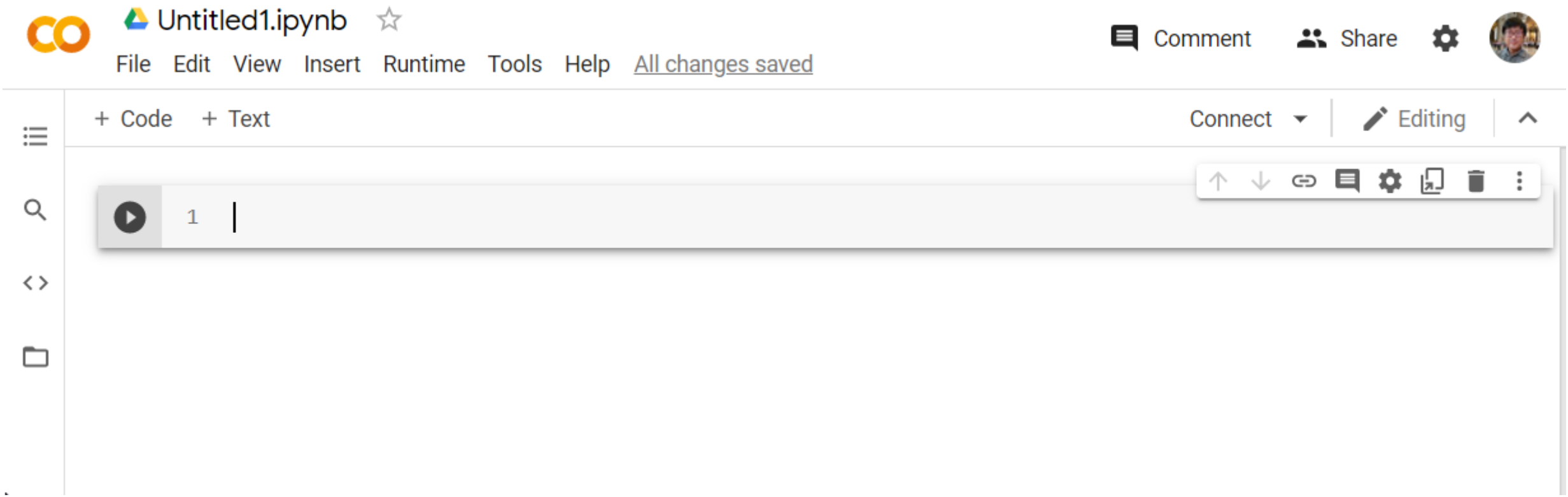
- ใช้ในเครื่องตัวเอง
  - ติดตั้ง R
  - ติดตั้งโปรแกรมอำนวยความสะดวกในการเขียนโปรแกรม (IDE) เช่น Rstudio, Rcommander, Jupyter, VScode เป็นต้น
- ใช้บริการ Cloud
  - Google Colab
  - Kaggle

ใน workshop นี้จะใช้ Google Colab

ดูวิธีการติดตั้ง R และ Rstudio ได้ที่:
https://www.youtube.com/watch?v=UaEtZ5XzVeE&list=PLoTScYm9OOGGSiUGzdWbjxIkZqEO-O6qZ

การใช้โปรแกรม R ด้วย Google Colab

- สร้างบัญชี Gmail และ login เข้าใช้บริการ
- ไปที่ url https://colab.to/r



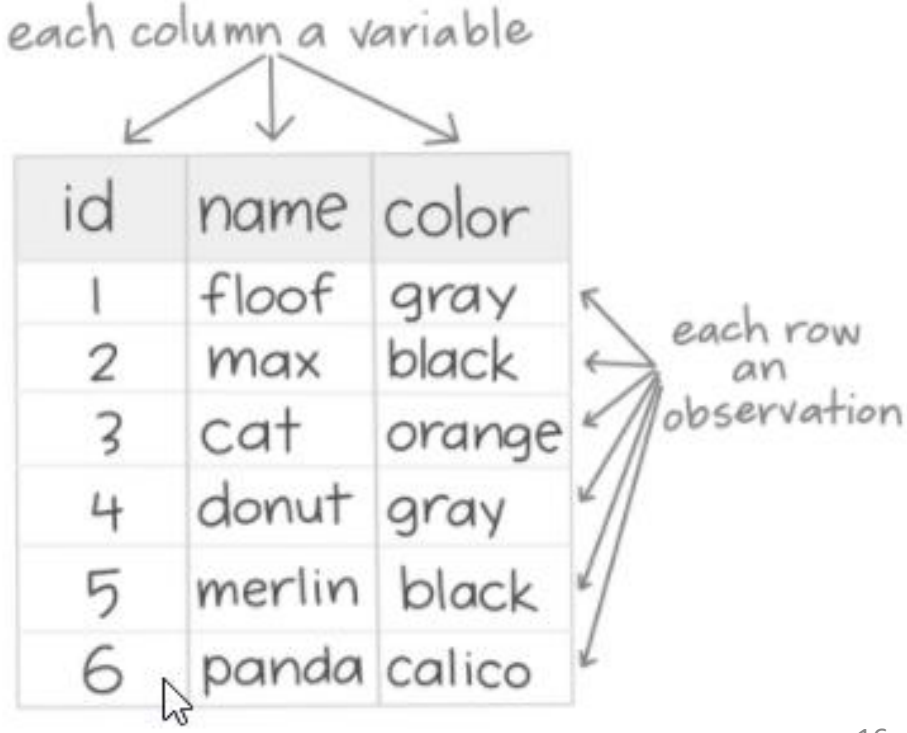
3 การจัดการข้อมูล

อะไรคือ Tidy data

Tidy Data

คือ ข้อมูลที่อยู่ในรูปแบบ/โครงสร้างที่พร้อมใช้ในการวิเคราะห์ข้อมูล

- แต่ละ Column คือ ตัวแปร (variable)
- แต่ละ Row คือ ตัวอย่าง (observation)
- แต่ละ Cell คือ ค่าที่วัดได้



