

# Intro to R

## Basic R Commands

```
2 + 3
```

```
## [1] 5
```

```
4 + 5 * 10
```

```
## [1] 54
```

```
sqrt(25)
```

```
## [1] 5
```

```
cos(pi)
```

```
## [1] -1
```

```
log10(100)
```

```
## [1] 2
```

## Variables and Vectors

```
x <- 5
```

```
x
```

```
## [1] 5
```

```
x = 10
```

```
x
```

```
## [1] 10
```

```
x + 2
```

```
## [1] 12
```

```
y <- x * 3
```

```
y
```

```
## [1] 30
```

```
x <- c(1, 2, 3, 4)
x + 2
```

```
## [1] 3 4 5 6
```

```
x * 3
```

```
## [1] 3 6 9 12
```

```
x <- 1:5
length(x)
```

```
## [1] 5
```

```
x <- rnorm(3)
x
```

```
## [1] 0.1461856 1.3770650 -0.1541668
```

```
x <- rnorm(3, mean = 4, sd = 20)
x
```

```
## [1] -15.923573 -13.455512 5.389015
```

## Built-In Functions in R

```
mean(c(1, 4, 5, 2, 8, 2))
```

```
## [1] 3.666667
```

```
max(c(1, 4, 5, 2, 8, 2))
```

```
## [1] 8
```

```
min(c(1, 4, 5, 2, 8, 2))
```

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

## Matrices

```
mat = matrix(data = 1:9, nrow = 3, ncol = 3, byrow = FALSE)
mat
```

```
##      [,1] [,2] [,3]
## [1,] 1    4    7
## [2,] 2    5    8
## [3,] 3    6    9
```

```
mat[1, 3]
```

```
## [1] 7
```

```
mat[1, ]
```

```
## [1] 1 4 7
```

```
mat[, 3]
```

```
## [1] 7 8 9
```

## Creating functions in R

```
testFunction <- function(x) {  
  print(x)  
}
```

```
testFunction(3011)
```

```
## [1] 3011
```

```
rm(testFunction)
```

```
testFunction <- function(x) {  
  cat(x, "is your input number")  
}
```

```
testFunction(3011)
```

```
## 3011 is your input number
```

```
testFunction <- function(x) {  
  count <- 0  
  for (i in 0:x) {  
    if (i%5 == 0) {  
      count <- count + 1  
    }  
  }  
  cat("The number of numbers divisible by 5 between 0 and", x, "is", count)  
}
```

```
testFunction(38)
```

```
## The number of numbers divisible by 5 between 0 and 38 is 8
```

```
rm(testFunction)

testFunction <- function(x, y) {
  output <- 0
  while (x > 0) {
    output <- output + y
    x <- x - 1
  }
  print(output)
}

testFunction(2, 5)
```

```
## [1] 10
```

## Working with Data Sets

```
# Install alr3 package
install.packages("alr3")
```

```
# Call alr3 package
library(alr3)
```

```
## Loading required package: car
```

```
# Name the htw data set to 'data'
data = htw
```

```
# Summarize data
summary(htwt)
```

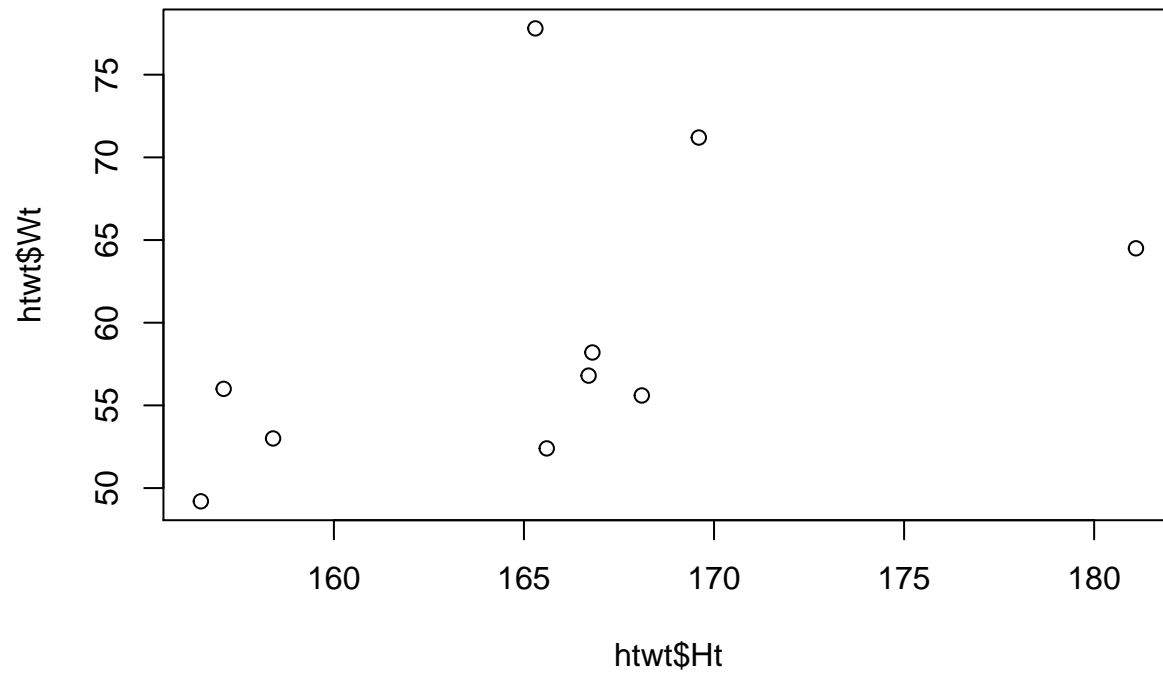
```
##           Ht           Wt
## Min.      :156.5   Min.      :49.20
## 1st Qu.:160.1   1st Qu.:53.65
## Median :166.2   Median :56.40
## Mean     :165.5   Mean      :59.47
## 3rd Qu.:167.8   3rd Qu.:62.92
## Max.     :181.1   Max.       :77.80
```

```
# Display first 7 lines of htw data set
head(htwt)
```

```
##      Ht   Wt
## 1 169.6 71.2
## 2 166.8 58.2
## 3 157.1 56.0
## 4 181.1 64.5
## 5 158.4 53.0
## 6 165.6 52.4
```

