Assignment 2

Xiang Li

2023/11/8

Exercise 1

1

```
raw_df = read.csv("data/Airbnb_Ragusa.csv")
```

2

```
valid_df = raw_df[, c("neighbourhood_cleansed", "latitude",
    "longitude", "price")]
head(valid_df, 5)
```

```
##   neighbourhood_cleansed latitude longitude   price
## 1                 Ragusa 36.92940  14.62523 $110.00
## 2                 Modica 36.85759  14.76121  $56.00
## 3                 Modica 36.86113  14.76306  $40.00
## 4                 Modica 36.83871  14.76097  $51.00
## 5                 Modica 36.85793  14.76054 $140.00
```

3

```
valid_df$price = gsub("[$]", "", valid_df$price)
valid_df$price = gsub("[,]", "", valid_df$price)
valid_df$price = as.numeric(valid_df$price)
head(valid_df, 5)
```

```
##   neighbourhood_cleansed latitude longitude price
## 1                 Ragusa 36.92940  14.62523   110
## 2                 Modica 36.85759  14.76121    56
## 3                 Modica 36.86113  14.76306    40
## 4                 Modica 36.83871  14.76097    51
## 5                 Modica 36.85793  14.76054   140
```

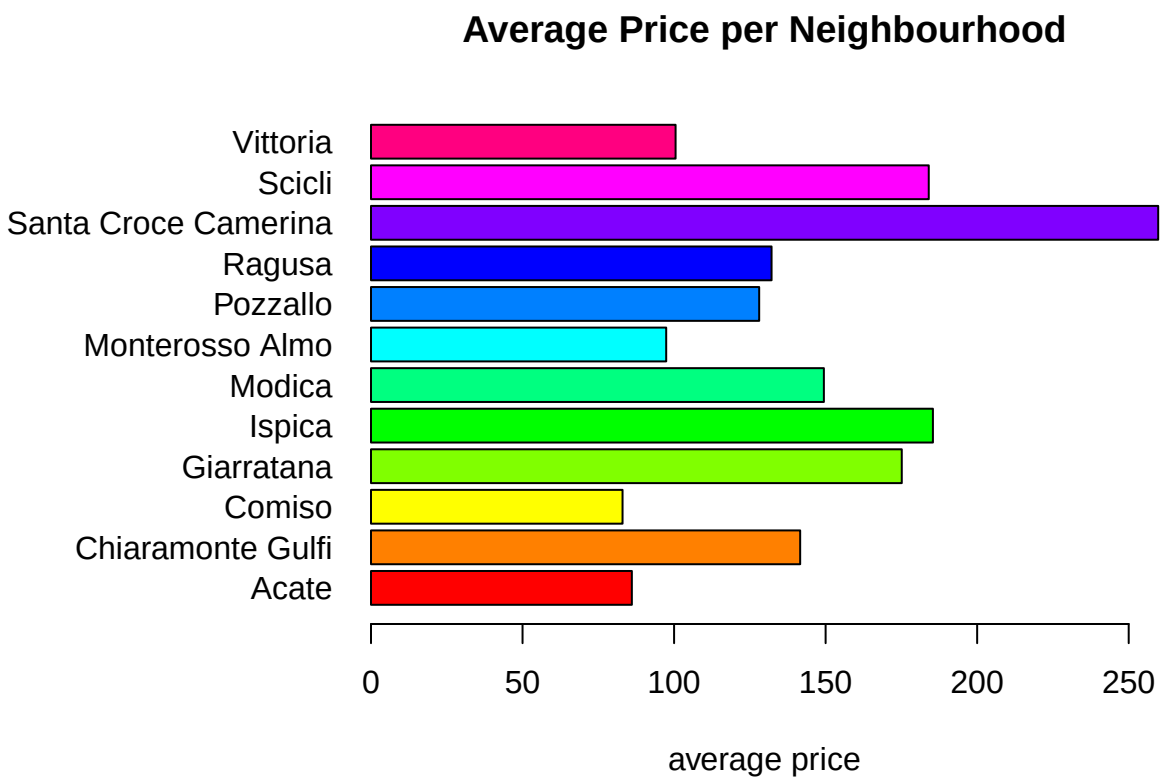
4

```
ave_price_df = aggregate(valid_df[, "price", drop = FALSE],
    by = list(neighbourhood_cleansed = valid_df$neighbourhood_cleansed),
    mean)
ave_price_df
```

```
##    neighbourhood_cleansed     price
## 1                   Acate  86.07143
## 2       Chiaramonte Gulfi 141.60000
## 3                  Comiso  83.00000
## 4              Giarratana 175.12500
## 5                  Ispica 185.40506
## 6                  Modica 149.42586
## 7         Monterosso Almo  97.40000
## 8                Pozzallo 128.07692
## 9                  Ragusa 132.16701
## 10   Santa Croce Camerina 259.73265
## 11                 Scicli 184.02929
## 12               Vittoria 100.49490
```