Package gapminder

- gapminder เป็น package ที่มีข้อมูลของ Hans Rosling ใช้ในการนำเสนอ Teds Talk
- มีหลายตาราง แต่จะขอใช้ 2 ตาราง ได้แก่ gapminder และ country_codes

ตาราง gapminder มีตัวแปรดังนี้

1. **country** คือ ชื่อประเทศ
2. **continent** คือ ทวีป
3. **year** คือ ปี ค.ศ.
4. **lifeExp** คือ อายุคาดเฉลี่ย (ปี)
5. **pop** คือ จำนวนประชากร (คน)
6. **gdpPercap** คือ รายได้เฉลี่ยต่อประชากรต่อปี (US dollars)

ติดตั้งและเรียกใช้

```
install.packages("gapminder")
library(gapminder)
```

```
1  gapminder::gapminder
```

A tibble: 1704 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Afghanistan	Asia	1952	28.801	8425333	779.4453
Afghanistan	Asia	1957	30.332	9240934	820.8530
Afghanistan	Asia	1962	31.997	10267083	853.1007
Afghanistan	Asia	1967	34.020	11537966	836.1971
Afghanistan	Asia	1972	36.088	13079460	739.9811
⋮	⋮	⋮	⋮	⋮	⋮
Zimbabwe	Africa	1987	62.351	9216418	706.1573
Zimbabwe	Africa	1992	60.377	10704340	693.4208
Zimbabwe	Africa	1997	46.809	11404948	792.4500
Zimbabwe	Africa	2002	39.989	11926563	672.0386
Zimbabwe	Africa	2007	43.487	12311143	469.7093

ข้อมูล Gapminder

ตาราง country_codes มีตัวแปรดังนี้

1. **country** คือ ชื่อประเทศ
2. **iso_alpha** คือ รหัสประเทศรูปแบบตัวอักษร 3 ตัว ของ ISO
3. **Iso_num** คือ รหัสประเทศรูปแบบตัวเลข

A tibble: 187 × 3

country	iso_alpha	iso_num
<chr>	<chr>	<int>
Afghanistan	AFG	4
Albania	ALB	8
Algeria	DZA	12
Angola	AGO	24
Argentina	ARG	32
⋮	⋮	⋮
Vietnam	VNM	704
West Bank and Gaza	PSE	275
Yemen, Rep.	YEM	887

filter

เป็นการเลือกข้อมูลตามเงื่อนไขที่กำหนด

ตัวอย่าง:

เลือกข้อมูลเฉพาะของประเทศไทยเท่านั้น

Syntax:

```
data %>%
    filter(Logical Expression)
```

```
1  gapminder %>%
2    filter(country == "Thailand")
```

A tibble: 12 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Thailand	Asia	1952	50.848	21289402	757.7974
Thailand	Asia	1957	53.630	25041917	793.5774
Thailand	Asia	1962	56.061	29263397	1002.1992
Thailand	Asia	1967	58.285	34024249	1295.4607
Thailand	Asia	1972	60.405	39276153	1524.3589
⋮	⋮	⋮	⋮	⋮	⋮
Thailand	Asia	1987	66.084	52910342	2982.654
Thailand	Asia	1992	67.298	56667095	4616.897
Thailand	Asia	1997	67.521	60216677	5852.625
Thailand	Asia	2002	68.564	62806748	5913.188
Thailand	Asia	2007	70.616	65068149	7458.396

arrange

เป็นการเรียงลำดับของข้อมูล

ตัวอย่าง:

เรียงลำดับข้อมูลด้วยทวีป

Syntax:

```
data %>%
    arrange(column1, column2, ...)
```

```
1  gapminder %>%
2    arrange(continent)
```

A tibble: 1704 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Algeria	Africa	1952	43.077	9279525	2449.008
Algeria	Africa	1957	45.685	10270856	3013.976
Algeria	Africa	1962	48.303	11000948	2550.817
Algeria	Africa	1967	51.407	12760499	3246.992
Algeria	Africa	1972	54.518	14760787	4182.664
⋮	⋮	⋮	⋮	⋮	⋮
New Zealand	Oceania	1987	74.320	3317166	19007.19
New Zealand	Oceania	1992	76.330	3437674	18363.32
New Zealand	Oceania	1997	77.550	3676187	21050.41
New Zealand	Oceania	2002	79.110	3908037	23189.80
New Zealand	Oceania	2007	80.204	4115771	25185.01

arrange

หากต้องการเรียงจากมากไปน้อย ให้ใส่ desc ที่หน้าตัวแปรที่ต้องการ

ตัวอย่าง:

เรียงลำดับประเทศและปี โดยให้ปีล่าสุดขึ้นก่อน

Syntax:

```
data %>%
    arrange(desc(column1), ...)
```

```
1  gapminder %>%
2    arrange(country, desc(year))
```

A tibble: 1704 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Afghanistan	Asia	2007	43.828	31889923	974.5803
Afghanistan	Asia	2002	42.129	25268405	726.7341
Afghanistan	Asia	1997	41.763	22227415	635.3414
Afghanistan	Asia	1992	41.674	16317921	649.3414
Afghanistan	Asia	1987	40.822	13867957	852.3959
⋮	⋮	⋮	⋮	⋮	⋮
Zimbabwe	Africa	1972	55.635	5861135	799.3622
Zimbabwe	Africa	1967	53.995	4995432	569.7951
Zimbabwe	Africa	1962	52.358	4277736	527.2722
Zimbabwe	Africa	1957	50.469	3646340	518.7643
Zimbabwe	Africa	1952	48.451	3080907	406.8841

select

เป็นการเลือกบางตัวแปร/column

ตัวอย่าง:

