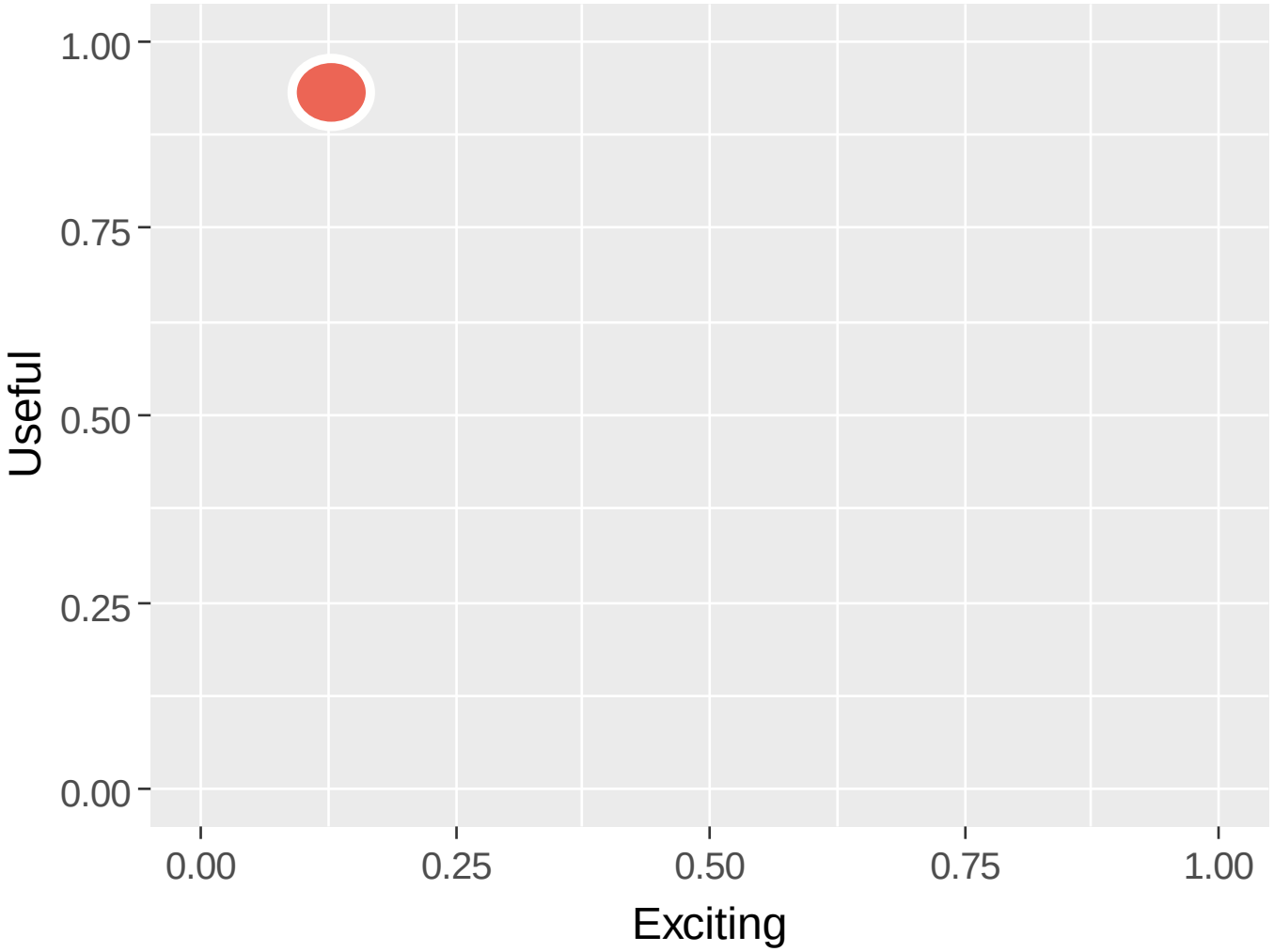
Session 9: Structures

Andrew Jones | Strategy Unit

Useful and Exciting



What is a data frame?

