

Intro2R

Data Wrangling and Visualisation

Song, Xiao Ping

xp.song@u.nus.edu

Course materials: <https://github.com/xp-song/Intro2R>

updated 2020-09-18

Outline

About our dataset

Data preparation

Survey Overview

Survey Analysis

It's your turn!

Further applications

About our Dataset

Kaggle Machine Learning and Data Science Survey 2018

- The industry-wide survey presents the state of data science and machine learning
- We will be analysing multiple choice responses `/data/kaggle-survey-2018_mcq.csv`

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Time from S	Q1	Q1_OTHER_	Q2	Q3	Q4	Q5	Q6	Q6_OTHER_	Q7	Q7_OTHER_	Q8	Q9	Q10	Q11_Part_1	Q11_Part_2
2	Duration (in	What is your	What is your	What is your	In which cou	What is the l	Which best c	Select the tit	Select the tit	In what indu	In what indu	How many y	What is your	Does your cu	Select any ac	Select any ac
3	710	Female	-1	45-49	United State	Doctoral deg	Other	Consultant	-1	Other	0			I do not know	Analyze and	Build and/or
4	434	Male	-1	30-34	Indonesia	Bachelor,Ã	Engineering	Other	0	Manufacturi	-1	5-Oct	10-20,000	No (we do not use ML methods)		
5	718	Female	-1	30-34	United State	Master,Ã	Computer sc	Data Scientis	-1	I am a stude	-1	0-1	0-10,000	I do not know	Analyze and understand d	
6	621	Male	-1	35-39	United State	Master,Ã	Social scienc	Not employe	-1		-1					
7	731	Male	-1	22-24	India	Master,Ã	Mathematic	Data Analyst	-1	I am a stude	-1	0-1	0-10,000	I do not know		
8	1142	Male	-1	25-29	Colombia	Bachelor,Ã	Physics or as	Data Scientis	-1	Computers/T	-1	0-1		I do not wish	We are explc	Analyze and understand d
9	959	Male	-1	35-39	Chile	Doctoral deg	Information	Other	1	Academics/E	-1	Oct-15	10-20,000	No (we do not use ML methods)		
10	1758	Male	-1	18-21	India	Master,Ã	Information	Other	2	Other	1	0-1	0-10,000	We recently started using	Build and/or	
11	641	Male	-1	25-29	Turkey	Master,Ã	Engineering	Not employe	-1		-1					
12	751	Male	-1	30-34	Hungary	Master,Ã	Engineering	Software Enj	-1	Online Servic	-1	3-Apr	20-30,000	We have well established	Build and/or	
13	2028	Male	-1	22-24	Ireland	Bachelor,Ã	Information	Student	-1	I am a stude	-1	3-Apr	I do not wish	I do not know		
14	823	Male	-1	40-44	United State	Master,Ã	Engineering	Data Scientis	-1	Other	2	5-Oct	125-150,000	We recently started using	Build and/or	
15	1091	Male	-1	25-29	France	Doctoral deg	Mathematic	Student	-1	I am a stude	-1		30-40,000	We have well established	ML methods	

About our Dataset

More about Kaggle

- An online community for data science
- Owned by Google (>1 mil users in 2017)
- Users can find/publish data and analysis, and take part in data science competitions

Our analysis includes code adapted from R Notebooks created by the Kaggle users [Heads or Tails](#) and [Jose Berengueres](#)

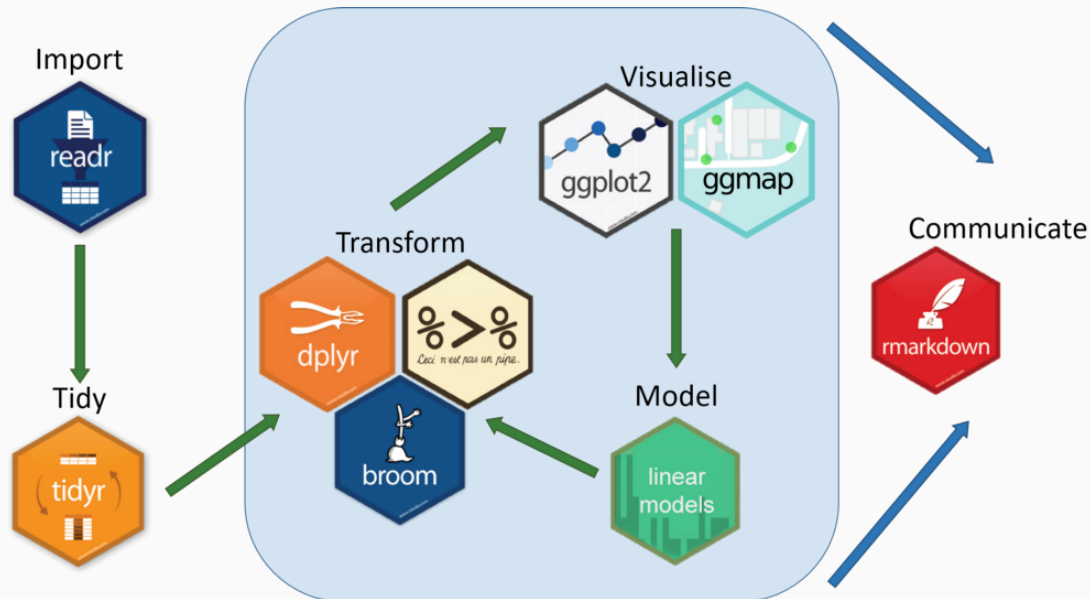
Before we begin...

Install and load packages

Before we begin...

Let's install the tidyverse collection of packages

[tidyverse](#): A collection of packages commonly used for data analyses



Example workflow (medium.com)

Before we begin...

Let's install the tidyverse collection of packages

```
install.packages("tidyverse", dependencies = TRUE) # don't forget quotes
```

- Type `n` if you get the following prompt:

```
Do you want to install from sources the package which needs compilation?
```

- Click 'Yes' if you are asked to restart R

Load these packages into R

```
library(tidyverse) # no need quotes
```

Before we begin...

"Tidy" data

- Tabular data (2D)
- Each variable is a column & each observation is a row
- Can be in long or wide format

country	year	key	value
Afghanistan	1999	infected	135
Afghanistan	1999	population	19839494
Afghanistan	2020	infected	384
Afghanistan	2020	population	21739203
Australia	1999	infected	34
Australia	1999	population	23423534
Australia	2020	infected	45
Australia	2020	population	23346436
Belgium	1999	infected	272
Belgium	1999	population	49273820
Belgium	2020	infected	274
Belgium	2020	population	48928472

country	year	infected	population
Afghanistan	1999	135	19839494
Afghanistan	2020	384	21739203
Australia	1999	34	23423534
Australia	2020	45	23346436
Belgium	1999	272	49273820
Belgium	1999	274	48928472

Outline

About our dataset

Data preparation

Survey Overview

Survey Analysis

It's your turn!

