01 Basic R for Finance

Boni

9/1/2020

R Basic

Working with Matrix

We can create matrix from one atomic vector.

```
x_vec <- c(9, 4, 6, 20, 19, 29)
x_mat <- matrix(data = x_vec, nrow = 2, ncol = 3, byrow = TRUE)
x_mat
```

```
##      [,1] [,2] [,3]
## [1,]    9    4    6
## [2,]   20   19   29
```

```
cor(x_mat)
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    1    1    1
## [3,]    1    1    1
```

We can also create matrix from 2 vectors.

```
y_vec = rnorm(6, 23, 98)

matrix_yx <- cbind(x_vec, y_vec)
matrix_yx
```

```
##      x_vec       y_vec
## [1,]     9 -103.566037
## [2,]     4   97.186556
## [3,]     6   93.648759
## [4,]    20  200.900375
## [5,]    19    4.389168
## [6,]    29  -82.531184
```

```
cor(matrix_yx)
```

```
##             x_vec      y_vec
## x_vec  1.0000000 -0.2311023
## y_vec -0.2311023  1.0000000
```

Working with DataFrame

```
cash_flow <- rnorm(10, 4, 9)
year <- runif(10, 5, 45)
company <- c("Google", "Google", "Microsoft", "Microsoft", "Apple", "Apple", "Google", "Google", "Micros
company_data <- data.frame(company, cash_flow, year)
company_data
```

```
##      company   cash_flow     year
## 1     Google   0.1179384 23.93724
## 2     Google -18.8401007 44.20072
## 3  Microsoft   5.8977452 16.19619
## 4  Microsoft  11.6878687 11.08862
## 5      Apple  13.0589085 20.34417
## 6      Apple  -3.2142581 23.29475
## 7     Google   8.1618710 42.19195
## 8     Google   8.5676989 25.44022
## 9  Microsoft  -6.4012936 34.10155
## 10 Microsoft   6.2224520 36.77092
```

```
# sub-setting the DF
company_data[1:3,1, drop=FALSE]
```

```
##     company
## 1    Google
## 2    Google
## 3 Microsoft
```

```
company_data$cash_flow
```

```
##  [1]   0.1179384 -18.8401007   5.8977452  11.6878687  13.0589085  -3.2142581
##  [7]   8.1618710   8.5676989  -6.4012936   6.2224520
```

```
subset(company_data, cash_flow < 6.00)
```

```
##     company   cash_flow     year
## 1    Google   0.1179384 23.93724
## 2    Google -18.8401007 44.20072
## 3 Microsoft   5.8977452 16.19619
## 6     Apple  -3.2142581 23.29475
## 9 Microsoft  -6.4012936 34.10155
```

```
# Delete a column
company_data$year = NULL
company_data
```

```
##      company   cash_flow
## 1     Google   0.1179384
## 2     Google -18.8401007
## 3  Microsoft   5.8977452
## 4  Microsoft  11.6878687
## 5      Apple  13.0589085
## 6      Apple  -3.2142581
## 7     Google   8.1618710
## 8     Google   8.5676989
## 9  Microsoft  -6.4012936
## 10 Microsoft   6.2224520
```

Working with Factor

```
# Factor will make R treat string as integer.
investment <- c("stock", "bonds", "stock", "bonds", "stock")
investment_factor <- factor(investment)
# Summary of factor is more informative.
summary(investment)
```

```
##    Length     Class      Mode 
##         5 character character
```

```
summary(investment_factor)
```

```
## bonds stock 
##     2     3
```

```
# Change the levels.
investment
```

```
## [1] "stock" "bonds" "stock" "bonds" "stock"
```