A data frame is a rectangular collection of variables (in columns) and observations (in rows).

id	gender	score
1	F	10.24
2	F	5.98
3	M	7.62

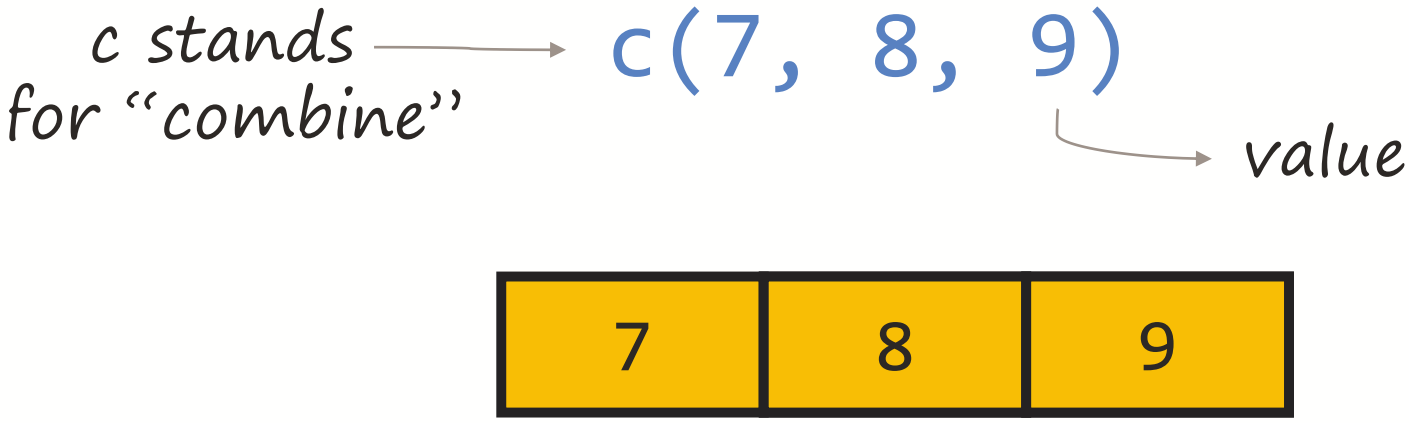
tibble = data frame

You may also come across the term "tibble". We'll take "tibble" to be synonymous with "data frame".

