R, R studio Analyzing Data — ggplot_knitr

The dplyr package contains functions to query and manipulate data frames.

```
1.select()
2.filter()
3.mutate()
4.group_by()
5.summarize()
```

Schedule:
talk : 9:50
exercise : 9:50 - 10:10
break : 10:10 - 10:20

data frame

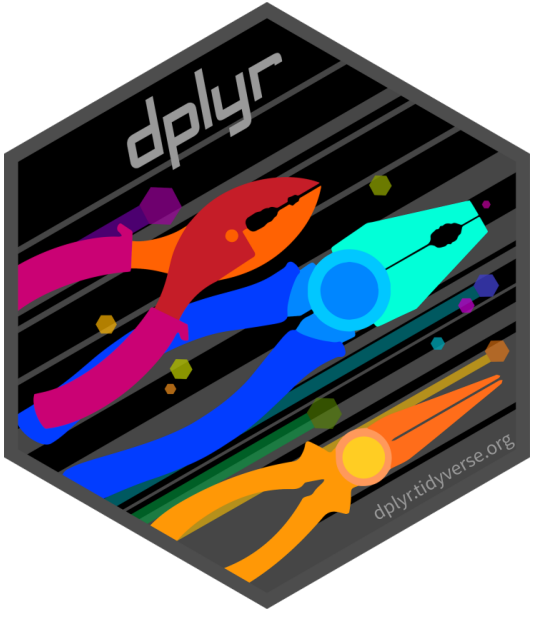
gapminder

Country	population
US	...
Afghan	...

function

dataframe

data frame in, data frame out



`select()` selects columns.

subset of gapminder

d

country	pop	continent	lifeExp
Canada	33390141	Americas	80.653
France	61083916	Europe	80.657
Mexico	108700891	Americas	76.195
United Kingdom	60776238	Europe	79.425
United States	301139947	Americas	78.242

continent	country
Americas	Canada
Europe	France
⋮	⋮

```
d %>% select(continent, country)
```

code using dplyr

pipe function columns we're selecting

123 GO – select just the "pop" column

```
d %>% select(pop)
```

`filter()` filters rows.

find rows where the continent is Europe

== "Europe"

country	pop	continent	lifeExp
Canada	33390141	Americas	80.653
France	61083916	Europe	80.657
Mexico	108700891	Americas	76.195
United Kingdom	60776238	Europe	79.425
United States	301139947	Americas	78.242

FALSE
TRUE

5 rows

```
d %>% filter(continent == "Europe")
```

d %>% filter(80 <= lifeExp)

Column

2 rows

country	pop	continent	lifeExp
[redacted]			
France	61083916	Europe	80.657
[redacted]			
United Kingdom	60776238	Europe	79.425
[redacted]			

123 GO – filter to countries with lifeExp at least 80

`mutate()` adds new columns.

d

country	pop	continent	lifeExp	pop_millions
Canada	33390141	Americas	80.653	33
France	61083916	Europe	80.657	61
Mexico	108700891	Americas	76.195	109
United Kingdom	60776238	Europe	79.425	61
United States	301139947	Americas	78.242	301

round(lifeExp, digits=2)

? round

name of a new column value

```
d %>% mutate(pop_millions = round(pop / 1e6))
```

1×10^6 1000000

123 GO – add a column called lifeExpInt by rounding lifeExp to the nearest integer.

d %>% mutate(lifeExpInt = round(lifeExp))

`group_by()` splits the data into groups.

```
       country       pop continent lifeExp
        Canada  33390141  Americas  80.653
        France  61083916    Europe  80.657
        Mexico 108700891  Americas  76.195
United Kingdom  60776238    Europe  79.425
 United States 301139947  Americas  78.242
```

Americas country pop conti

Canada
MX
US

Europe

country pop conti

FR
UK

```
d %>% group_by(continent)
```

`summarize()` applies functions to each group.

Americas country pop conti: life Exp

Canada 80.6

MX 76.2

US 78.2

mean is 78.4

d → group_by

Europe country pop conti Life Exp

FR 80.6

UK 79.2

80

Result

```
# A tibble: 2 x 2
  continent meanLifeExp
  <chr>           <dbl>
1 Americas         78.4
2 Europe           80.0
```

grouping column

new value we computed

```
d %>%
group_by(continent) %>%
summarize(meanLifeExp = mean(lifeExp))
```

new column

value

d → filter → select → mutate →

%>% (the pipe) allows you to more easily read function calls in the order they are applied.

1st arg "piped in"

```
d %>%
filter(continent == “Europe”) %>%
select(country, lifeExp) %>%
mutate(longlife = 80 < lifeExp)
```

with pipe

| - bash, shell

without pipe

1st argument

```
> temp = filter(d, continent == "Europe")
> temp = select(temp, country, lifeExp)
> temp = mutate(temp, longlife = 80 < lifeExp)
> temp
```

```
        country lifeExp longlife
         France  80.657     TRUE
United Kingdom  79.425    FALSE
```

Breakout Room plan

dplyr and SQL

Structured Query Language ≠ R

dplyr

```
d %>% select(continent, country)
```

SQL

```
SELECT continent, country FROM d
```