



Workshop: Intro to R and R Studio

Data Analysis Team:

- Albina Gibadullina (GAA)
- Amir Michalovich (GAA)
- Jeremy Buhler (Data Librarian)
- Sarah Parker (Data Librarian)

Land Acknowledgement

UBC is located on the traditional, ancestral, and unceded territory of the xʷməθkʷəy̓əm (Musqueam) people.

- Traditional: recognizes lands traditionally used and/or occupied by the Musqueam people or other First Nations in other parts of the country.
- Ancestral: recognizes land that is handed down from generation to generation.
- Unceded: refers to land that was not turned over to the Crown (government) by a treaty or other agreement.

Check out native-land.ca (<https://native-land.ca/>) to see the interactive map of Indigenous territories around the world.



Pre-workshop setup

Download and install R

For Windows:

1. Visit [R Project \(https://www.r-project.org/\)](https://www.r-project.org/) to learn about R versions.
2. Download and install R from your preferred CRAN mirror [here \(https://cran.r-project.org/mirrors.html\)](https://cran.r-project.org/mirrors.html)
 - A. Choose "0-Cloud" or a mirror site near you.

For Mac:

1. Check that your macOS system is up-to-date
2. Download and install R from [The Comprehensive R Archive Network \(https://cran.r-project.org/\)](https://cran.r-project.org/)

Download and install R studio

For Windows and Mac:

1. Download and install R Studio from [here \(https://rstudio.com/products/rstudio/download/#download\)](https://rstudio.com/products/rstudio/download/#download)

Learning Objectives

- Become familiar with R and R studio environment.
- Learn the basic R programming language.
- Learn how to explore data in R.

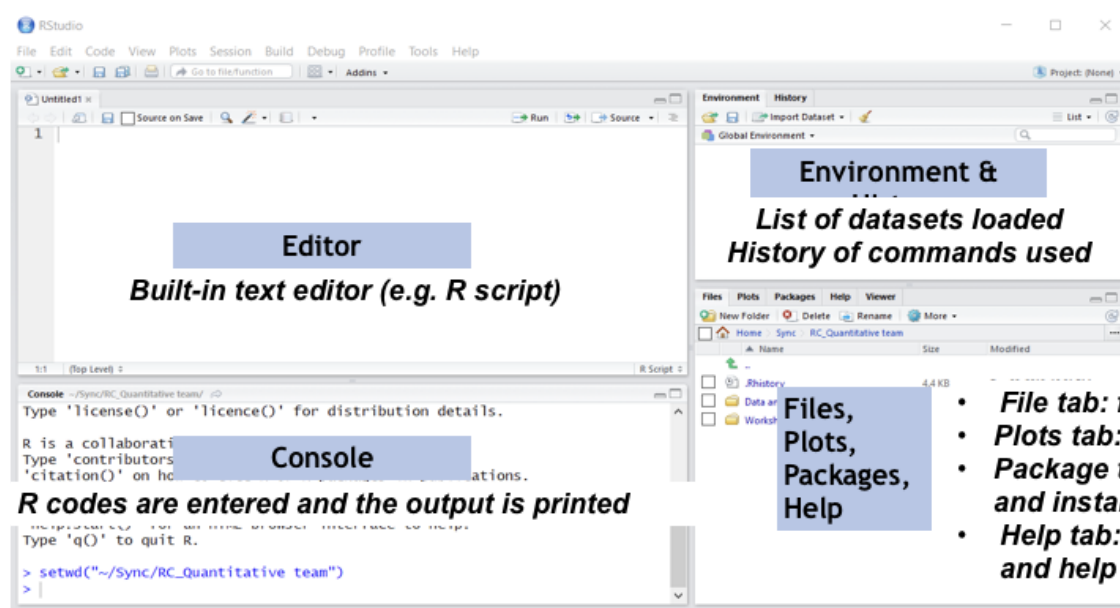
R vs. R Studio

- **R** is a programming language. We use R to run our codes and see their output.
- **R Studio** is another program which manages R in a user-friendly environment.

That is why, it is highly recommended to install both programs because they work together.

R Studio Environment

- Open R studio
- To create a new R script, click File > New file > R Script

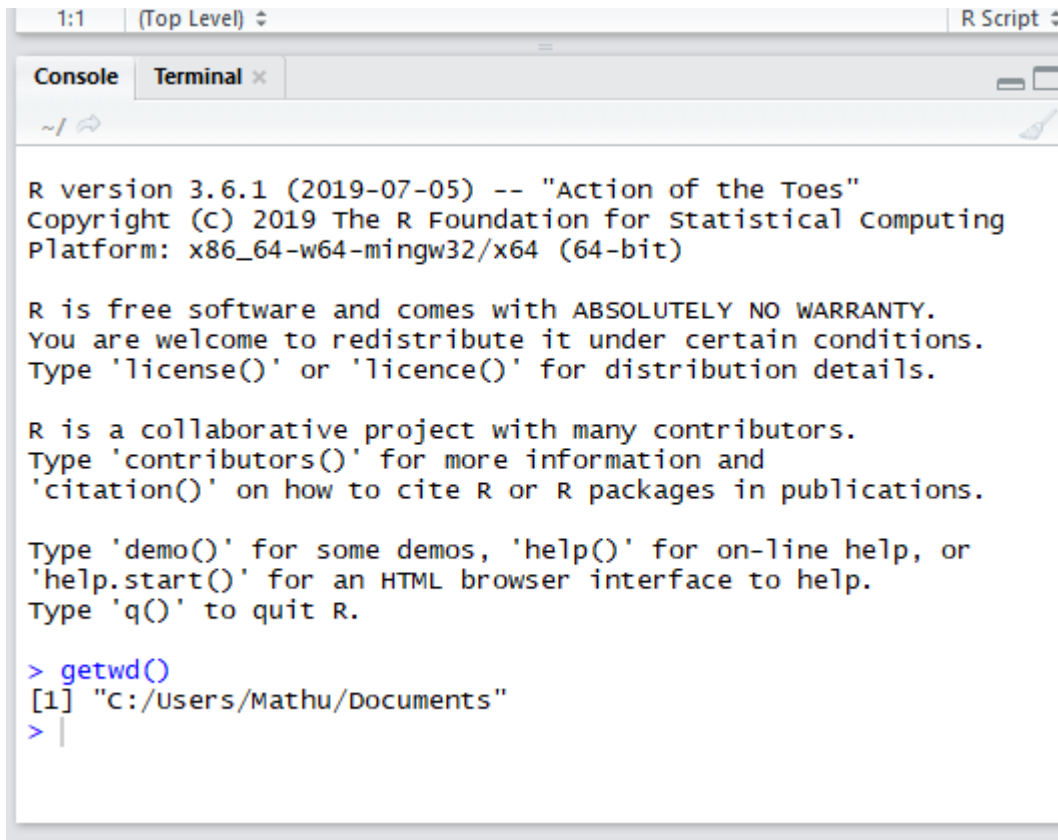


Working Directory

- Working directory is a folder/path where R reads and saves files.

Check your current working directory

- To check your current working directory, write and run `getwd()` in the console.



The screenshot shows the R Studio interface with the console pane active. The console displays the R version information and the output of the `getwd()` function. The output is `[1] "C:/Users/Mathu/Documents"`.

```
1:1 (Top Level) R Script
Console Terminal x
~/
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

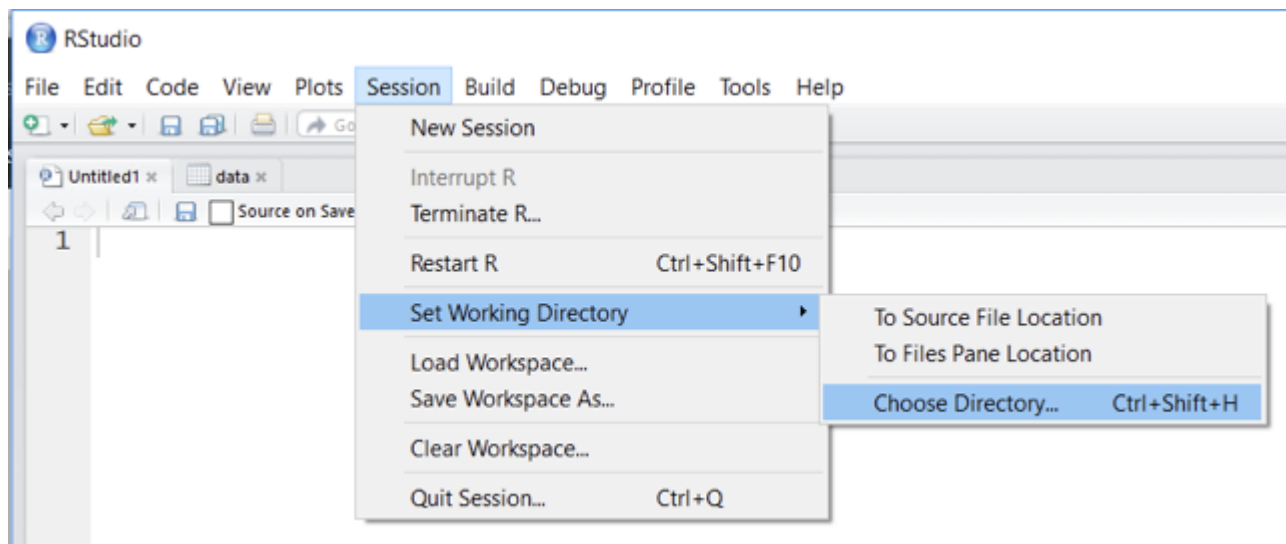
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> getwd()
[1] "C:/Users/Mathu/Documents"
> |
```

Set or Change your current working directory



- If you know the filepath to your new working directory, you can also write and run `setwd(dir)` in the console.

Data Input

Data types and structures

There are different types of data that can be stored in R. In many cases, our data is organized and stored as `DataFrame`.

A DataFrame:

Character	Numeric	Factor
A	1	High
B	2	Medium
C	3	Low

- Has multiple columns and rows
- Contains different types of variables
 - Numeric variables: numeric values with and without decimal places
 - Categorical variables: qualitative data that can be represented by characters or factors

Basic R commands

An operator is a symbol that instructs R to perform specific operation. Here are some basic operators which can be used in R:

Mathematical operations

Description: This allows R to perform mathematics operations.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
<- or =	Assign value from the right to the left

Relational operations

Description: This allows R to compare variables.

Operator	Description
==	Equal to
!=	Not Equal to
>	Great than
<	Less than
>=	Great than or equal to
<=	Less than or equal to

Exercise #1

Calculate

- $1+1$
- $4-9$
- $19/3$
- $14*6$
- 4^3

Assign values to variables

- $a <- 4$
- a
- $b <- 5$
- b

Question: What do you notice in the Environment pane?

Calculate

- $c <- 6 + a$
 - c
 - $d <- a + b + c$
 - d
 - $c==d$
-

Getting Started

R package

R package is a library of prewritten code designed for a particular task or a collection of tasks. For today, we will mainly use 'dplyr' and 'ggplot2'.