เลือกเฉพาะตัวแปรประเทศ ปี และ จำนวนประชากร

Syntax:

```
data %>%
    select(column1, column2, ...)
```

```
1  gapminder %>%
2    select(country, year, pop)
```

A tibble: 1704 × 3

country	year	pop
<fct>	<int>	<int>
Afghanistan	1952	8425333
Afghanistan	1957	9240934
Afghanistan	1962	10267083
Afghanistan	1967	11537966
Afghanistan	1972	13079460
⋮	⋮	⋮
Zimbabwe	1987	9216418
Zimbabwe	1992	10704340
Zimbabwe	1997	11404948
Zimbabwe	2002	11926563
Zimbabwe	2007	12311143

mutate

เป็นการสร้างตัวแปรใหม่

ตัวอย่าง:

การแปลงหน่วยประชากรจาก คน เป็น ล้านคน

Syntax:

```
data %>%
    mutate(column_name = expression)
```

```
gapminder %>%
  mutate(pop_m = pop/1000000) %>%
  filter(country == "Thailand")
```

A tibble: 12 × 7

country	continent	year	lifeExp	pop	gdpPercap	pop_m
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>	<dbl>
Thailand	Asia	1952	50.848	21289402	757.7974	21.28940
Thailand	Asia	1957	53.630	25041917	793.5774	25.04192
Thailand	Asia	1962	56.061	29263397	1002.1992	29.26340
Thailand	Asia	1967	58.285	34024249	1295.4607	34.02425
Thailand	Asia	1972	60.405	39276153	1524.3589	39.27615
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Thailand	Asia	1987	66.084	52910342	2982.654	52.91034
Thailand	Asia	1992	67.298	56667095	4616.897	56.66710
Thailand	Asia	1997	67.521	60216677	5852.625	60.21668
Thailand	Asia	2002	68.564	62806748	5913.188	62.80675
Thailand	Asia	2007	70.616	65068149	7458.396	65.06815

summarize

เป็นการสรุปข้อมูล เช่น ค่าเฉลี่ย ผลรวม เป็นต้น มักจะใช้ในการหาค่าสรุปของกลุ่ม/ประเภทที่สนใจ

ตัวอย่าง:

การหาจำนวนประชากรรวมของแต่ละทวีปในแต่ละปี

Syntax:

```
data %>%
  group_by(column1, ...) %>%
  summarize(column_name = expression)
```

```
1  gapminder %>%
2    group_by(continent, year) %>%
3    summarize(pop = sum(pop))
```

```
`summarise()` has grouped output by 'conti
```

A grouped_df: 60 × 3

continent	year	pop
<fct>	<int>	<dbl>
Africa	1952	237640501
Africa	1957	264837738
Africa	1962	296516865
Africa	1967	335289489
Africa	1972	379879541
⋮	⋮	⋮
Oceania	1987	19574415
Oceania	1992	20919651
Oceania	1997	22241430
Oceania	2002	23454829
Oceania	2007	24549947

joins

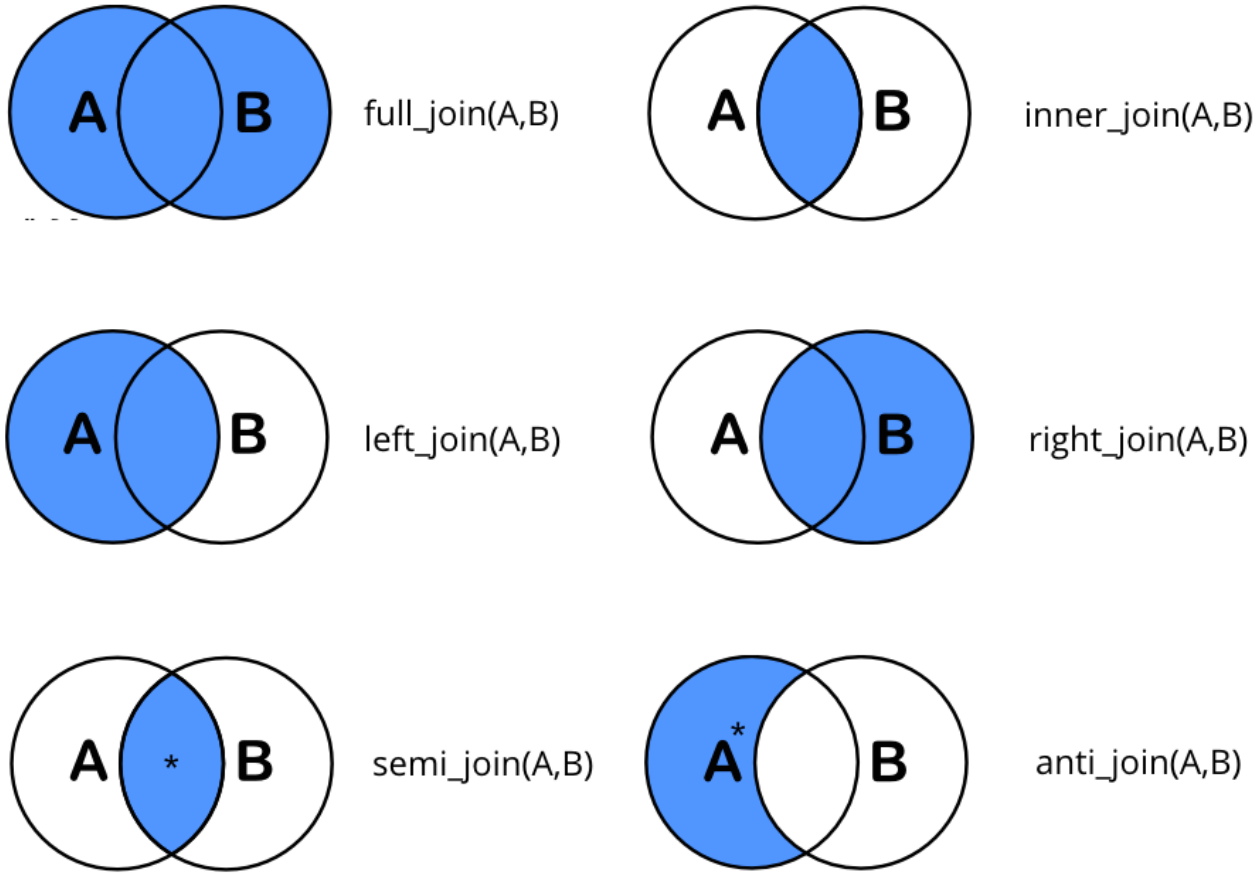
เป็นการเอา 2 ตาราง มาเชื่อมกัน โดยมี column เป็นตัวเชื่อม (key) การเชื่อมมีหลายรูปแบบ เช่น left_join