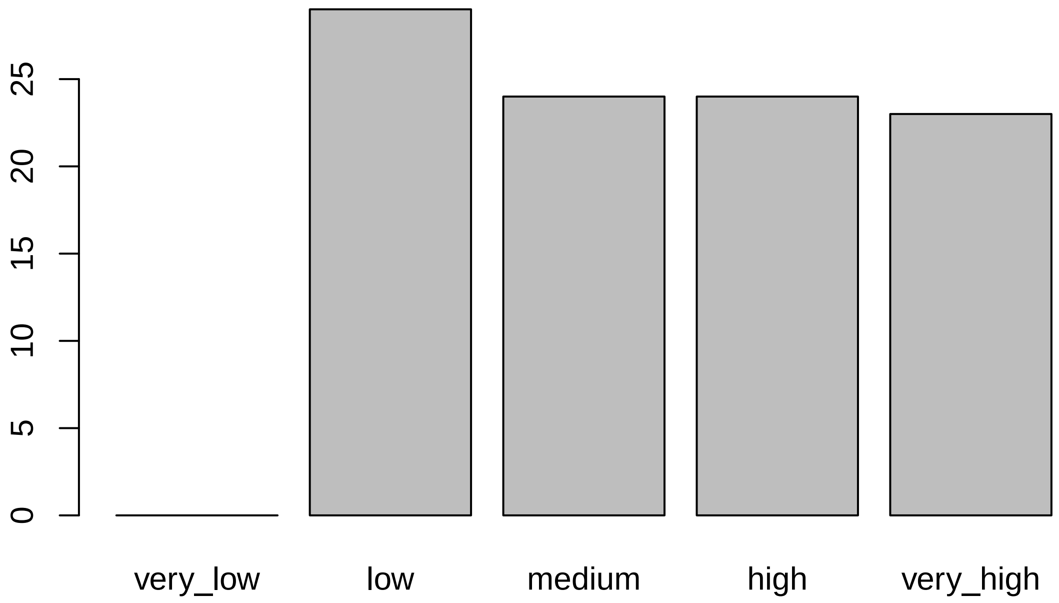
```
levels(investment) <- c("B", "S")
investment
```

```
## [1] "stock" "bonds" "stock" "bonds" "stock"
## attr(,"levels")
## [1] "B" "S"
```

```
# cut() will factoring the numbers based on interval.
price <- runif(100, 20, 100)
breaks <- c(0, 20, 40, 60, 80, 100)
grouped_price <- cut(price, breaks = breaks)
head(grouped_price)
```

```
## [1] (20,40]  (40,60]  (80,100] (80,100] (40,60]  (60,80]
## Levels: (0,20] (20,40] (40,60] (60,80] (80,100]
```

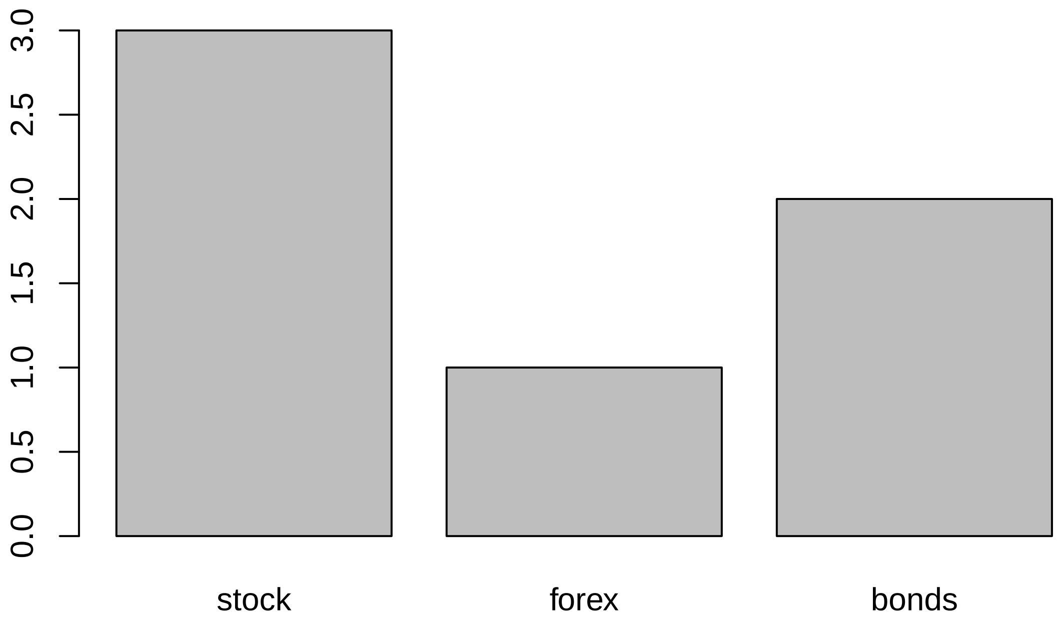
```
levels(grouped_price) <- c("very_low", "low", "medium", "high", "very_high")
# Plot the factor
plot(grouped_price)
```



```
# Order factor
investment <- c("stock", "bonds", "stock", "bonds", "stock", "forex")
ranked_investment <- factor(investment, ordered = TRUE, levels = c("stock", "forex", "bonds"))
ranked_investment
```

```
## [1] stock bonds stock bonds stock forex
## Levels: stock < forex < bonds
```

```
plot(ranked_investment)
```



```
# To remove unavailable level after subset, use drop=TRUE
summary(ranked_investment[1:3, drop = TRUE])
```

```
## stock bonds
##     2     1
```