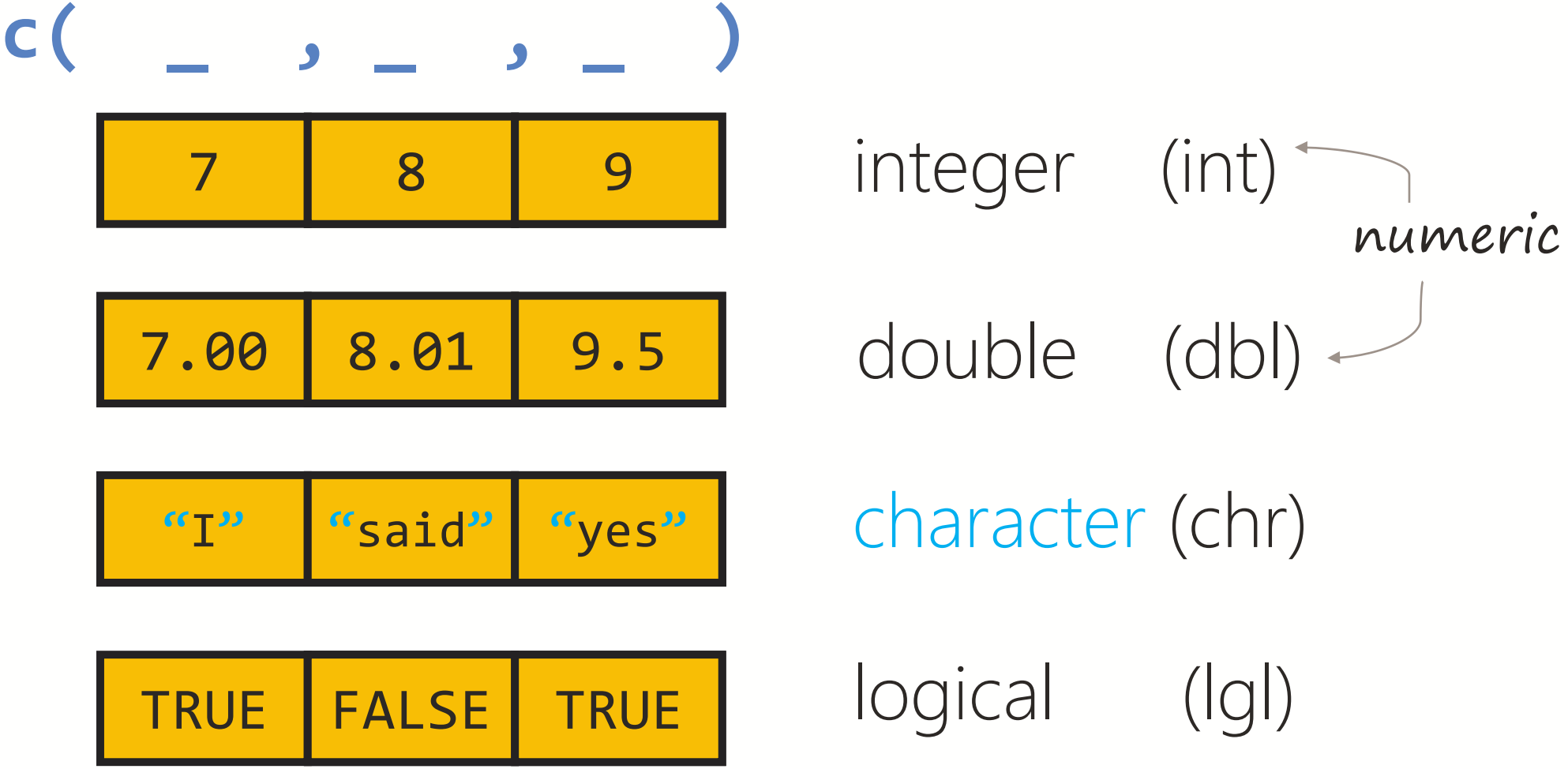
id	gender	score
1	F	10.24
2	F	5.98
3	M	7.62

Vectors

Vectors are the basic data structure in R. They are also the building blocks of data frames.

