

Tidyverse Webinar

Gapminder data

gapminder

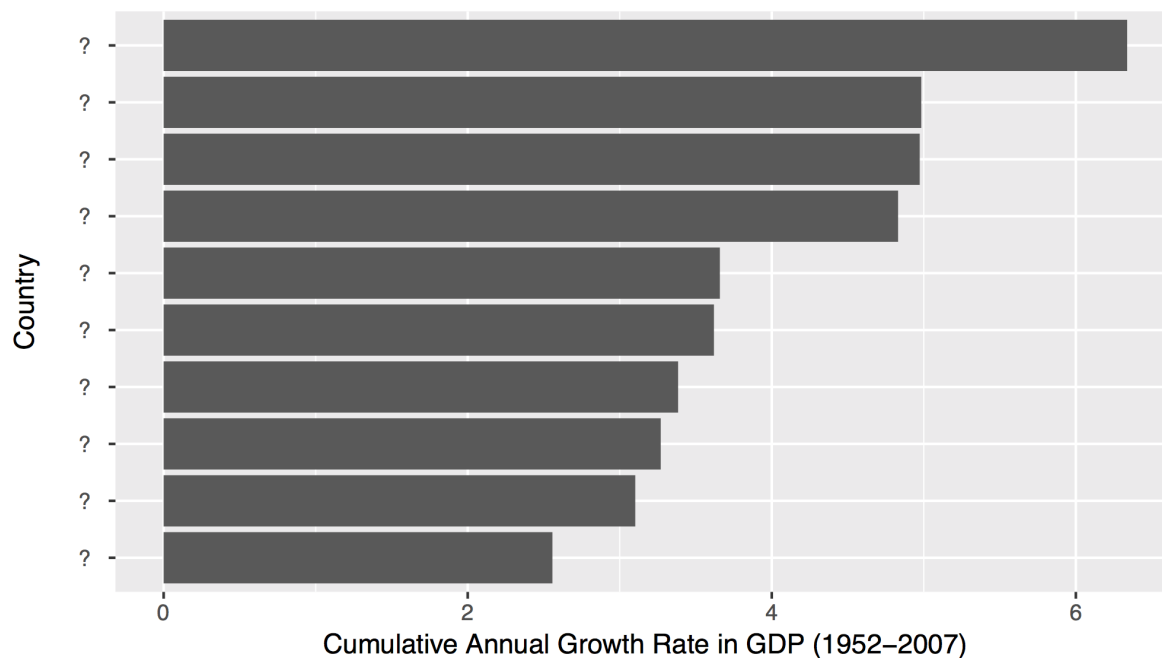
The gapminder data set contains demographic statistics popularized by Hans Rosling's TED talks.

```
library(gapminder)
gapminder
```

```
## # A tibble: 1,704 x 6
##   country continent year lifeExp      pop gdpPercap
##   <fctr>    <fctr> <int>   <dbl>    <int>    <dbl>
## 1 Afghanistan Asia  1952  28.801  8425333  779.4453
## 2 Afghanistan Asia  1957  30.332  9240934  820.8530
## 3 Afghanistan Asia  1962  31.997 10267083  853.1007
## 4 Afghanistan Asia  1967  34.020 11537966  836.1971
## 5 Afghanistan Asia  1972  36.088 13079460  739.9811
## 6 Afghanistan Asia  1977  38.438 14880372  786.1134
## 7 Afghanistan Asia  1982  39.854 12881816  978.0114
## 8 Afghanistan Asia  1987  40.822 13867957  852.3959
## 9 Afghanistan Asia  1992  41.674 16317921  649.3414
##10 Afghanistan Asia  1997  41.763 22227415  635.3414
## # ... with 1,694 more rows
```

Goal

Which countries had the fastest growing GDP's between 1952 and 2007?



The Tidyverse

Functions

In R, you manipulate data by passing the data to functions.

```
round(1234.567, digits = 2)
```

```
## [1] 1234.57
```

```
nrow(gapminder)
```

```
## [1] 1704
```

The tidyverse is a collection of R packages that contain functions. You must load the packages to use the functions.

