http://tryr.codeschool.com 에서 R을 배울 수 있습니다. 회원가입을 해야 진행사항을 기록하고 다음번에 이어서 할 수 있습니다. 답을 입력할 때에 입력 모드는 영문으로 되어 있어야 합니다. 한글일 경우에는 입력되지 않습니다.

배열은 1부터 시작합니다. 따라서 try accessing the sixth word of the sentence vector 의 답은 `sentence[6]` 입니다.

CHAPTER 2 - Vectors

Vector Names

```
> ranks <- 1:3
> names(ranks) <- c("first", "second", "third")
> ranks
 first second  third
     1      2      3
> ranks["first"]
first
    1
```

ggplot2

```
> install.packages("ggplot2")
> help(package = "ggplot2")
                Information on package 'ggplot2'

Description:

Package:            ggplot2
Type:               Package
Title:              An implementation of the Grammar of Graphics
Version:            0.9.1

...
```

```
> weights <- c(300, 200, 100, 250, 150)
> prices <- c(9000, 5000, 12000, 7500, 18000)
> chests <- c('gold', 'silver', 'gems', 'gold', 'gems')
> types <- factor(chests)
> library(ggplot2)
> qplot(weights, prices, color = types)
```

JTvMzN

CHAPTER 8 - What's Next

Plotting One Vector

```
> vesselsSunk <- c(4, 5, 1)
> barplot(vesselsSunk)
```

RyVejk

```
> barplot(1:100)
```

XlqRzx

Scatter Plots

```
x <- seq(1, 20, 0.1)
y <- sin(x)
plot(x,y)
---
values <- -10:10
absolutes <- abs(values)
plot(values, absolutes)
```

NA Values

```
> a <- c(1, 3, NA, 7, 9)
> sum(a)
[1] NA
> sum(a, na.rm = TRUE)
[1] 20
```

CHAPTER 3 - Matrices

```
> matrix(0, 3, 4)
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
```

```
> a <- 1:12
> print(a)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

```
> matrix(a, 3, 4)
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

```
> plank <- 1:8
> dim(plank) <- c(2,4)
> print(plank)
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```

```
> matrix(1, 5, 5)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1    1    1
[2,]    1    1    1    1    1
[3,]    1    1    1    1    1
[4,]    1    1    1    1    1
[5,]    1    1    1    1    1
```

```
> print(plank)
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
> plank[2,3]
[1] 6
> plank[1,4]
[1] 7
> plank[1,4] <- 0
> plank[2,]
[1] 2 4 6 8
> plank[,4]
[1] 7 8
> plank[, 2:4]
     [,1] [,2] [,3]
[1,]    3    5    7
[2,]    4    6    8
```

Matrix Plotting

```
> elevation <- matrix(1, 10, 10)
> elevation[4,6] <- 0
> contour(elevation)
```

contour

```
> persp(elevation)
```

persp

```
> persp(elevation, expand=0.2)
```

persp expand

```
> contour(volcano)
```

nj0A8b

```
> persp(volcano, expand=0.2)
```

rzG9ps

```
> image(volcano)
```

UkWseY

CHAPTER 4 - Summary Statistics

Mean

```
> limbs <- c(4, 3, 4, 3, 2, 4, 4, 4)
> names(limbs) <- c('One-Eye', 'Peg-Leg', 'Smitty', 'Hook', 'Scooter',
'Dan', 'Mikey', 'Blackbeard')
> mean(limbs)
[1] 3.5
> barplot(limbs)
```

barplot

```
> abline(h=mean(limbs))
```

abline

Median

```
> limbs <- c(4, 3, 4, 3, 2, 4, 4, 14)
> names(limbs) <- c('One-Eye', 'Peg-Leg', 'Smitty', 'Hook',
                    'Scooter', 'Dan', 'Mikey', 'Davy Jones')
```

```
> mean(limbs)
[1] 4.75
> median(limbs)
[1] 4
```

ohLzBU

Standard Deviation

```
> pounds <- c(45000, 50000, 35000, 40000, 35000, 45000, 10000, 15000)
> barplot(pounds)
> meanValue <- mean(pounds)
> abline(h= meanValue)
```

meanValue

```
> deviation <- sd(pounds)
> abline(h = meanValue + deviation)
```

iFMXvl

CHAPTER 5 - Factors

Creating Factors

```
> chests <- c('gold', 'silver', 'gems', 'gold', 'gems')
> types <- factor(chests)
> print(chests)
[1] "gold"   "silver" "gems"   "gold"   "gems"
> print(types)
[1] gold   silver gems   gold   gems
Levels: gems gold silver
> as.integer(types)
[1] 2 3 1 2 1
> levels(types)
[1] "gems"   "gold"   "silver"
```

Plots With Factors

```
> weights <- c(300, 200, 100, 250, 150)
> prices <- c(9000, 5000, 12000, 7500, 18000)
> plot(weights, prices)
```

Cmj0Wp

```
> plot(weights, prices, pch=as.integer(types))
```

CHAPTER 7 - Real-World Data

Some Real World Data

```
Country,Piracy
Australia,23
Bangladesh,90
Brunei,67
China,77
...
> piracy <- read.csv("piracy.csv")

Rank    Country         GDP
1       Liechtenstein   141100
2       Qatar           104300
3       Luxembourg      81100
4       Bermuda         69900
...
> gdp <- read.table("gdp.txt", sep="  ", header=TRUE)

> countries <- merge(x = gdp, y = piracy)
> plot(countries$GDP, countries$Piracy)
```

oMtDV8

관련 문서

- CodeSchool