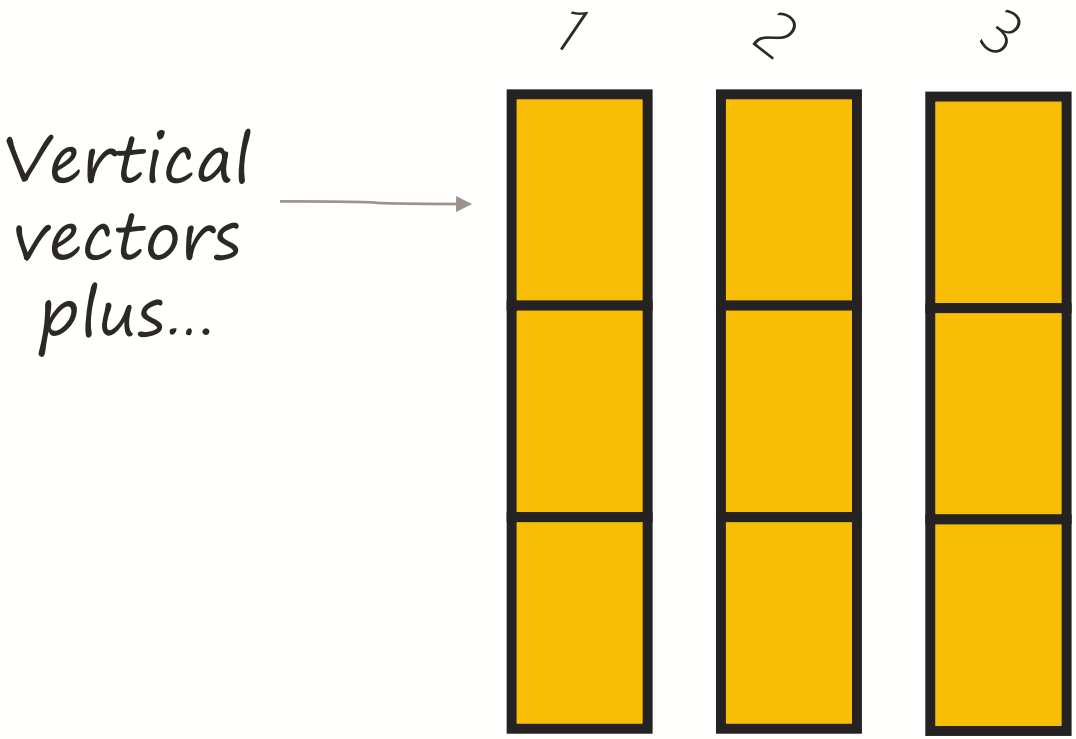


Vector types



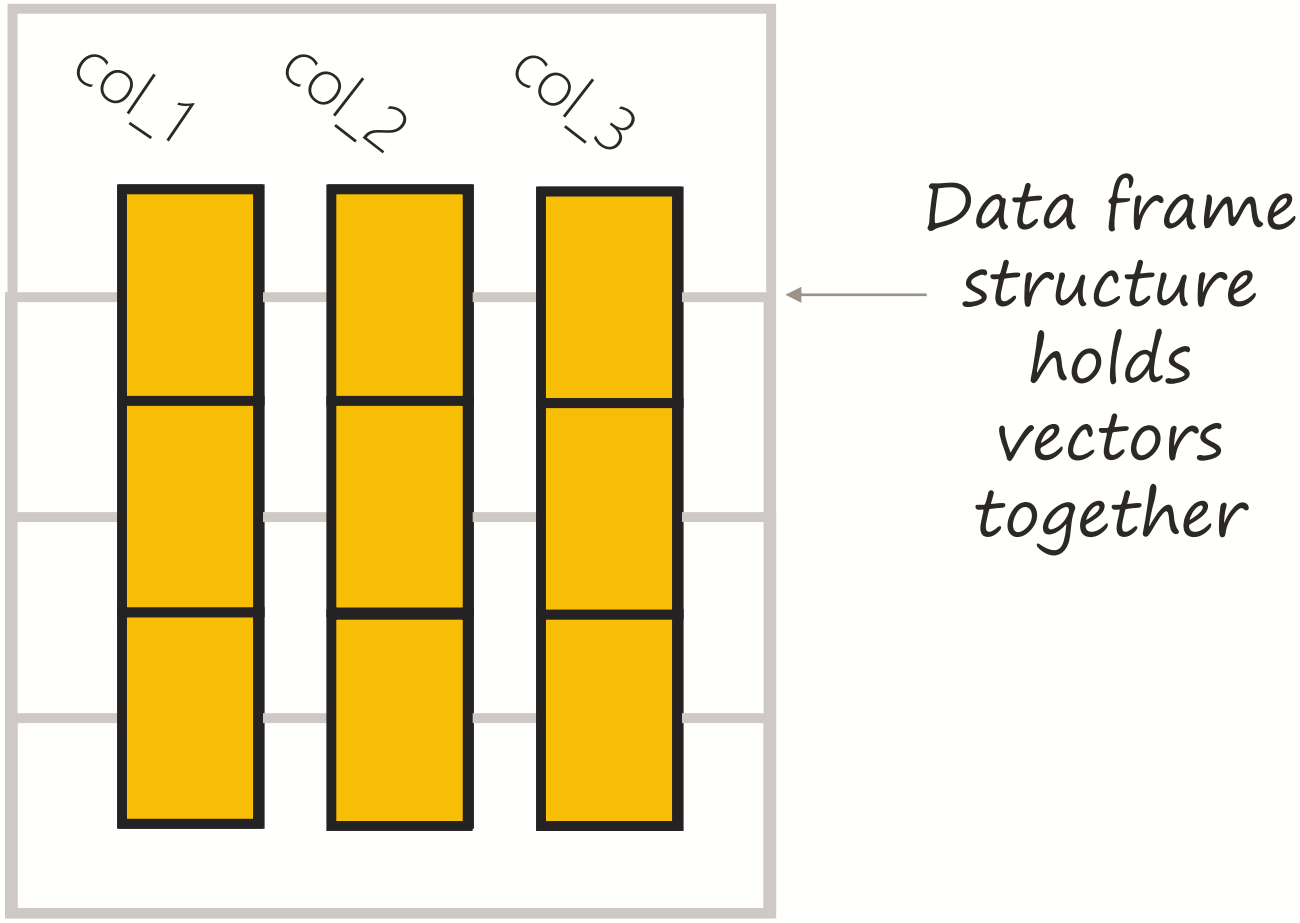
Data frames

