# Magrittr

## The Pipe %>%

```
# General operation moving forward

c(5,4,2,6,3,4,3,4,6) %>%
    mean()
```

```
## [1] 4.111111
```

```
# The pipe is like the unix command line pipe,
#   take output of left and pipe into the right
# The pipe in R differs in that you can break lines
#   and it doesnt need to be STDIN or STDOUT

#-------------------------------------------------#

# Taking a forward moving operation and assigning it to a variable

# read this as:
#   "X receives the value of the vector going through the mean function"

x <- c(5,4,2,6,3,4,3,4,6) %>%
    mean()
cat("the value of x is: ", x)
```

```
## the value of x is:  4.111111
```

## The Compound Pipe %<>%

```
# Normal pipe pushes results forward through processes
#   the compound pipe pushes the operation forward but also assigns the result
#       back to the initial object, basically it becomes '<-'

# first y contains this vector of numbers
y <- c(5,4,2,6,3,4,3,4,6)

# Then we compound pipe that vector into the mean function and the result of
#   the mean function is then re-assigned to y, akin to:
#       y = mean(y)
y %<>% mean()

cat("the value of y is: ", y)
```

```
## the value of y is:  4.111111
```

## The Exposition Pipe %$%

```
# When piping this tribble forward it assumes i want to
#   calculate the mean of both vectors

tibble::tribble(
    ~name, ~age,
    "john", 26,
    "karen", 54,
    "susan", 45,
    "joe", 16
) %>%
    mean(age) # this does not work
```

```
## Warning in mean.default(., age): argument is not numeric or logical: returning
## NA
```

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

```
#-------------------------------------------------#

# when using the exposition pipe %$% it allows you to use the names of the
#   content on the 'left hand side' of the piping operation and therefore
#       reference only the numeric vector in the mean function.

# so what appears in the mean function is equivalently exampleTribble$age

tibble::tribble(
    ~name, ~age,
    "john", 26,
    "karen", 54,
    "susan", 45,
    "joe", 16
) %$%
    mean(age) # and so this works
```

```
## [1] 35.25
```

## the Tee pipe

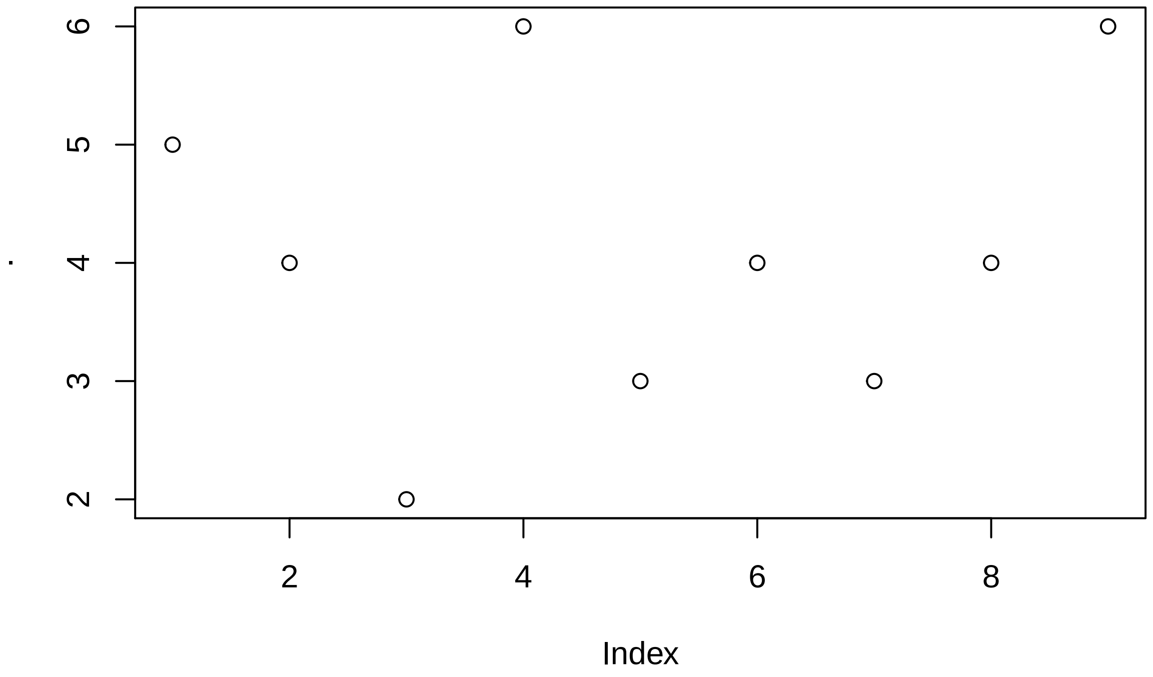
```
# pipe a a value forward but return the original value

#        value -------------> into a function (This is not returned)
#                    |
#                    |
#                    |
#                    |
#                    |
#                    V
#                    Returned Result


# value is piped into the mean function but only vector is returned
c(5,4,2,6,3,4,3,4,6) %T>% mean() # Shows only the vector
```

```
## [1] 5 4 2 6 3 4 3 4 6
```

```r
# This is useful when an expression is used for its side-effect, say plotting or printing.
c(5,4,2,6,3,4,3,4,6) %T>% plot() # shows the vector and the plot
```



```
## [1] 5 4 2 6 3 4 3 4 6
```