ตัวอย่าง:

การเชื่อมตาราง gapminder และ country_codes

Syntax:

```
dataA %>%
  join(dataB,
       by = c("columnA" = "columnB"))
```



joins

```
1  gapminder %>%
2    left_join(country_codes, by = c("country" = "country"))
```

A tibble: 1704 × 8

country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
<chr>	<fct>	<int>	<dbl>	<int>	<dbl>	<chr>	<int>
Afghanistan	Asia	1952	28.801	8425333	779.4453	AFG	4
Afghanistan	Asia	1957	30.332	9240934	820.8530	AFG	4
Afghanistan	Asia	1962	31.997	10267083	853.1007	AFG	4
Afghanistan	Asia	1967	34.020	11537966	836.1971	AFG	4
Afghanistan	Asia	1972	36.088	13079460	739.9811	AFG	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Zimbabwe	Africa	1987	62.351	9216418	706.1573	ZWE	716
Zimbabwe	Africa	1992	60.377	10704340	693.4208	ZWE	716
Zimbabwe	Africa	1997	46.809	11404948	792.4500	ZWE	716
Zimbabwe	Africa	2002	39.989	11926563	672.0386	ZWE	716
Zimbabwe	Africa	2007	43.487	12311143	469.7093	ZWE	716

reshape

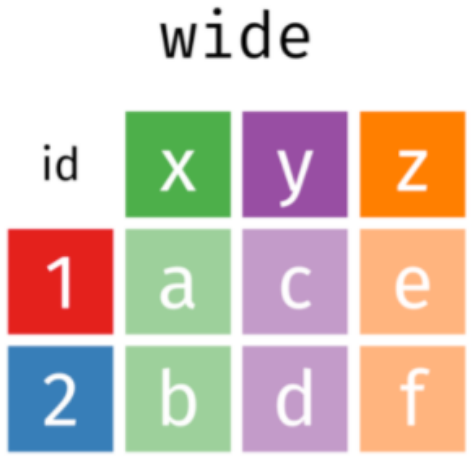
เป็นการจัดรูปแบบของข้อมูลซึ่งมี 2 รูปแบบ คือ

wide format

เป็นข้อมูลที่ตัวแปรอยู่ในแต่ละ column โดย cell ในการบอกค่า

long format

เป็นข้อมูลที่มี column key ในการบอกตัวแปร และมี column value ในการบอกค่า



pivot_wider

แปลงจาก long format เป็น wide format

ตัวอย่าง:

จะทำตารางอายุคาดเฉลี่ยของแต่ละประเทศรายปี โดยให้ปีเป็น column

Syntax:

```
data %>%
  pivot_wider(
    id_cols = c("column", ...),
    names_from = "column",
    values_from = "column"
)
```

```
gapminder %>%
  pivot_wider(
    id_cols = "country",
    names_from = "year",
    values_from = "lifeExp"
  )
```

A tibble: 142 × 13

country	1952	1957	1962	1967	1972	1977	1982
<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Afghanistan	28.801	30.332	31.997	34.020	36.088	38.438	39.854
Albania	55.230	59.280	64.820	66.220	67.690	68.930	70.420
Algeria	43.077	45.685	48.303	51.407	54.518	58.014	61.368
Angola	30.015	31.999	34.000	35.985	37.928	39.483	39.942
Argentina	62.485	64.399	65.142	65.634	67.065	68.481	69.942
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Vietnam	40.412	42.887	45.363	47.838	50.254	55.764	58.816
West Bank and Gaza	43.160	45.671	48.127	51.631	56.532	60.765	64.406
Yemen, Rep.	32.548	33.970	35.180	36.984	39.848	44.175	49.113
Zambia	42.038	44.077	46.023	47.768	50.107	51.386	51.821
Zimbabwe	48.451	50.469	52.358	53.995	55.635	57.674	60.363

pivot_longer

แปลงจาก wide format เป็น long format

ตัวอย่าง:

จะแปลงข้อมูล lifeExp, pop, gdpPercap เป็น long format

Syntax:

```
data %>%
  pivot_longer(
    cols = c(“column”, ...),
    names_to = “variable”,
    values_to = “values”
)
```