We can think of data frames like this:

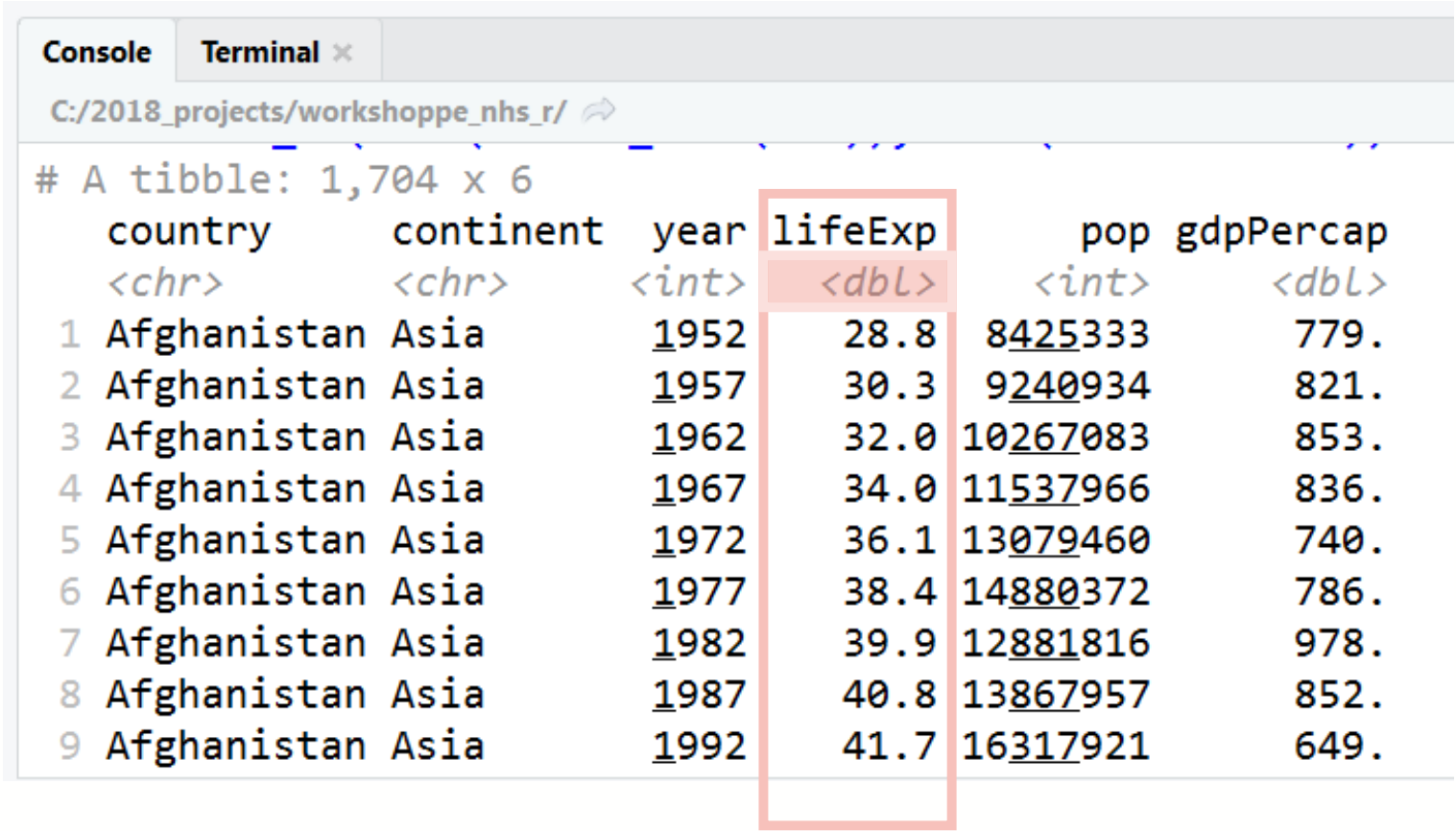


Data frames

We can think of data frames like this:



