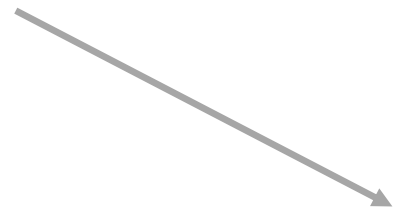# 1. Tabular Data

Tabular Data

The primary way that we will organize data is called the tabular data model. Data are arranged into a grid of rows and columns.

Tabular Data

The primary way that we will organize data is called the tabular data model. Data are arranged into a grid of rows and columns.

the individual entries are called **values**

breed	weight	height
Shih-Tzu	5.5	24
Labrador	33	56
Beagle	10.2	34
Newfoundland	70	69
Chihuahua	1.3	20
Affenpinscher	9.6	27

Tabular Data

The primary way that we will organize data is called the tabular data model. Data are arranged into a grid of rows and columns.

the individual entries are called **values**

Rows ⇒ Observations

each row represents one observation

breed	weight	height
Shih-Tzu	5.5	24
Labrador	33	56
Beagle	10.2	34
Newfoundland	70	69
Chihuahua	1.3	20
Affenpinscher	9.6	27

Tabular Data

The primary way that we will organize data is called the tabular data model. Data are arranged into a grid of rows and columns.

Columns ⇒ Features

each feature has name and a data type

the individual entries are called **values**

Rows ⇒ Observations

each row represents one observation

breed	weight	height
Shih-Tzu	5.5	24
Labrador	33	56
Beagle	10.2	34
Newfoundland	70	69
Chihuahua	1.3	20
Affenpinscher	9.6	27

Tabular Data

Each feature (column) in the tabular data model has a **data type** associated with it. This is not always explicit in way the dataset is saved, but will be defined when we are working with it in R.

The two most common data types we will see are:

↣ **numeric:** everything can be represented by a number
↣ **character:** arbitrary sequences of any characters

There is a single type for each feature; we cannot mix and match data types. We will see other data types as they arise in our work.

Tabular Data

Implied data types in our example:

character **<chr>**	numeric **<dbl>**	numeric **<int>**
↓	↓	↓
breed	**weight**	**height**
Shih-Tzu	5.5	24
Labrador	33	56
Beagle	10.2	34
Newfoundland	70	69
Chihuahua	1.3	20
Affenpinscher	9.6	27

2. Intro to R

Objects

As described in the notes, everything in R is an object. We can save new objects by assigning a value to a name with the arrow (<-) operator:

almost_pi <- 3.14

almost_e <- 2.718

almost_phi <- 1.618

Functions

Functions in R are used to take a number of input objects and generate an output. These range from very simple mathematical functions (sin()) to functions to run long, complex modelling computations.

The inputs to a functions can be set by name or position. Some inputs will have default values that are used if we do not overide them. For example, all of these forms are the same:

```
read_csv(file = "input.csv")
read_csv("input.csv")
read_csv("input.csv", col_names = TRUE, skip = 0)
```

You can see the arguments, names, and default values looking at the documention in the help tab (lower right-side in RStudio).

Pipes

The pipe operator (|>) allows us to pass the output of one function to the first input of another function. It is very useful in code for data science because we often apply a chain of different functions to a dataset. Conside this example that applies a sequence of functions to the number 8:

```
abs(tan(log(exp(8), base = 2)))
```

We can re-write this using pipes as follows:

```
8 |>
  exp() |>
  log(base = 2) |>
  tan() |>
  abs()
```

This is perhaps overkill here, but will be very helpful in future applications.

Formating Code

As mentioned in the notes, formatting code correctly is **very** important. This is perhaps the most common thing I take points off for on the first exam. We'll follow these rules:

    put one space before and after an equal's sign or assignment arrow
    put one space after a comma, but no space before a comma
    put one space around mathematical operations (such as + and *)

And three more dealing with pipes:

    start a chain of pipes with the input dataset
    use a new line after a pipe
    indent every line that follows a pipe with two extra spaces

The best way to follow these formatting rules is to just mimic they way I write code in the notes. It will quickly become second nature.

Naming Things

When doing data science one often has to create names for things. For example, files, features, functions, and objects. To simplify things, we will use a small set of **rules** that we will apply to everything:

  ↣ only use lowercase letters, numbers, and underscores
  ↣ start all names with lowercase letters

You should also try to make names short and simple while being easy to understand and memorize.

Bad   → **WeightOfDogInKilograms**
Better →   **weight_of_dog_in_kgs**
Best   → **weight**

R Markdown

There are three ways to run R code in an R Markdown file:

1. Run an entire chunk of code with the green arrow button or:
   Ctrl + Shift + Enter (Windows) or ⌘ + Shift + Enter (macOS)
2. To run a highlighted chunk of code:
   Ctrl + Enter (Windows) or ⌘ + Enter (macOS)
3. To create an *knit* HTML output of the entire file, click the ball of yarn.