“psych” R package as
a general toolbox for
psychological research

“dplyr” R package
for data management

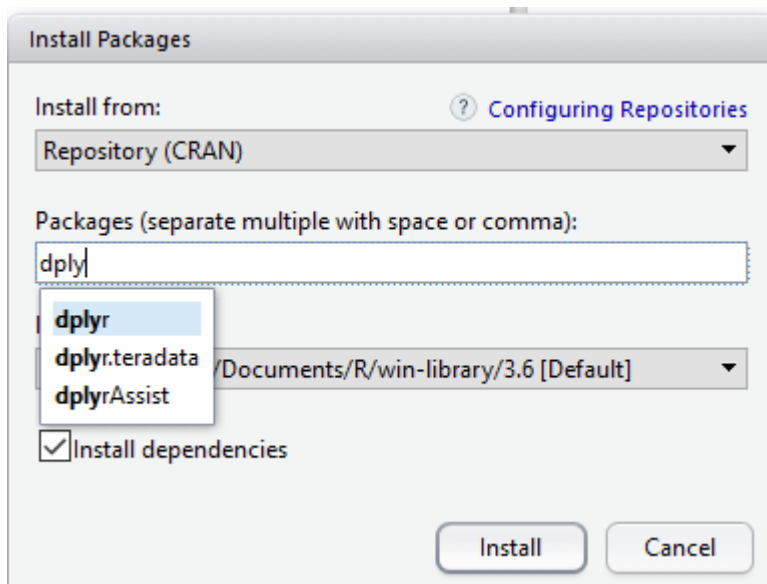


**“ggplot2”
R package**
for data
visualization

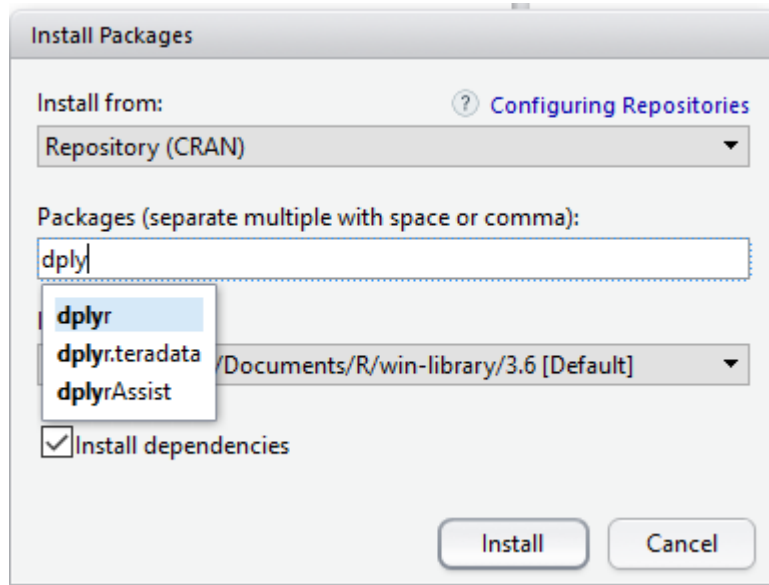
Installing a new package

There are three options to install a new package.

1. Under Tools -> Install Package -> Search for “dplyr”, “ggplot2”, and “gapminder”



2. In Packages Tab (bottom right corner) -> Install -> Search for “dplyr”, “ggplot2”, and “gapminder” -> Install



3. Write and run the following codes:

```
install.packages("dplyr")
```

```
install.packages("ggplot2")
```

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

```
In [41]: install.packages("dplyr")  
  
install.packages("ggplot2")  
  
install.packages("gapminder")
```

```
Installing package into ‘/home/jupyter/R/x86_64-pc-linux-gnu-library/4.0’  
(as ‘lib’ is unspecified)
```

```
Installing package into ‘/home/jupyter/R/x86_64-pc-linux-gnu-library/4.0’  
(as ‘lib’ is unspecified)
```

```
Warning message:
```

```
“package ‘ggplot2’ is not available (for R version 4.0.2)”
```

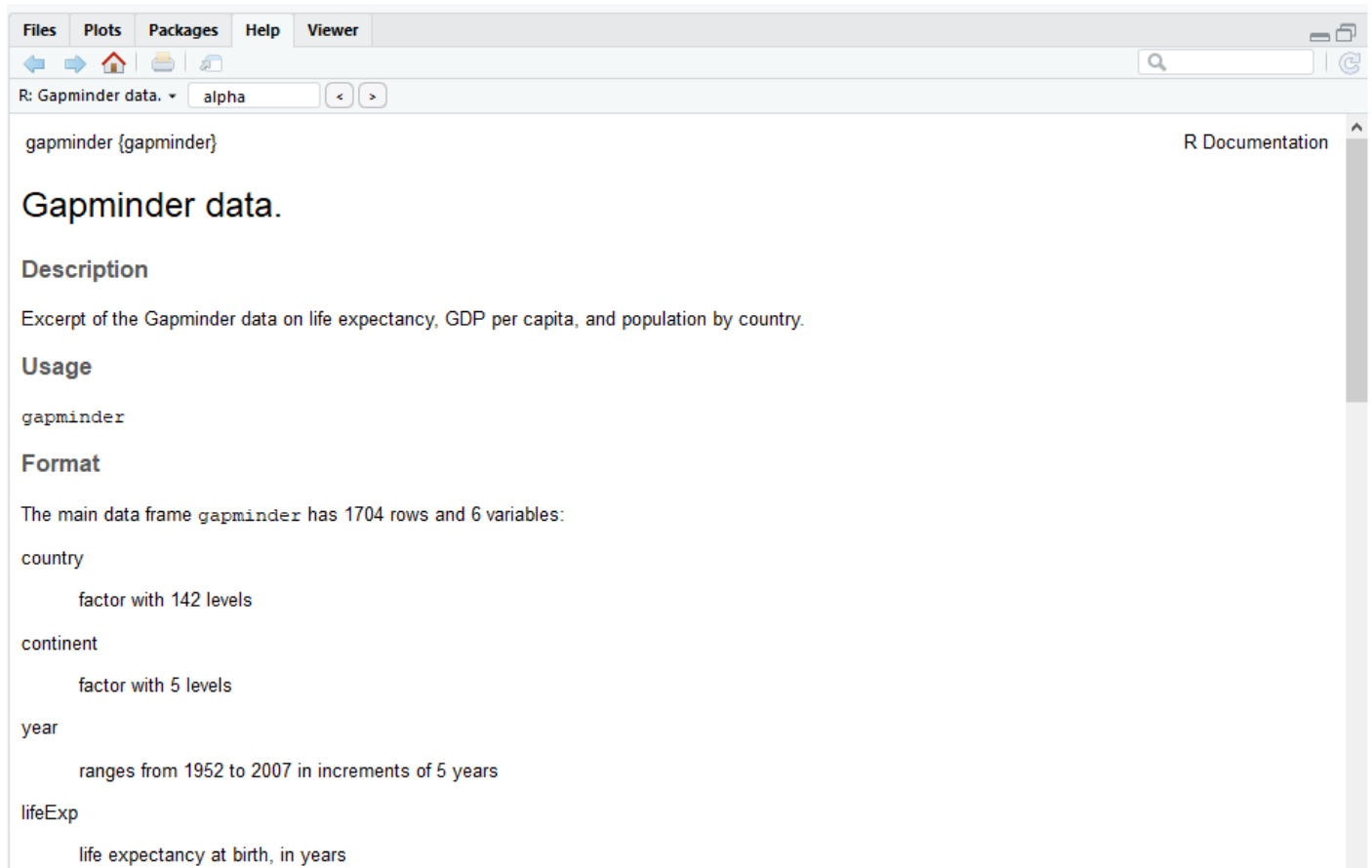
```
Installing package into ‘/home/jupyter/R/x86_64-pc-linux-gnu-library/4.0’  
(as ‘lib’ is unspecified)
```

Loading installed packages

```
In [42]: # Use the library command to load any installed packages.  
  
library(dplyr)  
  
library(ggplot2)  
  
library(gapminder)
```

Execute this code to check the description of the dataset

?gapminder



The screenshot shows the RStudio interface with the 'Viewer' tab active. The title bar indicates 'R: Gapminder data, alpha'. The main content area displays the documentation for the 'gapminder' dataset, which includes a description, usage instructions, and a detailed format section. The format section lists the variables: 'country' (factor with 142 levels), 'continent' (factor with 5 levels), 'year' (ranging from 1952 to 2007 in 5-year increments), and 'lifeExp' (life expectancy at birth in years). A search bar and navigation icons are visible at the top of the viewer window.

gapminder {gapminder}

R Documentation

Gapminder data.

Description

Excerpt of the Gapminder data on life expectancy, GDP per capita, and population by country.

Usage

```
gapminder
```

Format

The main data frame `gapminder` has 1704 rows and 6 variables:

`country`

factor with 142 levels

`continent`

factor with 5 levels

`year`

ranges from 1952 to 2007 in increments of 5 years

`lifeExp`

life expectancy at birth, in years

Importing data from built-in R datasets

```
In [3]: # Import data from the built-in "gapminder" dataset

countries <- gapminder
head(countries) #see the first six rows of the dataframe
```

A tibble: 6 × 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
Afghanistan	Asia	1977	38.438	14880372	786.1134

How can you see the entire dataset?

View(countries)

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Intro to R workshop.Rmd* x countries x

Filter

	country	continent	year	lifeExp	pop	gdpPercap
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
11	Afghanistan	Asia	2002	42.129	25268405	726.7341
12	Afghanistan	Asia	2007	43.828	31889923	974.5803
13	Albania	Europe	1952	55.230	1282697	1601.0561
14	Albania	Europe	1957	59.280	1476505	1942.2842
15	Albania	Europe	1962	64.820	1728137	2312.8890
16	Albania	Europe	1967	66.220	1984060	2760.1969
17	Albania	Europe	1972	67.690	2263554	3313.4222
18	Albania	Europe	1977	68.930	2509048	3533.0039
19	Albania	Europe	1982	70.420	2780097	3630.8807
20	Albania	Europe	1987	72.000	3075321	3738.9327
21	Albania	Europe	1992	71.581	3326498	2497.4379
22	Albania	Europe	1997	72.950	3428038	3193.0546
23	Albania	Europe	2002	75.651	3588513	4604.3117

Showing 1 to 23 of 1,704 entries

In [4]: *# How can you check the data structure?*

```
str(countries)
```

```
tibble [1,704 × 6] (S3: tbl_df/tbl/data.frame)
 $ country  : Factor w/ 142 levels "Afghanistan",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ continent: Factor w/ 5 levels "Africa","Americas",...: 3 3 3 3 3 3 3 3 3 3 ...
...
 $ year      : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997
...
 $ lifeExp   : num [1:1704] 28.8 30.3 32 34 36.1 ...
 $ pop       : int [1:1704] 8425333 9240934 10267083 11537966 13079460 1488037
2 12881816 13867957 16317921 22227415 ...
 $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

In [5]: *# Count the number of observations (measured by the number of rows) and the number of variables (measured by the number of columns)*

```
nrow(countries) #To check the number of rows
ncol(countries) #To check the number of columns
```

```
1704
```

```
6
```

In [6]: *# Summarize the dataframe*

```
summary(countries)
```

country	continent	year	lifeExp
Afghanistan: 12	Africa :624	Min. :1952	Min. :23.60
Albania : 12	Americas:300	1st Qu.:1966	1st Qu.:48.20
Algeria : 12	Asia :396	Median :1980	Median :60.71
Angola : 12	Europe :360	Mean :1980	Mean :59.47
Argentina : 12	Oceania : 24	3rd Qu.:1993	3rd Qu.:70.85
Australia : 12		Max. :2007	Max. :82.60
(Other) :1632			

pop	gdpPercap
Min. :6.001e+04	Min. : 241.2
1st Qu.:2.794e+06	1st Qu.: 1202.1
Median :7.024e+06	Median : 3531.8
Mean :2.960e+07	Mean : 7215.3
3rd Qu.:1.959e+07	3rd Qu.: 9325.5
Max. :1.319e+09	Max. :113523.1

In [8]: *# Check if the dataframe is complete:*

```
# Count the number of observations for each available year
table(countries$year)
```

```
1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
142  142  142  142  142  142  142  142  142  142  142  142
```

In [9]: *# Count the number of observations in each continent for each available year*

```
table(countries$continent, countries$year)
```

	1952	1957	1962	1967	1972	1977	1982	1987	1992	1997	2002	2007
Africa	52	52	52	52	52	52	52	52	52	52	52	52
Americas	25	25	25	25	25	25	25	25	25	25	25	25
Asia	33	33	33	33	33	33	33	33	33	33	33	33
Europe	30	30	30	30	30	30	30	30	30	30	30	30
Oceania	2	2	2	2	2	2	2	2	2	2	2	2

Extra Info: Importing data from external data sources

CSV data file

- `read.csv(file="countries.csv")` # make sure the file is located in your working directory

SPSS, SAS, or Stata data file

- `install.packages("foreign")`
- `library(foreign)`
- `read.spss()` # SPSS
- `read.dta()` # Stata
- `read.ssd()` # SAS

Data Manipulation with dplyr

Using dplyr verbs to manage data

select ()

- To select columns based on column names
- Useful when your data has many columns and you only need a subset of them

In [10]: *# Select specific columns such as country, year, lifeExp*

