

# Introduction to data: 1 - **Language of data**

Viorel Munteanu

# Welcome

- In this tutorial we will take you through concepts and R code that are essential for getting started with data analysis.
- Scientists seek to answer questions using rigorous methods and careful observations.
- These observations form the backbone of a Data Science investigation and are called data.
- Data Science is the study of how best to collect, analyze, and draw conclusions from data. It is helpful to put statistics in the context of a general process of investigation

# Welcome

- **Step 1: Identify a question or problem.**
- **Step 2: Collect relevant data on the topic**
- **Step 3:** Data wrangling. Transform data
- **Step 4:** Analyze the data (EDA). Data visualization
- **Step 5:** Modeling Process. Model selection
- **Step 6:** Communication

We will focus on **steps 1 and 2** of this process in this tutorial.

## Case study

---

# Treating Chronic Fatigue Syndrome

- **Objective:** Evaluate the effectiveness of cognitive-behavior therapy for chronic fatigue syndrome.
- **Participant pool:** 142 patients who were recruited from referrals by primary care physicians and consultants to a hospital clinic specializing in chronic fatigue syndrome.
- **Actual participants:** Only 60 of the 142 referred patients entered the study. Some were excluded because they didn't meet the diagnostic criteria, some had other health issues, and some refused to be a part of the study.

# Study design

- Patients randomly assigned to treatment and control groups, 30 patients in each group:
  - *Treatment*: Cognitive behavior therapy – collaborative, educative, and with a behavioral emphasis. Patients were shown on how activity could be increased steadily and safely without exacerbating symptoms.
  - *Control*: Relaxation – No advice was given about how activity could be increased. Instead progressive muscle relaxation, visualization, and rapid relaxation skills were taught.

# Results

The table below shows the distribution of patients with good outcomes at 6-month follow-up. Note that 7 patients dropped out of the study: 3 from the treatment and 4 from the control group.

		<i>Good outcome</i>		
		Yes	No	Total
<i>Group</i>	Treatment	19	8	27
	Control	5	21	26
	Total	24	29	53

# Results

The table below shows the distribution of patients with good outcomes at 6-month follow-up. Note that 7 patients dropped out of the study: 3 from the treatment and 4 from the control group.

		<i>Good outcome</i>		
		Yes	No	Total
<i>Group</i>	Treatment	19	8	27
	Control	5	21	26
	Total	24	29	53

- Proportion with **good outcomes** in treatment group:

$$19/27 \approx 0.70 \rightarrow 70\%$$



# Results

The table below shows the distribution of patients with good outcomes at 6-month follow-up. Note that 7 patients dropped out of the study: 3 from the treatment and 4 from the control group.

<i>Group</i>	<i>Good outcome</i>		Total
	Yes	No	
Treatment	19	8	27
Control	5	21	26
Total	24	29	53

- Proportion with **good outcomes** in treatment group:

$$19/27 \approx 0.70 \rightarrow 70\%$$

- Proportion with **good outcomes** in control group:

$$5/26 \approx 0.19 \rightarrow 19\%$$

# Understanding the results

Do the data show a “real” difference between the groups?

# Understanding the results

Do the data show a “real” difference between the groups?

- Suppose you flip a coin 100 times. While the chance a coin lands heads in any given coin flip is 50%, we probably won't observe exactly 50 heads. This type of fluctuation is part of almost any type of data generating process.
- The observed difference between the two groups ( $70 - 19 = 51\%$ ) may be real or may be due to natural variation.
- Since the difference is quite large, it is more believable that the difference is real.
- ***We need statistical tools to determine if the difference is so large that we should reject the notion that it was due to chance.***

# Generalizing the results

Are the results of this study generalizable to all patients with chronic fatigue syndrome?

# Generalizing the results

Are the results of this study generalizable to all patients with chronic fatigue syndrome?

These patients had specific characteristics and **volunteered** to be a part of this study, Therefore they may **not be representative** of all patients with chronic fatigue syndrome. While we cannot immediately generalize the results to all patients, this first study is encouraging. The method works for patients with some narrow set of characteristics, and that gives hope that it will work, at least to some degree, with other patients.

# Aim

Our learning goals for the tutorial are *to internalize the language of data, load and view a dataset in R and distinguish between various variable types, classify a study as observational or experimental, and determine the scope modeling process, distinguish between various sampling strategies, and identify the principles of experimental design*

# Types of variables

- When you first start working with a dataset, it's good practice to take a note of its dimensions; how many rows or observations and how many columns or variables the data frame has.
- We can do this using the `glimpse()` function, that also give us a quick look at the list of variables in the dataset.

# Types of variables

- Now, we will delve deeper into the categorization of variables as **numerical and categorical**.
- This is an important step, as the type of variable helps us determine *what summary statistics to calculate*, *what type of visualizations to make*, and *what models will be appropriate* to answer the research questions we're exploring.



# Types of variables

There are two types of variables: numerical and categorical.

- **Numerical**, in other words, quantitative, variables take on numerical values. It is sensible to add, subtract, take averages, and so on, with these values.
- **Categorical**, or qualitative, variables, take on a limited number of distinct categories. These categories can be identified with numbers, for example, it is customary to see likert variables (strongly agree to strongly disagree) coded as 1 through 5, but it wouldn't be sensible to do arithmetic operations with these values. They are merely placeholders for the levels of the categorical variable.

# Numerical data

Numerical variables can be further categorized as **continuous or discrete**.

- **Continuous numerical** variables are usually measured, such as height. These variables can take on an infinite number of values within a given range.
- **Discrete numerical** variables are those that take on one of a specific set of numeric values where we are able to count or enumerate all of the possibilities. One example of a discrete variable is number of pets in a household. In general, count data are an example of discrete variables.

# Numerical data

When determining whether a numerical variable is continuous or discrete, it is important to think about the nature of the variable and not just the observed value, as rounding of continuous variables can make them appear to be discrete. For example, height is a continuous variable, however we tend to report our height rounded to the nearest unit of measure, like inches or centimeters.

# Categorical data data

Categorical variables that have ordered levels are called **ordinal**. Think about a survey question where you're asked how satisfied you are with the customer service you received and the options are ***very unsatisfied, unsatisfied, neutral, satisfied, and very satisfied***. These levels have an inherent ordering, hence the variable would be called ordinal.

If the levels of a categorical variable do not have an inherent ordering to them, then the variable is simply called **nominal**. For example, do you consume caffeine or not?

# Variables in hsb2

Let's take a moment to go through the variables in the High School and Beyond dataset:

```
glimpse(hsb2)
```

```
Rows: 200
```

```
Columns: 11
```

```
$ id      <int> 70, 121, 86, 141, 172, 113, 50, 11, 84, 48, 75, 60, 95, 104, 3...
$ gender  <chr> "male", "female", "male", "male", "male", "male", "male", "mal...
$ race    <chr> "white", "white", "white", "white", "white", "white", "african...
$ ses     <fct> low, middle, high, high, middle, middle, middle, middle, middl...
$ schtyp  <fct> public, public, public, public, public, public, public, public...
$ prog    <fct> general, vocational, general, vocational, academic, academic, ...
$ read    <int> 57, 68, 44, 63, 47, 44, 50, 34, 63, 57, 60, 57, 73, 54, 45, 42...
$ write   <int> 52, 59, 33, 44, 52, 52, 59, 46, 57, 55, 46, 65, 60, 63, 57, 49...
$ math    <int> 41, 53, 54, 47, 57, 51, 42, 45, 54, 52, 51, 51, 71, 57, 50, 43...
$ science <int> 47, 63, 58, 53, 53, 63, 53, 39, 58, 50, 53, 63, 61, 55, 31, 50...
$ socst   <int> 57, 61, 31, 56, 61, 61, 61, 36, 51, 51, 61, 61, 71, 46, 56, 56...
```

## Variables in hsb2

Using the *glimpse()* function, we can obtain a list of the variables in the dataset and also see what the values stored in these variables look like.

The first variable is **id**, which is an identifier variable for the student.

```
int [1:200] 70 121 86 141 172 113 50 11 84  
48 ...
```

Strictly speaking, **this is a categorical variable**, though the labeling of this variable is likely not that useful since we would not use this variable in an analysis of relationships between the variables in the dataset. You can think of this variable as being an anonymized version to having the names of the students in the dataset.

## Variables in hsb2

The next variable is gender, a categorical variable, with levels "male" and "female"

```
chr [1:200] "male" "female" "male" "male"  
"male" "male" "male" "male" ...
```

## Variables in hsb2

There is no inherent ordering to the levels of this variable, no matter what anyone tells you! So, this is just a categorical variable. The same is true for the race variable, which has levels of "white", "african american", "hispanic", and "asian".

```
chr [1:200] "white" "white" "white"  
"white" "white" "white" ...
```



## Variables in hsb2

Socio-economic status, on the other hand, has three levels "low", "middle", and "high" that have an inherent ordering, hence this variable is an *ordinal* categorical variable.

```
Factor w/ 3 levels "low", "middle", ...: 1 2  
3 3 2 2 2 2 2 2 ...
```

## Variables in hsb2

School type and program are also both categorical variables, with no inherent ordering to their levels.

```
Factor w/ 2 levels "public","private": 1 1 1  
1 1 1 1 1 1 1 ...
```

```
Factor w/ 3 levels "general","academic",...:  
1 3 1 3 2 2 1 2 1 2 ...
```

The remaining variables are scores that these students received in reading, writing, math, science, and social studies tests. Since these scores are all whole numbers, and assuming that it is not possible to obtain a non-whole number score on these tests, these variables are discrete numerical.

# Categorical data in R: factors

- There are various data classes in R. One of these classes is a ***factor***, which is what R often stores categorical variables as.
- An important use of factors is in statistical and ML modeling, since categorical variables enter into models differently than numerical variables.