Load the Tidyverse

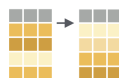
```
## install.packages("tidyverse")  
library("tidyverse")
```

Tidy Tools

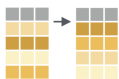
Tidyverse functions are designed to be:

1. **Simple** - They do one thing, and they do it well
2. **Composable** - They can be combined with other functions for multi-step operations

Which countries have the largest populations?



arrange(.data, ...)
Order rows by values of a column (low to high), use
with **desc()** to order from high to low.



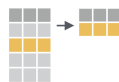
arrange(.data, desc(...))
Order rows by values of a column (low to high),
use with **desc()** to order from high to low.

```
arrange(gapminder, desc(pop))
```

```
## # A tibble: 1,704 x 6  
##   country continent  year  lifeExp      pop gdpPercap  
##   <fctr>      <fctr> <int>    <dbl>    <int>    <dbl>  
## 1  China      Asia   2007  72.96100 1318683096 4959.1149  
## 2  China      Asia   2002  72.02800 1280400000 3119.2809  
## 3  China      Asia   1997  70.42600 1230075000 2289.2341  
## 4  China      Asia   1992  68.69000 1164970000 1655.7842  
## 5  India      Asia   2007  64.69800 1110396331 2452.2104  
## 6  China      Asia   1987  67.27400 1084035000 1378.9040
```

```
## 7 India Asia 2002 62.87900 1034172547 1746.7695
## 8 China Asia 1982 65.52500 1000281000 962.4214
## 9 India Asia 1997 61.76500 959000000 1458.8174
## 10 China Asia 1977 63.96736 943455000 741.2375
## # ... with 1,694 more rows
```

Which countries had the largest population *in 2007*?



filter(.data, ...)
Extract rows that meet logical criteria.

```
gapminder2007 <- filter(gapminder, year == 2007)
arrange(gapminder2007, desc(pop))
```

```
## # A tibble: 142 x 6
##   country continent year lifeExp      pop gdpPercap
##   <fctr>    <fctr> <int>   <dbl>    <int>    <dbl>
## 1 China      Asia   2007  72.961 1318683096 4959.115
## 2 India      Asia   2007  64.698 1110396331 2452.210
## 3 United States Americas 2007  78.242 301139947 42951.653
## 4 Indonesia Asia    2007  70.650 223547000 3540.652
## 5 Brazil    Americas 2007  72.390 190010647 9065.801
## 6 Pakistan  Asia    2007  65.483 169270617 2605.948
## 7 Bangladesh Asia    2007  64.062 150448339 1391.254
## 8 Nigeria   Africa  2007  46.859 135031164 2013.977
## 9 Japan     Asia    2007  82.603 127467972 31656.068
## 10 Mexico   Americas 2007  76.195 108700891 11977.575
## # ... with 132 more rows
```

A better way

Use the pipe operator (`%>%`) to compose tidyverse functions.

```
gapminder %>%
  filter(year == 2007) %>%
  arrange(desc(pop))
```

```
## # A tibble: 142 x 6
##   country continent year lifeExp      pop gdpPercap
##   <fctr>    <fctr> <int>   <dbl>    <int>    <dbl>
## 1 China      Asia   2007  72.961 1318683096 4959.115
## 2 India      Asia   2007  64.698 1110396331 2452.210
## 3 United States Americas 2007  78.242 301139947 42951.653
## 4 Indonesia Asia    2007  70.650 223547000 3540.652
## 5 Brazil    Americas 2007  72.390 190010647 9065.801
## 6 Pakistan  Asia    2007  65.483 169270617 2605.948
## 7 Bangladesh Asia    2007  64.062 150448339 1391.254
## 8 Nigeria   Africa  2007  46.859 135031164 2013.977
## 9 Japan     Asia    2007  82.603 127467972 31656.068
## 10 Mexico   Americas 2007  76.195 108700891 11977.575
## # ... with 132 more rows
```

Which countries had the largest life expectancy in 2007?