Vectors within data frames




```
# A tibble: 1,704 x 6
   country     continent  year lifeExp      pop gdpPercap
   <chr>       <chr>     <int>   <dbl>    <int>     <dbl>
 1 Afghanistan Asia       1952    28.8  8425333       779.
 2 Afghanistan Asia       1957    30.3  9240934       821.
 3 Afghanistan Asia       1962    32.0 10267083       853.
 4 Afghanistan Asia       1967    34.0 11537966       836.
 5 Afghanistan Asia       1972    36.1 13079460       740.
 6 Afghanistan Asia       1977    38.4 14880372       786.
 7 Afghanistan Asia       1982    39.9 12881816       978.
 8 Afghanistan Asia       1987    40.8 13867957       852.
 9 Afghanistan Asia       1992    41.7 16317921       649.
```

Extracting vectors

Data frame columns = vectors = a series of values.

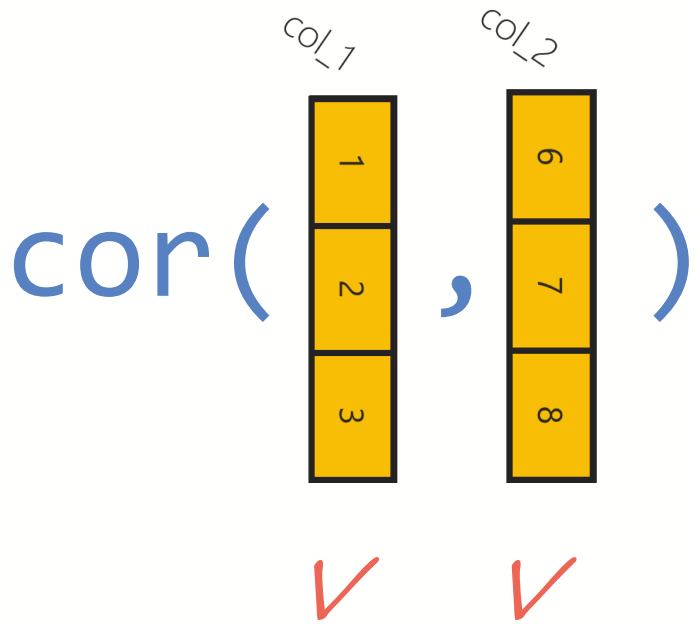
Extracting vectors

Data frame columns = vectors = a series of values.

Many excellent R tools work with vectors, but will not work with the extra structure found around data frame columns.

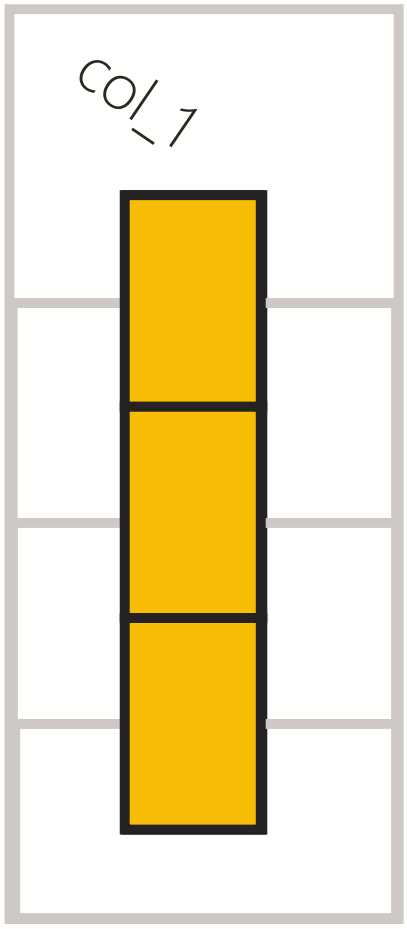
Using vectors

Take, for instance, the function **`cor()`** from base R . It will return the correlation between two vectors.