# Categorical Data

- Often stored as factor in R
  - Important use: statistical modeling
  - Sometimes undesirable, sometimes essential
- Common in subgroup analysis
  - Only interested in a subset of the data
  - Filter for specific levels (values) of categorical variable

# Categorical Data

- A common step in many analyses that involve categorical data is a [subgroup analysis](#), where we work with only a subset of the data.
- For example, analyzing data only from students in public schools or only for students who identified as female. We can obtain these subsets by filtering for the specific levels we're interested in.
- Suppose we want to do an analysis of only the students in public schools in the High School and Beyond dataset. Let's first find out how many such students there are.

# Categorical Data

One option for obtaining this information in R uses the `count()` function from the **dplyr** package, one of the packages included in the tidyverse. This package provides a variety of functions for wrangling and summarizing data.

Once such function is `count()` which gives the frequencies of occurrence of the unique values in a given column. In this case we're interested in the number of students for each level of the `schtyp` (school type) column.

# Variables in hsb2

Let's take a moment to go through the variables in the High School and Beyond dataset:

```
glimpse(hsb2)
```

```
Rows: 200
```

```
Columns: 11
```

```
$ id      <int> 70, 121, 86, 141, 172, 113, 50, 11, 84, 48, 75, 60, 95, 104, 3...
$ gender  <chr> "male", "female", "male", "male", "male", "male", "male", "mal...
$ race    <chr> "white", "white", "white", "white", "white", "white", "african...
$ ses     <fct> low, middle, high, high, middle, middle, middle, middle, middl...
$ schtyp  <fct> public, public, public, public, public, public, public, public...
$ prog    <fct> general, vocational, general, vocational, academic, academic, ...
$ read    <int> 57, 68, 44, 63, 47, 44, 50, 34, 63, 57, 60, 57, 73, 54, 45, 42...
$ write   <int> 52, 59, 33, 44, 52, 52, 59, 46, 57, 55, 46, 65, 60, 63, 57, 49...
$ math    <int> 41, 53, 54, 47, 57, 51, 42, 45, 54, 52, 51, 51, 71, 57, 50, 43...
$ science <int> 47, 63, 58, 53, 53, 63, 53, 39, 58, 50, 53, 63, 61, 55, 31, 50...
$ socst   <int> 57, 61, 31, 56, 61, 61, 61, 36, 51, 51, 61, 61, 71, 46, 56, 56...
```

# Categorical Data

```
hsb2 %>%  
  count(schtyp)  
# A tibble: 2 × 2  
  schtyp      n  
  <fct>    <int>  
1 public    168  
2 private   32
```



# Categorical Data

- There are 168 students in public schools and 32 in private schools.
- We can read the code as: “take the `hsb2` data frame and **pipe it** into the `count()` function, then `count()` the occurrences of unique values in the `schtyp` variable.”
- You might be wondering what we mean by “*pipe it into the `count()` function*”?

# Classroom survey

A survey was conducted on students in an introductory statistics course. Below are a few of the questions on the survey, and the corresponding variables the data from the responses were stored in:

- gender: What is your gender?
- intro extra: Do you consider yourself introverted or extraverted?
- sleep: How many hours do you sleep at night, on average?
- bedtime: What time do you usually go to bed?
- countries: How many countries have you visited?
- dread: On a scale of 1-5, how much do you dread being here?

# Data frame

Data collected on students in a statistics class on a variety of variables:

*variable*

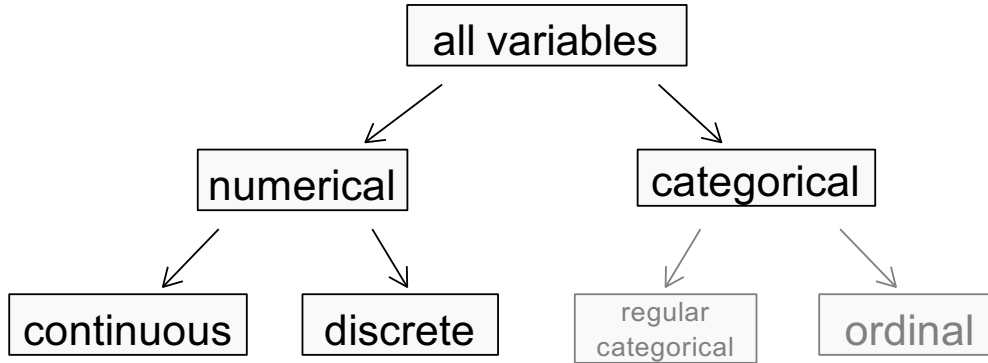
↓

Stu.	gender	introextra	...	dread
1	male	extravert	..	3
2	female	extravert	..	2
3	female	· <sup>3</sup>		4
4	female	introvert <sup>5</sup>	..	2
⋮	⋮	⋮	⋮	⋮
86	male	extravert	⋮	3

← *observation*

...

# Types of variables



# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender:

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep:

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep: *numerical, continuous*



# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep: *numerical, continuous*
- bedtime:

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep: *numerical, continuous*
- bedtime: *categorical, ordinal*

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep: *numerical, continuous*
- bedtime: *categorical, ordinal*
- countries:

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep: *numerical, continuous*
- bedtime: *categorical, ordinal*
- countries: *numerical, discrete*

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep: *numerical, continuous*
- bedtime: *categorical, ordinal*
- countries: *numerical, discrete*
- dread:

# Types of variables (cont.)

	gender	sleep	bedtime	countries	dread
1	male	5	12-2	13	3
2	female	7	10-12	7	2
3	female	5.5	12-2	1	4
4	female	7	12-2		2
5	female	3	12-2	1	3
6	female	3	12-2	9	4

- gender: *categorical*
- sleep: *numerical, continuous*<sup>4</sup>
- bedtime: *categorical, ordinal*
- countries: *numerical, discrete*
- dread: *categorical, ordinal* - could also be used as numerical

# Practice

What type of variable is a telephone area code?

- (a) numerical, continuous
- (b) numerical, discrete
- (c) categorical
- (d) categorical, ordinal

# Practice

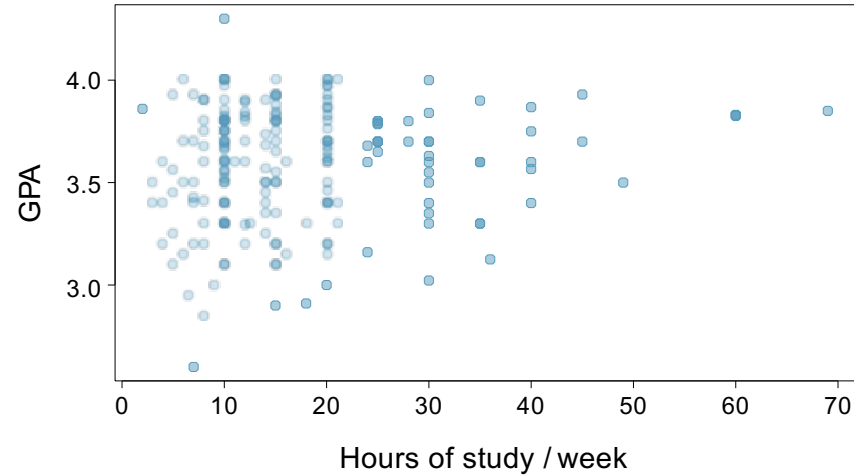
What type of variable is a telephone area code?

- (a) numerical, continuous
- (b) numerical, discrete
- (c) *categorical*
- (d) categorical, ordinal



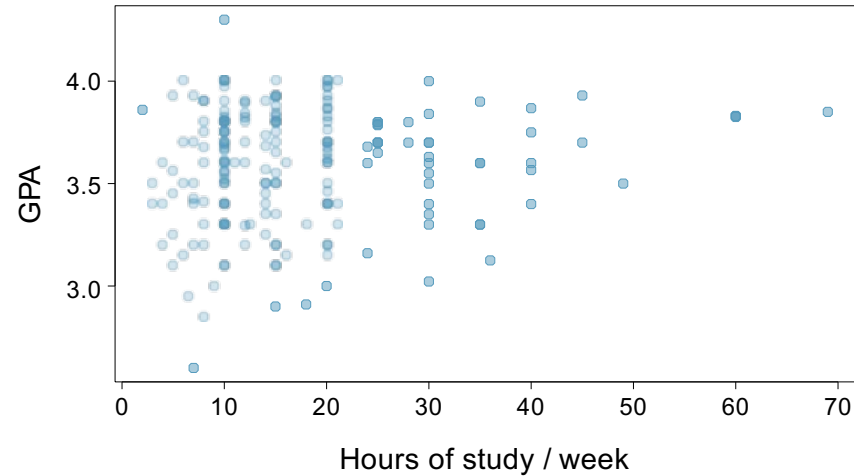
# Relationships among variables

Does there appear to be a relationship between GPA and number of hours students study per week?



# Relationships among variables

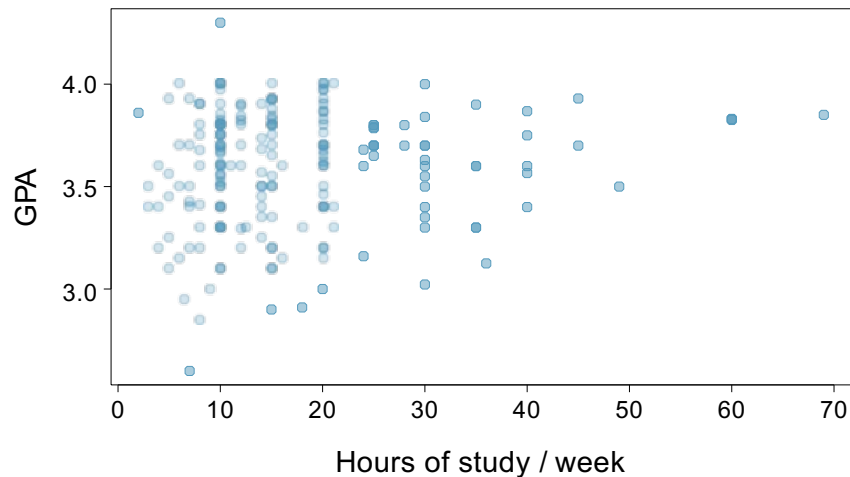
Does there appear to be a relationship between GPA and number of hours students study per week?