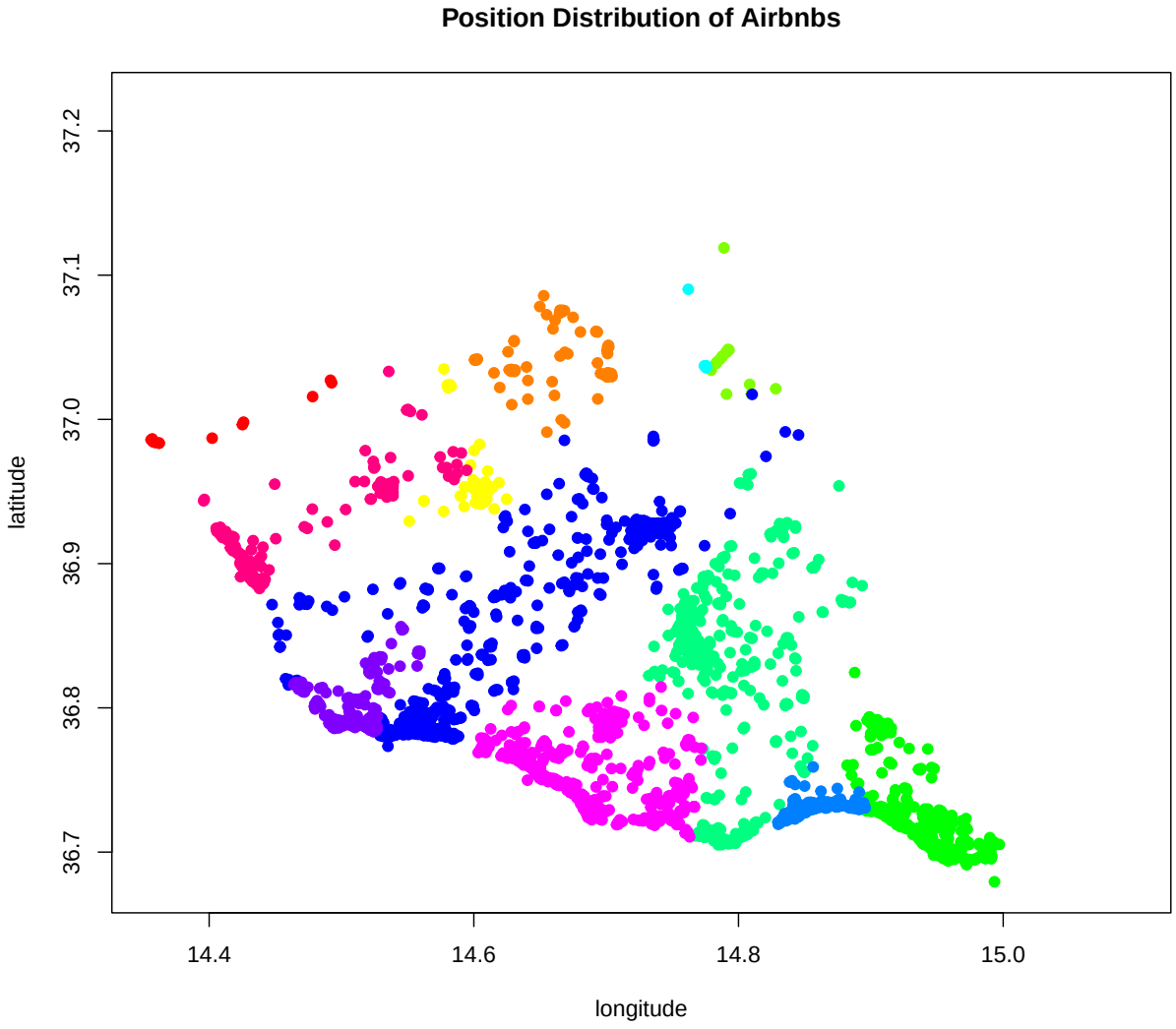
5

```
coloursvector = rainbow(12)
par(mar = c(5, 10, 4, 2))
barplot(height = ave_price_df$price, names.arg = ave_price_df$neighbourhood_cleansed,
    horiz = TRUE, col = coloursvector, main = "Average Price per Neighbourhood",
    xlim = c(0, max(ave_price_df$price)), xlab = "average price",
    ylab = "", las = 1)
```