Further applications

Load data

Import tabular data as *tibbles* using `readr::read_csv()`

Tibbles are dataframes with stricter rules that avoid hassle/errors often associated with conventional dataframes.

```
multi <- read_csv("data/kaggle-survey-2018_mcq.csv", skip = 1)
```

```
head(multi)
```

Load data

Let's compare `read_csv()` **with** `read.csv()` **in base R**

```
multi2 <- read.csv("data/kaggle-survey-2018_mcq.csv", skip = 1)
head(multi2)
```

Examine data

Examine column names

```
head(colnames(multi))
```

```
## [1] "Duration (in seconds)"
## [2] "What is your gender? - Selected Choice"
## [3] "What is your gender? - Prefer to self-describe - Text"
## [4] "What is your age (# years)?"
## [5] "In which country do you currently reside?"
## [6] "What is the highest level of formal education that you have attained or plan to attain within the next 12 months?"
```

- Analysing colnames as entire sentences is not very feasible at scale (we'll abbreviate the colnames later)

Print the first column by name

```
multi$`Duration (in seconds)`
```

- Have to wrap colname with backticks (because of white spaces)

Convert data

The pipe operator `%>%`

Frequently used to manipulate data in stages/sequence

E.g.:

```
round(exp(diff(log(x))), 1) #using nested brackets
```

```
x %>% log() %>% #using the pipe operator  
  diff() %>%  
  exp() %>%  
  round(1)
```

Convert data

Multiple choice questions have categorical answers with discrete levels—i.e. Factors!

Convert columns with *character* data into *factors* using `mutate_if()`

```
multi <- multi %>%  
  mutate_if(is.character, as.factor)
```

Outline

About our dataset

Data preparation

Survey Overview

Survey Analysis

It's your turn!

Further applications

Sample Size

How many respondents are there?

```
nrow(multi)
```

```
## [1] 23859
```


Survey duration

Abbreviate the colname `Duration (in seconds)` **to** `duration` **using** `rename()`

```
multi <- multi %>%  
  rename(duration = `Duration (in seconds)`)
```

Change the units from seconds to minutes using `mutate()`

```
multi <- multi %>%  
  mutate(duration = duration/60) # override the colname
```

Print out first few rows of `multi$duration`

```
head(multi$duration)
```

```
## [1] 11.833333  7.233333 11.966667 10.350000 12.183333 19.033333
```

Survey duration

Plot a histogram using the `ggplot2::ggplot()` function

Three basic steps:

1. Provide *data*
2. Assign your data *variables* to *aesthetics*
3. Assign the graphical *primitives*

```
multi %>%  
  ggplot(aes(duration)) +  
  geom_histogram()
```

Survey duration

Plot a histogram using the `ggplot2::ggplot()` function

Three basic steps:

1. Provide *data*
2. Assign your data *variables* to *aesthetics*
3. Assign the graphical *primitives*

```
multi %>%  
  ggplot(aes(duration)) +  
  geom_histogram() +  
  geom_vline(xintercept = median(multi$duration))
```

Survey duration

Plot a histogram using the `ggplot2::ggplot()` function

Three basic steps:

1. Provide *data*
2. Assign your data *variables* to *aesthetics*
3. Assign the graphical *primitives*

```
multi %>%  
  ggplot(aes(duration)) +  
  geom_histogram() +  
  geom_vline(xintercept = median(multi$duration)) +  
  scale_x_log10()
```

Survey duration

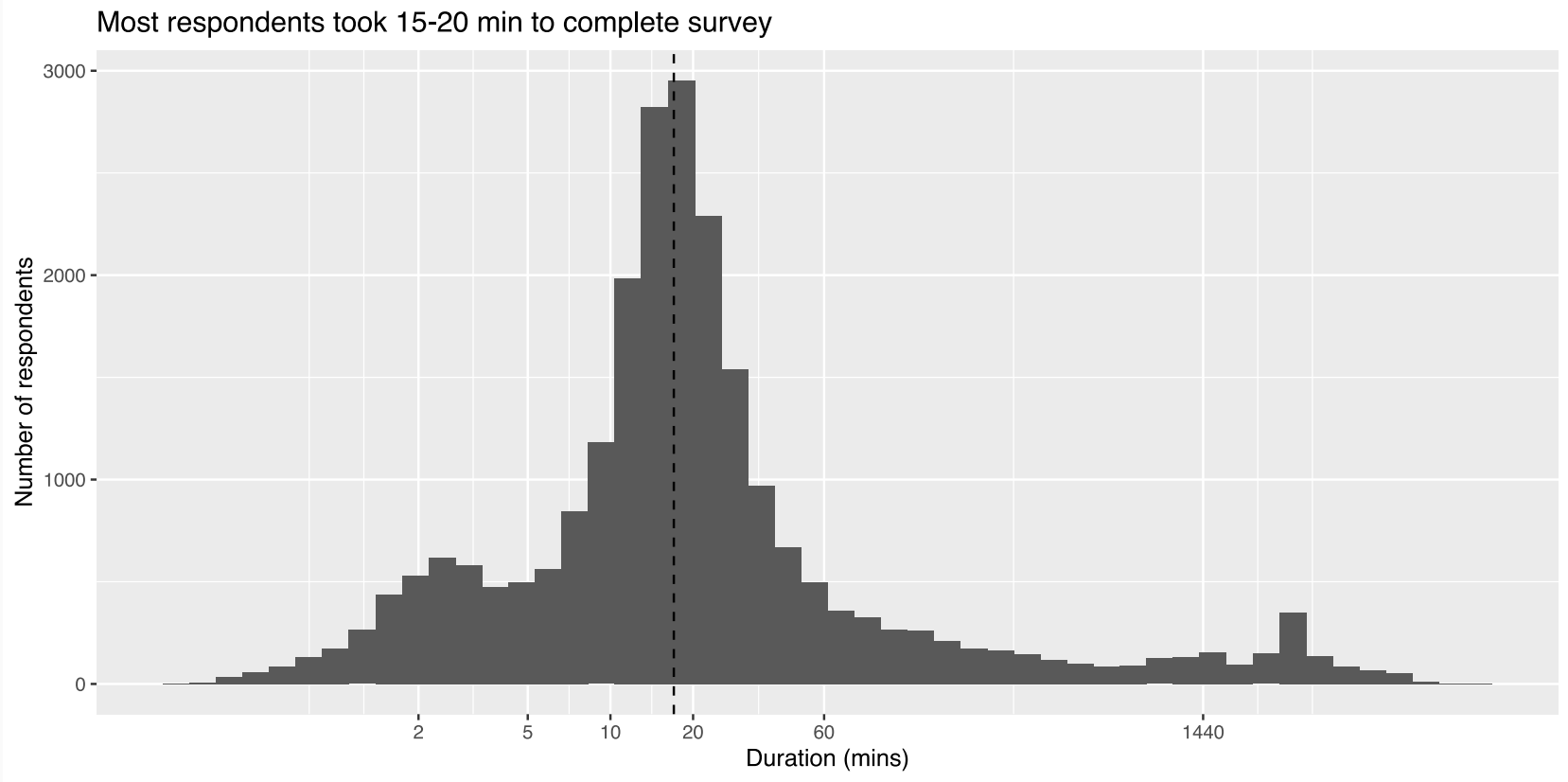
Plot a histogram using the `ggplot2::ggplot()` function