Can you spot anything unusual about any of the data points?

# Relationships among variables

Does there appear to be a relationship between GPA and number of hours students study per week?



Can you spot anything unusual about any of the data points?

*There is one student with  $GPA > 4.0$ , this is likely a data error.*

# Explanatory and response variables

- To identify the explanatory variable in a pair of variables, identify which of the two is suspected of affecting the other:

explanatory variable  $\overset{\text{might affect}}{\longrightarrow}$  response variable

- Labeling variables as explanatory and response does not guarantee the relationship between the two is actually causal, even if there is an association identified between the two variables. We use these labels only to keep track of which variable we suspect affects the other.

# Two primary types of data collection

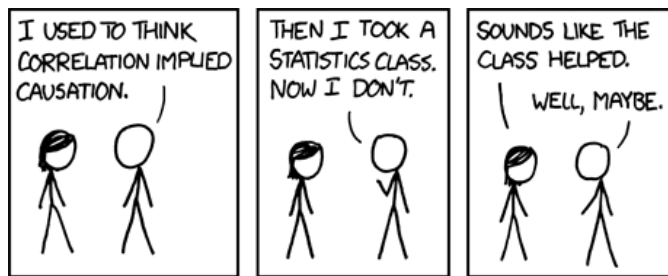
- *Observational studies*: Collect data in a way that does not directly interfere with how the data arise (e.g. surveys).
  - Can provide evidence of a naturally occurring association between variables, but they cannot by themselves show a causal connection.

# Two primary types of data collection

- *Observational studies*: Collect data in a way that does not directly interfere with how the data arise (e.g. surveys).
  - Can provide evidence of a naturally occurring association between variables, but they cannot by themselves show a causal connection.
- *Experiment*: Researchers randomly assign subjects to various treatments in order to establish causal connections between the explanatory and response variables.

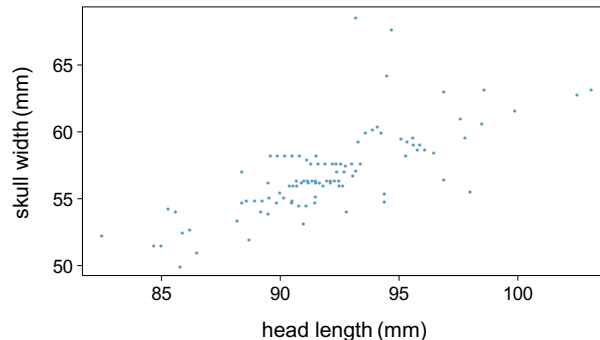
# Association vs. causation

- When two variables show some connection with one another, they are called *associated* variables.
- Associated variables can also be called *dependent* variables and vice-versa.
- If two variables are not associated, i.e. there is no evident connection between the two, then they are said to be *independent*.
- In general, **association does not imply causation**, and causation can only be inferred from a randomized experiment.



# Practice

Based on the scatterplot on the right, which of the following statements is correct about the head and skull lengths of possums?

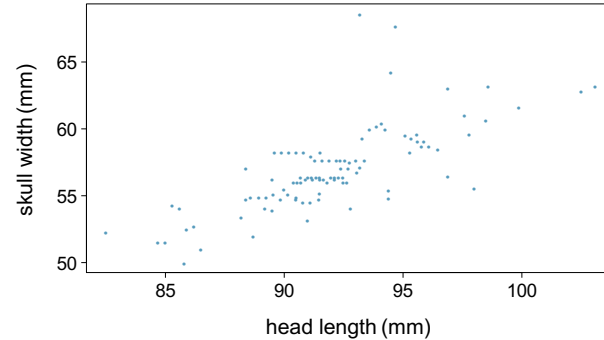


- (a) There is no relationship between head length and skull width, i.e. the variables are independent.
- (b) Head length and skull width are positively associated.
- (c) Skull width and head length are negatively associated.
- (d) A longer head causes the skull to be wider.
- (e) A wider skull causes the head to be longer.



# Practice

Based on the scatterplot on the right, which of the following statements is correct about the head and skull lengths of possums?



- (a) There is no relationship between head length and skull width, i.e. the variables are independent.
- (b) *Head length and skull width are positively associated.*
- (c) Skull width and head length are negatively associated.
- (d) A longer head causes the skull to be wider.
- (e) A wider skull causes the head to be longer.

## **Sampling principles and strategies**

---

# Populations and samples

PHYS ED | AUGUST 29, 2012, 12:01 AM | 21 Comments

## Finding Your Ideal Running Form

By GRETCHEN REYNOLDS



David De Loss/Getty Images

<http://well.blogs.nytimes.com/2012/08/29/>

[finding-your-ideal-running-form](#)

*Research question:* Can people become better, more efficient runners on their own, merely by running?

# Populations and samples

PHYS ED | AUGUST 29, 2012, 12:01 AM | 21 Comments

## Finding Your Ideal Running Form

By GRETCHEN REYNOLDS



David De Lossy/Getty Images

<http://well.blogs.nytimes.com/2012/08/29/>

[finding-your-ideal-running-form](#)

*Research question:* Can people become better, more efficient runners on their own, merely by running?

*Population of interest:*

# Populations and samples

PHYS ED | AUGUST 29, 2012, 12:01 AM | 21 Comments

## Finding Your Ideal Running Form

By GRETCHEN REYNOLDS



David De Loss/Getty Images

<http://well.blogs.nytimes.com/2012/08/29/>

[finding-your-ideal-running-form](#)

*Research question:* Can people become better, more efficient runners on their own, merely by running?

*Population of interest:* All people

# Populations and samples

PHYS ED | AUGUST 29, 2012, 12:01 AM | 21 Comments

## Finding Your Ideal Running Form

By GRETCHEN REYNOLDS



David De Lossy/Getty Images

*Research question:* Can people become better, more efficient runners on their own, merely by running?

*Population of interest:* All people

<http://well.blogs.nytimes.com/2012/08/29/>

[finding-your-ideal-running-form](#)

*Sample:* Group of adult women who recently joined a running group

# Populations and samples

PHYS ED | AUGUST 29, 2012, 12:01 AM | 21 Comments

## Finding Your Ideal Running Form

By GRETCHEN REYNOLDS



David De Lossy/Getty Images

<http://well.blogs.nytimes.com/2012/08/29/>

[finding-your-ideal-running-form](#)

*Research question:* Can people become better, more efficient runners on their own, merely by running?

*Population of interest:* All people

*Sample:* Group of adult women who recently joined a running group

*Population to which results can be generalized:*

# Populations and samples

PHYS ED | AUGUST 29, 2012, 12:01 AM | 21 Comments

## Finding Your Ideal Running Form

By GRETCHEN REYNOLDS



David De Loss/Getty Images

<http://well.blogs.nytimes.com/2012/08/29/>

[finding-your-ideal-running-form](#)

*Research question:* Can people become better, more efficient runners on their own, merely by running?

*Population of interest:* All people

*Sample:* Group of adult women who recently joined a running group

*Population to which results can be generalized:* Adult women, if the data are randomly sampled



# Anecdotal evidence and early smoking research

- Anti-smoking research started in the 1930s and 1940s when cigarette smoking became increasingly popular. While some smokers seemed to be sensitive to cigarette smoke, others were completely unaffected.
- Anti-smoking research was faced with resistance based on *anecdotal evidence* such as “My uncle smokes three packs a day and he’s in perfectly good health”, evidence based on a limited sample size that might not be representative of the population.
- It was concluded that “smoking is a complex human behavior, by its nature difficult to study, confounded by human variability.”
- In time researchers were able to examine larger samples of cases (smokers), and trends showing that smoking has negative health impacts became much clearer.

# Census

- Wouldn't it be better to just include everyone and “sample” the entire population?
  - This is called a *census*.

# Census

- Wouldn't it be better to just include everyone and “sample” the entire population?
  - This is called a *census*.
- There are problems with taking a census:
  - It can be difficult to complete a census: there always seem to be some individuals who are hard to locate or hard to measure. *And these difficult-to-find people may have certain characteristics that distinguish them from the rest of the population.*
  - Populations rarely stand still. Even if you could take a census, the population changes constantly, so it's never possible to get a perfect measure.
  - Taking a census may be more complex than sampling.

# Observational studies

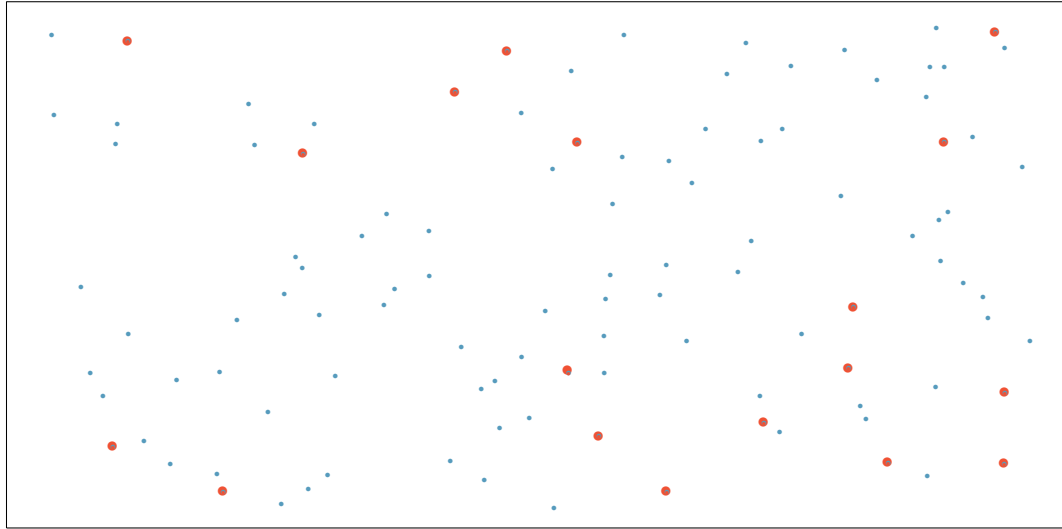
- Researchers collect data in a way that does not directly interfere with how the data arise.
- Results of an observational study can generally be used to establish an association between the explanatory and response variables.