## Simple Plot

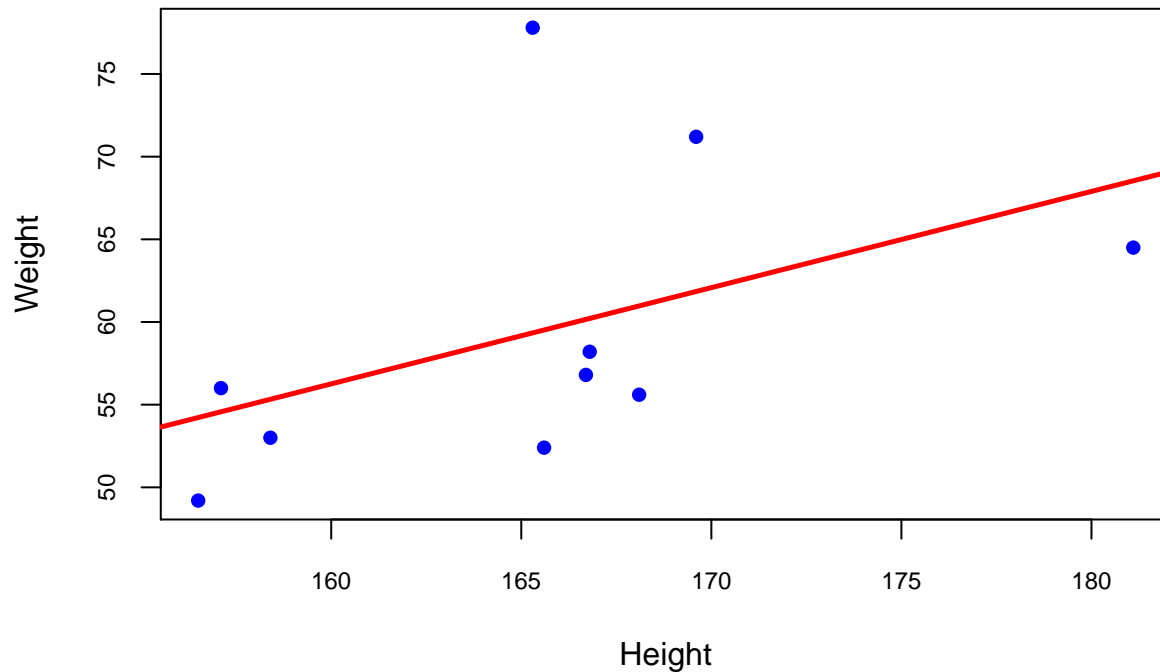
```
# Plot variable 'Ht' on the x-axis and 'Wt' on the y-axis  
plot(htwt$Ht, htwt$Wt)
```



## Detailed Plot

```
# Detailed plot  
plot(htwt$Wt ~ htwt$Ht, main = "Plot of Height vs. Weight", xlab = "Height",  
      ylab = "Weight", pch = 16, col = "blue", type = "p", cex.axis = 0.75, cex.lab = 1,  
      cex.main = 1)  
linebestfit = lm(Wt ~ Ht, data = data)  
abline(linebestfit, col = "red", lwd = 2.5)
```

Plot of Height vs. Weight



```
# Summarize line of best fit
summary(linebestfit)
```

```
##
## Call:
## lm(formula = Wt ~ Ht, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.1166 -4.7744 -2.8412  0.5696 18.4581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.8759    64.4728  -0.572   0.583
## Ht           0.5821     0.3892   1.496   0.173
##
## Residual standard error: 8.456 on 8 degrees of freedom
## Multiple R-squared:  0.2185, Adjusted R-squared:  0.1208
## F-statistic: 2.237 on 1 and 8 DF, p-value: 0.1731
```

## Practice Using R

Need more practice? Download the “swirl” package.

Swirl is an interactive courses that teaches you how to program in R.

```
# Install package  
install.packages("swirl")
```

First try the “R Programming” lesson

List of lessons:

- R Programming
- Data Analysis
- Regression Models
- Getting and Cleaning Data
- Statistical Inference

```
# Call package  
library(swirl)  
  
# Download lessons  
install_from_swirl("R Programming")
```

Once the lessons are installed...

```
# Activate interactive course  
swirl()
```

Follow the prompts and have fun!

Questions regarding Swirl: [swirlstats.com](http://swirlstats.com)

More questions? Feel free to get in touch.

gall0441 at umn dot edu