Three basic steps:

1. Provide *data*
2. Assign your data *variables* to *aesthetics*
3. Assign the graphical *primitives*

```
multi %>%  
  ggplot(aes(duration)) +  
  geom_histogram(bins = 50) +  
  geom_vline(xintercept = median(multi$duration), linetype = 2) +  
  
  scale_x_log10(breaks = c(2, 5, 10, 20, 60, 1440)) + #address extreme x-values  
  
  #customisation  
  labs(x = "Duration (mins)", y = "Number of respondents") + #change axis labels  
  ggtitle("Most respondents took 15-20 min to complete survey") #add figure title
```

Survey duration



Note: The dashed line denotes the median survey duration. The x-axis has been transformed to a logarithmic scale.

Outline

About our dataset

Data preparation

Survey Overview

Survey Analysis

It's your turn!

Further applications

Geographical distribution

Abbreviate the colname In which country do you currently reside? **to** country

```
multi <- multi %>%  
  rename(country = `In which country do you currently reside?`)
```

Examine the column

```
head(multi$country)
```

```
## [1] "United States of America" "Indonesia"  
## [3] "United States of America" "United States of America"  
## [5] "India"                    "Colombia"
```


Geographical distribution

Summarise the number of respondents per country

```
ctry_n <- multi %>%  
  count(country)
```

Geographical distribution

Summarise the number of respondents per country

Show entries

Search:

country	n
Argentina	119
Australia	330
Austria	62
Bangladesh	107
Belarus	91
Belgium	111

Showing 1 to 6 of 58 entries

Previous

1

2

3

4

5

...

10

Next

Geographical distribution

Remove rows with certain answers in our summary table `ctry_n`

```
ctry_n <- ctry_n %>%  
  filter(!(country %in% c("Other", "I do not wish to disclose my location")))
```

```
ctry_n
```

Geographical distribution

Map the country name to the ISO3 country code using the function `countrycode()`

- `install.packages("countrycode")`
- Add data as a new column named `iso3` using `mutate()`

```
ctry_n <- ctry_n %>%  
  mutate(iso3 = countrycode(country, origin = "country.name", destination = "iso3c"))
```

```
ctry_n
```

Geographical distribution

Dealing with duplicates

Let's check if the number of country names & country codes match

```
length(unique(ctry_n$country))
```

```
## [1] 56
```

```
length(unique(ctry_n$iso3))
```

```
## [1] 55
```

Check which elements are duplicated with `duplicated()`

```
duplicated(ctry_n$iso3)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [34] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [45] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [56] FALSE
```

Geographical distribution

Dealing with duplicates

Subset all rows with duplicates

```
ctry_n[duplicated(ctry_n$iso3) | duplicated(ctry_n$iso3, fromLast=TRUE),]
```

Geographical distribution

Dealing with duplicates

Group the dataframe by `iso3`, then add up the no. of respondents `n`

```
ctry_n <- ctry_n %>%  
  group_by(iso3) %>%  
  summarise(country = first(country), #get the first value for country name  
            n = sum(n)) #sum up duplicates
```

