# Create a vector

Feeling lucky? You better, because this part takes you on a trip to the City of Sins, also known as Statisticians Paradise!

Thanks to R and your new data-analytical skills, you will learn how to uplift your performance at the tables and fire off your career as a professional gambler. This chapter will show how you can easily keep track of your betting progress and how you can do some simple analyses on past actions. Next stop, VEGAS!!

##Instruction

Do you still remember what you have learned in the first part? Assign the value "Go!" to the variable vegas. Remember: R is case sensitive!

# Create a vector (2)

Let us focus first!

On your way from rags to riches, you will make extensive use of vectors. Vectors are one-dimension arrays that can hold numeric data, character data, or logical data. In other words, a vector is a simple tool to store data. For example, you can store your daily gains and losses in the casinos.

In R, you create a vector with the combine function [**c()**](http://www.rdocumentation.org/packages/base/functions/c). You place the vector elements separated by a comma between the parentheses. For example:

numeric\_vector <- c(1, 2, 3)

character\_vector <- c("a", "b", "c")

Once you have created these vectors in R, you can use them to do calculations.

##Instructions

Complete the code such that boolean\_vector contains the three elements: TRUE, FALSE and TRUE (in that order).

**Create a vector (3)**

After one week in Las Vegas and still zero Ferraris in your garage, you decide that it is time to start using your data analytical superpowers.

Before doing a first analysis, you decide to first collect all the winnings and losses for the last week:

For poker\_vector:

* On Monday you won $140
* Tuesday you lost $50
* Wednesday you won $20
* Thursday you lost $120
* Friday you won $240

For roulette\_vector:

* On Monday you **lost** $24
* Tuesday you **lost** $50
* Wednesday you won $100
* Thursday you **lost** $350
* Friday you won $10

You only played poker and roulette, since there was a delegation of mediums that occupied the craps tables. To be able to use this data in R, you decide to create the variables poker\_vector and roulette\_vector.

#Instructions

Assign the winnings/losses for roulette to the variable roulette\_vector.

# Naming a vector

As a data analyst, it is important to have a clear view on the data that you are using. Understanding what each element refers to is therefore essential.

In the previous exercise, we created a vector with your winnings over the week. Each vector element refers to a day of the week but it is hard to tell which element belongs to which day. It would be nice if you could show that in the vector itself.

You can give a name to the elements of a vector with the names() function. Have a look at this example:

some\_vector <- c("John Doe", "poker player")

names(some\_vector) <- c("Name", "Profession")

This code first creates a vector some\_vector and then gives the two elements a name. The first element is assigned the name Name, while the second element is labeled Profession. Printing the contents to the console yields following output:

Name Profession

"John Doe" "poker player"

#Instruction

The code on the right names the elements in poker\_vector with the days of the week. Add code to do the same thing for roulette\_vector

# Naming a vector (2)

If you want to become a good biological data analyst, you have to become lazy. (If you are already lazy, chances are high you are one of those exceptional, natural-born Bioinformatician talents.)

In the previous exercises you probably experienced that it is boring and frustrating to type and retype information such as the days of the week. However, when you look at it from a higher perspective, there is a more efficient way to do this, namely, to assign the days of the week vector to a **variable**!

Just like you did with your poker and roulette returns, you can also create a variable that contains the days of the week. This way you can use and re-use it.

##Instruction

* A variable days\_vector that contains the days of the week has already been created for you.
* Use days\_vector to set the names of poker\_vector and roulette\_vector.

**Calculating total winnings**

Now that you have the poker and roulette winnings nicely as named vectors, you can start doing some data analytical magic.

You want to find out the following type of information:

* How much has been your overall profit or loss per day of the week?
* Have you lost money over the week in total?
* Are you winning/losing money on poker or on roulette?

To get the answers, you have to do arithmetic calculations on vectors.

It is important to know that if you sum two vectors in R, it takes the element-wise sum. For example, the following three statements are completely equivalent:

c(1, 2, 3) + c(4, 5, 6)

c(1 + 4, 2 + 5, 3 + 6)

c(5, 7, 9)

You can also do the calculations with variables that represent vectors:

a <- c(1, 2, 3)

b <- c(4, 5, 6)

c <- a + b

#Instruction

* Take the sum of the variables A\_vector and B\_vector and assign it to total\_vector.
* Inspect the result by printing out total\_vector.

# Calculating total winnings (2)

Now you understand how R does arithmetic with vectors, it is time to get those Ferraris in your garage! First, you need to understand what the overall profit or loss per day of the week was. The total daily profit is the sum of the profit/loss you realized on poker per day, and the profit/loss you realized on roulette per day.

In R, this is just the sum of roulette\_vector and poker\_vector.

#Instruction

Assign to the variable total\_daily how much you won or lost on each day in total (poker and roulette combined).

# Calculating total winnings (3)

Based on the previous analysis, it looks like you had a mix of good and bad days. This is not what your ego expected, and you wonder if there may be a very tiny chance you have lost money over the week in total?

A function that helps you to answer this question is [**sum()**](http://www.rdocumentation.org/packages/base/functions/sum). It calculates the sum of all elements of a vector. For example, to calculate the total amount of money you have lost/won with poker you do:

total\_poker <- sum(poker\_vector)

#Instruction

* Calculate the total amount of money that you have won/lost with roulette and assign to the variable total\_roulette.
* Now that you have the totals for roulette and poker, you can easily calculate total\_week (which is the sum of all gains and losses of the week).
* Print out total\_week.