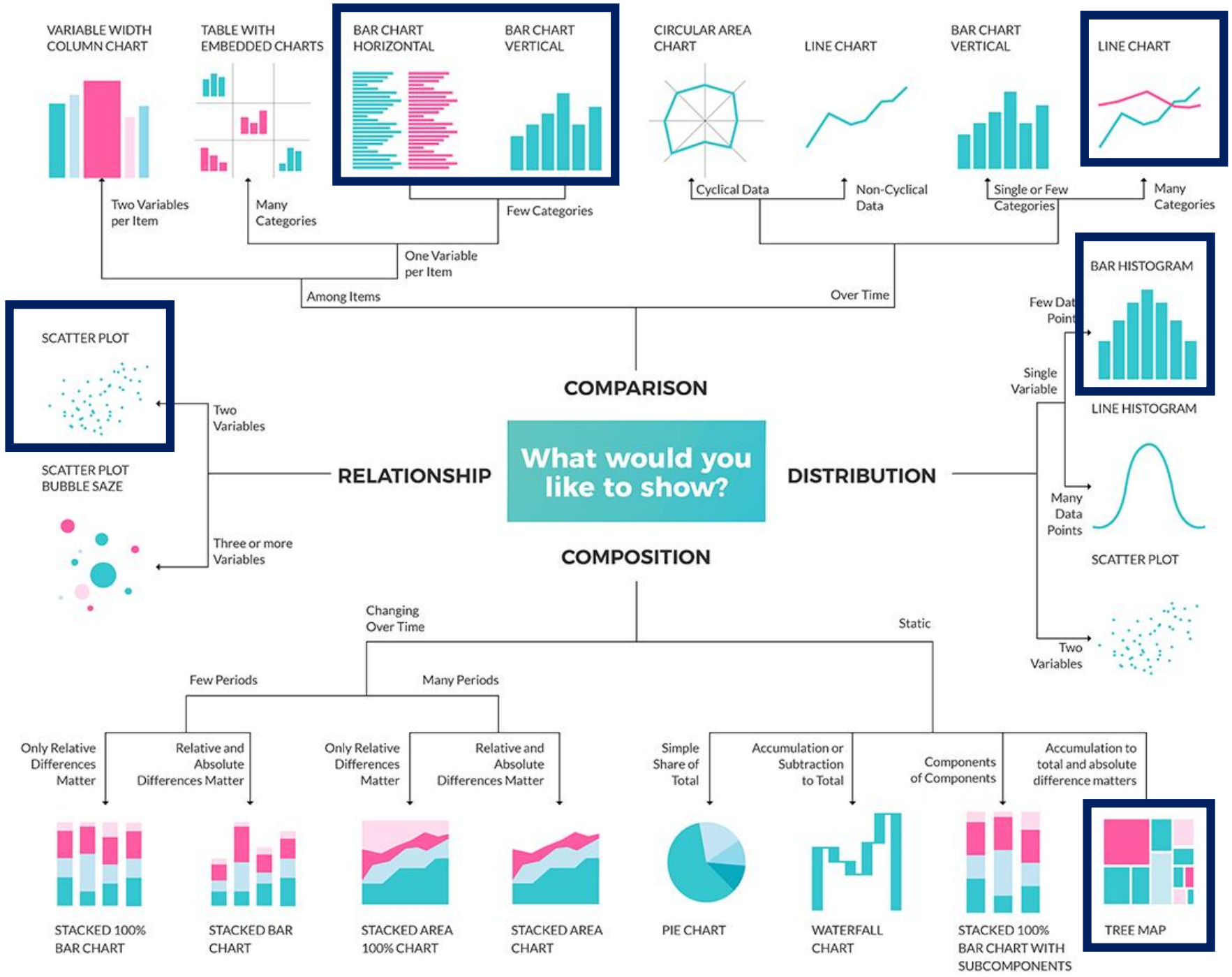
```
gapminder %>%
  pivot_longer(
    cols = c("lifeExp", "pop", "gdpPercap"),
    names_to = "variable",
    values_to = "values"
  )
```

A tibble: 5112 × 5

country	continent	year	variable	values
<fct>	<fct>	<int>	<chr>	<dbl>
Afghanistan	Asia	1952	lifeExp	28.8010
Afghanistan	Asia	1952	pop	8425333.0000
Afghanistan	Asia	1952	gdpPercap	779.4453
Afghanistan	Asia	1957	lifeExp	30.3320
Afghanistan	Asia	1957	pop	9240934.0000
⋮	⋮	⋮	⋮	⋮
Zimbabwe	Africa	2002	pop	1.192656e+07
Zimbabwe	Africa	2002	gdpPercap	6.720386e+02
Zimbabwe	Africa	2007	lifeExp	4.348700e+01
Zimbabwe	Africa	2007	pop	1.231114e+07
Zimbabwe	Africa	2007	gdpPercap	4.697093e+02