# Obtaining good samples

- Almost all statistical methods are based on the notion of implied randomness.
- If observational data are not collected in a random framework from a population, these statistical methods – the estimates and errors associated with the estimates – are not reliable.
- Most commonly used random sampling techniques are *simple*, *stratified*, and *cluster* sampling.

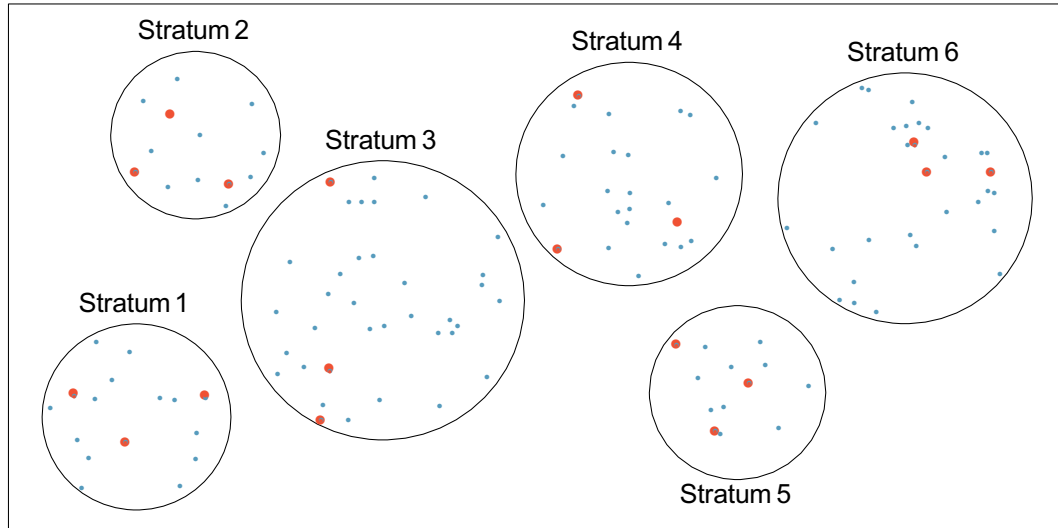
# Simple random sample

Randomly select cases from the population, where there is no implied connection between the points that are selected.



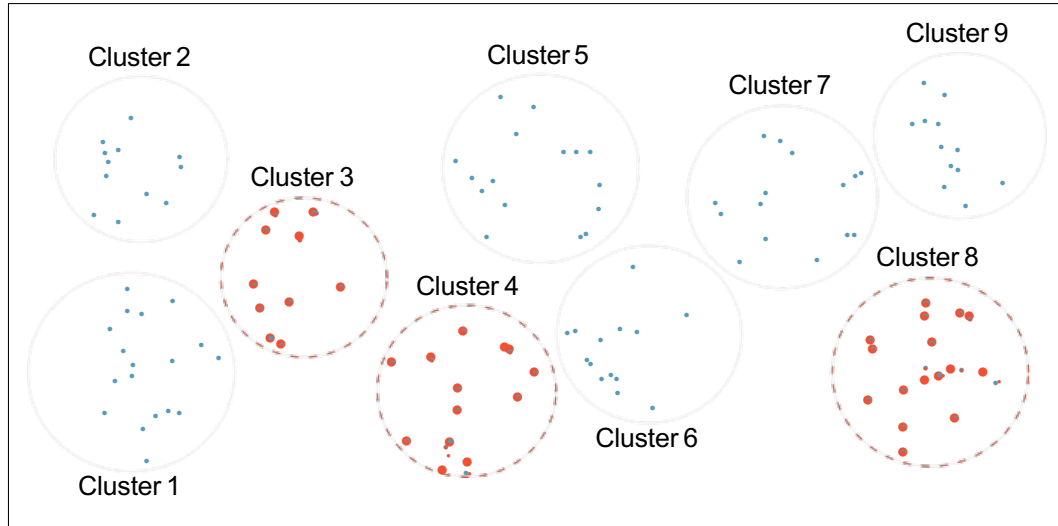
# Stratified sample

*Strata* are made up of **similar observations**. We take a simple random sample from each stratum.



# Cluster sample

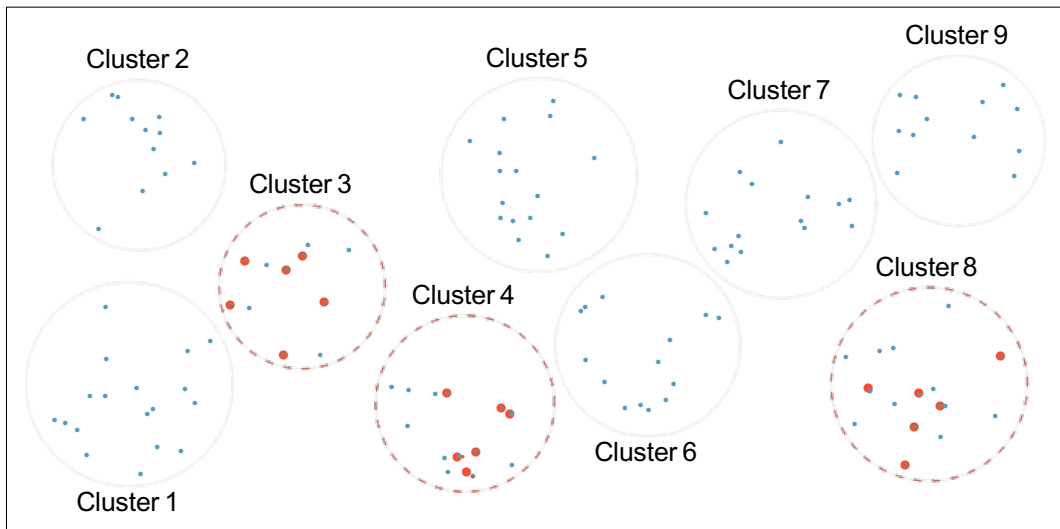
*Clusters* are usually not made up of **homogeneous observations**. We take a simple random sample of clusters, and then sample all observations in that cluster. Usually preferred for economical reasons.





# Multistage sample

*Clusters* are usually not made up of **homogeneous observations**. We take a simple random sample of clusters, and then take a simple random sample of observations from the sampled clusters.



# Practice

A city council has requested a household survey be conducted in a suburban area of their city. The area is broken into many distinct and unique neighborhoods, some including large homes, some with only apartments. Which approach would likely be the least effective?

- (a) Simple random sampling
- (b) Cluster sampling
- (c) Stratified sampling
- (d) Blocked sampling

# Practice

A city council has requested a household survey be conducted in a suburban area of their city. The area is broken into many distinct and unique neighborhoods, some including large homes, some with only apartments. Which approach would likely be the least effective?

- (a) Simple random sampling
- (b) *Cluster sampling*
- (c) Stratified sampling
- (d) Blocked sampling

## R coding

---

# Primer for R and Data Science

This tutorial does not assume any previous R experience, but if you would like an introduction to R first, we recommend the [RStudio Primers](#) or the [R Bootcamp](#).

# Packages

Packages are the fundamental units of reproducible R code. They include reusable functions, the documentation that describes how to use them, and sample data. In this lesson we will make use of two packages:

- **tidyverse:** Tidyverse is a collection of R packages for data science that adhere to a common philosophy of data and R programming syntax, and are designed to work together naturally. You can learn more about tidyverse [here](#). But no need to go digging through the package documentation, we will walk you through what you need to know about these packages as they become relevant.
- **openintro:** The openintro package contains datasets used in openintro resources. You can find out more about the package [here](#).

# Packages

Once we have installed the packages, we use the `library()` function to load packages into R.

Let's load these two packages to be used in the remainder of this lesson.

```
install.packages('tidyverse')  
library(tidyverse)  
library(openintro)
```

# Data in R

- One of the datasets that we will work with in this tutorial comes from the High School and Beyond Survey, which is a survey conducted on high school seniors by the National Center of Education Statistics.
- The data are organized in what we call a *data frame*, where each row represents an *observation* or a *case* and each column represents a *variable*. If you ever use spreadsheets, such as a Google sheet or Excel, this representation should be familiar to you.



# Loading data into R

Un pachet R este o colecție de funcții, date și documente care extind capacitățile bazei R. Utilizarea pachetelor este esențială pentru utilizarea cu succes a R-ului. Colecția de pachete ***tidyverse*** împărtășesc o filozofie comună a modelării datelor și sunt concepute pentru a lucra în mod natural împreună.

```
install.packages('tidyverse')  
library(tidyverse)
```

# Loading data into R

- There are many ways of loading data into R depending on where your data are stored. In this lesson we're using a dataset that is included in an R package so we can access this dataset by loading the package with the `library()` function.
- Other commonly used formats of data are plain text, comma separated values (CSV), Excel files (XLS or XLSX), or RData (the file format R uses to store data).
- A resource we recommend for learning more about importing data into R is the Data Import chapter in [R 4 Data Science](#) by Grolemund and Wickham.