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

```
ctry_n
```

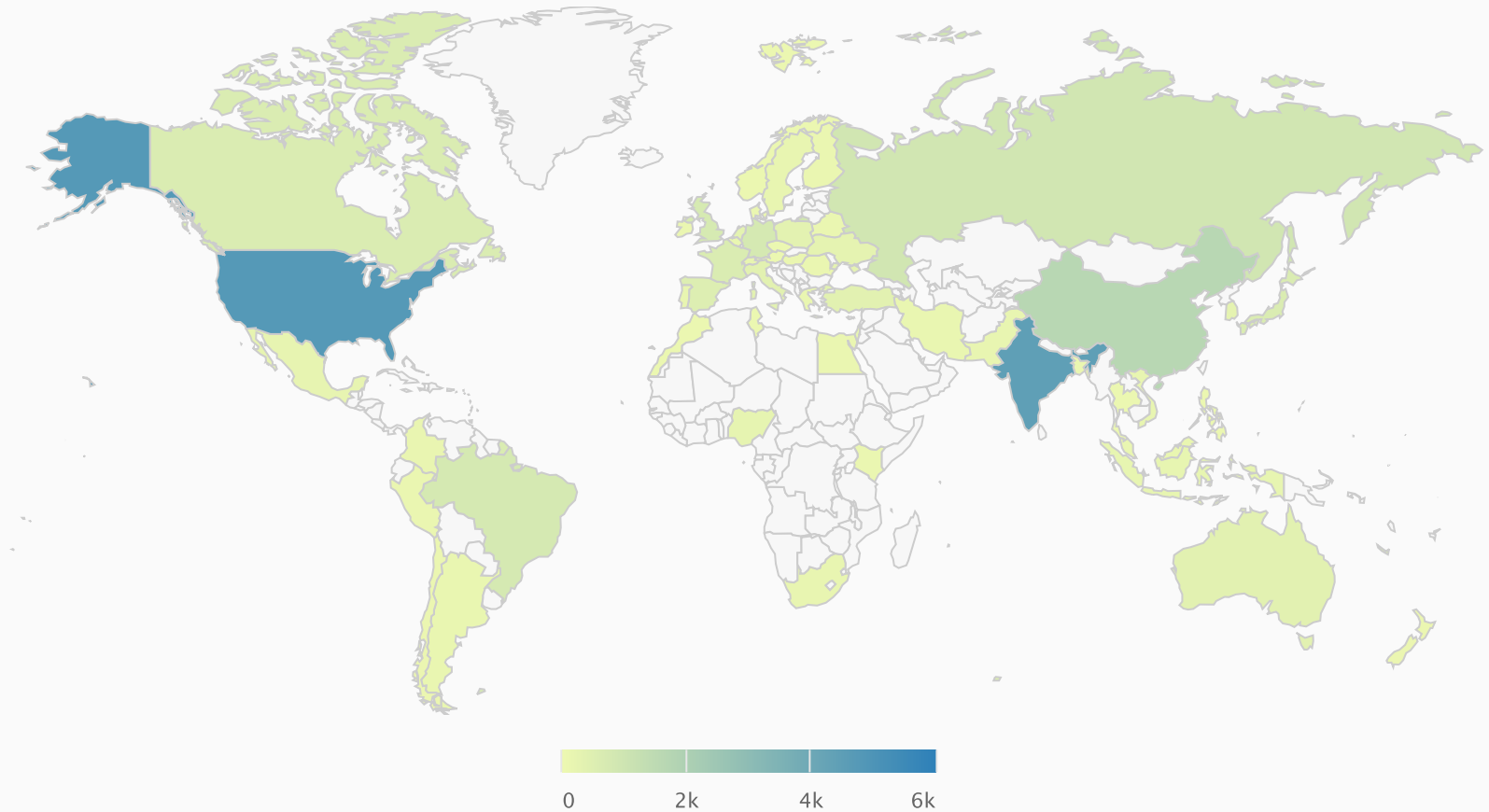
Geographical distribution

Plot `ctry_n` **as an interactive map using** `highcharter::highchart()`

```
highchart() %>%  
  hc_add_series_map(worldgeojson, #map data to add  
    ctry_n, #our data  
    value = 'n', #colname of interest in ctry_n  
    joinBy = 'iso3') %>% #both datasets have this colname (country code)  
  
  #customisation  
  hc_title(text = 'Geographical distribution of survey respondents') %>%  
  hc_colorAxis(minColor = "#edf8b1", maxColor = "#2c7fb8") %>%  
  hc_tooltip(useHTML = TRUE, headerFormat = "", pointFormat = "{point.country}: {point.n} respondents")
```


Geographical distribution

Geographical distribution of survey respondents



Survey Analysis: Country-level Data

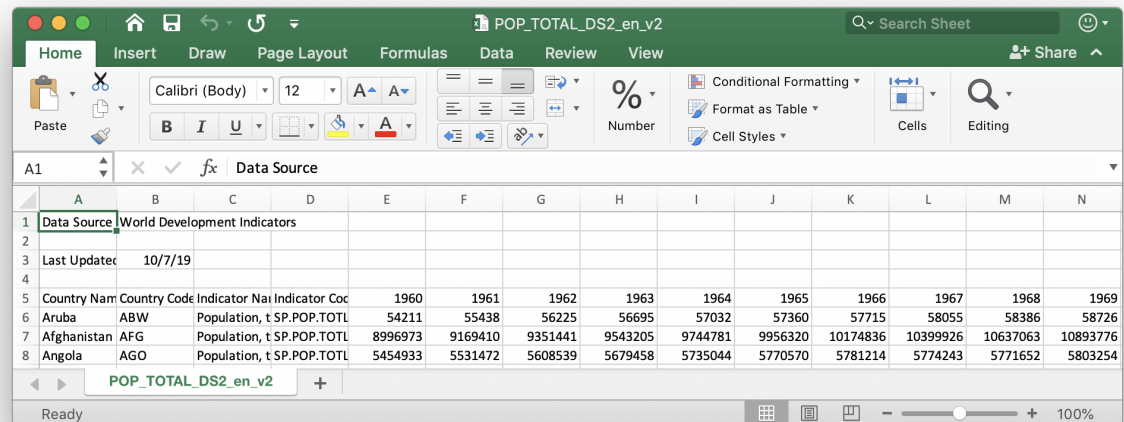
Country-level Data

Datasets

Total population count (The World Bank)

```
pops <- read_csv("data/POP_TOTAL_DS2_en_v2.csv", skip = 4)

pops <- pops %>%
  rename(country = "Country Name", population = "2018") %>%
  select(country, population) %>% #we are only interested in these 2 columns
  mutate(iso3 = countrycode(country, origin = "country.name",
                             destination = "iso3c"))
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Data Source	World Development Indicators												
2														
3	Last Updated	10/7/19												
4														
5	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969
6	Aruba	ABW	Population, total	SP.POP.TOTL	54211	55438	56225	56695	57032	57360	57715	58055	58386	58726
7	Afghanistan	AFG	Population, total	SP.POP.TOTL	8996973	9169410	9351441	9543205	9744781	9956320	10174836	10399926	10637063	10893776
8	Angola	AGO	Population, total	SP.POP.TOTL	5454933	5531472	5608539	5679458	5735044	5770570	5781214	5774243	5771652	5803254

Country-level Data

Datasets

Total population count ([The World Bank](#))

pops

Country-level Data

Datasets

Total population count ([The World Bank](#))

- Check for NA values using `is.na()`

```
pops[is.na(pops$iso3),]
```

Country-level Data

Datasets

Total population count ([The World Bank](#))

- **Remove rows with NA values**

```
pops <- pops[!is.na(pops$iso3),]
```

```
pops
```

Country-level Data

Datasets

Global Innovation Index ([INSEAD](#))

```
innov <- read_csv("data/INSEADGlobalInnovationIndex2018.csv")

innov <- innov %>%
  rename(country = "Economy", index = "Score") %>%
  select(country, index) %>%
  mutate(iso3 = countrycode(country, origin = "country.name",
                             destination = "iso3c"))
```

Country-level Data

Datasets

Global Innovation Index ([INSEAD](#))

innov

Country-level Data

Datasets

We have 3 summary tables with '*countries*' as data points (each row):

Join

1. `ctry_n` Number of survey respondents
2. `pops` Total population
3. `innov` Innovation index

Country-level Data

Datasets

Join

Combine the three tables using `inner_join()`, based on the variable `iso3`

```
ctry_data <- ctry_n %>%  
  inner_join(innov, by = "iso3") %>%  
  inner_join(pops, by = "iso3")
```

Country-level Data

Datasets

Combined table `ctry_data`

Join

`ctry_data`

Country-level Data

Datasets

Remove `country.x` **and** `country.y`

Join

```
ctry_data <- ctry_data %>%  
  select(-c(country.x, country.y))
```

Country-level Data

Datasets

ctry_data

Join

Country-level Data

Datasets

Let's save `ctry_data` **on our computer!**

Join