```
select(countries, country, year, lifeExp)
```


A tibble: 1704 × 3

country	year	lifeExp
<fct>	<int>	<dbl>
Afghanistan	1952	28.801
Afghanistan	1957	30.332
Afghanistan	1962	31.997
Afghanistan	1967	34.020
Afghanistan	1972	36.088
Afghanistan	1977	38.438
Afghanistan	1982	39.854
Afghanistan	1987	40.822
Afghanistan	1992	41.674
Afghanistan	1997	41.763
Afghanistan	2002	42.129
Afghanistan	2007	43.828
Albania	1952	55.230
Albania	1957	59.280
Albania	1962	64.820
Albania	1967	66.220
Albania	1972	67.690
Albania	1977	68.930
Albania	1982	70.420
Albania	1987	72.000
Albania	1992	71.581
Albania	1997	72.950
Albania	2002	75.651
Albania	2007	76.423
Algeria	1952	43.077
Algeria	1957	45.685
Algeria	1962	48.303
Algeria	1967	51.407
Algeria	1972	54.518
Algeria	1977	58.014
:	:	:
Yemen, Rep.	1982	49.113
Yemen, Rep.	1987	52.922
Yemen, Rep.	1992	55.599

country	year	lifeExp
<fct>	<int>	<dbl>
Yemen, Rep.	1997	58.020
Yemen, Rep.	2002	60.308
Yemen, Rep.	2007	62.698
Zambia	1952	42.038
Zambia	1957	44.077
Zambia	1962	46.023
Zambia	1967	47.768
Zambia	1972	50.107
Zambia	1977	51.386
Zambia	1982	51.821
Zambia	1987	50.821
Zambia	1992	46.100
Zambia	1997	40.238
Zambia	2002	39.193
Zambia	2007	42.384
Zimbabwe	1952	48.451
Zimbabwe	1957	50.469
Zimbabwe	1962	52.358
Zimbabwe	1967	53.995
Zimbabwe	1972	55.635
Zimbabwe	1977	57.674
Zimbabwe	1982	60.363
Zimbabwe	1987	62.351
Zimbabwe	1992	60.377
Zimbabwe	1997	46.809
Zimbabwe	2002	39.989
Zimbabwe	2007	43.487

```
In [11]: # Select all columns except for continent and gdpPercap  
select(countries, -continent, -gdpPercap) # use subtraction operator
```

A tibble: 1704 × 4

country	year	lifeExp	pop
<fct>	<int>	<dbl>	<int>
Afghanistan	1952	28.801	8425333
Afghanistan	1957	30.332	9240934
Afghanistan	1962	31.997	10267083
Afghanistan	1967	34.020	11537966
Afghanistan	1972	36.088	13079460
Afghanistan	1977	38.438	14880372
Afghanistan	1982	39.854	12881816
Afghanistan	1987	40.822	13867957
Afghanistan	1992	41.674	16317921
Afghanistan	1997	41.763	22227415
Afghanistan	2002	42.129	25268405
Afghanistan	2007	43.828	31889923
Albania	1952	55.230	1282697
Albania	1957	59.280	1476505
Albania	1962	64.820	1728137
Albania	1967	66.220	1984060
Albania	1972	67.690	2263554
Albania	1977	68.930	2509048
Albania	1982	70.420	2780097
Albania	1987	72.000	3075321
Albania	1992	71.581	3326498
Albania	1997	72.950	3428038
Albania	2002	75.651	3508512
Albania	2007	76.423	3600523
Algeria	1952	43.077	9279525
Algeria	1957	45.685	10270856
Algeria	1962	48.303	11000948
Algeria	1967	51.407	12760499
Algeria	1972	54.518	14760787
Algeria	1977	58.014	17152804
:	:	:	:
Yemen, Rep.	1982	49.113	9657618
Yemen, Rep.	1987	52.922	11219340
Yemen, Rep.	1992	55.599	13367997

country	year	lifeExp	pop
<fct>	<int>	<dbl>	<int>
Yemen, Rep.	1997	58.020	15826497
Yemen, Rep.	2002	60.308	18701257
Yemen, Rep.	2007	62.698	22211743
Zambia	1952	42.038	2672000
Zambia	1957	44.077	3016000
Zambia	1962	46.023	3421000
Zambia	1967	47.768	3900000
Zambia	1972	50.107	4506497
Zambia	1977	51.386	5216550
Zambia	1982	51.821	6100407
Zambia	1987	50.821	7272406
Zambia	1992	46.100	8381163
Zambia	1997	40.238	9417789
Zambia	2002	39.193	10595811
Zambia	2007	42.384	11746035
Zimbabwe	1952	48.451	3080907
Zimbabwe	1957	50.469	3646340
Zimbabwe	1962	52.358	4277736
Zimbabwe	1967	53.995	4995432
Zimbabwe	1972	55.635	5861135
Zimbabwe	1977	57.674	6642107
Zimbabwe	1982	60.363	7636524
Zimbabwe	1987	62.351	9216418
Zimbabwe	1992	60.377	10704340
Zimbabwe	1997	46.809	11404948
Zimbabwe	2002	39.989	11926563
Zimbabwe	2007	43.487	12311143

filter ()

Filter rows based on conditions

In [12]: *# Filter data of countries from 1997 and with Oceania continent*

```
filter(countries, year==1997, continent=="Oceania")
```

A tibble: 2 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Australia	Oceania	1997	78.83	18565243	26997.94
New Zealand	Oceania	1997	77.55	3676187	21050.41

In [13]: *# Filter data of countries from 2007 and with gdpPercap greater than \$40,000*

```
filter(countries, year==2007, gdpPercap>40000)
```

A tibble: 5 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Ireland	Europe	2007	78.885	4109086	40676.00
Kuwait	Asia	2007	77.588	2505559	47306.99
Norway	Europe	2007	80.196	4627926	49357.19
Singapore	Asia	2007	79.972	4553009	47143.18
United States	Americas	2007	78.242	301139947	42951.65

Question: What relational operator should you use for greater than and equal to \$40,000?

In [14]: *# Solution:*

```
filter(countries, year==2007, gdpPercap>=40000)
```

A tibble: 5 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Ireland	Europe	2007	78.885	4109086	40676.00
Kuwait	Asia	2007	77.588	2505559	47306.99
Norway	Europe	2007	80.196	4627926	49357.19
Singapore	Asia	2007	79.972	4553009	47143.18
United States	Americas	2007	78.242	301139947	42951.65

```
In [15]: # Filter observations from year 2002 or year 2007  
  
filter(countries, year==2002 | year==2007) # Use vertical line for Or condition
```

A tibble: 284 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Afghanistan	Asia	2002	42.129	25268405	726.7341
Afghanistan	Asia	2007	43.828	31889923	974.5803
Albania	Europe	2002	75.651	3508512	4604.2117
Albania	Europe	2007	76.423	3600523	5937.0295
Algeria	Africa	2002	70.994	31287142	5288.0404
Algeria	Africa	2007	72.301	33333216	6223.3675
Angola	Africa	2002	41.003	10866106	2773.2873
Angola	Africa	2007	42.731	12420476	4797.2313
Argentina	Americas	2002	74.340	38331121	8797.6407
Argentina	Americas	2007	75.320	40301927	12779.3796
Australia	Oceania	2002	80.370	19546792	30687.7547
Australia	Oceania	2007	81.235	20434176	34435.3674
Austria	Europe	2002	78.980	8148312	32417.6077
Austria	Europe	2007	79.829	8199783	36126.4927
Bahrain	Asia	2002	74.795	656397	23403.5593
Bahrain	Asia	2007	75.635	708573	29796.0483
Bangladesh	Asia	2002	62.013	135656790	1136.3904
Bangladesh	Asia	2007	64.062	150448339	1391.2538
Belgium	Europe	2002	78.320	10311970	30485.8838
Belgium	Europe	2007	79.441	10392226	33692.6051
Benin	Africa	2002	54.406	7026113	1372.8779
Benin	Africa	2007	56.728	8078314	1441.2849
Bolivia	Americas	2002	63.883	8445134	3413.2627
Bolivia	Americas	2007	65.554	9119152	3822.1371
Bosnia and Herzegovina	Europe	2002	74.090	4165416	6018.9752
Bosnia and Herzegovina	Europe	2007	74.852	4552198	7446.2988
Botswana	Africa	2002	46.634	1630347	11003.6051
Botswana	Africa	2007	50.728	1639131	12569.8518
Brazil	Americas	2002	71.006	179914212	8131.2128
Brazil	Americas	2007	72.390	190010647	9065.8008
:	:	:	:	:	:
Thailand	Asia	2002	68.564	62806748	5913.1875
Thailand	Asia	2007	70.616	65068149	7458.3963
Togo	Africa	2002	57.561	4977378	886.2206

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Togo	Africa	2007	58.420	5701579	882.9699
Trinidad and Tobago	Americas	2002	68.976	1101832	11460.6002
Trinidad and Tobago	Americas	2007	69.819	1056608	18008.5092
Tunisia	Africa	2002	73.042	9770575	5722.8957
Tunisia	Africa	2007	73.923	10276158	7092.9230
Turkey	Europe	2002	70.845	67308928	6508.0857
Turkey	Europe	2007	71.777	71158647	8458.2764
Uganda	Africa	2002	47.813	24739869	927.7210
Uganda	Africa	2007	51.542	29170398	1056.3801
United Kingdom	Europe	2002	78.471	59912431	29478.9992
United Kingdom	Europe	2007	79.425	60776238	33203.2613
United States	Americas	2002	77.310	287675526	39097.0995
United States	Americas	2007	78.242	301139947	42951.6531
Uruguay	Americas	2002	75.307	3363085	7727.0020
Uruguay	Americas	2007	76.384	3447496	10611.4630
Venezuela	Americas	2002	72.766	24287670	8605.0478
Venezuela	Americas	2007	73.747	26084662	11415.8057
Vietnam	Asia	2002	73.017	80908147	1764.4567
Vietnam	Asia	2007	74.249	85262356	2441.5764
West Bank and Gaza	Asia	2002	72.370	3389578	4515.4876
West Bank and Gaza	Asia	2007	73.422	4018332	3025.3498
Yemen, Rep.	Asia	2002	60.308	18701257	2234.8208
Yemen, Rep.	Asia	2007	62.698	22211743	2280.7699
Zambia	Africa	2002	39.193	10595811	1071.6139
Zambia	Africa	2007	42.384	11746035	1271.2116
Zimbabwe	Africa	2002	39.989	11926563	672.0386
Zimbabwe	Africa	2007	43.487	12311143	469.7093

arrange ()

Arrange rows by ascending or decending order

In [16]: *# Arrange countries based on life expectancy (from smallest to largest)*

```
arrange(countries, lifeExp)
```

A tibble: 1704 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Rwanda	Africa	1992	23.599	7290203	737.0686
Afghanistan	Asia	1952	28.801	8425333	779.4453
Gambia	Africa	1952	30.000	284320	485.2307
Angola	Africa	1952	30.015	4232095	3520.6103
Sierra Leone	Africa	1952	30.331	2143249	879.7877
Afghanistan	Asia	1957	30.332	9240934	820.8530
Cambodia	Asia	1977	31.220	6978607	524.9722
Mozambique	Africa	1952	31.286	6446316	468.5260
Sierra Leone	Africa	1957	31.570	2295678	1004.4844
Burkina Faso	Africa	1952	31.975	4469979	543.2552
Afghanistan	Asia	1962	31.997	10267083	853.1007
Angola	Africa	1957	31.999	4561361	3827.9405
Gambia	Africa	1957	32.065	323150	520.9267
Guinea-Bissau	Africa	1952	32.500	580653	299.8503
Yemen, Rep.	Asia	1952	32.548	4963829	781.7176
Sierra Leone	Africa	1962	32.767	2467895	1116.6399
Somalia	Africa	1952	32.978	2526994	1135.7498
Guinea-Bissau	Africa	1957	33.489	601095	431.7905
Guinea	Africa	1952	33.609	2664249	510.1965
Mali	Africa	1952	33.685	3838168	452.3370
Mozambique	Africa	1957	33.779	7038035	495.5868
Gambia	Africa	1962	33.896	374020	599.6503
Yemen, Rep.	Asia	1957	33.970	5498090	804.8305
Angola	Africa	1962	34.000	4826015	4269.2767
Afghanistan	Asia	1967	34.020	11537966	836.1971
Ethiopia	Africa	1952	34.078	20860941	362.1463
Sierra Leone	Africa	1967	34.113	2662190	1206.0435
Equatorial Guinea	Africa	1952	34.482	216964	375.6431
Guinea-Bissau	Africa	1962	34.488	627820	522.0344
Guinea	Africa	1957	34.558	2876726	576.2670
:	:	:	:	:	:
Greece	Europe	2007	79.483	10706290	27538.41
France	Europe	2002	79.590	59925035	28926.03
Israel	Asia	2002	79.696	6029529	21905.60

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Netherlands	Europe	2007	79.762	16570613	36797.93
Canada	Americas	2002	79.770	31902268	33328.97
Spain	Europe	2002	79.780	40152517	24835.47
Austria	Europe	2007	79.829	8199783	36126.49
Singapore	Asia	2007	79.972	4553009	47143.18
Hong Kong, China	Asia	1997	80.000	6495918	28377.63
Sweden	Europe	2002	80.040	8954175	29341.63
Norway	Europe	2007	80.196	4627926	49357.19
New Zealand	Oceania	2007	80.204	4115771	25185.01
Italy	Europe	2002	80.240	57926999	27968.10
Australia	Oceania	2002	80.370	19546792	30687.75
Iceland	Europe	2002	80.500	288030	31163.20
Italy	Europe	2007	80.546	58147733	28569.72
Switzerland	Europe	2002	80.620	7361757	34480.96
Canada	Americas	2007	80.653	33390141	36319.24
France	Europe	2007	80.657	61083916	30470.02
Japan	Asia	1997	80.690	125956499	28816.58
Israel	Asia	2007	80.745	6426679	25523.28
Sweden	Europe	2007	80.884	9031088	33859.75
Spain	Europe	2007	80.941	40448191	28821.06
Australia	Oceania	2007	81.235	20434176	34435.37
Hong Kong, China	Asia	2002	81.495	6762476	30209.02
Switzerland	Europe	2007	81.701	7554661	37506.42
Iceland	Europe	2007	81.757	301931	36180.79
Japan	Asia	2002	82.000	127065841	28604.59
Hong Kong, China	Asia	2007	82.208	6980412	39724.98
Japan	Asia	2007	82.603	127467972	31656.07