# Reading data in R

- `data <- read_csv('path/to/file/file.csv')`
- `data <- read_delim('path/to/file/file.csv')`
- `data <- read_tsv('path/to/file/file.csv')`
- `head(data)`
- `glimpse(data)`
- `data2 <- data %>%  
 select(column1, column2, column3)`
- `head(data2)`

# Data in R

In this lesson we'll work with the High School and Beyond dataset, stored in the openintro package. The data are stored in a **data frame** called `hsb2`. You can read more about this dataset [here](#). Below is a preview of the dataset.

```
# A tibble: 6 × 11
```

	id	gender	race	ses	schtyp	prog	read	write	math	science	socst
	<int>	<chr>	<chr>	<fct>	<fct>	<fct>	<int>	<int>	<int>	<int>	<int>
1	70	male	white	low	public	general	57	52	41	47	57
2	121	female	white	middle	public	vocational	68	59	53	63	61
3	86	male	white	high	public	general	44	33	54	58	31
4	141	male	white	high	public	vocational	63	44	47	53	56
5	172	male	white	middle	public	academic	47	52	57	53	61
6	113	male	white	middle	public	academic	44	52	51	63	61

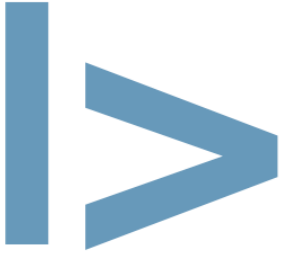
# Take a peek

When you want to work with data in R, a good first step is to take a peek at what the data look like. The `glimpse()` function is one good way of doing this. Click on the blue “Run Code” button to run the code below, and take a look at the output of the `glimpse()` function.

```
Rows: 200
Columns: 11
$ id      <int> 70, 121, 86, 141, 172, 113, 50, 11, 84, 48, 75, 60, 95, 104, 3...
$ gender  <chr> "male", "female", "male", "male", "male", "male", "male", "mal...
$ race    <chr> "white", "white", "white", "white", "white", "white", "african...
$ ses     <fct> low, middle, high, high, middle, middle, middle, middle, middl...
$ schtyp  <fct> public, public, public, public, public, public, public, public...
$ prog    <fct> general, vocational, general, vocational, academic, academic, ...
$ read    <int> 57, 68, 44, 63, 47, 44, 50, 34, 63, 57, 60, 57, 73, 54, 45, 42...
$ write   <int> 52, 59, 33, 44, 52, 52, 59, 46, 57, 55, 46, 65, 60, 63, 57, 49...
$ math    <int> 41, 53, 54, 47, 57, 51, 42, 45, 54, 52, 51, 51, 71, 57, 50, 43...
$ science <int> 47, 63, 58, 53, 53, 63, 53, 39, 58, 50, 53, 63, 61, 55, 31, 50...
$ socst   <int> 57, 61, 31, 56, 61, 61, 61, 36, 51, 51, 61, 61, 71, 46, 56, 56...
```

The output of `glimpse()` tells us that the data frame includes 200 observations (rows) and 11 variables (columns). It also lists the variables and their types, along with values of the first few observations.

# The pipe operator



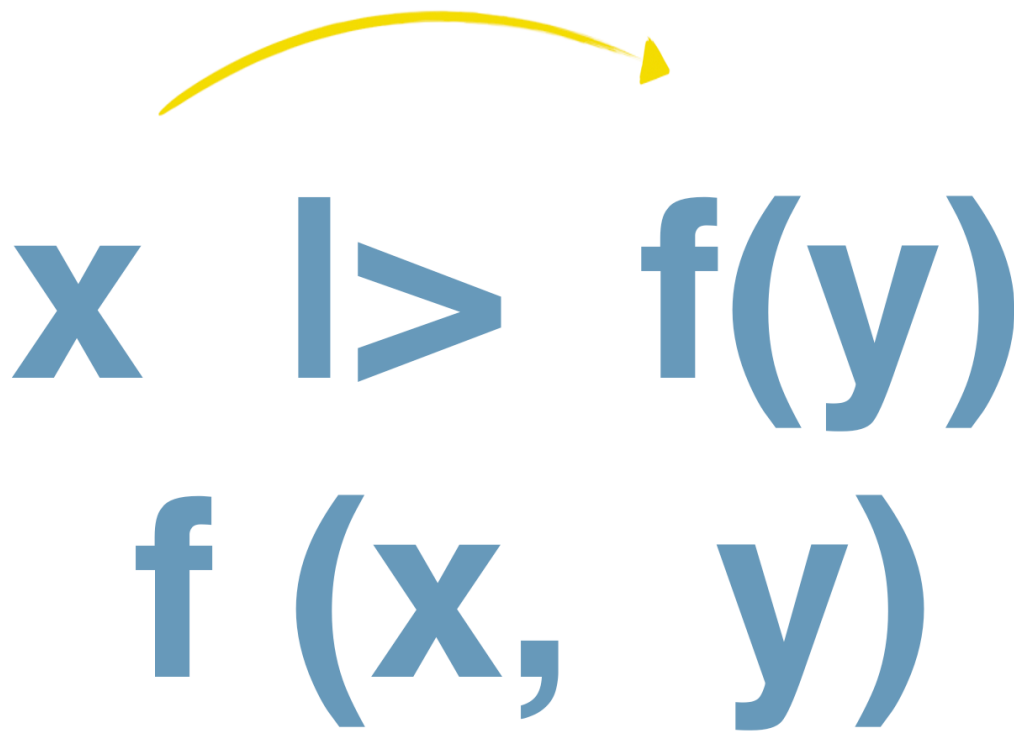
or

%>%

# The pipe operator

The **pipe operator**, which is **vertical bar greater than**, tells  $\mathbb{R}$  to pass the object that comes before it into the first argument of the function that comes after it.

Mathematically, **x pipe f(y)** becomes  $f(x, y)$ , since  $x$  is piped into the first argument of the function  $f()$ .



# The pipe operator

For example, one way of adding numbers in R is using the `sum()` function. The `sum()` function has as many arguments as there are numbers you wish to add, where each number is separated with a comma. For example, to add 3 and 4, we would use the following code. Notice, 3 and 4 are separated by a comma, indicating that these are the two numbers we wish for the `sum()` function to add.

```
sum(3, 4)
```

```
[1] 7
```



# The pipe operator

If we wanted to do the same operation with a pipe, we would instead use the code below. The pipe operator inserts 3 as the first argument into the `sum()` function, which looks like `sum(3, 4)`.

```
3 %>% sum(4)
```

```
[1] 7
```

Piping 3 into the `sum()` function may seem a bit silly, especially since it's not much easier than typing `sum(3, 4)`.

However, as we progress through these tutorials you will see that the piping operator will allow us to sequentially link together data wrangling operations. This can save us a great deal of headache later on, as the format of operations that use the pipe are far simpler to read!

# Filter

Next, let's use the `filter()` function to filter the data to only include public school students.

```
hsb2_public <- hsb2 %>%  
  filter(schtyp == 'public')
```

```
# A tibble: 6 × 11
```

	id	gender	race	ses	schtyp	prog	read	write	math	science	socst
	<int>	<chr>	<chr>	<fct>	<fct>	<fct>	<int>	<int>	<int>	<int>	<int>
1	70	male	white	low	public	general	57	52	41	47	57
2	121	female	white	middle	public	vocational	68	59	53	63	61
3	86	male	white	high	public	general	44	33	54	58	31
4	141	male	white	high	public	vocational	63	44	47	53	56
5	172	male	white	middle	public	academic	47	52	57	53	61
6	113	male	white	middle	public	academic	44	52	51	63	61

# Filter

- We can read the above code as: “take the `hsb2` data frame and **pipe it** into the `filter()` function. Next, `filter()` the data **for cases where school type is equal to public**. Then, **assign the resulting data frame** to a new data frame called **`hsb2_underscore_public`**.”
- We should take note of two pieces of R syntax: *the double equal sign* (`==`) and quotations (`“”`). In R, `==` is a logical test for “*is equal to*”. R uses this logical test to search for observations (rows) in the data frame where school type is equal to public, and returns a data frame where this comparison is `TRUE` for every row.

# Filter

- In R, variables that are categorical use characters (rather than numbers) for values. To indicate to R that you want your logical test to compare the values of a categorical variable to a specific level of that variable, you need to surround the name of the level in quotations (e.g. `schtyp == "public"`).
- The quotations tell R that the value of the variable is a character, not a number. If you forget to use quotations, R will give you an error message!

## Pipe & Filter

Now, if we make another frequency table of school type in the filtered dataset, we should only see public school in the output.

```
hsb2_public |>
  count(schtyp)

# A tibble: 1 × 2
  schtyp      n
  <fct>   <int>
1 public   168
```

# Discretize variables

- A common way of creating a new variable from an existing variable is discretizing, that is converting a numerical variable to a categorical variable based on certain criteria.
- For example, suppose we are not interested in the actual reading score of students, but instead whether their reading score is below average or at or above average.

# Discretize variables

First, we need to calculate the average reading score with the `mean()` function. This will give us the mean value, 52.23.

```
# Calculate average reading score and show  
the value  
mean(hsb2$read)  
[1] 52.23
```

## Discretize variables

However, in order to be able to refer back to this value later on, we might want to store it as an object that we can refer to by name. So instead of just printing the result, let's save it as a new object called **avg underscore read**.

```
# Calculate average reading score and  
store as avg_read  
avg_read <- mean(hsb2$read)  
[1] 52.23
```



## Discretize variables

Before we move on, a quick tip: most often we want to do both; see the value and also store it for later use. The approach we used here, running the `mean()` function twice, is redundant. A less redundant way to accomplish this task is to wrap your assignment code in parentheses so that R will both assign the average value of reading test scores to `avg_read`, and print out the value assigned to `avg_read`.

```
(avg_read <- mean(hsb2$read))
```

```
[1] 52.23
```

## Discretize variables

Next, in order to create the two groups of interest, we need to determine whether each student is either (1) below or (2) at or above average. For example, a reading score of 57 is above average, so is 68, but 44 is below. Obviously, going through each record like this would be tedious and error prone, so let's explore another option!