4 การทำกราฟ

การเลือกกราฟ



1. เปรียบเทียบ
2. การกระจาย
3. องค์ประกอบ
4. ความสัมพันธ์

ที่มา:
https://education.microsoft.com/en-us/course/0a60eeb6/1

ggplot2

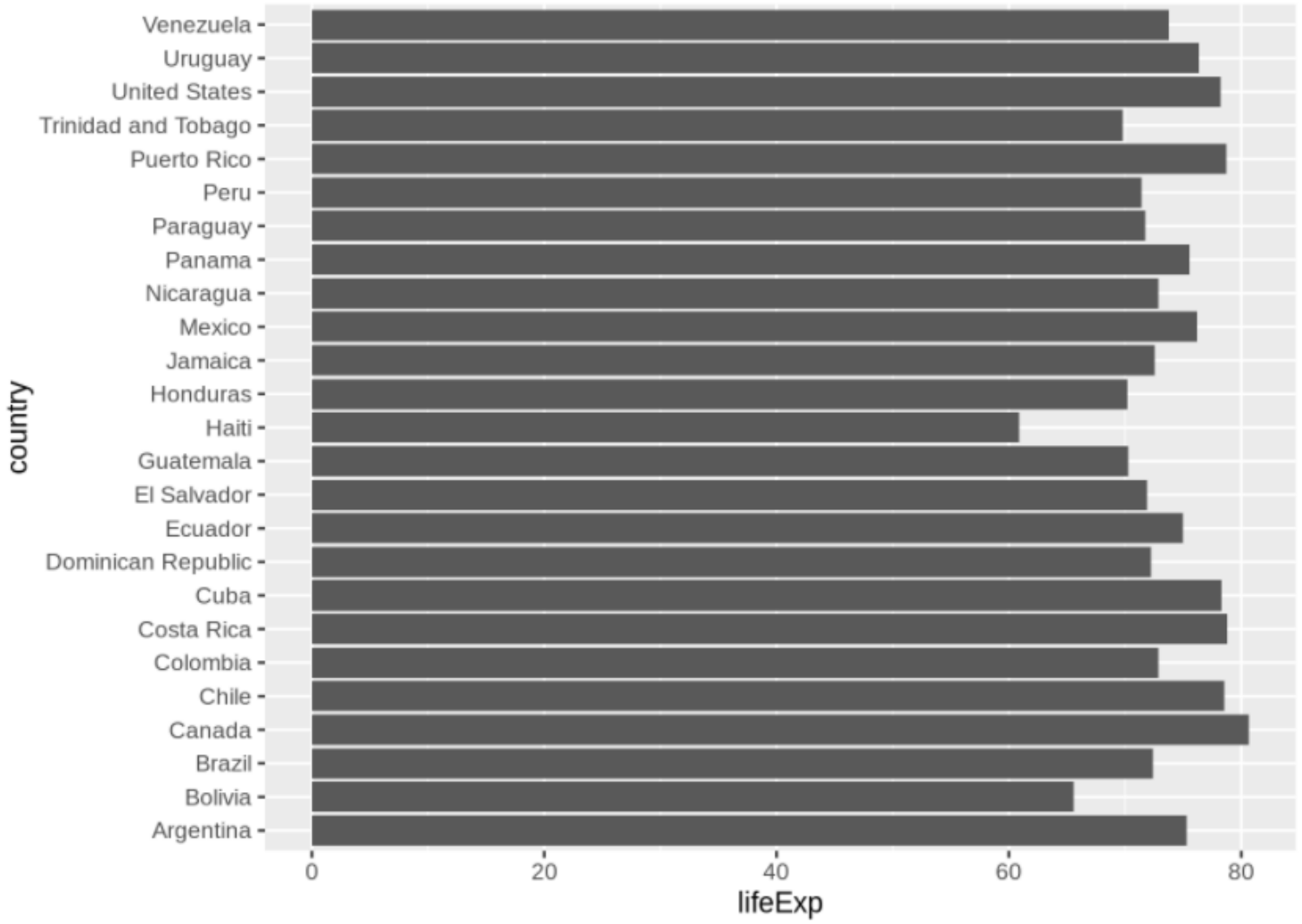
- ggplot2 เป็น แพกเกจ สำหรับทำกราฟที่เป็นที่นิยม
- ใช้ concept ที่เรียกว่า The Grammar of Graphics

ggplot2 เบื้องต้น

1. การสร้างกราฟ คือ การเชื่อมโยง ตัวแปร (ข้อมูล) กับการแสดงผล **(Aesthetic mapping)**
2. กำหนดรูปแบบการแสดงผล **geoms** เช่น กราฟเส้น กราฟแท่ง
3. สร้างกราฟเป็นชั้น **(layers)** ซ้อนไปเรื่อยๆ