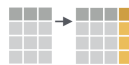


select(.data, ...)
Extract columns by name.

```
gapminder %>%  
  filter(year == 2007) %>%  
  arrange(desc(lifeExp)) %>%  
  select(country, lifeExp)
```

```
## # A tibble: 142 x 2  
##       country lifeExp  
##       <fctr>   <dbl>  
## 1      Japan  82.603  
## 2 Hong Kong, China 82.208  
## 3      Iceland 81.757  
## 4 Switzerland 81.701  
## 5      Australia 81.235  
## 6       Spain  80.941  
## 7       Sweden 80.884  
## 8       Israel 80.745  
## 9       France 80.657  
## 10      Canada 80.653  
## # ... with 132 more rows
```

What is the gdp of each country?



mutate(.data, ...)
Compute new column(s).

```
gapminder %>%  
  mutate(gdp = pop * gdpPercap)
```

```
## # A tibble: 1,704 x 7  
##       country continent year lifeExp      pop gdpPercap      gdp  
##       <fctr>    <fctr> <int>   <dbl>   <int>    <dbl>    <dbl>  
## 1 Afghanistan   Asia  1952  28.801  8425333  779.4453 6567086330  
## 2 Afghanistan   Asia  1957  30.332  9240934  820.8530 7585448670  
## 3 Afghanistan   Asia  1962  31.997 10267083  853.1007 8758855797  
## 4 Afghanistan   Asia  1967  34.020 11537966  836.1971 9648014150  
## 5 Afghanistan   Asia  1972  36.088 13079460  739.9811 9678553274  
## 6 Afghanistan   Asia  1977  38.438 14880372  786.1134 11697659231  
## 7 Afghanistan   Asia  1982  39.854 12881816  978.0114 12598563401  
## 8 Afghanistan   Asia  1987  40.822 13867957  852.3959 11820990309  
## 9 Afghanistan   Asia  1992  41.674 16317921  649.3414 10595901589  
## 10 Afghanistan   Asia  1997  41.763 22227415  635.3414 14121995875  
## # ... with 1,694 more rows
```

What was the maximum gdp?



summarise(.data, ...)
Compute table of summaries. Use **group_by()** to compute groupwise summaries.

```
gapminder %>%  
  mutate(gdp = pop * gdpPerCap) %>%  
  summarise(max_gdp = max(gdp))
```

```
## # A tibble: 1 x 1  
##       max_gdp  
##       <dbl>  
## 1 1.293446e+13
```

What was the first gdp? The last gdp?

```
gapminder %>%  
  mutate(gdp = pop * gdpPerCap) %>%  
  summarise(first_gdp = first(gdp), last_gdp = last(gdp))
```

```
## # A tibble: 1 x 2  
##   first_gdp last_gdp  
##   <dbl>    <dbl>  
## 1 6567086330 5782658337
```

What was the first and last gdp for each country?



group_by(.data, ..., add = FALSE)
Returns a table with recognized groups.

```
gapminder %>%  
  mutate(gdp = pop * gdpPerCap) %>%  
  group_by(country) %>%  
  summarise(first_gdp = first(gdp), last_gdp = last(gdp))
```

```
## # A tibble: 142 x 3  
##   country first_gdp last_gdp  
##   <fctr>    <dbl>    <dbl>  
## 1 Afghanistan 6567086330 31079291949  
## 2 Albania      2053669902 21376411360  
## 3 Algeria      22725632678 207444851958  
## 4 Angola       14899557133 59583895818  
## 5 Argentina    105676319105 515033625357  
## 6 Australia     87256254102 703658358894  
## 7 Austria       42516266683 296229400691
```

```
## 8      Bahrain  1188460759  21112675360
## 9  Bangladesh  32082059995  209311822134
## 10     Belgium  72838686716  350141166520
## # ... with 132 more rows
```

