

# SPEC Lab REU R Resources: Intro to R- Vectors

*Alix Ziff with reference to earlier versions by Miriam Barnum and Therese Anders*

*Summer 2020*

## 1 Walk Through Work: Vectors

A **vector** is the simplest type of data you can work with in R. “A vector or a one-dimensional array simply represents a collection of information stored in a specific order.” (Imai 2016: 6) It is essentially a list of data of a single type (either numerical, character, or logical). (See also, Minions <https://www.youtube.com/watch?v=A05n32Bl0aY>).

### 1.1 Variable types

There are four main variable types that you should be familiar with:

- **Numerical:** Well, any number.
- **Character:** We typically store any alphanumeric data that is not ordered as a character vector.
- **Logical:** A collection of TRUE and FALSE values.
- **Factor:** Think about it as an ordinal variable, i.e. an ordered categorical variable.

### 1.2 Vectors

To create a vector we use the function `c()` (“concatenate”) to combine separate data points. The general format for creating a vector in R is as follows:

```
name_of_vector <- c("what you want to put into the vector")
```

Suppose, we have data on the population in millions for 5 the five most populous countries in 2016. We will create a vector with numerical values separated by a comma.

```
c(1379, 1331, 325, 258, 207)
```

```
## [1] 1379 1331 325 258 207
```

Yay our vector now exists! But where? And how do we access it? When we name a vector, it is saved as an *object* (how R stores information) which we can later modify or use with other functions. To assign values to objects, we use the assignment operator `<-`.

```
pop1 <- c(1379, 1331, 325, 258, 207)
```

```
pop1
```

```
## [1] 1379 1331 325 258 207
```

That’s better.

We can use the function `c()` to combine two vectors. Suppose we had data on 5 additional countries.

```
pop2 <- c(194, 187, 161, 142, 127)
```

```
pop <- c(pop1, pop2)
```

```
pop
```

```
## [1] 1379 1331 325 258 207 194 187 161 142 127
```

*What if we want to know more about or add data to our vector? ##Tell me about it* First, lets check which variable type was used to store our population data. The below output tells us that the object `pop` is of class `numeric`, and has the dimensions `[1:10]`, that is 10 elements in one dimension.

```
str(pop)
```

```
## num [1:10] 1379 1331 325 258 207 ...
```

### 1.2.1 Functions

There are a number of special functions that operate on vectors and allow us to compute measures of location and dispersion.

	Function
<code>min()</code>	Returns the minimum of the values or object.
<code>max()</code>	Returns the maximum of the values or object.
<code>sum()</code>	Returns the sum of the values or object.
<code>length()</code>	Returns the length of the values or object.
<code>mean()</code>	Returns the average of the values or object.
<code>median()</code>	Returns the median of the values or object.
<code>var()</code>	Returns the variance of the values or object.
<code>sd()</code>	Returns the standard deviation of the values or object.

What is the average population size from these countries?

```
mean(pop)
```

```
## [1] 431.1
```

How many people total live in all these places?

```
sum(pop)
```

```
## [1] 4311
```

### 1.3 More please!

Suppose, we wanted to add data on the country names. We enter this data in character format. To save time, we will only do this for the five most populous countries.

```
cname <- c("CHN", "IND", "USA", "IDN", "BRA")
str(cname)
```

```
## chr [1:5] "CHN" "IND" "USA" "IDN" "BRA"
```

Now, lets code a logical variable that shows whether the country is in Asia or not.

```
asia <- c(TRUE, TRUE, F, T, F)
str(asia)
```

```
## logi [1:5] TRUE TRUE FALSE TRUE FALSE
```

Lastly, we define a factor variable for the regime type that can take one of four values (based on Economist Intelligence Unit): Full Democracy, Flawed Democracy, Hybrid Regimes, Autocracies. Note that empirically, we don't have a "hybrid category" here. We could define an empty factor level, but we will skip this step here.

```
regime <- c("Autocracy", "FlawedDem", "FullDem", "FlawedDem", "FlawedDem")
regime <- as.factor(regime)
str(regime)
```

```
## Factor w/ 3 levels "Autocracy","FlawedDem",...: 1 2 3 2 2
```