```
In [17]: # Arrange countries based on life expectancy (from smallest to largest), sorted first by year (from earliest year available to most recent)  
  
arrange(countries, year, lifeExp)
```

A tibble: 1704 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Afghanistan	Asia	1952	28.801	8425333	779.4453
Gambia	Africa	1952	30.000	284320	485.2307
Angola	Africa	1952	30.015	4232095	3520.6103
Sierra Leone	Africa	1952	30.331	2143249	879.7877
Mozambique	Africa	1952	31.286	6446316	468.5260
Burkina Faso	Africa	1952	31.975	4469979	543.2552
Guinea-Bissau	Africa	1952	32.500	580653	299.8503
Yemen, Rep.	Asia	1952	32.548	4963829	781.7176
Somalia	Africa	1952	32.978	2526994	1135.7498
Guinea	Africa	1952	33.609	2664249	510.1965
Mali	Africa	1952	33.685	3838168	452.3370
Ethiopia	Africa	1952	34.078	20860941	362.1463
Equatorial Guinea	Africa	1952	34.482	216964	375.6431
Djibouti	Africa	1952	34.812	63149	2669.5295
Central African Republic	Africa	1952	35.463	1291695	1071.3107
Eritrea	Africa	1952	35.928	1438760	328.9406
Nepal	Asia	1952	36.157	9182536	545.8657
Malawi	Africa	1952	36.256	2917802	369.1651
Myanmar	Asia	1952	36.319	20092996	331.0000
Nigeria	Africa	1952	36.324	33119096	1077.2819
Madagascar	Africa	1952	36.681	4762912	1443.0117
Gabon	Africa	1952	37.003	420702	4293.4765
Senegal	Africa	1952	37.278	2755589	1450.3570
India	Asia	1952	37.373	372000000	546.5657
Niger	Africa	1952	37.444	3379468	761.8794
Indonesia	Asia	1952	37.468	82052000	749.6817
Bangladesh	Asia	1952	37.484	46886859	684.2442
Oman	Asia	1952	37.578	507833	1828.2303
Haiti	Americas	1952	37.579	3201488	1840.3669
Chad	Africa	1952	38.092	2682462	1178.6659
:	:	:	:	:	:
United States	Americas	2007	78.242	301139947	42951.653
Cuba	Americas	2007	78.273	11416987	8948.103
Denmark	Europe	2007	78.332	5468120	35278.419

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Taiwan	Asia	2007	78.400	23174294	28718.277
Chile	Americas	2007	78.553	16284741	13171.639
Korea, Rep.	Asia	2007	78.623	49044790	23348.140
Puerto Rico	Americas	2007	78.746	3942491	19328.709
Costa Rica	Americas	2007	78.782	4133884	9645.061
Ireland	Europe	2007	78.885	4109086	40675.996
Finland	Europe	2007	79.313	5238460	33207.084
Germany	Europe	2007	79.406	82400996	32170.374
United Kingdom	Europe	2007	79.425	60776238	33203.261
Belgium	Europe	2007	79.441	10392226	33692.605
Greece	Europe	2007	79.483	10706290	27538.412
Netherlands	Europe	2007	79.762	16570613	36797.933
Austria	Europe	2007	79.829	8199783	36126.493
Singapore	Asia	2007	79.972	4553009	47143.180
Norway	Europe	2007	80.196	4627926	49357.190
New Zealand	Oceania	2007	80.204	4115771	25185.009
Italy	Europe	2007	80.546	58147733	28569.720
Canada	Americas	2007	80.653	33390141	36319.235
France	Europe	2007	80.657	61083916	30470.017
Israel	Asia	2007	80.745	6426679	25523.277
Sweden	Europe	2007	80.884	9031088	33859.748
Spain	Europe	2007	80.941	40448191	28821.064
Australia	Oceania	2007	81.235	20434176	34435.367
Switzerland	Europe	2007	81.701	7554661	37506.419
Iceland	Europe	2007	81.757	301931	36180.789
Hong Kong, China	Asia	2007	82.208	6980412	39724.979
Japan	Asia	2007	82.603	127467972	31656.068

In [18]: *# Arrange countries based on life expectancy (from largest to smallest)*

```
arrange(countries, desc(lifeExp))
```


A tibble: 1704 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Japan	Asia	2007	82.603	127467972	31656.07
Hong Kong, China	Asia	2007	82.208	6980412	39724.98
Japan	Asia	2002	82.000	127065841	28604.59
Iceland	Europe	2007	81.757	301931	36180.79
Switzerland	Europe	2007	81.701	7554661	37506.42
Hong Kong, China	Asia	2002	81.495	6762476	30209.02
Australia	Oceania	2007	81.235	20434176	34435.37
Spain	Europe	2007	80.941	40448191	28821.06
Sweden	Europe	2007	80.884	9031088	33859.75
Israel	Asia	2007	80.745	6426679	25523.28
Japan	Asia	1997	80.690	125956499	28816.58
France	Europe	2007	80.657	61083916	30470.02
Canada	Americas	2007	80.653	33390141	36319.24
Switzerland	Europe	2002	80.620	7361757	34480.96
Italy	Europe	2007	80.546	58147733	28569.72
Iceland	Europe	2002	80.500	288030	31163.20
Australia	Oceania	2002	80.370	19546792	30687.75
Italy	Europe	2002	80.240	57926999	27968.10
New Zealand	Oceania	2007	80.204	4115771	25185.01
Norway	Europe	2007	80.196	4627926	49357.19
Sweden	Europe	2002	80.040	8954175	29341.63
Hong Kong, China	Asia	1997	80.000	6495918	28377.63
Singapore	Asia	2007	79.972	4553009	47143.18
Austria	Europe	2007	79.829	8199783	36126.49
Spain	Europe	2002	79.780	40152517	24835.47
Canada	Americas	2002	79.770	31902268	33328.97
Netherlands	Europe	2007	79.762	16570613	36797.93
Israel	Asia	2002	79.696	6029529	21905.60
France	Europe	2002	79.590	59925035	28926.03
Greece	Europe	2007	79.483	10706290	27538.41
:	:	:	:	:	:
Guinea	Africa	1957	34.558	2876726	576.2670
Guinea-Bissau	Africa	1962	34.488	627820	522.0344
Equatorial Guinea	Africa	1952	34.482	216964	375.6431

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Sierra Leone	Africa	1967	34.113	2662190	1206.0435
Ethiopia	Africa	1952	34.078	20860941	362.1463
Afghanistan	Asia	1967	34.020	11537966	836.1971
Angola	Africa	1962	34.000	4826015	4269.2767
Yemen, Rep.	Asia	1957	33.970	5498090	804.8305
Gambia	Africa	1962	33.896	374020	599.6503
Mozambique	Africa	1957	33.779	7038035	495.5868
Mali	Africa	1952	33.685	3838168	452.3370
Guinea	Africa	1952	33.609	2664249	510.1965
Guinea-Bissau	Africa	1957	33.489	601095	431.7905
Somalia	Africa	1952	32.978	2526994	1135.7498
Sierra Leone	Africa	1962	32.767	2467895	1116.6399
Yemen, Rep.	Asia	1952	32.548	4963829	781.7176
Guinea-Bissau	Africa	1952	32.500	580653	299.8503
Gambia	Africa	1957	32.065	323150	520.9267
Angola	Africa	1957	31.999	4561361	3827.9405
Afghanistan	Asia	1962	31.997	10267083	853.1007
Burkina Faso	Africa	1952	31.975	4469979	543.2552
Sierra Leone	Africa	1957	31.570	2295678	1004.4844
Mozambique	Africa	1952	31.286	6446316	468.5260
Cambodia	Asia	1977	31.220	6978607	524.9722
Afghanistan	Asia	1957	30.332	9240934	820.8530
Sierra Leone	Africa	1952	30.331	2143249	879.7877
Angola	Africa	1952	30.015	4232095	3520.6103
Gambia	Africa	1952	30.000	284320	485.2307
Afghanistan	Asia	1952	28.801	8425333	779.4453
Rwanda	Africa	1992	23.599	7290203	737.0686

```
In [19]: # Arrange countries based on life expectancy (from largest to smallest), sorted first by year (from earliest year available to most recent)  
  
arrange(countries, year, desc(lifeExp))
```

A tibble: 1704 × 6

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Norway	Europe	1952	72.670	3327728	10095.422
Iceland	Europe	1952	72.490	147962	7267.688
Netherlands	Europe	1952	72.130	10381988	8941.572
Sweden	Europe	1952	71.860	7124673	8527.845
Denmark	Europe	1952	70.780	4334000	9692.385
Switzerland	Europe	1952	69.620	4815000	14734.233
New Zealand	Oceania	1952	69.390	1994794	10556.576
United Kingdom	Europe	1952	69.180	50430000	9979.508
Australia	Oceania	1952	69.120	8691212	10039.596
Canada	Americas	1952	68.750	14785584	11367.161
United States	Americas	1952	68.440	157553000	13990.482
Belgium	Europe	1952	68.000	8730405	8343.105
Germany	Europe	1952	67.500	69145952	7144.114
France	Europe	1952	67.410	42459667	7029.809
Ireland	Europe	1952	66.910	2952156	5210.280
Czech Republic	Europe	1952	66.870	9125183	6876.140
Austria	Europe	1952	66.800	6927772	6137.076
Finland	Europe	1952	66.550	4090500	6424.519
Uruguay	Americas	1952	66.071	2252965	5716.767
Italy	Europe	1952	65.940	47666000	4931.404
Greece	Europe	1952	65.860	7733250	3530.690
Slovenia	Europe	1952	65.570	1489518	4215.042
Israel	Asia	1952	65.390	1620914	4086.522
Spain	Europe	1952	64.940	28549870	3834.035
Slovak Republic	Europe	1952	64.360	3558137	5074.659
Puerto Rico	Americas	1952	64.280	2227000	3081.960
Hungary	Europe	1952	64.030	9504000	5263.674
Japan	Asia	1952	63.030	86459025	3216.956
Paraguay	Americas	1952	62.649	1555876	1952.309
Argentina	Americas	1952	62.485	17876956	5911.315
:	:	:	:	:	:
Mali	Africa	2007	54.467	12031795	1042.5816
Kenya	Africa	2007	54.110	35610177	1463.2493
Ethiopia	Africa	2007	52.947	76511887	690.8056

country	continent	year	lifeExp	pop	gdpPercap
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
Namibia	Africa	2007	52.906	2055080	4811.0604
Tanzania	Africa	2007	52.517	38139640	1107.4822
Burkina Faso	Africa	2007	52.295	14326203	1217.0330
Equatorial Guinea	Africa	2007	51.579	551201	12154.0897
Uganda	Africa	2007	51.542	29170398	1056.3801
Botswana	Africa	2007	50.728	1639131	12569.8518
Chad	Africa	2007	50.651	10238807	1704.0637
Cameroon	Africa	2007	50.430	17696293	2042.0952
Burundi	Africa	2007	49.580	8390505	430.0707
South Africa	Africa	2007	49.339	43997828	9269.6578
Cote d'Ivoire	Africa	2007	48.328	18013409	1544.7501
Malawi	Africa	2007	48.303	13327079	759.3499
Somalia	Africa	2007	48.159	9118773	926.1411
Nigeria	Africa	2007	46.859	135031164	2013.9773
Congo, Dem. Rep.	Africa	2007	46.462	64606759	277.5519
Guinea-Bissau	Africa	2007	46.388	1472041	579.2317
Rwanda	Africa	2007	46.242	8860588	863.0885
Liberia	Africa	2007	45.678	3193942	414.5073
Central African Republic	Africa	2007	44.741	4369038	706.0165
Afghanistan	Asia	2007	43.828	31889923	974.5803
Zimbabwe	Africa	2007	43.487	12311143	469.7093
Angola	Africa	2007	42.731	12420476	4797.2313
Lesotho	Africa	2007	42.592	2012649	1569.3314
Sierra Leone	Africa	2007	42.568	6144562	862.5408
Zambia	Africa	2007	42.384	11746035	1271.2116
Mozambique	Africa	2007	42.082	19951656	823.6856
Swaziland	Africa	2007	39.613	1133066	4513.4806

summarize()

Available sub-functions:

- Center: mean(), median()
- Spread: sd(), IQR(), mad()
- Range: min(), max(), quantile()
- Position: first(), last(), nth(),
- Count: n(), n_distinct()
- Logical: any(), all()

