

Solutions

First assignment

What is the sum of the first 100 positive integers?

```
n = 100
result = n*(n+1)/2
Result
5050
```

Use the accessor \$ to extract the state abbreviations and assign them to the object a. What is the class of this object?

```
a = murders$abb
class(a)
[1] "character"
```

Now use the square brackets to extract the state abbreviations and assign them to the object b. Use the identical function to determine if a and b are the same.

b = murders[names(murders) == 'abb']	b = murders[,2]
class(b)	class(b)
[1] "data.frame"	[1] "character"
class(a)==class(b)	class(a)==class(b)
[1] FALSE	[1] TRUE

First assignment

The function `table` takes a vector and returns the frequency of each element. You can quickly see how many states are in each region by applying this function. Use this function in one line of code to create a table of states per region.

```
table(murders$region)
## ## Northeast South North Central West
## 9 17 12 13
```

Create two vectors of different dimensions and insert the second one in the first one between the 2nd and 3rd elements.

```
x = c(1,2,3,4)
y = c(5,6,7)
x = c(x[1:2],y,x[3:length(x)])
x
## [1] 1 2 5 6 7 3 4
```

First assignment

Draw 100 numbers from a Uniform distribution on [0,1] and count how many values are larger than 0.5

```
sum(runif(100)>0.5)
```

Compute the per 100,000 murder rate for each state and store it in the object `murder_rate`. Then compute the average murder rate for the US using the function `mean`. What is the average?

```
murder_rate = murders$total/murders$population*100000  
mean(murder_rate)  
## [1] 2.779125
```

Write a script allowing to load a vector file and “remove” the missing values.

```
vector<-read_csv("./vector.csv")  
vector  
## 2,3NA,5,4)  
na.omit(vector)  
vector=vector[!is.na(vector)]  
vector  
## 2,3,5,4
```

First assignment

Create a histogram of the state populations.

```
hist(murders$population)
```

Generate boxplots of the state populations by region.

```
boxplot(population~region,data=murders)
```

Create a function that normalizes a vector:

```
norm_func= function(x){  
  return ((x-mean(x))/sd(x))  
}
```

Use this function on the iris dataset so that each column is normalized

```
data("iris")  
sapply(iris[1:4], function(x) norm_func(x))
```

```
iris1 = norm_func(iris[,1])  
iris2 = norm_func(iris[,2])  
iris3 = norm_func(iris[,3])  
iris4 = norm_func(iris[,4])  
Iris_norm = data.frame(iris1,iris2,iris3,iris4,iris[,5])
```