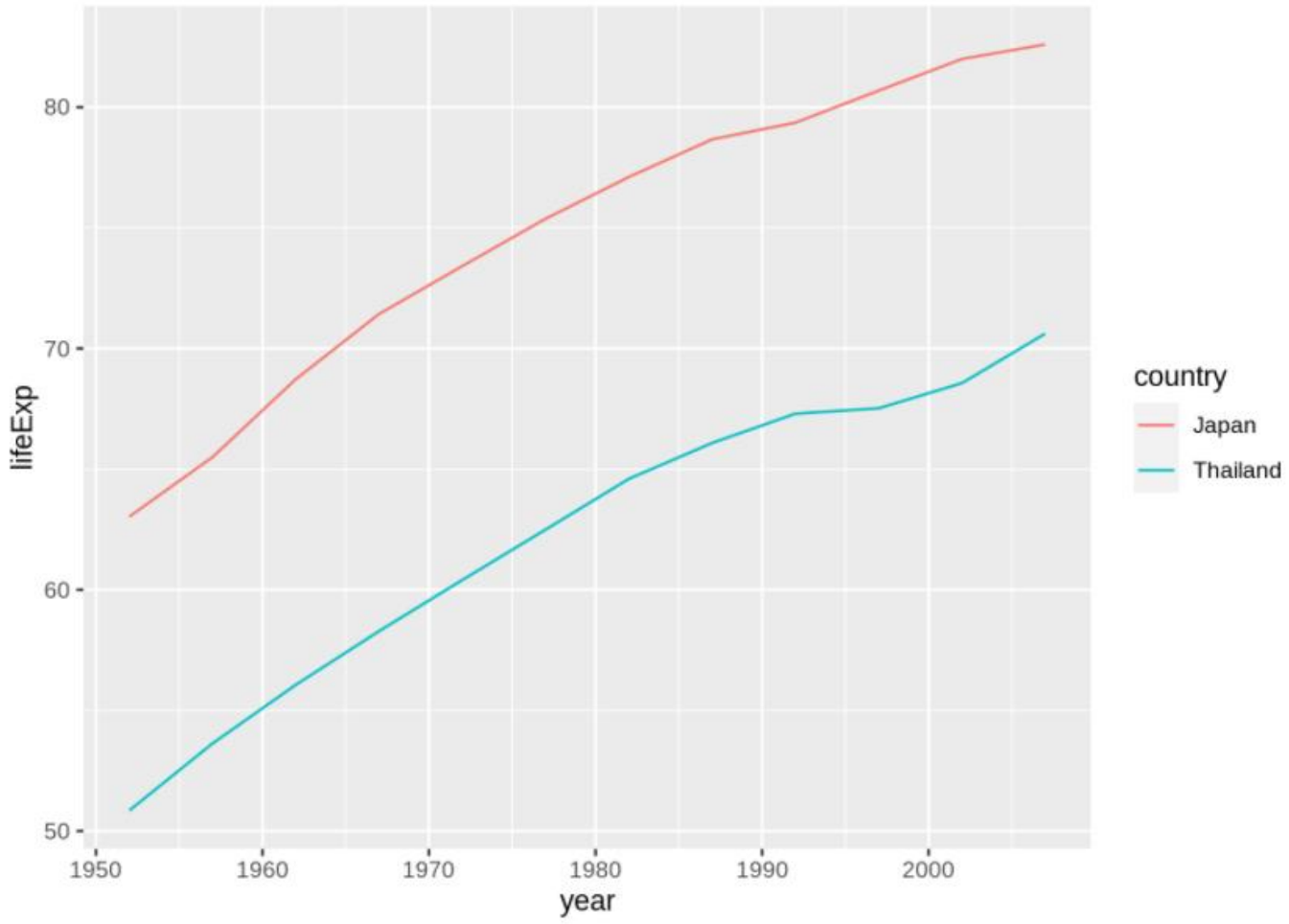
Bar Chart

```
ggplot(
  data = gapminder %>%
    filter(
      year == 2007,
      continent == "Americas"),
  aes(
    x = lifeExp,
    y = country)) +
  geom_col()
```



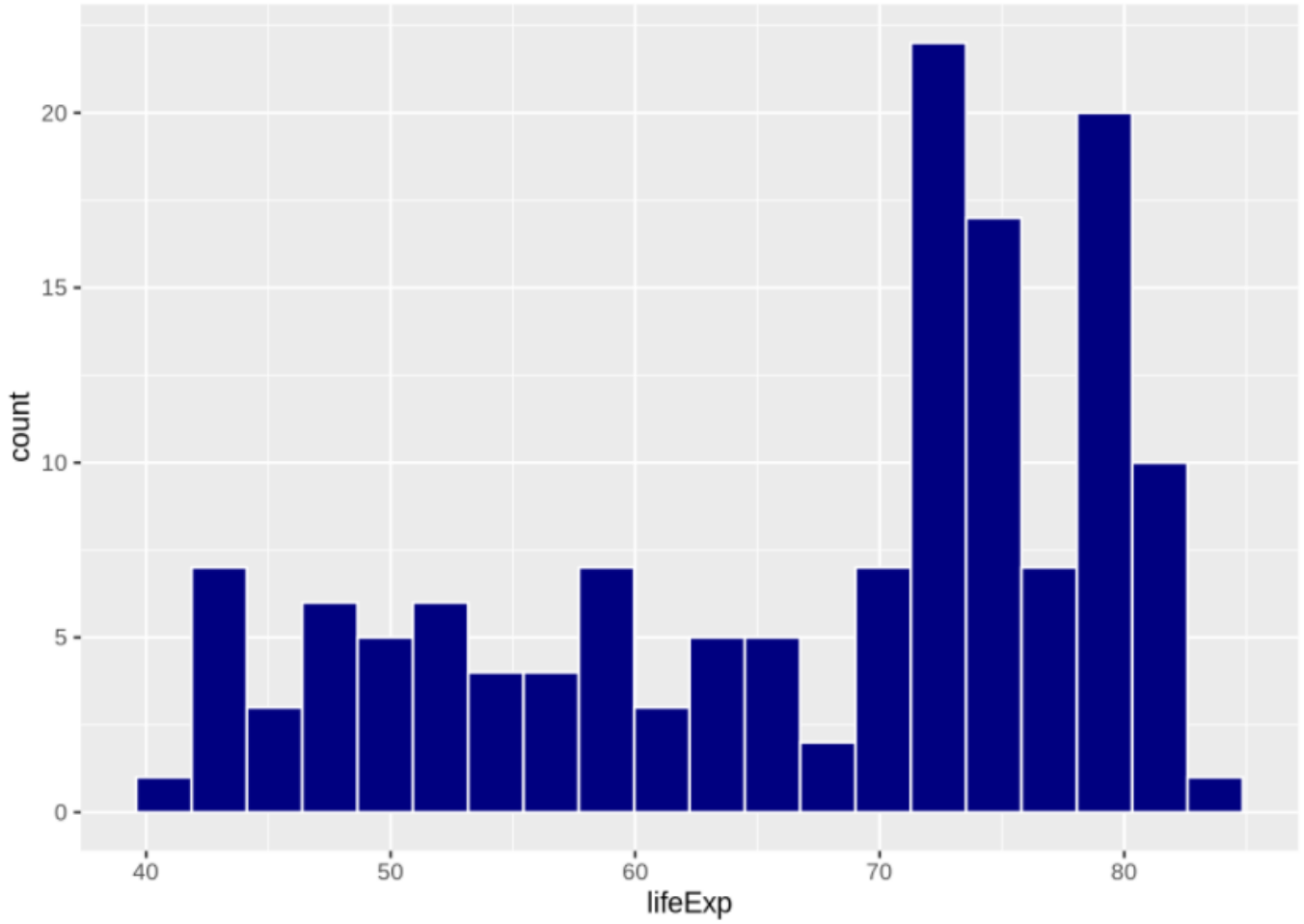
Line Chart

```r
ggplot(
  data = gapminder %>%
    filter(country %in% c("Thailand", "Japan")),
  aes(
    x = year,
    y = lifeExp,
    color = country)) +
  geom_line()
```