notes R automatically treat string as factor, when using data.frame(), to remove this behavior we can add argument stringsAsFactors = FALSE.

Working with List

```
# List can contain multiple data type.
list1 <- list("Boni", investment)
list1
```

```
## [[1]]
## [1] "Boni"
##
## [[2]]
## [1] "stock" "bonds" "stock" "bonds" "stock" "forex"
```

```
# To subset list we use [[]] instead of []
list1[[2]]
```

```
## [1] "stock" "bonds" "stock" "bonds" "stock" "forex"
```

```
# Add names to list.
names(list1) <- c("name", "investment")
list1$name
```

```
## [1] "Boni"
```

```
# Split DataFrame to list
cash_flow <- rnorm(10, 4, 9)
year <- runif(10, 5, 45)
company <- c("Google", "Google", "Microsoft", "Microsoft", "Apple", "Apple", "Google", "Google", "Micros
company_data <- data.frame(company, cash_flow, year)
company_data
```

```
##      company cash_flow      year
## 1     Google 11.867618 41.051566
## 2     Google  2.265627  6.732927
## 3  Microsoft 15.444524 34.807107
## 4  Microsoft  6.401880  7.013473
## 5      Apple  1.474655 38.589851
## 6      Apple 10.615811 13.101394
## 7     Google -3.387839 43.017917
## 8     Google -5.398433  7.639633
## 9  Microsoft -4.814427 42.193334
## 10 Microsoft 20.938536 29.225908
```

```
new_list <- split(company_data, company_data$company)
new_list
```

```
## $Apple
##   company cash_flow     year
## 5   Apple  1.474655 38.58985
## 6   Apple 10.615811 13.10139
##
## $Google
##   company cash_flow      year
## 1  Google 11.867618 41.051566
## 2  Google  2.265627  6.732927
## 7  Google -3.387839 43.017917
## 8  Google -5.398433  7.639633
##
## $Microsoft
##      company cash_flow      year
## 3  Microsoft 15.444524 34.807107
## 4  Microsoft  6.401880  7.013473
## 9  Microsoft -4.814427 42.193334
## 10 Microsoft 20.938536 29.225908
```

```
# Unsplit list
unsplit(new_list, company)
```

```
##      company cash_flow      year
## 1     Google 11.867618 41.051566
## 2     Google  2.265627  6.732927
## 3  Microsoft 15.444524 34.807107
## 4  Microsoft  6.401880  7.013473
## 5      Apple  1.474655 38.589851
## 6      Apple 10.615811 13.101394
## 7     Google -3.387839 43.017917
## 8     Google -5.398433  7.639633
## 9  Microsoft -4.814427 42.193334
## 10 Microsoft 20.938536 29.225908
```

```
# Getting attributes
attributes(new_list)
```

```
## $names
## [1] "Apple"     "Google"    "Microsoft"
```

```
attr(new_list, "names")
```

```
## [1] "Apple"     "Google"    "Microsoft"
```