What was the cumulative annual growth for each country between 1952 and 1957?

```
gapminder %>%
  mutate(gdp = pop * gdpPercap) %>%
  group_by(country) %>%
  summarise(gdp1952 = first(gdp), gdp2007 = last(gdp)) %>%
  mutate(cagr = ((gdp2007 / gdp1952) ^ (1/55) - 1) * 100) %>%
  arrange(desc(cagr)) %>%
  select(country, cagr)
```

```
## # A tibble: 142 x 2
##       country      cagr
##       <fctr>    <dbl>
## 1      Singapore 8.348304
## 2 Equatorial Guinea 8.346729
## 3          Oman 8.218950
## 4          Taiwan 7.869795
## 5        Botswana 7.548170
## 6      Korea, Rep. 7.487215
## 7 Hong Kong, China 7.064369
## 8        Thailand 6.384445
## 9          Libya 6.372590
## 10         China 6.337334
## # ... with 132 more rows
```

Tidy data

Each tidyverse function expects and returns the same type of data: *tidy data*. A tabular data set is tidy iff:

1. Each variable is in its own column
2. Each observation is in its own row

Visualization

What did GDP growth look like?

Let's focus on the 10 biggest economies (in 1952). What are they?

```
gapminder %>%
  filter(year == 1952) %>%
  mutate(gdp = pop * gdpPercap) %>%
  arrange(desc(gdp)) %>%
  select(country, gdp)
```

```
## # A tibble: 142 x 2
##       country      gdp
##       <fctr>    <dbl>
## 1 United States 2.204242e+12
```

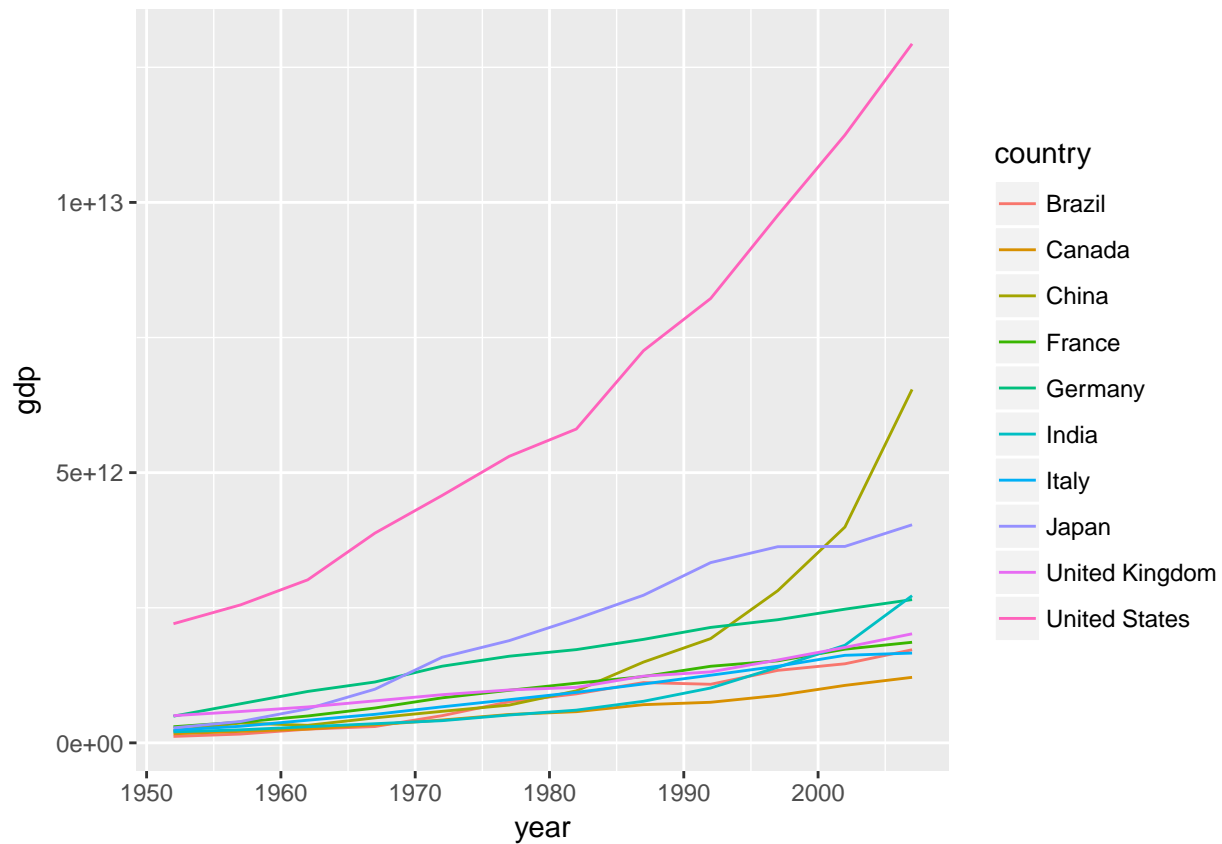
```
## 2 United Kingdom 5.032666e+11
## 3      Germany 4.939866e+11
## 4      France 2.984834e+11
## 5      Japan 2.781349e+11
## 6      Italy 2.350603e+11
## 7      China 2.227550e+11
## 8      India 2.033225e+11
## 9      Canada 1.680701e+11
## 10     Brazil 1.193716e+11
## # ... with 132 more rows
```