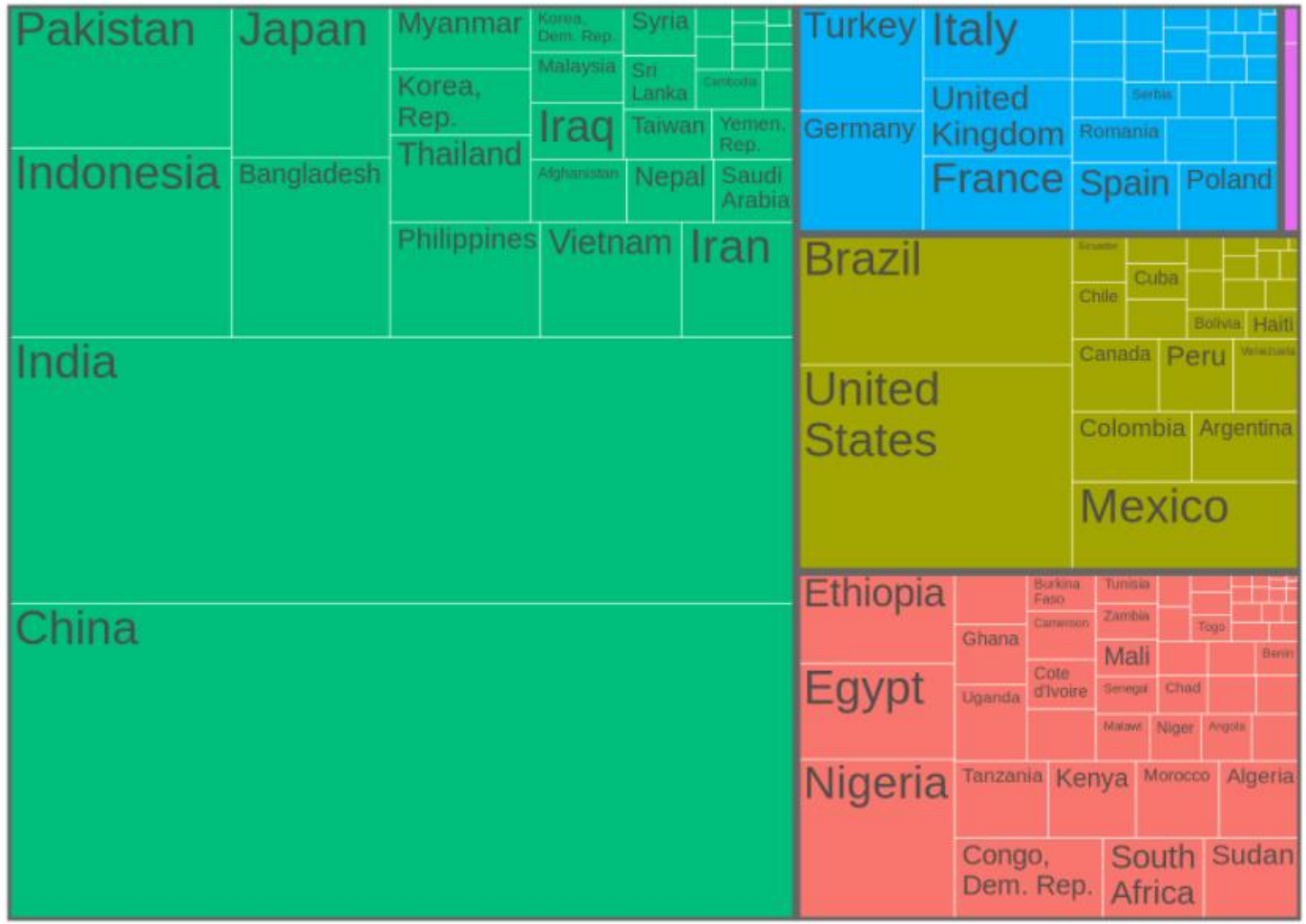
Histogram

```
1  ggplot(
2    data = gapminder %>% filter(year == 2007),
3    aes(x = lifeExp)) +
4    geom_histogram(
5      bins = 20,
6      fill = "navyblue",
7      color = "white")
```



Treemap

```
library(treemapify)
ggplot(gapminder %>% filter(year == "2007"),
       aes(area = pop,
           fill = continent,
           subgroup = continent,
           label = country
          )) +
  geom_treemap(color = "white") +
  geom_treemap_subgroup_border(color = "grey40") +
  geom_treemap_text(
    colour = "grey30",
    place = "topleft",
    reflow = T) +
  theme(legend.position = "none")
```



Scatter Plot

```
1  ggplot(data = gapminder,
2         aes(x = gdpPercap,
3             y = lifeExp,
4             size = pop,
5             col = continent)) +
6    geom_point(alpha = 0.3)
```