```
In [20]: # Find average value of the lifeExp variable

summarize(countries, mean_lifeExp=mean(lifeExp))
```

A tibble: 1 × 1

mean_lifeExp
<dbl>
59.47444

```
In [21]: #Find range of the lifeExp variable

summarize(countries, range_lifeExp=max(lifeExp)-min(lifeExp))
```

A tibble: 1 × 1

range_lifeExp
<dbl>
59.004

mutate()

Add new columns on previously existing data with mutate ().

```
In [22]: # Create a new variable called 'GDP' (a product of 'gdpPercap' and 'pop')  
mutate(countries, GDP=gdpPercap*pop)
```

A tibble: 1704 × 7

country	continent	year	lifeExp	pop	gdpPercap	GDP
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>	<dbl>
Afghanistan	Asia	1952	28.801	8425333	779.4453	6567086330
Afghanistan	Asia	1957	30.332	9240934	820.8530	7585448670
Afghanistan	Asia	1962	31.997	10267083	853.1007	8758855797
Afghanistan	Asia	1967	34.020	11537966	836.1971	9648014150
Afghanistan	Asia	1972	36.088	13079460	739.9811	9678553274
Afghanistan	Asia	1977	38.438	14880372	786.1134	11697659231
Afghanistan	Asia	1982	39.854	12881816	978.0114	12598563401
Afghanistan	Asia	1987	40.822	13867957	852.3959	11820990309
Afghanistan	Asia	1992	41.674	16317921	649.3414	10595901589
Afghanistan	Asia	1997	41.763	22227415	635.3414	14121995875
Afghanistan	Asia	2002	42.129	25268405	726.7341	18363410424
Afghanistan	Asia	2007	43.828	31889923	974.5803	31079291949
Albania	Europe	1952	55.230	1282697	1601.0561	2053669902
Albania	Europe	1957	59.280	1476505	1942.2842	2867792398
Albania	Europe	1962	64.820	1728137	2312.8890	3996988985
Albania	Europe	1967	66.220	1984060	2760.1969	5476396323
Albania	Europe	1972	67.690	2263554	3313.4222	7500110047
Albania	Europe	1977	68.930	2509048	3533.0039	8864476394
Albania	Europe	1982	70.420	2780097	3630.8807	10094200603
Albania	Europe	1987	72.000	3075321	3738.9327	11498418358
Albania	Europe	1992	71.581	3326498	2497.4379	8307722183
Albania	Europe	1997	72.950	3428038	3193.0546	10945912519
Albania	Europe	2002	75.651	3508512	4604.2117	16153932130
Albania	Europe	2007	76.423	3600523	5937.0295	21376411360
Algeria	Africa	1952	43.077	9279525	2449.0082	22725632678
Algeria	Africa	1957	45.685	10270856	3013.9760	30956113720
Algeria	Africa	1962	48.303	11000948	2550.8169	28061403854
Algeria	Africa	1967	51.407	12760499	3246.9918	41433235247
Algeria	Africa	1972	54.518	14760787	4182.6638	61739408943
Algeria	Africa	1977	58.014	17152804	4910.4168	84227416174
:	:	:	:	:	:	:
Yemen, Rep.	Asia	1982	49.113	9657618	1977.5570	19098490176
Yemen, Rep.	Asia	1987	52.922	11219340	1971.7415	22121638707
Yemen, Rep.	Asia	1992	55.599	13367997	1879.4967	25125105886

country	continent	year	lifeExp	pop	gdpPercap	GDP
<fct>	<fct>	<int>	<dbl>	<int>	<dbl>	<dbl>
Yemen, Rep.	Asia	1997	58.020	15826497	2117.4845	33512362498
Yemen, Rep.	Asia	2002	60.308	18701257	2234.8208	41793958635
Yemen, Rep.	Asia	2007	62.698	22211743	2280.7699	50659874994
Zambia	Africa	1952	42.038	2672000	1147.3888	3065822956
Zambia	Africa	1957	44.077	3016000	1311.9568	3956861606
Zambia	Africa	1962	46.023	3421000	1452.7258	4969774845
Zambia	Africa	1967	47.768	3900000	1777.0773	6930601540
Zambia	Africa	1972	50.107	4506497	1773.4983	7992264611
Zambia	Africa	1977	51.386	5216550	1588.6883	8287471946
Zambia	Africa	1982	51.821	6100407	1408.6786	8593512579
Zambia	Africa	1987	50.821	7272406	1213.3151	8823720129
Zambia	Africa	1992	46.100	8381163	1210.8846	10148621483
Zambia	Africa	1997	40.238	9417789	1071.3538	10089784202
Zambia	Africa	2002	39.193	10595811	1071.6139	11354618752
Zambia	Africa	2007	42.384	11746035	1271.2116	14931695864
Zimbabwe	Africa	1952	48.451	3080907	406.8841	1253572117
Zimbabwe	Africa	1957	50.469	3646340	518.7643	1891590901
Zimbabwe	Africa	1962	52.358	4277736	527.2722	2255531194
Zimbabwe	Africa	1967	53.995	4995432	569.7951	2846372532
Zimbabwe	Africa	1972	55.635	5861135	799.3622	4685169626
Zimbabwe	Africa	1977	57.674	6642107	685.5877	4553746742
Zimbabwe	Africa	1982	60.363	7636524	788.8550	6024110454
Zimbabwe	Africa	1987	62.351	9216418	706.1573	6508240905
Zimbabwe	Africa	1992	60.377	10704340	693.4208	7422611852
Zimbabwe	Africa	1997	46.809	11404948	792.4500	9037850590
Zimbabwe	Africa	2002	39.989	11926563	672.0386	8015110972
Zimbabwe	Africa	2007	43.487	12311143	469.7093	5782658337

Piping with Multiple Functions

What if you want to perform multiple functions in R? Use Pipe operator (`%>%`) in the dplyr package. It allows you to perform multiple functions without using nested parentheses.

This is how piping looks like:

DataFrame%>%

```
#function to execute first %>%
#function to execute second %>%
#function to execute third
```

Example of Piping

Finding average life expectancy in each continent in 2007

- Select country, continent, year, lifeExp
- Filter observations from 2007
- Split data by continents
- Summarize the mean life expectancy within each continent
- Round the average life expectancy to 2 decimal points

```
In [24]: # Answer

countries %>%
  select(country, continent, year, lifeExp) %>% #select country, continent, year, lifeExp
  filter(year==2007) %>% #filter observations from 2007
  group_by(continent) %>% # split data by continents
  summarise(mean_lifeExp=mean(lifeExp)) %>% #summarize the mean life expectancy within each continent
  mutate(mean_lifeExp=round(mean_lifeExp,2)) # Use mutate to modify existing variable, round mean weight to 2 decimal points

`summarise()` ungrouping output (override with `.groups` argument)
```

A tibble: 5 × 2

continent	mean_lifeExp
<fct>	<dbl>
Africa	54.81
Americas	73.61
Asia	70.73
Europe	77.65
Oceania	80.72

Exercise #2

Finding max, min, and average of GDP (in millions USD) of European countries in each year for all the years before 2000

- Filter observations of European countries before 2000
- Use mutate to create a GDP variable (measured in millions USD)
- Group by year
- Summarize the min, mean, max, and range of GDP

In [25]: *# Answers:*