```
write_csv(ctry_data, "output/country_data.csv")
```

Save

Respondents per population

Calculate the no. of respondents as a *proportion of the total population*

- Save it in a new colname `respop10k`

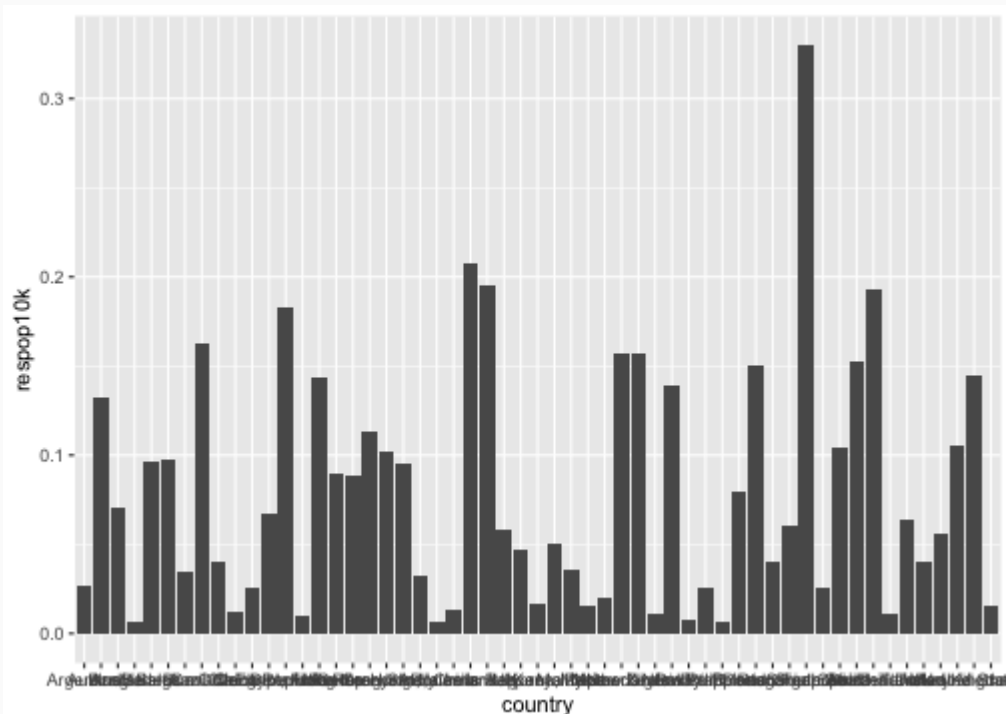
```
ctry_data$respop10k <- ctry_data$n / ctry_data$population * 10000 #respondents per 10k ppl
```

```
ctry_data
```

Respondents per population

Plot a bar chart of `respop10k` for each `country`, using `geom_col` as a graphical primitive

```
ctry_data %>%  
  ggplot(aes(x = country, y = respop10k)) +  
  geom_col()
```



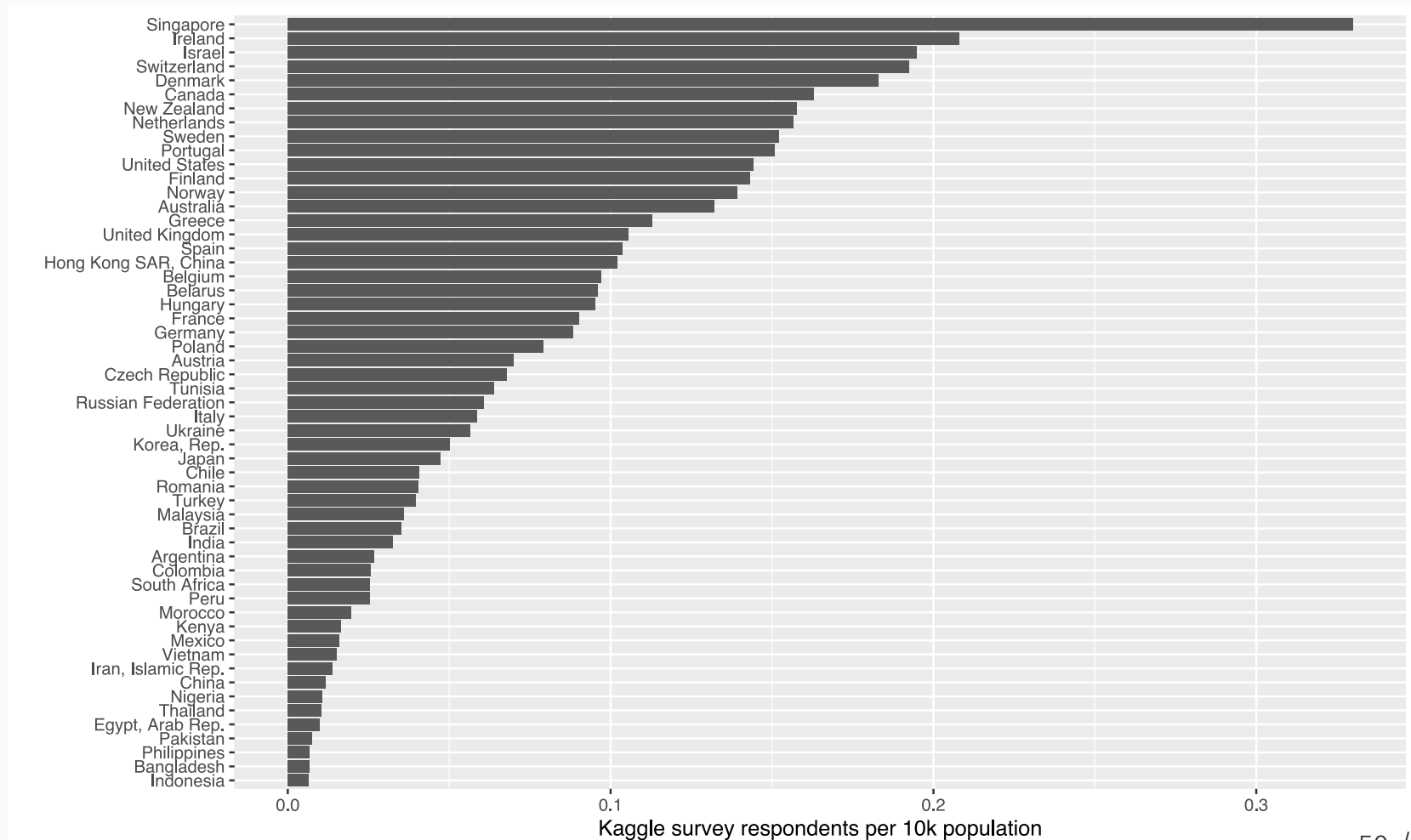
Respondents per population

Arrange countries in descending order using `reorder()`, and swap the axes using `coord_flip()`

```
ctryplot <- ctry_data %>%  
  ggplot(aes(x = reorder(country, respop10k), #plot countries in desc order  
            y = respop10k)) +  
  geom_col() +  
  