But, notice:

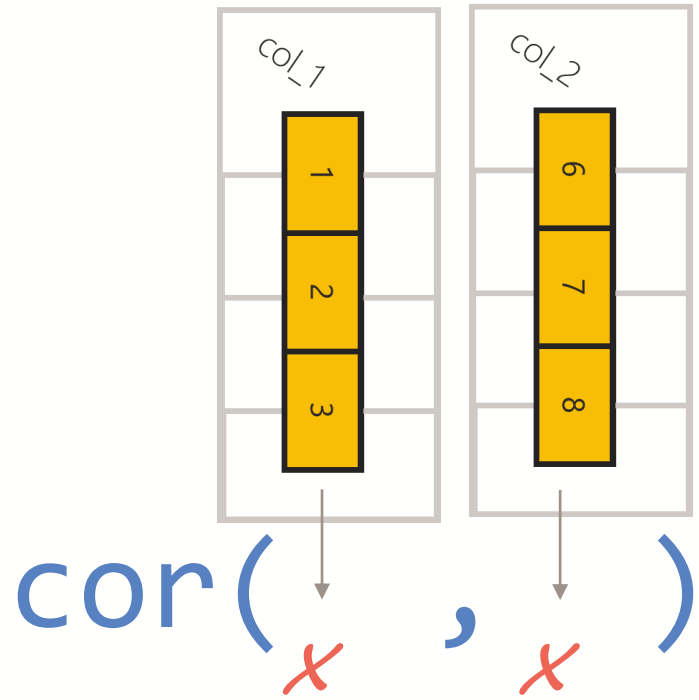
```
df %>%
  select(col_1)
```



Data frame structure remains

Extra structure

If our columns have data frame structure, they won't "fit".



Extracting vectors

```
df$col_1
```

base R syntax

```
df %>%
  pull(col_1)
```

tidyverse style

Data frame structure removed

Your turn

Load the gapminder library. First, use the dplyr command, **`select`**, to print the year column to the console. Next, extract the year column with the `$` operator and print to console. Note the difference.

Assign this vector to an object named **`yr`** .

Your turn (2)

Using Gapminder (observations for 2007) find the correlation **`cor()`** between GDP and population.

You'll need to `filter()` and `mutate()` then assign the resulting data frame to an object: `gdp_df`

Use `$` to extract vectors and `cor()`to find the correlation (single value) between the GDP and population vectors.

Your turn (3)

Now let's try plotting population against GDP with geom point. Add a geom_smooth layer with a linear fit and assign to an object: **p1**

```
geom_smooth(method = "lm")
```

Your turn (4)

Install the package "gridExtra" (note capital E). Make plot objects p2 and p3 for the same graphic but different smooth methods: "glm" and "loess" (default). Use grid arrange to plot them in a grid:

```
grid.arrange(p1, p2, p3, nrow = 2)
```

Solution

Create the data frame and assign to object gdp_df:

```
gdp_df <- gapminder %>%
    mutate(gdp = pop*gdpPercap) %>%
    filter(year == 2007)
```

Solution

Create plot. E.g. for plot 3:

```
p3 <- ggplot(gdp_df, aes(pop, gdp))+
        geom_point()+
        ylim(0, 5e12)+
        xlim(0, 2.5e8)+
        geom_smooth(method = "lm")

grid.arrange(p1, p2, p3, nrow = 2)
```

I'll zoom in on this region. e is scientific notation

Addendum: Lists

Lists are a special type of vector. They can store data types of different kinds:

Addendum: Lists

Lists can store plots, and even whole data frames:

This work is licensed as

Creative Commons

Attribution-ShareAlike 4.0

International

To view a copy of this license, visit

https://creativecommons.org/licenses/by-sa/4.0/

End