# Comparing total winnings

Oops, it seems like you are losing money. Time to rethink and adapt your strategy! This will require some deeper analysis...

After a short brainstorm in your hotel's jacuzzi, you realize that a possible explanation might be that your skills in roulette are not as well developed as your skills in poker. So maybe your total gains in poker are higher (or > ) than in roulette.

##Instruction

* Calculate total\_poker and total\_roulette as in the previous exercise. Use the sum() function twice.
* Check if your total gains in poker are higher than for roulette by using a comparison. Simply print out the result of this comparison. What do you conclude, should you focus on roulette or on poker?

# Vector selection: the good times

Your hunch seemed to be right. It appears that the poker game is more your cup of tea than roulette.

Another possible route for investigation is your performance at the beginning of the working week compared to the end of it. You did have a couple of Margarita cocktails at the end of the week...

To answer that question, you only want to focus on a selection of the total\_vector. In other words, our goal is to select specific elements of the vector. To select elements of a vector (and later matrices, data frames, ...), you can use square brackets. Between the square brackets, you indicate what elements to select. For example, to select the first element of the vector, you type poker\_vector[1]. To select the second element of the vector, you type poker\_vector[2], etc. Notice that the first element in a vector has index 1, not 0 as in many other programming languages.

##Instructions

Assign the poker results of Wednesday to the variable poker\_wednesday.

# Vector selection: the good times (2)

How about analyzing your midweek results?

To select multiple elements from a vector, you can add square brackets at the end of it. You can indicate between the brackets what elements should be selected. For example: suppose you want to select the first and the fifth day of the week: use the vector c(1, 5) between the square brackets. For example, the code below selects the first and fifth element of poker\_vector:

poker\_vector[c(1, 5)]

#Instructions

Assign the poker results of Tuesday, Wednesday and Thursday to the variable poker\_midweek.

# Vector selection: the good times (3)

Selecting multiple elements of poker\_vector with c(2, 3, 4) is not very convenient. Many bioinformaticians are lazy people by nature, so they created an easier way to do this: c(2, 3, 4) can be abbreviated to2:4, which generates a vector with all natural numbers from 2 up to 4.

So, another way to find the mid-week results is poker\_vector[2:4]. Notice how the vector 2:4 is placed between the square brackets to select element 2 up to 4.

##Instruction

Assign to roulette\_selection\_vector the roulette results from Tuesday up to Friday; make use of : if it makes things easier for you.

# Vector selection: the good times (4)

Another way to tackle the previous exercise is by using the names of the vector elements (Monday, Tuesday, ...) instead of their numeric positions. For example,

poker\_vector["Monday"]

will select the first element of poker\_vector since "Monday" is the name of that first element.

Just like you did in the previous exercise with numerics, you can also use the element names to select multiple elements, for example:

poker\_vector[c("Monday","Tuesday")]

#Instruction

* Select the first three elements in poker\_vector by using their names: "Monday", "Tuesday" and "Wednesday". Assign the result of the selection to poker\_start.
* Calculate the average of the values in poker\_start with the [**mean()**](http://www.rdocumentation.org/packages/base/functions/mean) function. Simply print out the result so you can inspect it.

**Selection by comparison - Step 1**

By making use of comparison operators, we can approach the previous question in a more proactive way.

The (logical) comparison operators known to R are:

* < for less than
* > for greater than
* <= for less than or equal to
* >= for greater than or equal to
* == for equal to each other
* != not equal to each other

As seen in the previous part, stating 6 > 5 returns TRUE. The nice thing about R is that you can use these comparison operators also on vectors. For example:

> c(4, 5, 6) > 5

[1] FALSE FALSE TRUE

This command tests for every element of the vector if the condition stated by the comparison operator is TRUE or FALSE.

#Instructions

* Check which elements in poker\_vector are positive (i.e. > 0) and assign this to selection\_vector.
* Print out selection\_vector so you can inspect it. The printout tells you whether you won (TRUE) or lost (FALSE) any money for each day.

# Selection by comparison - Step 2

Working with comparisons will make your data analytical life easier. Instead of selecting a subset of days to investigate yourself (like before), you can simply ask R to return only those days where you realized a positive return for poker.

In the previous exercises you used selection\_vector <- poker\_vector > 0 to find the days on which you had a positive poker return. Now, you would like to know not only the days on which you won, but also how much you won on those days.

You can select the desired elements, by putting selection\_vector between the square brackets that follow poker\_vector:

poker\_vector[selection\_vector]

R knows what to do when you pass a logical vector in square brackets: it will only select the elements that correspond to TRUE in selection\_vector.

#Instructions

Use selection\_vector in square brackets to assign the amounts that you won on the profitable days to the variable poker\_winning\_days.

# Advanced selection

Just like you did for poker, you also want to know those days where you realized a positive return for roulette.

#Instruction

* Create the variable selection\_vector, this time to see if you made profit with roulette for different days.
* Assign the amounts that you made on the days that you ended positively for roulette to the variable roulette\_winning\_days. This vector thus contains the positive winnings of roulette\_vector.