  #customisation  
  labs(x = "", y = "Kaggle survey respondents per 10k population") +  
  coord_flip() #switch x & y axis
```

Respondents per population

ctryplot



Respondents per population

Save the plot to an image file using `jpeg()`

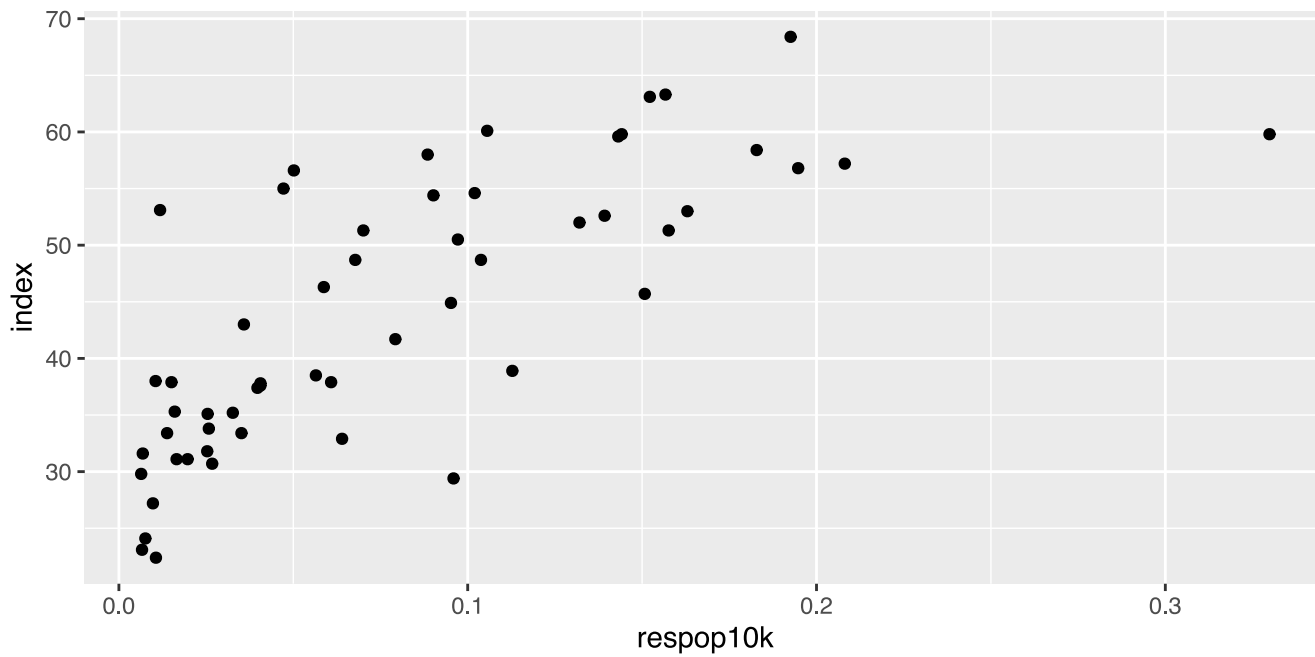
- Run the following code chunk at one go

```
jpeg(filename = "output/ctryplot.jpeg", width = 1800, height = 2000, res = 300)
ctryplot
dev.off() #finish creating the image file
```

Innovation

Make a scatter plot of `respop10k` against the innovation `index`, using `geom_point()` as graphical primitive

```
ggplot(ctry_data, aes(x = respop10k, y = index)) +  
  geom_point()
```

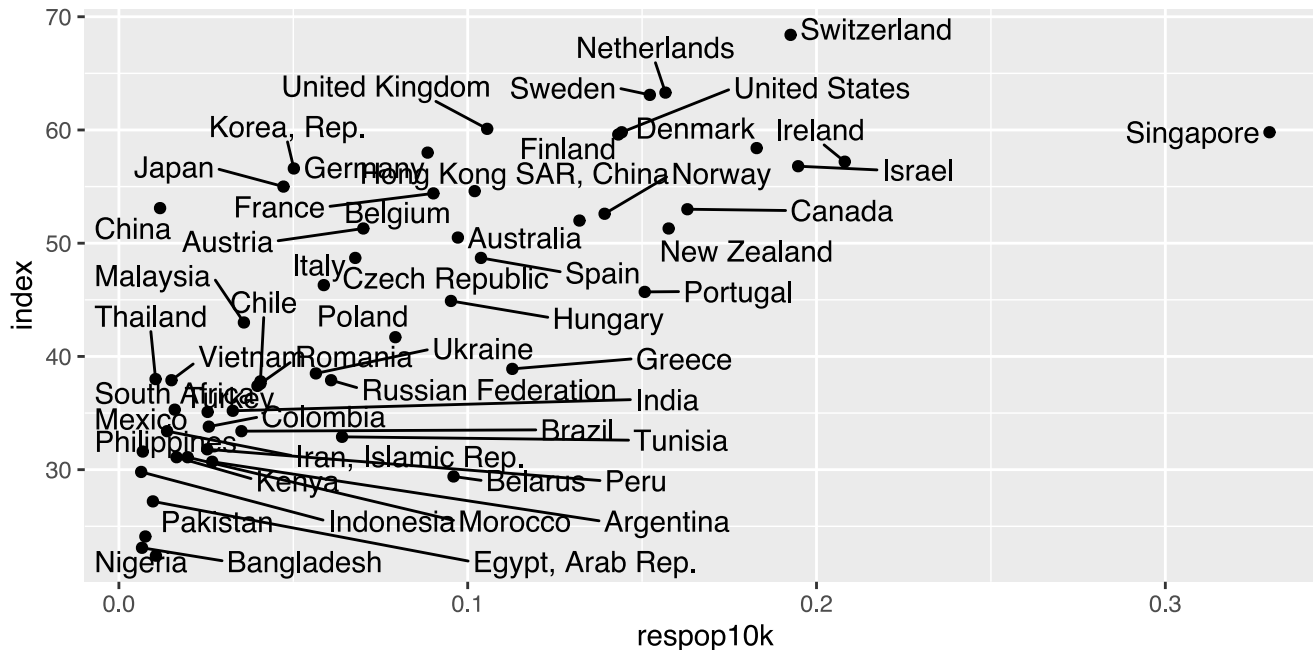


Innovation

Add the `label` aesthetic to the `aes()` argument, and add text labels using