# New variable: read\_cat

id	...	read	read_cat
70	...	57 →	at or above avg
121	...	68 →	at or above avg
86	...	44 →	below avg
...	...	...	...
137	...	63 →	at or above avg



## New variable: read\_cat

Instead we can create a new variable, named `read_cat`, with the `mutate()` function and the helpful `if else()` function.

```
hsb2 <- hsb2 |>
  mutate(read_cat = if_else(read < avg_read,
                             "below average",
                             "at or above
average"))
)
```

hsb2

# New variable: read\_cat

```
# A tibble: 200 × 12
  id gender race      ses schtyp prog  read write  math science socst
  <int> <chr> <chr>      <fct> <fct> <fct> <int> <int> <int>    <int> <int>
1    70 male  white      low  public gene...    57    52    41        47    57
2   121 female white    midd... public voca...    68    59    53        63    61
3    86 male  white      high  public gene...    44    33    54        58    31
4   141 male  white      high  public voca...    63    44    47        53    56
5   172 male  white    midd... public acad...    47    52    57        53    61
6   113 male  white    midd... public acad...    44    52    51        63    61
7    50 male  african amer... midd... public gene...    50    59    42        53    61
8    11 male  hispanic    midd... public acad...    34    46    45        39    36
9    84 male  white      midd... public gene...    63    57    54        58    51
10   48 male  african amer... midd... public acad...    57    55    52        50    51
#  190 more rows
#  1 more variable: read_cat <chr>
```

## New variable: `read_cat`

First, we start with the data frame, `hsb2`, and pipe it into the `mutate()` function. We use the `mutate()` function to create a new variable called `read_cat`. Note that we are using a new variable name here, so that we do not overwrite the existing reading score variable, called `read`.

The values of this new variable are simple: if the reading score of the student is below the average reading score, the variable will have the label “below average”, otherwise, the label will be “at or above average”.

## New variable: `read_cat`

This discretization can be accomplished using the `if_else()` function in R:

- **The first** argument of the function is the logical test we wish to perform: `read < avg_read`.
- **The second** argument is what we want the function to do if the result of the logical test is `TRUE`, in other words, if the student's score is below the average score: `"below average"`.
- **The third** argument is what we want the function to do if the result of the logical test is `FALSE`, in other words, if the student's score is above the average score: `"at or above average"`.

Next, it's your turn to discretize a different variable.

# Visualizing numerical data

- The most logical and most useful first step of any data analysis is an exploratory analysis. And a very important and informative component of exploratory data analysis is visualization.
- We will learn a lot more about data visualization in the tutorial on Summarizing and Visualizing Data, so we won't go into too much detail on data visualization in this tutorial.
- Let's, however, make a simple scatterplot to visualize the relationship between two numerical variables so that you can get some exposure to constructing plots in R and how to interpret them.



# Visualizing numerical data

- There are many methods for visualizing data in R, but in this tutorial we will focus on using the ggplot2 package, which is part of the tidyverse.
- We chose ggplot2 because this package makes *modern looking hassle-free plots* that take care of fiddly details like drawing legends.
- Additionally, once you learn how to make simple bivariate plots, with ggplot2 it is easy to extend your code to create a visualization that displays the relationship between many variables at once without having to learn too much more syntax.

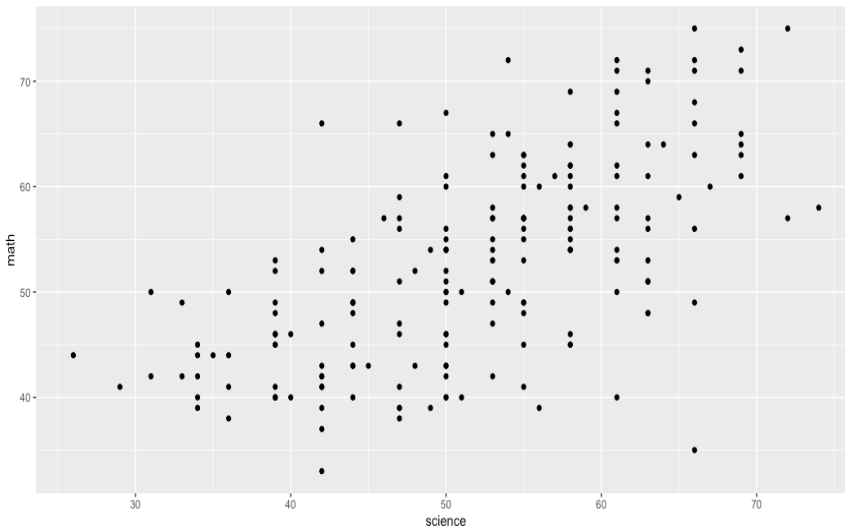
# Visualizing numerical data

- Another attractive feature of ggplot2 is that *you can build your plots in layers*, e.g. you can start with a layer showing the raw data and then add layers of annotations and statistical summaries.
- This is an attractive feature for learning the syntax, as we can go step-by-step, starting with a simple plot and slowly building up to more complex ones.

# Visualizing numerical data

We'll visualize the relationship between the math and science scores of the students in the High School and Beyond dataset.

```
ggplot(data = hsb2, aes(x =  
  science, y = math)) +  
  geom_point()
```



# Visualizing numerical data

Let's pause for a moment and review what's going on in the code above.

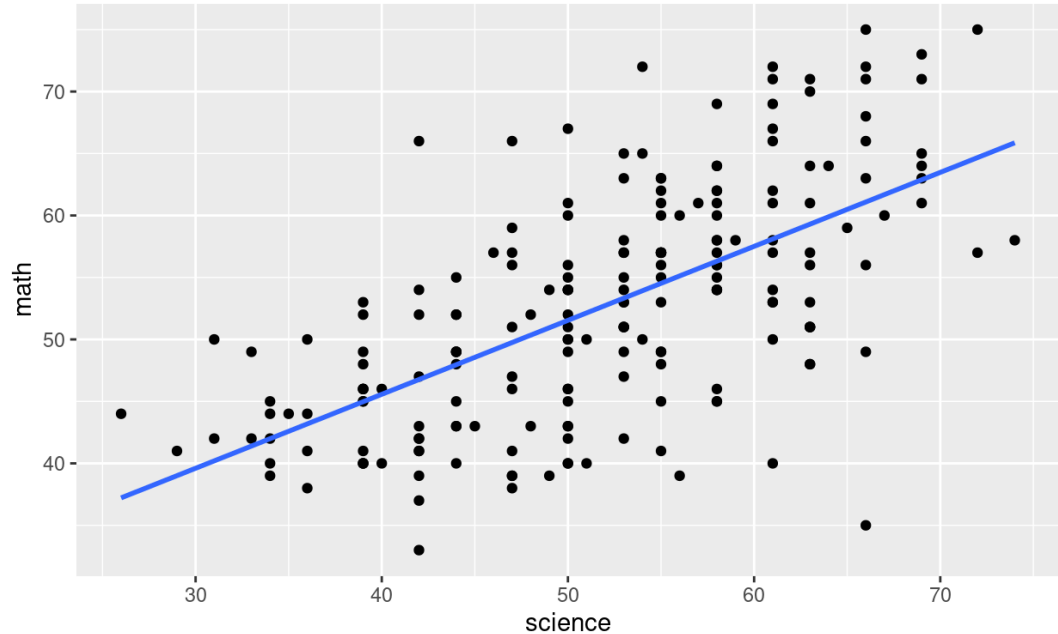
- We use the `ggplot()` function to create plots.
- The first argument is the data frame containing the data we wish to plot: `data = hsb2`.
- In the `aes`thetics argument, we map variables from the data frame to certain components of the plot. In this case we want to plot science test scores on the x and math test scores on the y axis: `aes(x = science, y = math)`.
- Lastly, we specify what `geom`etric shapes should be used to represent each observation. In this case, we want to make a scatterplot, so we want each observation to be represented by a “point” on the plot, hence we add use the `geom_point()` function to add a points layer to the plot.

# Visualizing numerical data

- In summary, the main function is `ggplot()`, the first argument is the data to use, then the `aes` maps the variables to certain features of the plot, and finally the `geom` informs the type of plot you want to make.
- Another important aspect to note here is that the `geom_XXX()` function is separated from the `ggplot()` function with a plus, `+`.
- As we mentioned earlier, `ggplot2` plots are constructed in series of layers. The plus sign separates these layers. Generally, the `+` sign can be thought of as the end of a line, so you should always hit enter/return after it. While it is not mandatory to move to the next line for each layer, doing so makes the code a lot easier to organize and read.

# Interpreting visualization

We can see that there is a positive relationship between the science and math scores of students, meaning that students who score highly in science tend to also score highly in math. Probably not that surprising a result.



# Math, science, and program

We also mentioned earlier that extending from bivariate to multivariate plots is easy in ggplot2.

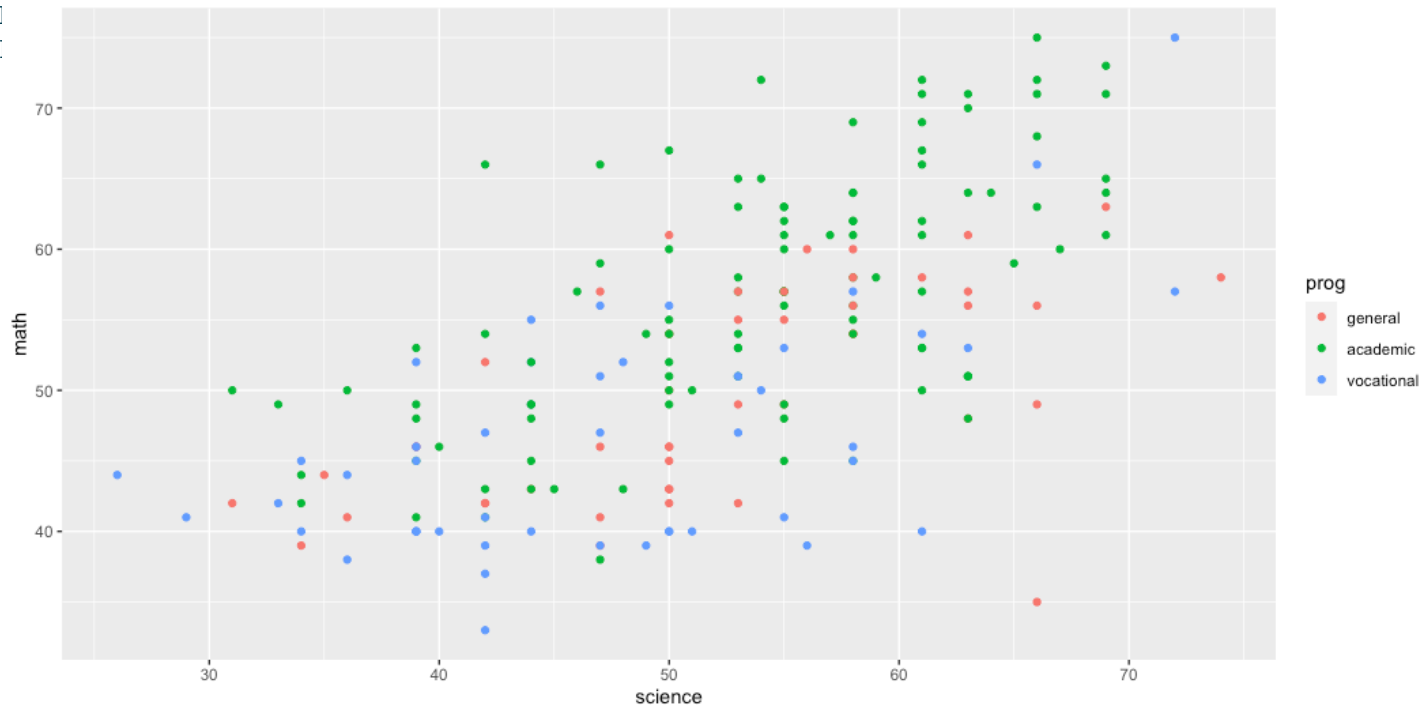
Let's plot the same math and science test scores, but this time let's also consider the program that the student is in: general, academic, or vocational.

```
ggplot(data = hsb2, aes(x = science, y = math,  
color = prog)) +  
  geom_point()
```