```
countries %>%
  filter(continent=="Europe", year<2000) %>%
  mutate(GDP=gdpPercap*pop/1000000) %>%
  group_by(year) %>%
  summarise(min_GDP=min(GDP),
            mean_GDP=mean(GDP),
            max_GDP=max(GDP),
            range=max(GDP)-min(GDP))
```

`summarise()` ungrouping output (override with `.groups` argument)

A tibble: 10 × 5

year	min_GDP	mean_GDP	max_GDP	range
<int>	<dbl>	<dbl>	<dbl>	<dbl>
1952	1075.342	84971.34	503266.6	502191.3
1957	1526.277	109989.51	723530.0	722003.7
1962	1884.278	138984.69	951416.2	949531.9
1967	2646.344	173366.64	1126100.6	1123454.3
1972	3306.140	218691.46	1418181.2	1414875.1
1977	4359.923	255367.52	1603305.8	1598945.9
1982	5445.018	279484.08	1725846.0	1720401.0
1987	6587.462	316507.47	1914915.6	1908328.1
1992	4353.423	342703.25	2136268.2	2131914.7
1997	4478.414	383606.93	2278996.2	2274517.8

Data Visualization with ggplot

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and geoms—visual marks that represent data points.



Data Visualization with ggplot2 :: CHEAT SHEET



Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (aesthetics) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
    stat = <STAT>, position = <POSITION>) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

required

Not required, sensible defaults supplied

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings data geom

qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5 x 5 file named "plot.png" in working directory. Matches file type to file extension.

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank()
(Useful for expanding limits)

b + geom_curve()
aes(yend = lat + 1, xend = long + 1, curvature = 1) - x, yend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path()
(lineend = "butt", linejoin = "round", linemitre = 1)
x, y, alpha, color, group, linetype, size

a + geom_polygon()
aes(group = group)
x, y, alpha, color, fill, group, linetype, size

b + geom_rect()
aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon()
aes(ymin = unemploy - 900, ymax = unemploy + 900) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline()
aes(intercept = 0, slope = 1)
b + geom_hline()
aes(yintercept = lat)
b + geom_vline()
aes(xintercept = long)

b + geom_segment()
aes(yend = lat + 1, xend = long + 1)
b + geom_spoke()
aes(angle = 1:1155, radius = 1)

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); **c2** <- ggplot(mpg)

c + geom_area()
stat = "bin"
x, y, alpha, color, fill, linetype, size

c + geom_density()
kernel = "gaussian"
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly()
x, y, alpha, color, group, linetype, size

c + geom_histogram()
binwidth = 5
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq()
aes(sample = hwy)
x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fill))
d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

continuous x, continuous y
e <- ggplot(mpg, aes(cty, hwy))

e + geom_label()
aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter()
height = 2, width = 2
x, y, alpha, color, fill, shape, size

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug()
sides = "bl"
x, y, alpha, color, linetype, size

e + geom_smooth()
method = lm
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text()
aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

discrete x, continuous y

f <- ggplot(mpg, aes(class, hwy))
f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot()
binaxis = "y", stackdir = "center"
x, y, alpha, color, fill, group

f + geom_violin()
scale = "area"
x, y, alpha, color, fill, group, linetype, size, weight

discrete x, discrete y

g <- ggplot(diamonds, aes(cut, color))
g + geom_count()
x, y, alpha, color, fill, shape, size, stroke

THREE VARIABLES

seals\$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); **l** <- ggplot(seals, aes(long, lat))
l + geom_contour()
aes(z = z)
x, y, z, alpha, colour, group, linetype, size, weight

l + geom_raster()
aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE
x, y, alpha, fill

l + geom_tile()
aes(fill = z), x, y, alpha, color, fill, linetype, size, width

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))
h + geom_bin2d()
binwidth = c(0.25, 500)
x, y, alpha, color, fill, linetype, size, weight

h + geom_density2d()
x, y, alpha, colour, group, linetype, size

h + geom_hex()
x, y, alpha, colour, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))
i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step()
direction = "hv"
x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))
j + geom_crossbar()
fatten = 2
x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar()
x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))
k + geom_map()
aes(map_id = state), map = map, expand_limits(x = map\$long, y = map\$lat), map_id, alpha, color, fill, linetype, size

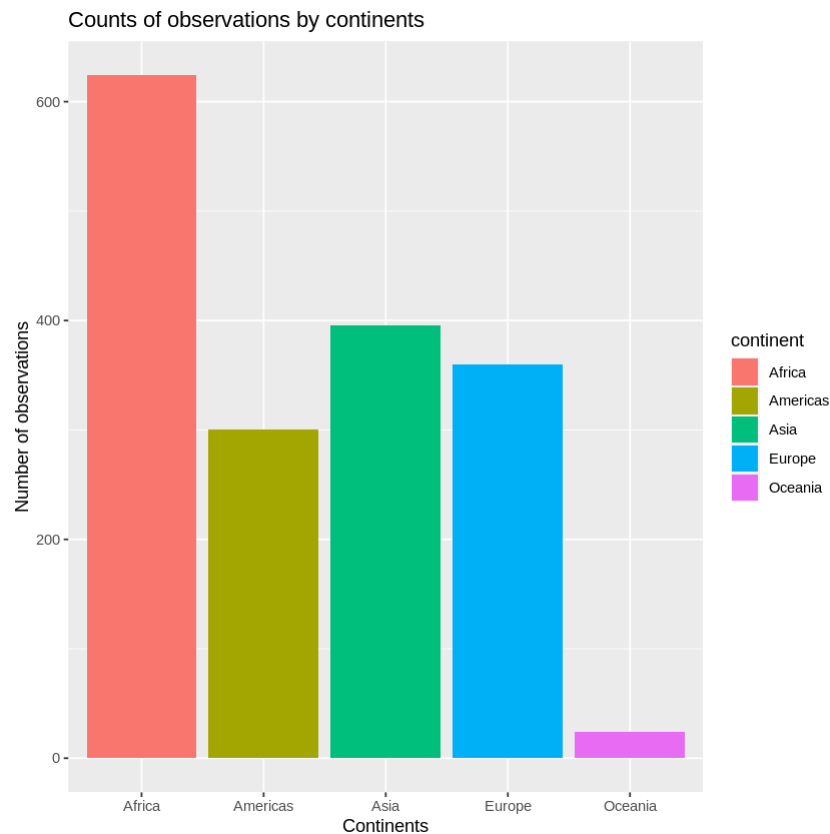


Barchart

Shows the distribution of a categorical variable

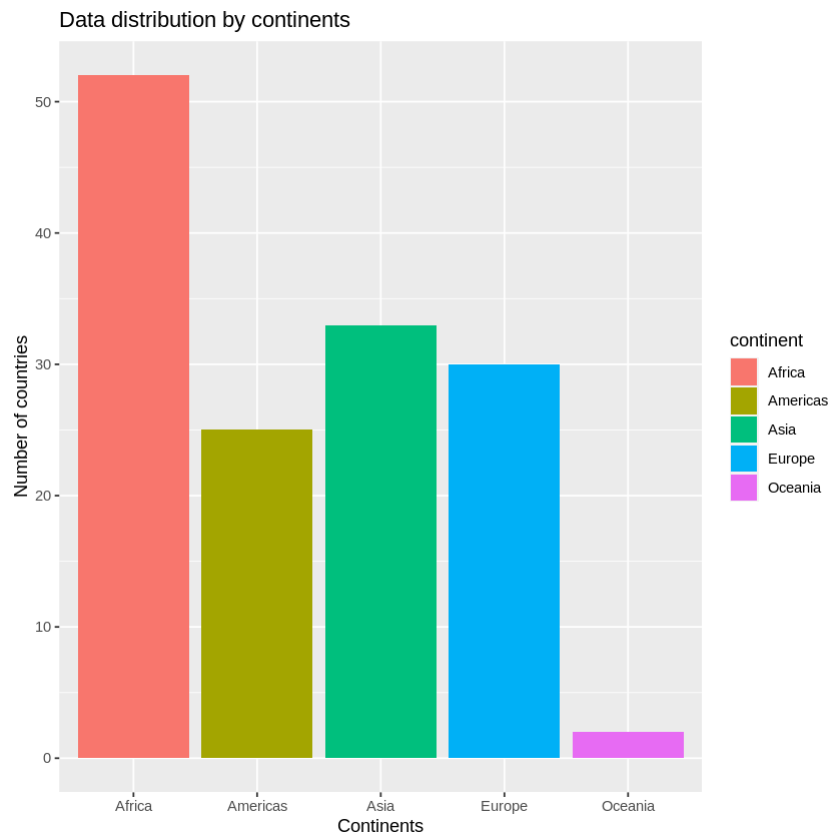
```
In [26]: # A barchart showing the number of observations in each continent

ggplot(data=countries, aes(x=continent, fill=continent))+
  geom_bar()+
  labs(title="Counts of observations by continents", x="Continents", y="Number of observations")
```



```
In [27]: # Modify count to measure the number of countries by dividing by 12

ggplot(data=countries, aes(x=continent, fill=continent))+
  geom_bar(aes(y=..count../12))+
  labs(title="Data distribution by continents", x="Continents", y="Number of
countries")
```

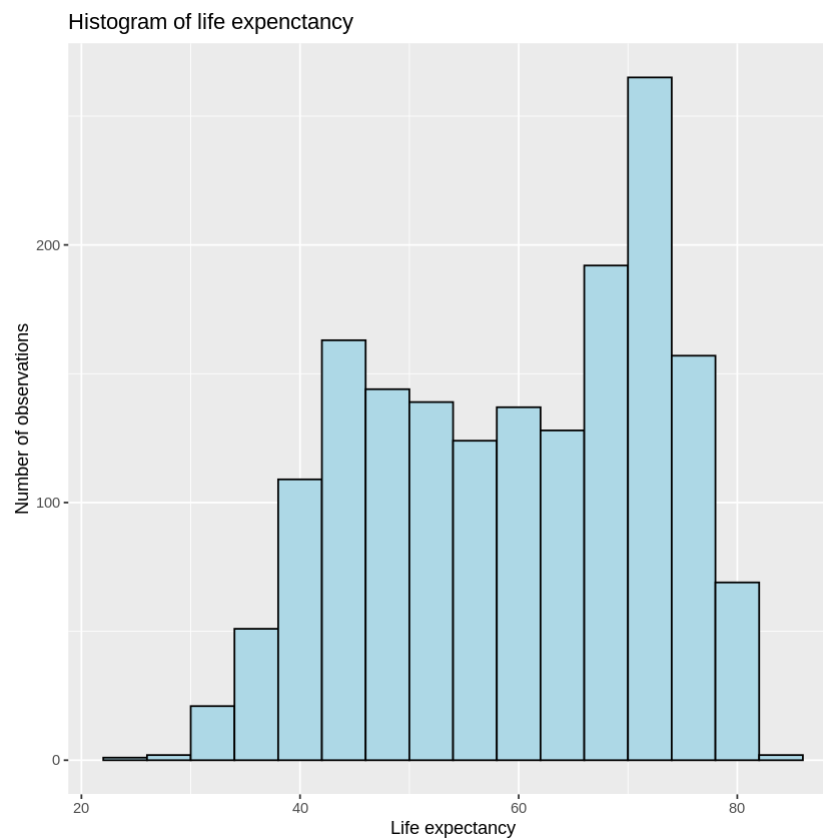


Histogram and density plots

Visualizes the distribution of one quantitative variable

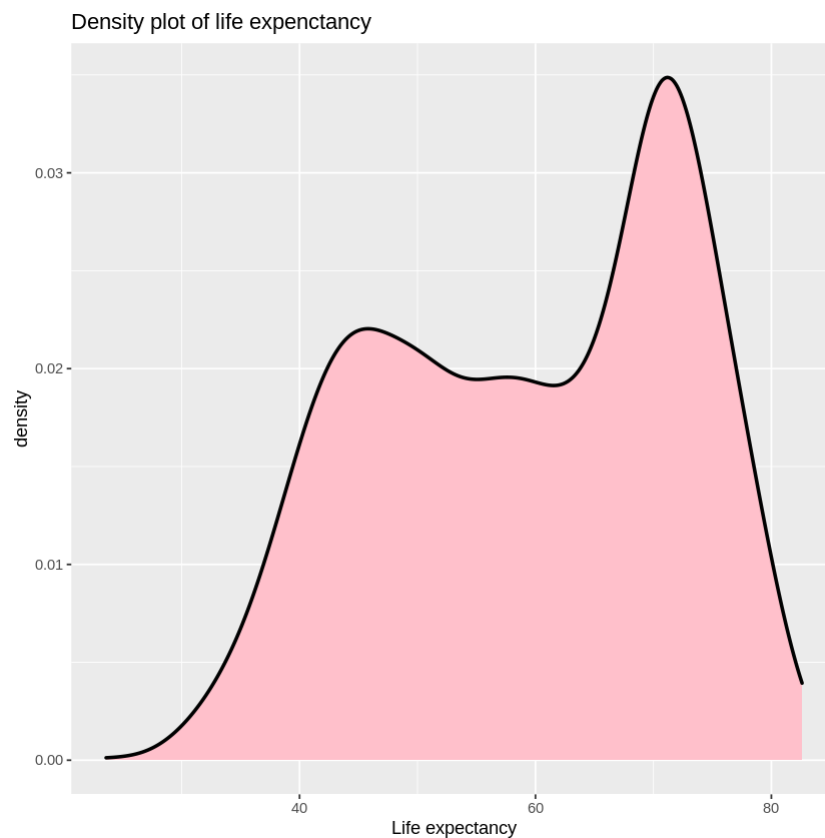
```
In [28]: # Visualize the distribution of life expectancy

ggplot(data=countries, aes(x=lifeExp)) +
  geom_histogram(binwidth=4,color="black", fill="lightblue") +
  labs(title="Histogram of life expectancy ", x="Life expectancy", y="Number of observations")
```



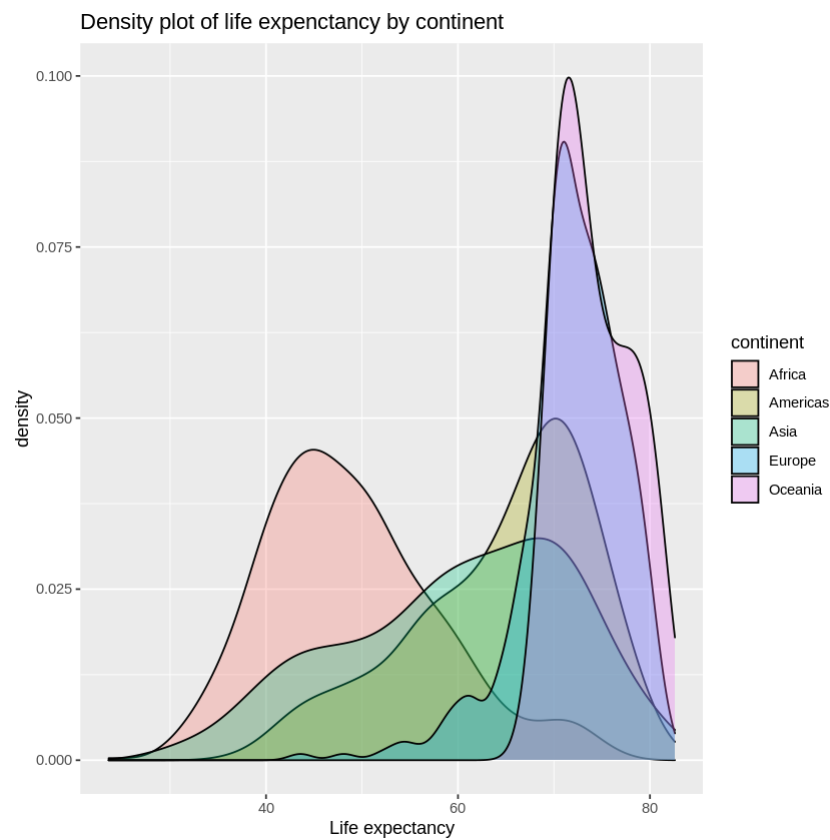
In [29]: *# Alternative to histograms - density plots*