Visualize the Top 10

```
top_10 <- c("United States", "United Kingdom", "Germany", "France",
            "Japan", "Italy", "China", "India", "Canada", "Brazil")
```

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

```
gapminder %>%
  filter(country %in% top_10) %>%
  mutate(gdp = pop * gdpPercap) %>%
  ggplot() +
    geom_line(mapping = aes(x = year, y = gdp, color = country))
```



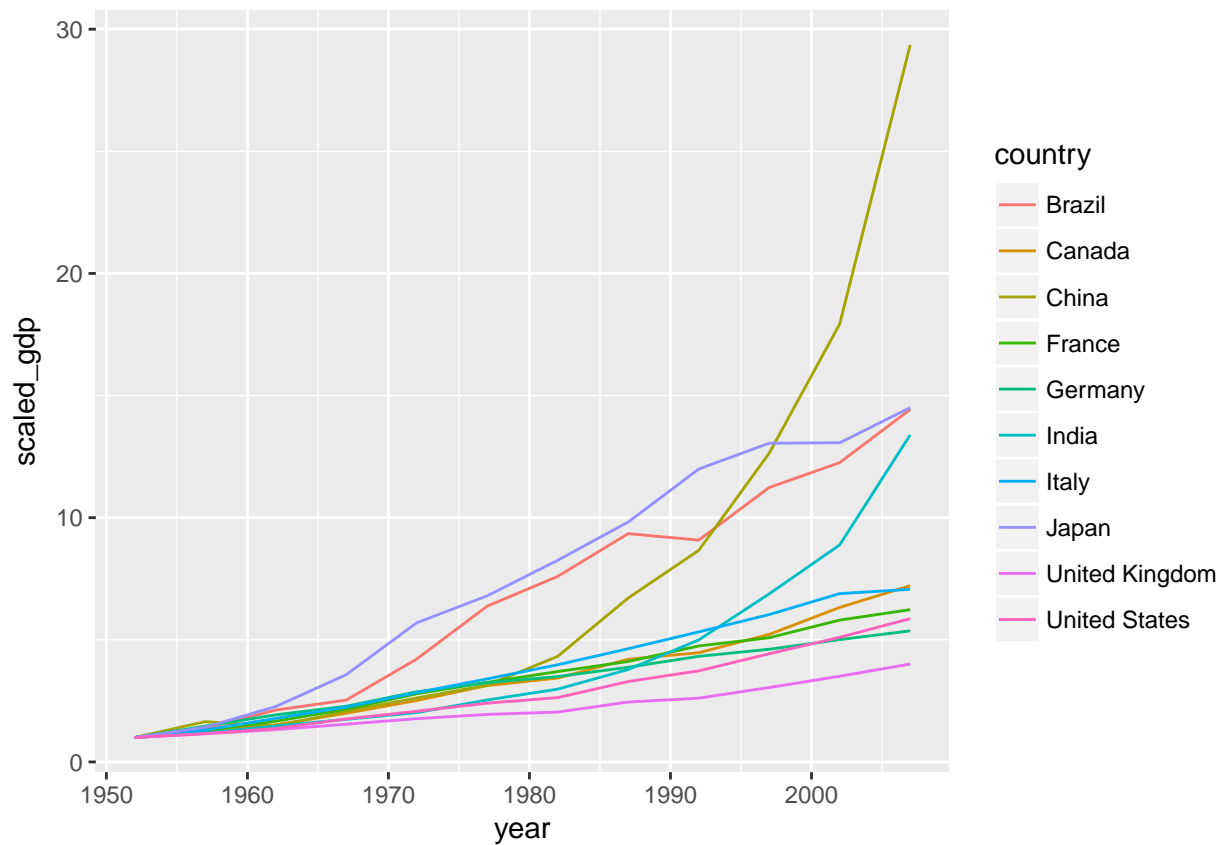
Scaled data

Let's scale the data within each country to make growth easier to compare

```
gapminder %>%  
  filter(country %in% top_10) %>%  
  mutate(gdp = pop * gdpPercap)
```

```
## # A tibble: 120 x 7  
##   country continent year lifeExp      pop gdpPercap      gdp  
##   <fctr>      <fctr> <int>   <dbl>    <int>    <dbl>    <dbl>  
## 1 Brazil  Americas  1952  50.917  56602560  2108.944  1.193716e+11  
## 2 Brazil  Americas  1957  53.285  65551171  2487.366  1.630498e+11  
## 3 Brazil  Americas  1962  55.665  76039390  3336.586  2.537119e+11  
## 4 Brazil  Americas  1967  57.632  88049823  3429.864  3.019989e+11  
## 5 Brazil  Americas  1972  59.504  100840058 4985.711  5.027594e+11  
## 6 Brazil  Americas  1977  61.489  114313951 6660.119  7.613445e+11  
## 7 Brazil  Americas  1982  63.336  128962939 7030.836  9.067173e+11  
## 8 Brazil  Americas  1987  65.205  142938076 7807.096  1.115931e+12  
## 9 Brazil  Americas  1992  67.057  155975974 6950.283  1.084077e+12  
## 10 Brazil Americas  1997  69.388  168546719 7957.981  1.341292e+12  
## # ... with 110 more rows
```

```
gapminder %>%  
  filter(country %in% top_10) %>%  
  mutate(gdp = pop * gdpPercap) %>%  
  group_by(country) %>%  
  mutate(scaled_gdp = gdp / first(gdp)) %>%  
  ggplot() +  
    geom_line(mapping = aes(x = year, y = scaled_gdp, color = country))
```