The code looks very similar to what we used before, except that we now have one other aesthetic mapping between the program variable and the `color` of the points that represent the observations. Note that we type the name of the variable as it appears in the data frame: `prog`.

# Math, science, and program

```
ggplot(data = hsl)  
  science, y = math  
  geom_point()
```





# Math, science, and program. Results

- The same positive relationship between math and science scores is still apparent.
- But we can also see that students in academic programs, shown with green points, tend to have higher math scores relative to their science scores than those in vocational programs, in blue, and general programs, in red.

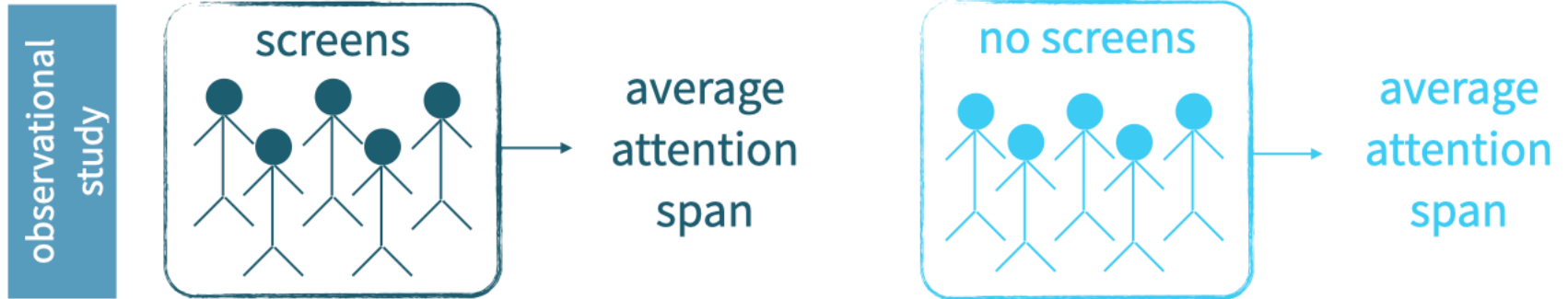
# Observational studies and experiments

- In an *observational study*, researchers collect data in a way that does not directly interfere with how the data arise, in other words, they merely "observe". And based on an observational study we can only establish an association between the *explanatory* and *response* variables.
- In an *experiment*, on the other hand, researchers randomly assign subjects to various treatments and can therefore establish causal connections between the *explanatory* and *response* variables.

# Observational studies

- Suppose we want to evaluate the relationship between using screens at bedtime such as a computer, tablet, or phone and attention span during the day.
- We can design this study as an observational study or as an experiment.
- In an observational study, we sample two types of people from the population: those who choose to use screens at bedtime and those who don't.
- Then, we find the average attention span for the two groups of people and compare.

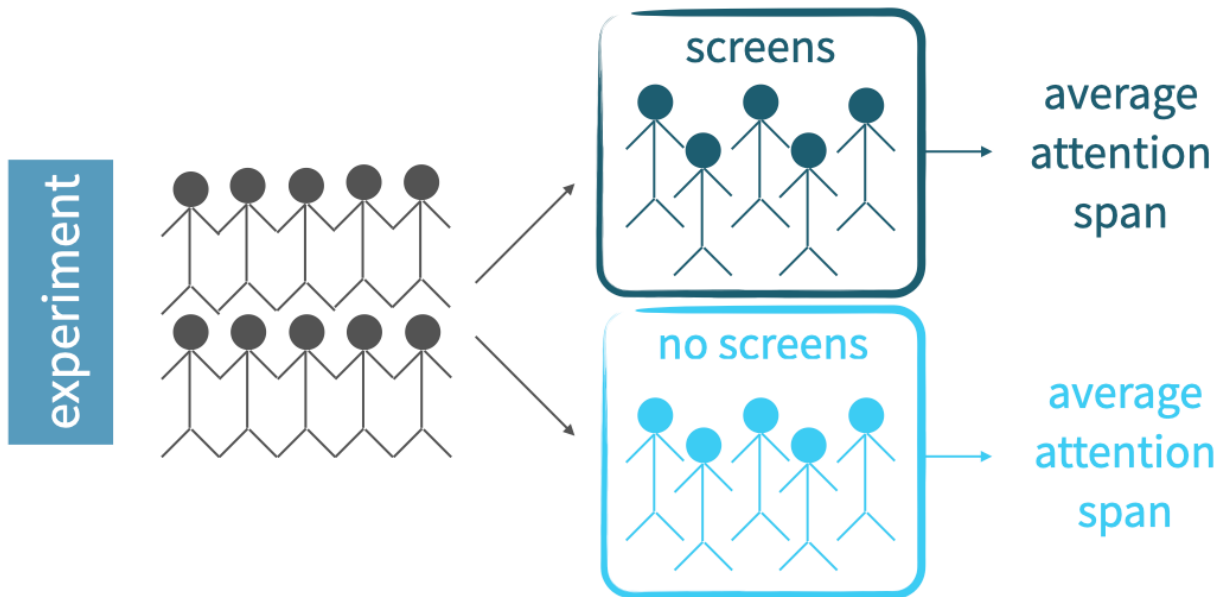
# Observational studies



# Experiments

- In an experiment, we sample a group of people from the population and then we randomly assign these people into two groups: those who are asked to use screens at bedtime and those who asked not to use them.
- The difference is that the decision of whether to use screens or not is not left up to the subjects, as it was in the observational study, but is instead imposed by the researcher.
- At the end, we compare the attention spans of the two groups.

# Experiments



# Experiments

- Based on the observational study, even if we find a difference between the average attention span of these two groups of people, we can't attribute this difference solely to using screens because **there may be other variables that we didn't control for in this study** that contribute to the observed difference.
- For example, people who use screens at night might also be using screens for longer time periods during the day and their attention span might be affected by the daytime usage as well.

# Experiments

- However, in the experiment, such variables that might also contribute to the outcome, called **confounding variables**, are most likely represented equally in the two groups due to random assignment.
- Therefore, if we find a difference between the two averages, we can indeed make a causal statement attributing this difference to bedtime screen usage.



# Random sampling assignment

**Random sampling** occurs when *subjects are being selected for a study*. If subjects are selected randomly from the population of interest, then the resulting sample is likely representative of that population and therefore the study's results can be generalizable to that population.

**Random assignment** occurs only *in experimental settings where subjects are being assigned to various treatments*. Random assignment allows for causal conclusions about the relationship between the treatment and the response.

## Random sampling assignment

Here is a quick summary of how random sampling and random assignment affect the scope of inference of a study's results.

	Random assignment	No random assignment	
Random sampling	Causal and generalizable	Not causal, but generalizable	Generalizable
No random sampling	Causal, but not generalizable	Neither causal nor generalizable	Not generalizable
	Causal	Not causal	

# Random sampling assignment

A study that employs (1) **random sampling** and (2) **random assignment** can be used to make causal conclusions and these conclusions can be generalized to the whole population.

This would be an ideal experiment, but such studies are usually difficult to carry out, especially if the experimental units are humans, since it may be difficult to randomly sample people from the population and then impose treatments on them.

This is why most experiments recruit volunteer subjects. You may have seen ads for these on a university campus or in a newspaper.

# Random sampling assignment

Such human experiments that rely on volunteers employ (1) **random assignment**, but (2) **not random sampling**. These studies can be used to make causal conclusions, but the conclusions only apply to the sample and the results cannot be generalized to the population.

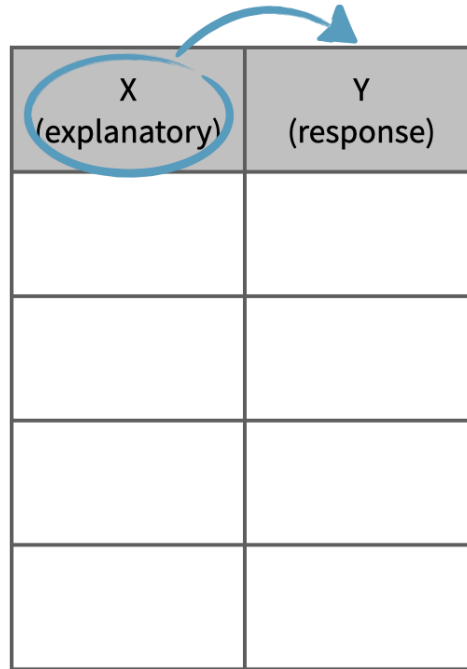
A study that uses (1) **no random assignment**, but does (2) **use random sampling** is your typical observational study. Results can only be used to make association statements, but they can be generalized to the whole population.