```
ggrepel::geom_text_repel()
```

```
ggplot(ctry_data, aes(x = respop10k, y = index,  
                      label = country)) +  
  geom_point() +  
  geom_text_repel()
```

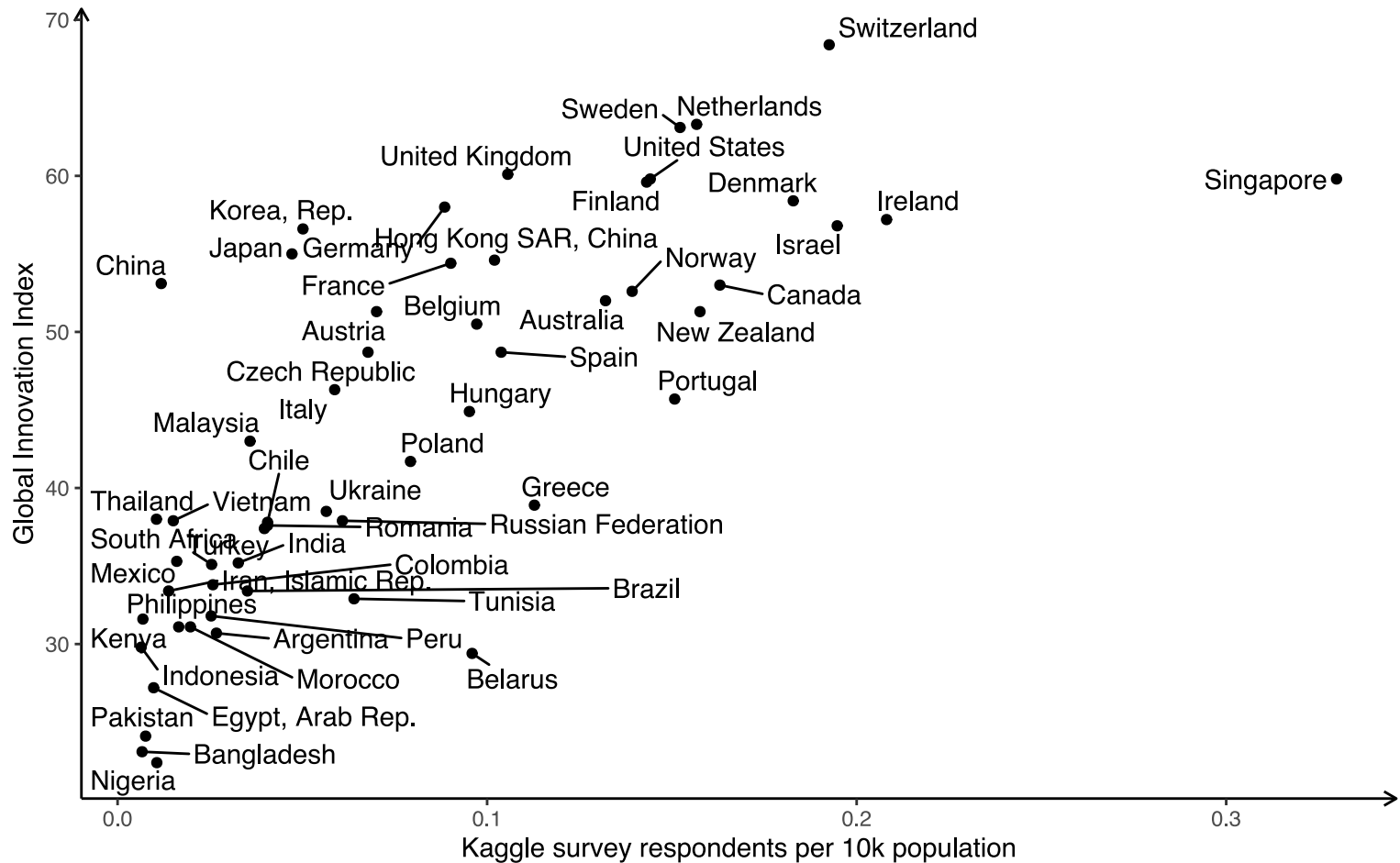


Innovation

Even more customisation:

```
ggplot(ctry_data, aes(x = respop10k,  
                      y = index,  
                      label = country)) +  
  
  geom_point() +  
  geom_text_repel() +  
  
  #customisation  
  labs(y = "Global Innovation Index",  
        x = "Kaggle survey respondents per 10k population") +  
  theme(legend.position = "none",  
        panel.grid.major = element_blank(),  
        panel.grid.minor = element_blank(),  
        panel.background = element_blank(),  
        axis.line = element_line(colour = "black",  
                                  arrow = arrow(length = unit(0.08, "inches"), type = "open")))
```

Innovation



Innovation

Test the correlation between `respop10k` **and the innovation** `index` **using** `cor.test()`

```
cor.test(ctry_data$respop10k, ctry_data$index,  
         method = "spearman") #ranked comparisons
```

```
##  
##      Spearman's rank correlation rho  
##  
## data:  ctry_data$respop10k and ctry_data$index  
## S = 5627.6, p-value = 3.364e-13  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
##      rho  
## 0.7969838
```


Outline

About our dataset

Data preparation

Survey Overview

Survey Analysis

It's your turn!

Further applications

It's your turn!

Explore and visualise `data(diamonds, package = "ggplot2")`

Filter diamonds that are less than \$3000 with a Premium cut

Hint: Use `summary()` for a summary of the dataset

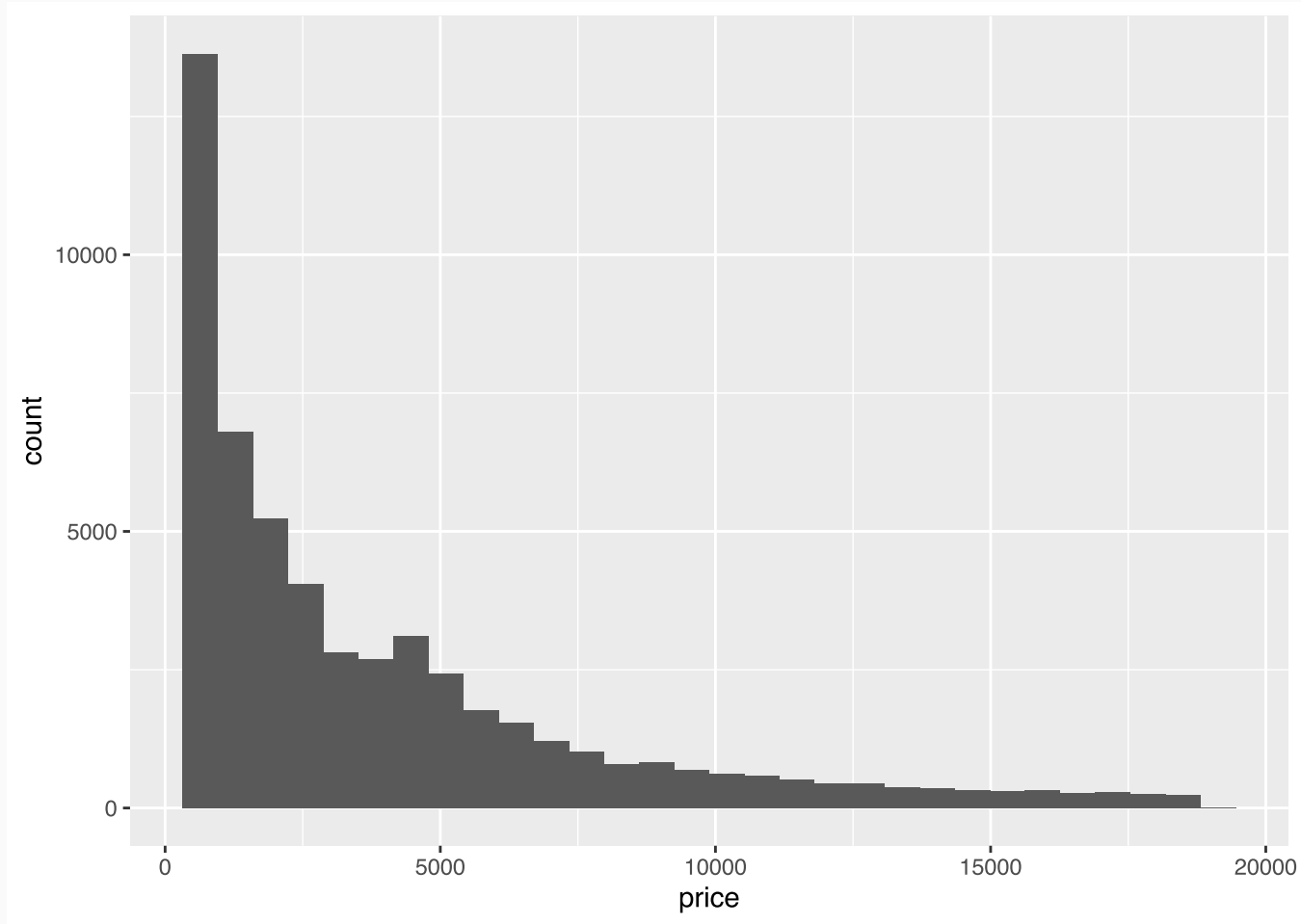
It's your turn!

Filter diamonds that are less than \$3000 with a Premium cut