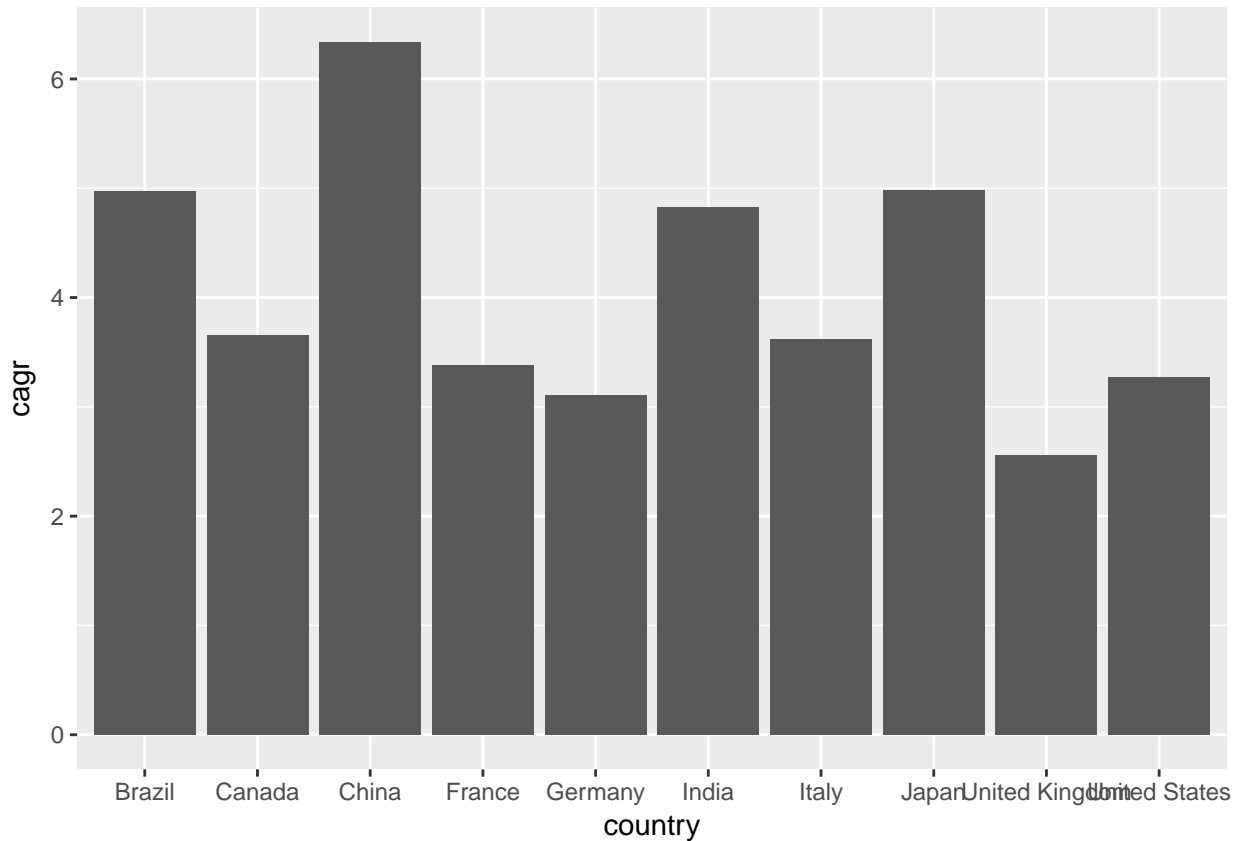



Cumulative Annual Growth Rates

```
gapminder %>%
  filter(country %in% top_10) %>%
  mutate(gdp = pop * gdpPerCap) %>%
  group_by(country) %>%
  summarise(start = first(gdp), end = last(gdp)) %>%
  mutate(cagr = ((end/start) ^ (1 / 55) - 1) * 100) %>%
  arrange(desc(cagr)) %>%
  select(country, cagr)
```

```
## # A tibble: 10 x 2
##   country      cagr
##   <fctr>      <dbl>
## 1 China 6.337334
## 2 Japan 4.983258
## 3 Brazil 4.973063
## 4 India 4.830628
## 5 Canada 3.658473
## 6 Italy 3.619402
## 7 France 3.383767
## 8 United States 3.269607
## 9 Germany 3.101929
## 10 United Kingdom 2.557105
```

```
gapminder %>%
  filter(country %in% top_10) %>%
  mutate(gdp = pop * gdpPerCap) %>%
  group_by(country) %>%
  summarise(start = first(gdp), end = last(gdp)) %>%
  mutate(cagr = ((end/start) ^ (1 / 55) - 1) * 100) %>%
  arrange(desc(cagr)) %>%
  select(country, cagr) %>%
  ggplot() +
    geom_col(mapping = aes(x = country, y = cagr))
```



Aspirational

```
library(forcats)

gapminder %>%
  filter(country %in% top_10) %>%
  mutate(gdp = pop * gdpPerCap) %>%
  group_by(country) %>%
  summarise(start = first(gdp), end = last(gdp)) %>%
  mutate(cagr = ((end/start) ^ (1 / 55) - 1) * 100) %>%
  arrange(desc(cagr)) %>%
  ggplot() +
    geom_col(mapping = aes(x = fct_reorder(country, cagr), y = cagr)) +
    labs(x = "Country", y = "Cumulative Annual Growth Rate in GDP (1952-2007)") +
```

```
coord_flip()
```