A final type of study, one that (1) **doesn't use random assignment** or **random sampling**, can only be used to make non causal association statements. This is an unideal observational study.

# Explanatory and response variables

- Often when one mentions "a relationship between variables" we think of a relationship between just two variables, say a so called **explanatory variable**,  $x$ , and **response variable**,  $y$ .
- Labeling variables as explanatory and response does not guarantee the relationship between the two is actually causal, even if there is an association identified.
- We use these labels only to keep track of which variable we suspect affects the other.

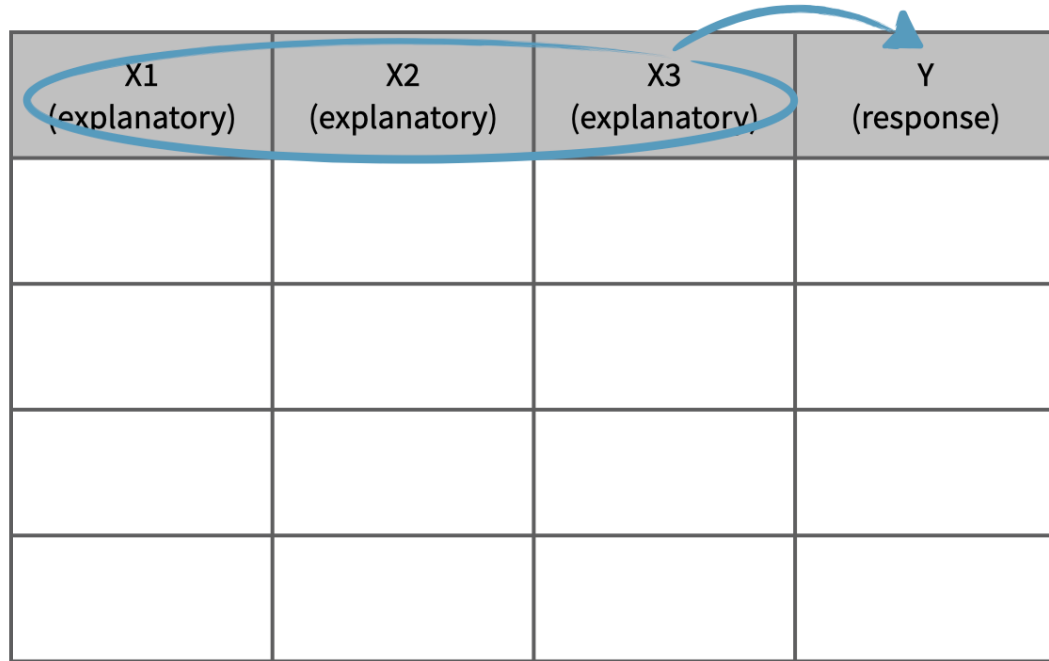
# Explanatory and response variables



A diagram illustrating the relationship between explanatory and response variables. It features a table with two columns. The first column is labeled 'X (explanatory)' and the second column is labeled 'Y (response)'. A blue arrow points from the 'X' column to the 'Y' column, indicating a causal or explanatory relationship. The 'X' column is circled in blue.

X (explanatory)	Y (response)

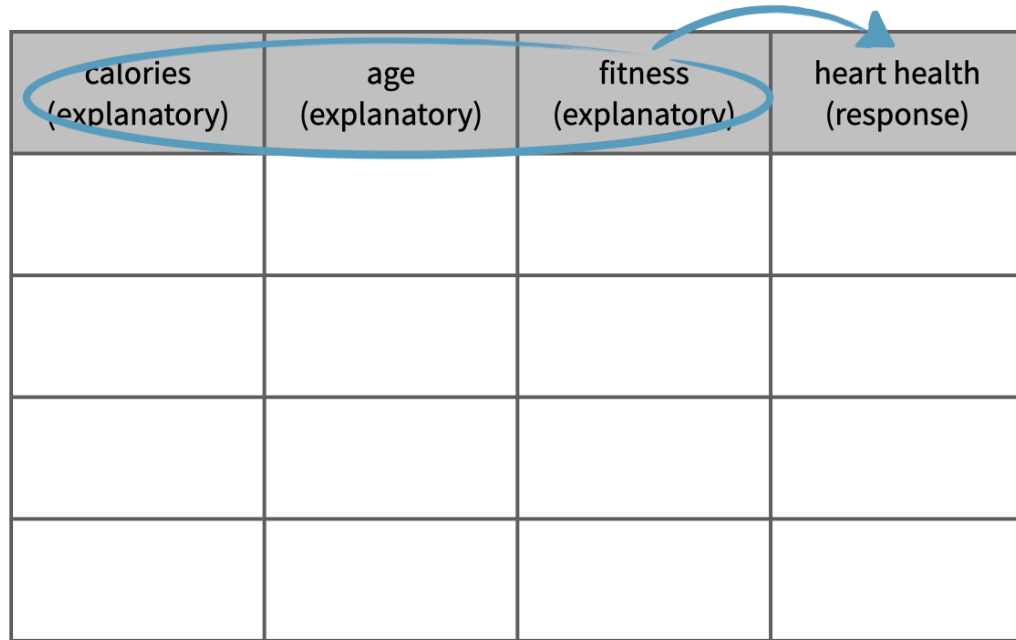
# Multivariate relationships



A diagram illustrating multivariate relationships. It features a table with four columns: X1 (explanatory), X2 (explanatory), X3 (explanatory), and Y (response). The first three columns are grouped by a blue oval, and a blue arrow points from this group to the Y column, indicating that X1, X2, and X3 collectively explain Y.

X1 (explanatory)	X2 (explanatory)	X3 (explanatory)	Y (response)

# Multivariate relationships



The diagram illustrates multivariate relationships using a table with four columns. The first three columns represent explanatory variables: 'calories (explanatory)', 'age (explanatory)', and 'fitness (explanatory)'. The fourth column represents the response variable: 'heart health (response)'. A blue oval encircles the three explanatory variable headers, and a blue arrow points from this oval to the response variable header, indicating that all three variables collectively influence the response.

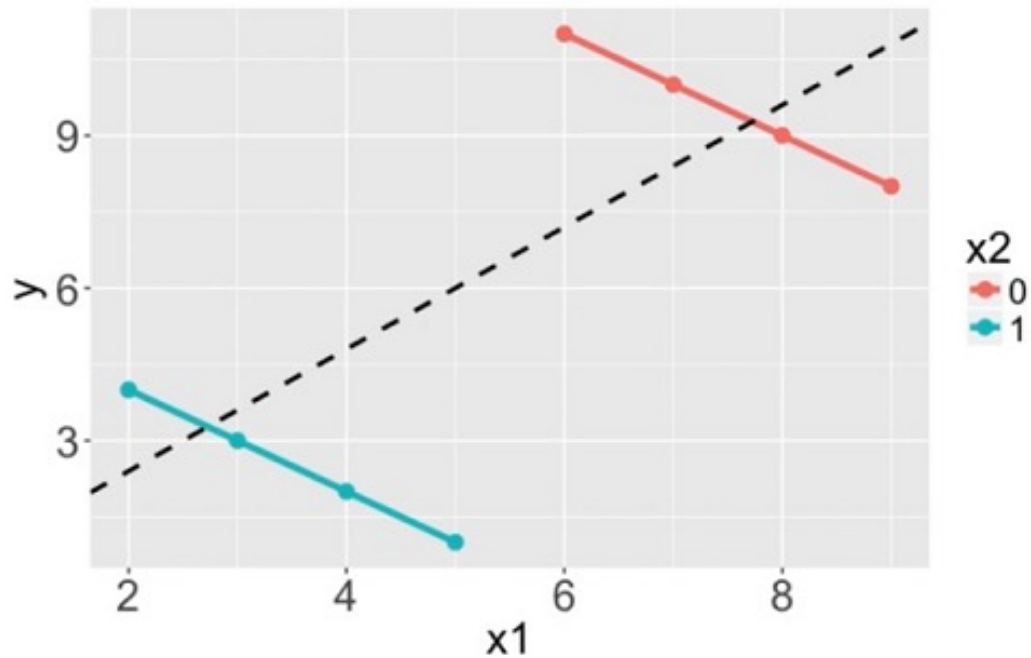
calories (explanatory)	age (explanatory)	fitness (explanatory)	heart health (response)



# Simpson's paradox

- Not considering an important variable when studying a relationship can result in what we call a Simpson's paradox.
- This paradox illustrates the effect the omission of an explanatory variable can have on the measure of association between another explanatory variable and the response variable.
- In other words, the inclusion of a third variable in the analysis can change the apparent relationship between the other two variables.

# Simpson's paradox



# Berkeley admission data

- The goal of this exercise is to determine the number of male and female applicants who got admitted and rejected.
- Specifically, we want to find out how many males are admitted and how many are rejected.
- And similarly we want to find how many females are admitted and how many are rejected.

# Number of males and females admitted

```
library(openintro)
ucb_admit <- as.data.frame(UCBAdmissions)
head(ucb_admit)
```

	Admit	Gender	Dept	Freq
1	Admitted	Male	A	512
2	Rejected	Male	A	313
3	Admitted	Female	A	89
4	Rejected	Female	A	19
5	Admitted	Male	B	353
6	Rejected	Male	B	207

# Number of males and females admitted

- To do so we will use the `count()` function. In one step, `count()` groups the data and then tallies the number of observations in each level of the grouping variable. These counts are available under a new variable called `n`.
- Pass the `Gender` and `Admit` columns from the `ucb_admit` dataset (which is already pre-loaded) into the `count()` function, to count how many students of each gender are admitted and how many are rejected.

# Number of males and females admitted

```
ucb_admit %>%  
  group_by(Gender) %>%  
  count(Admit)  
# A tibble: 4 × 3  
# Groups:   Gender [2]  
  Gender Admit      n  
  <fct>   <fct>  <int>  
1 Male    Admitted    6  
2 Male    Rejected    6  
3 Female  Admitted    6  
4 Female  Rejected    6
```

# Take home message

1. **Identify a data set that you prefer to work with during the course**
2. **Chapter 4 and 5 from the course support**