```
diamonds[diamonds$price < 3000 & diamonds$cut == "Premium", ]
```

It's your turn!

Example plot

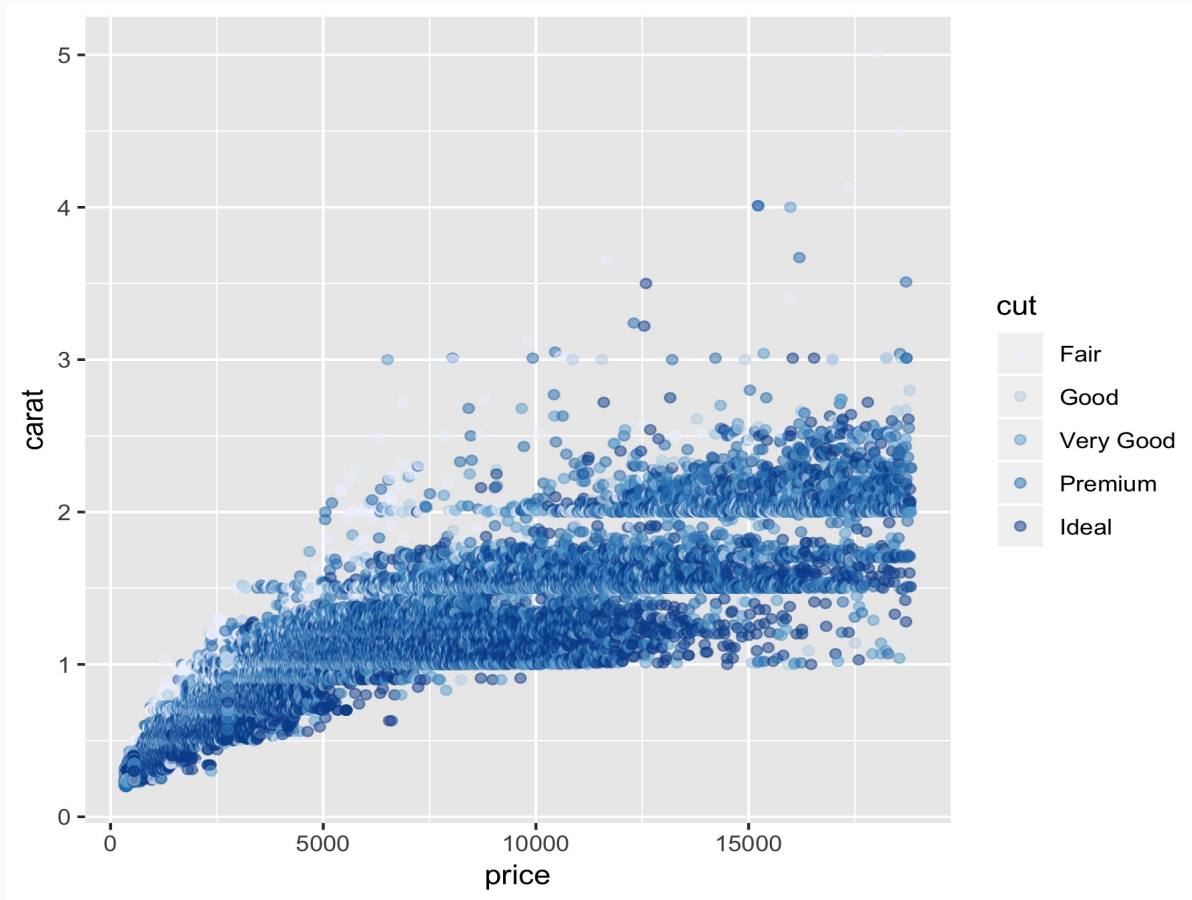


It's your turn!

```
diamonds %>%  
  ggplot(aes(x = price)) +  
  geom_histogram()
```

It's your turn!

Example plot



It's your turn!

```
diamonds %>%  
  ggplot(aes(x = price,  
             y = carat,  
             color = cut)) +  
  geom_point(alpha = 0.5) +  
  
  scale_color_brewer(type = "seq")
```

Questions?

About our dataset

Data preparation

Survey Overview

Survey Analysis

It's your turn!

Further applications

Further applications

Data communication

- Interactive plots with [Plotly](#)
- Visualisation with the [rCharts](#) package
- Interactive web apps with [Shiny](#)
- Alternative outputs for [R Markdown](#) documents

Statistics in R

- [r-statistics.co](#) by Selva Prabhakaran
- [R Tutorial: An Introduction to Statistics](#)
- [Learning Statistics with R](#) by Danielle Navarro
- [Statistics Fundamentals with R](#) by Datacamp
- [Statistics and R](#) by Havard University

Other resources

- [Skill tracks in R](#) by Datacamp