```
ggplot(data=countries, aes(x=lifeExp)) +  
  geom_density(size=1, fill="pink") +  
  labs(title="Density plot of life expenctancy ", x="Life expectancy")
```



```
In [30]: # Contrasting lifeExp distributionns of continents using density plots

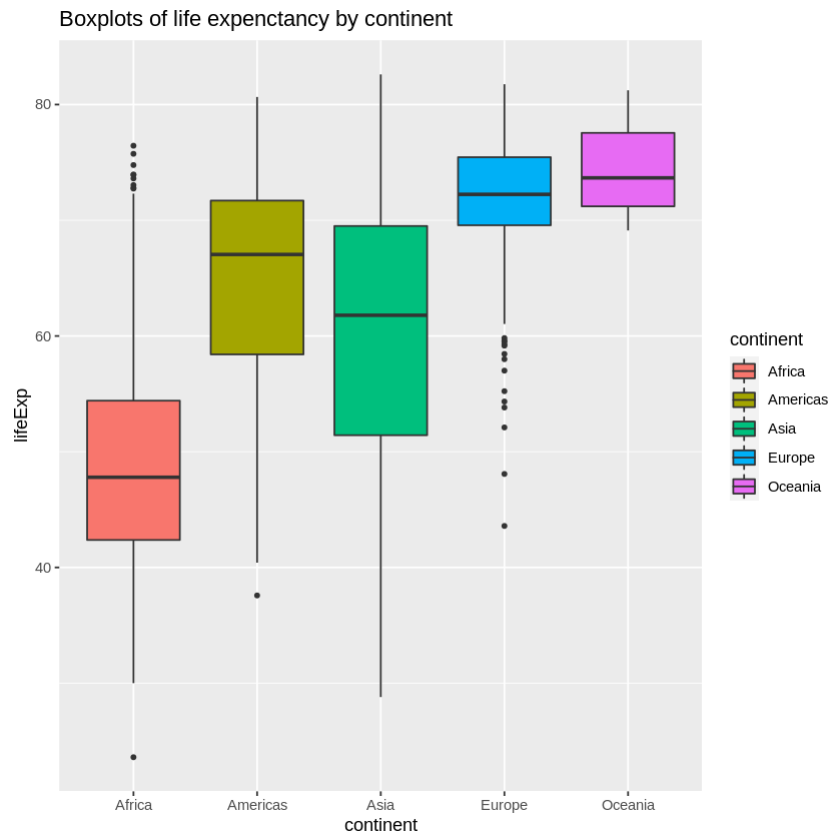
ggplot(data=countries, aes(x=lifeExp, fill=continent)) +
  geom_density(alpha=0.3) + #alpha measures color saturation (0 to 1)
  labs(title="Density plot of life expectancy by continent ", x="Life expectancy")
```



Boxplots

Boxplots are used to compare distributions of one quantitative variable across multiple categories (a visualization alternative to density plots)

```
In [31]: # Comparing distribution of lifeExp variable by continent:  
  
ggplot(data=countries, aes(x=continent, y=lifeExp, fill=continent))+  
  geom_boxplot(outlier.size=1)+  
  labs(title="Boxplots of life expenctancy by continent")
```

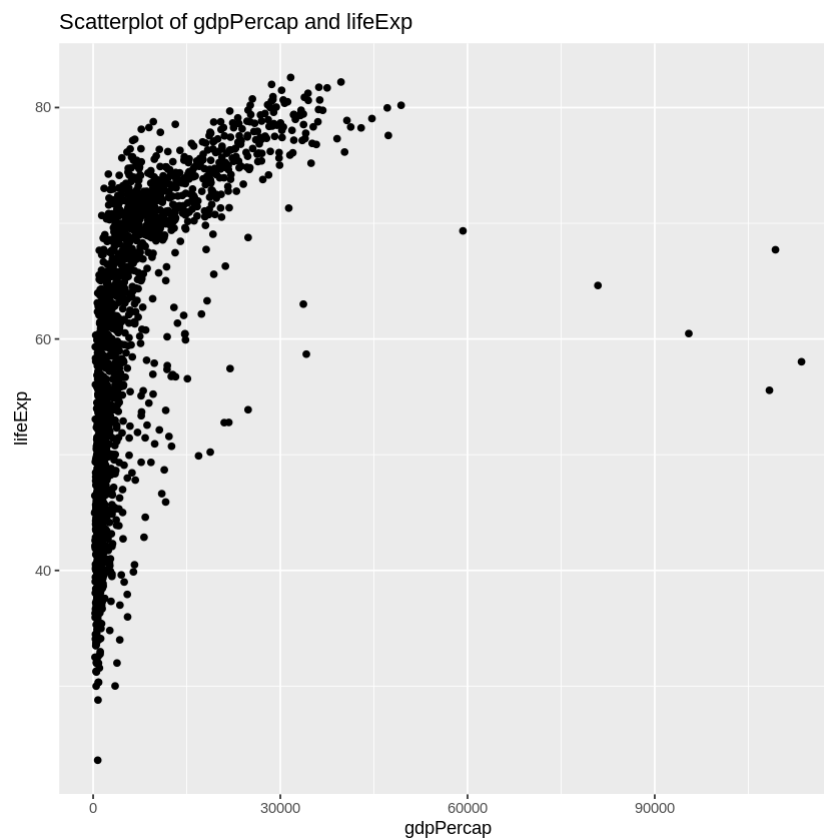


Scatterplot

Scatterplots show the relationship between two quantitative variables

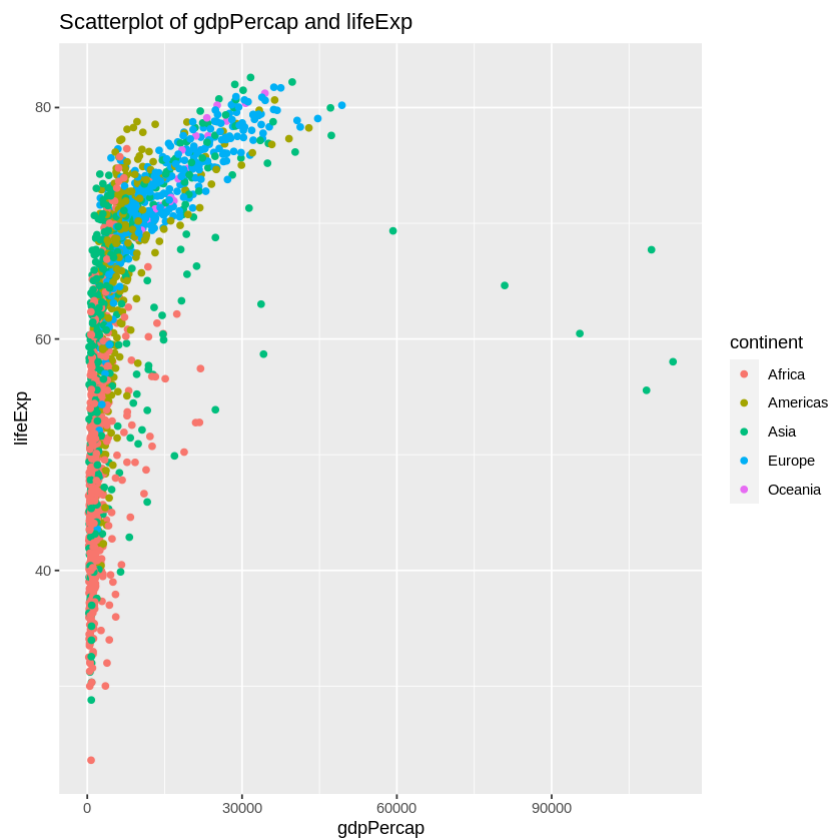
In [32]: *# Make a scatterplot with gdpPercap and lifeExp variables*

```
ggplot(data=countries, aes(x=gdpPercap, y=lifeExp))+  
  geom_point()+  
  labs(title="Scatterplot of gdpPercap and lifeExp")
```



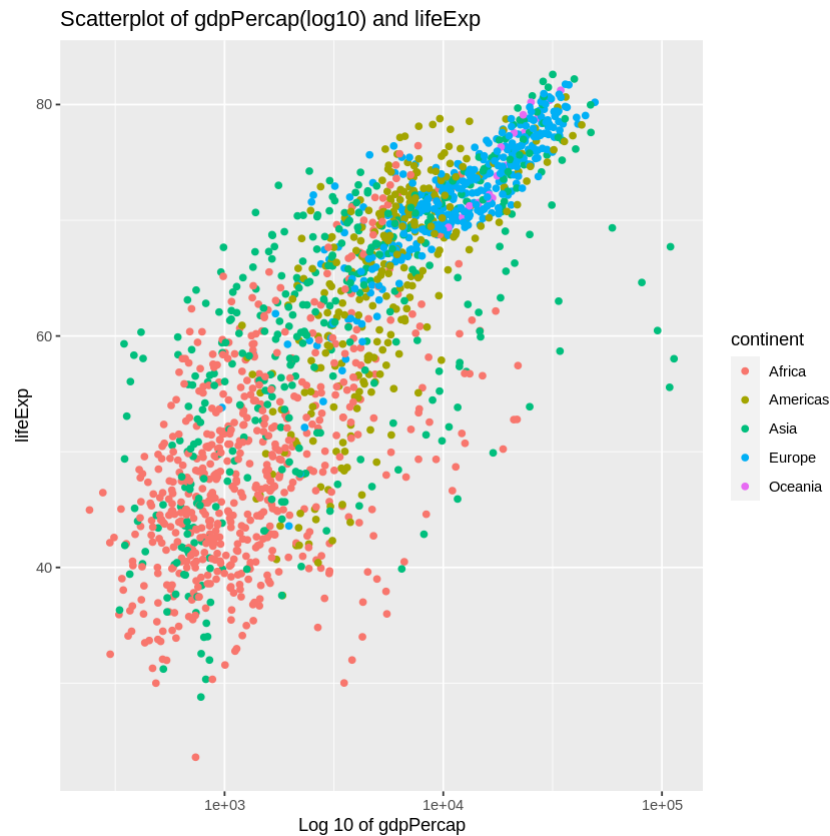
```
In [33]: # Add color to represent different continents

ggplot(data=countries, aes(x=gdpPercap, y=lifeExp))+
  geom_point(aes(color=continent))+
  labs(title="Scatterplot of gdpPercap and lifeExp")
```



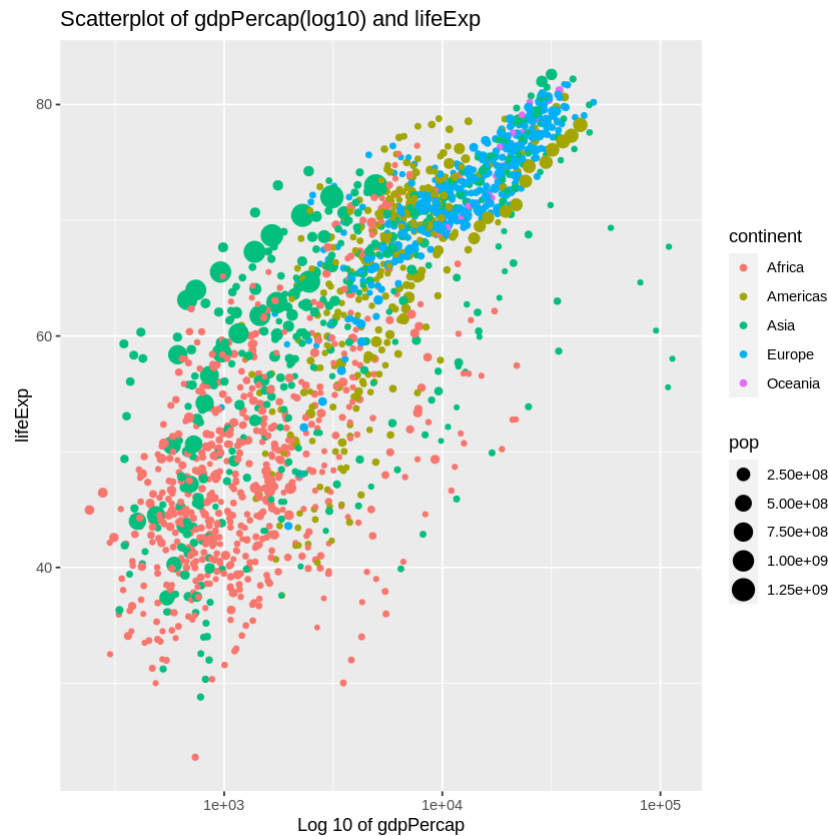
```
In [34]: # Transform the gdpPercap variable using a log10 scale

ggplot(data=countries, aes(x=gdpPercap, y=lifeExp))+
  geom_point(aes(color=continent))+
  scale_x_log10()+
  labs(title="Scatterplot of gdpPercap(log10) and lifeExp", x="Log 10 of gdp
Percap")
```



```
In [35]: # Change the size of data points to measure the size of population

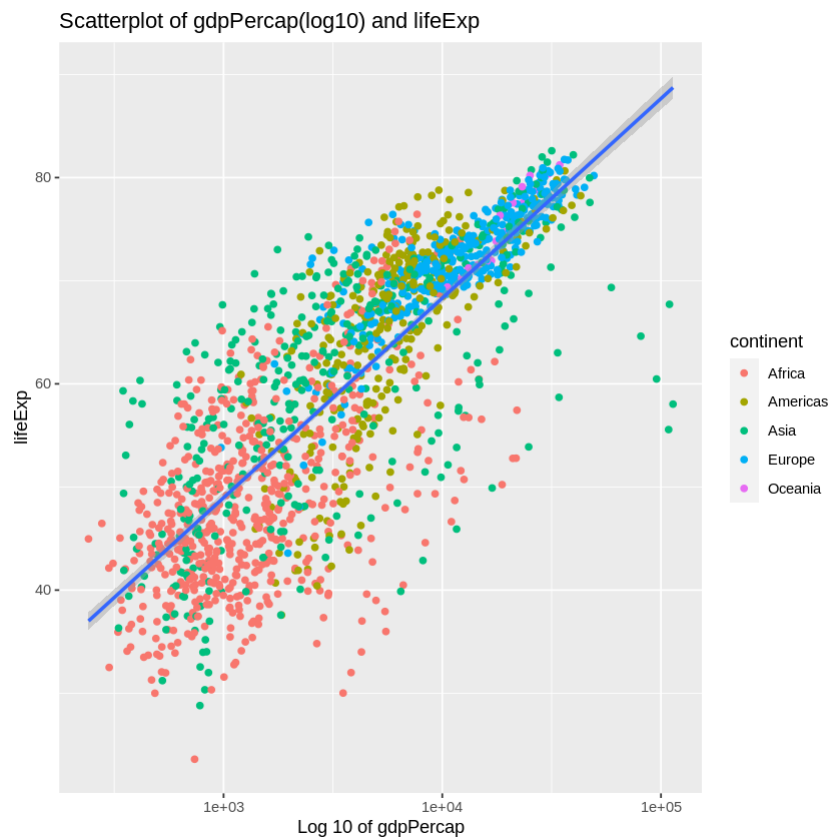
ggplot(data=countries, aes(x=gdpPercap, y=lifeExp))+
  geom_point(aes(color=continent, size=pop))+
  scale_x_log10()+
  labs(title="Scatterplot of gdpPercap(log10) and lifeExp", x="Log 10 of gdp
Percap")
```



```
In [36]: # Add a best-fit line to the scatterplot

ggplot(data=countries, aes(x=gdpPercap, y=lifeExp))+
  geom_point(aes(color=continent))+
  geom_smooth(method="lm")+
  scale_x_log10()+
  labs(title="Scatterplot of gdpPercap(log10) and lifeExp", x="Log 10 of gdp
Percap")
```

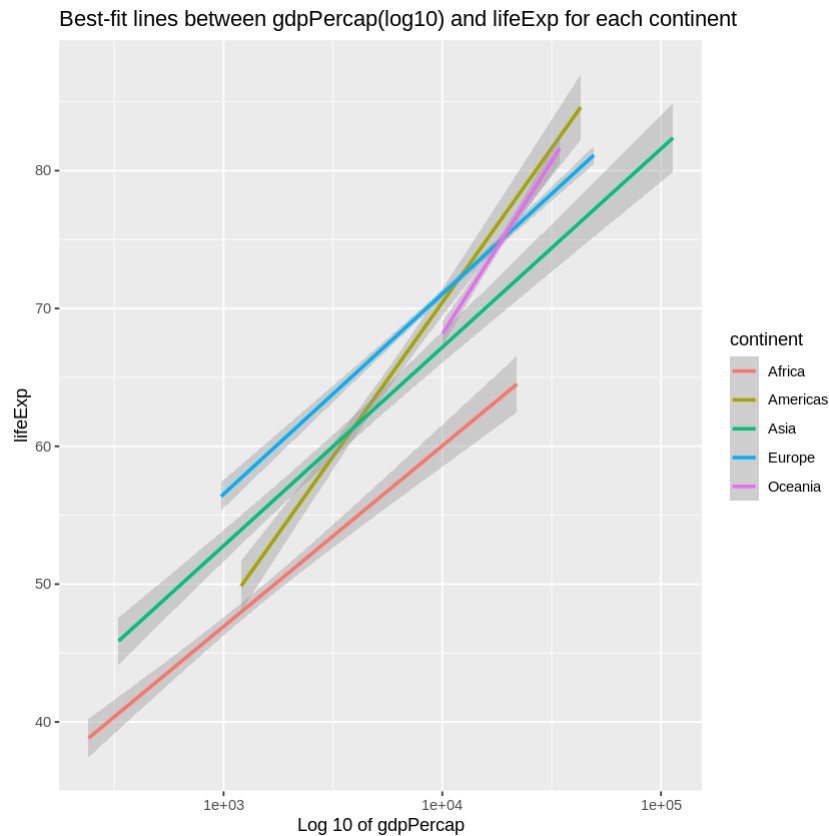
`geom_smooth()` using formula 'y ~ x'




```
In [37]: # Add a best-fit line for each continent

ggplot(data=countries, aes(x=gdpPercap, y=lifeExp))+
  geom_smooth(aes(color=continent), method="lm")+
  scale_x_log10()+
  labs(title="Best-fit lines between gdpPercap(log10) and lifeExp for each c
ontinent", x="Log 10 of gdpPercap")
```

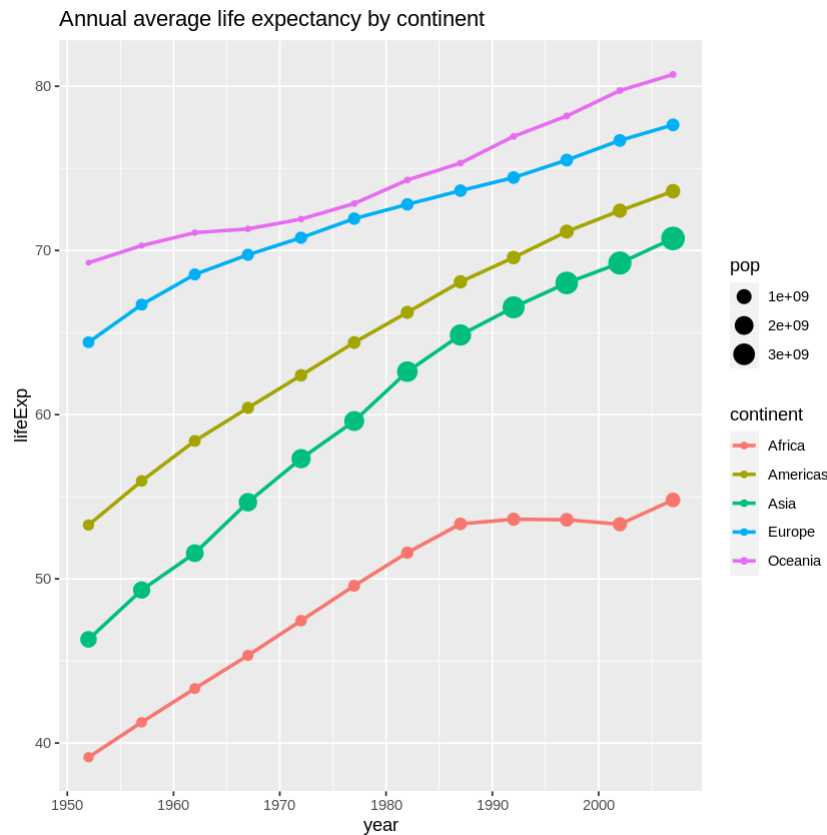
`geom_smooth()` using formula 'y ~ x'



Piping in ggplot

```
In [38]: # Visualize changes in annual average life expectancy by continent with points
#         representing the size of populations
countries %>%
  group_by(continent, year) %>%
  summarise(lifeExp=mean(lifeExp), pop=sum(pop)) %>%
  ggplot(aes(x=year, y=lifeExp, color=continent))+
  geom_line(size=1)+
  geom_point(aes(size=pop))+
  labs(title="Annual average life expectancy by continent")
```

`summarise()` regrouping output by 'continent' (override with `.groups` argument)



Exercise #3

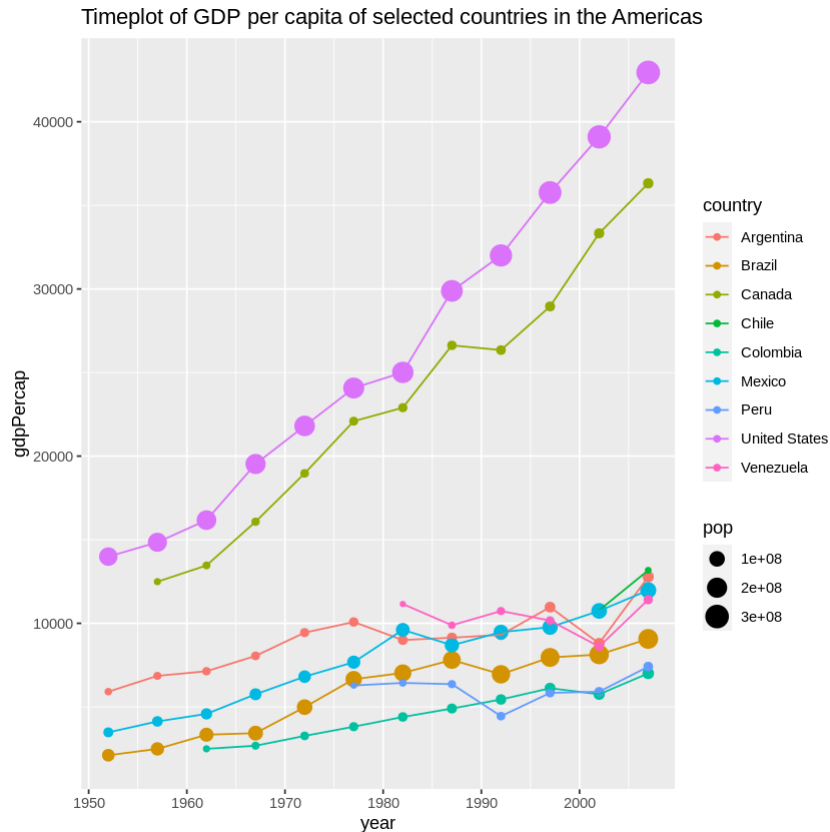
Make a timeplot of `gdpPercap` for countries in Americas with population above 15m (points should represent the size of population)

Hints:

- Use filter function to limit the selection of data to Americas, pop > 15000000
- Use ggplot, geom_point, and geom_line to make a graph

In [39]: *# Answer*

```
countries %>%
  filter(continent=="Americas",pop>15000000) %>%
  ggplot(aes(x=year, y=gdpPercap, color=country))+
  geom_point(aes(size=pop))+
  geom_line()+
  labs(title="Timeplot of GDP per capita of selected countries in the Americas")
```



Save dataframe as CSV file

In [40]: `write.csv(countries, file= "countries.csv")`

Questions?

Research Commons has a team of GAAs happy to help you analyze your data. Book a consultation online.

UBC Library Research Commons

Search[Home](#)[Workshops](#)[Consultations](#)[Calendar](#)[News](#)[Spaces and Software](#)[About the Team](#)

Consultations

All of our consultations occur online.

Graduate Student Expert

Get help with Thesis Formatting, Citation management (RefWorks, Zotero, Mendeley), Data Analysis (R, Python, SPSS, NVivo). For more personalized assistance, you can request to book a one-on-one consultation with one of our Graduate experts.

[Book a Consultation](#)

Digital Scholarship

Get in touch to learn more about digital scholarship or get help with a project, schedule a consultation. Learn more about the Digital Scholarship team.

Eka Grguric, Digital Scholarship Specialist
eka.grguric@ubc.ca

Reference

R-core@R-project.org.(2020,June 15). *mtcars*.

<https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/mtcars>

(<https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/mtcars>)