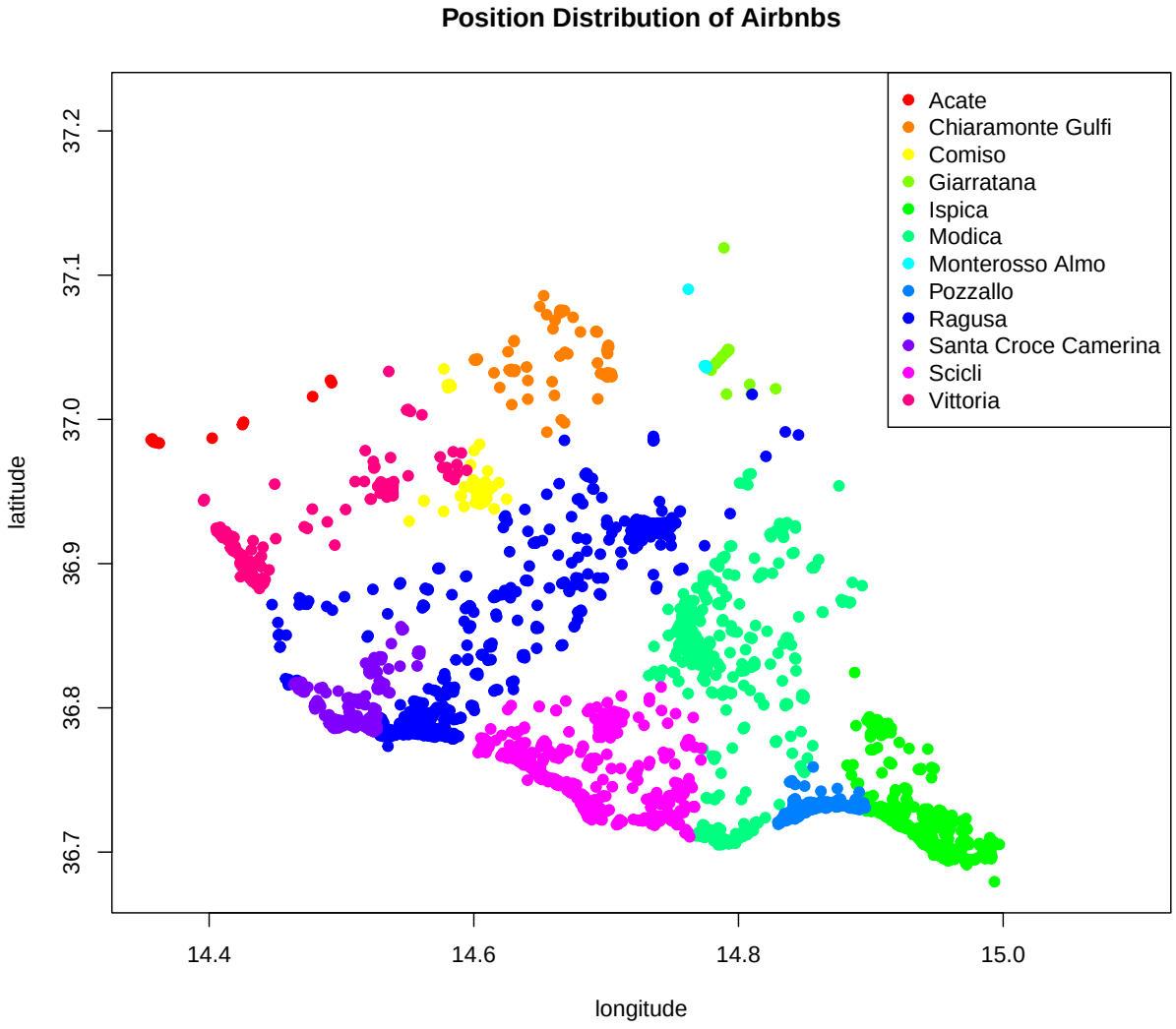
6

```
ave_price_df$color = coloursvector
valid_df = merge(valid_df, ave_price_df[, c("neighbourhood_cleansed",
    "color")], by = "neighbourhood_cleansed", all = TRUE)
plot(valid_df$longitude, valid_df$latitude, type = "p",
    col = valid_df$color, ylim = c(min(valid_df$latitude),
        max(valid_df$latitude) + 0.1), xlim = c(min(valid_df$longitude),
        max(valid_df$longitude) + 0.1), ylab = "latitude",
    xlab = "longitude", main = "Position Distribution of Airbnbs",
    pch = 19)
```



7

```
plot(valid_df$longitude, valid_df$latitude, type = "p",
    col = valid_df$color, ylim = c(min(valid_df$latitude),
        max(valid_df$latitude) + 0.1), xlim = c(min(valid_df$longitude),
        max(valid_df$longitude) + 0.1), ylab = "latitude",
    xlab = "longitude", main = "Position Distribution of Airbnbs",
    pch = 19)
legend(x = "topright", legend = ave_price_df$neighbourhood_cleansed,
    pch = 19, col = ave_price_df$color)
```

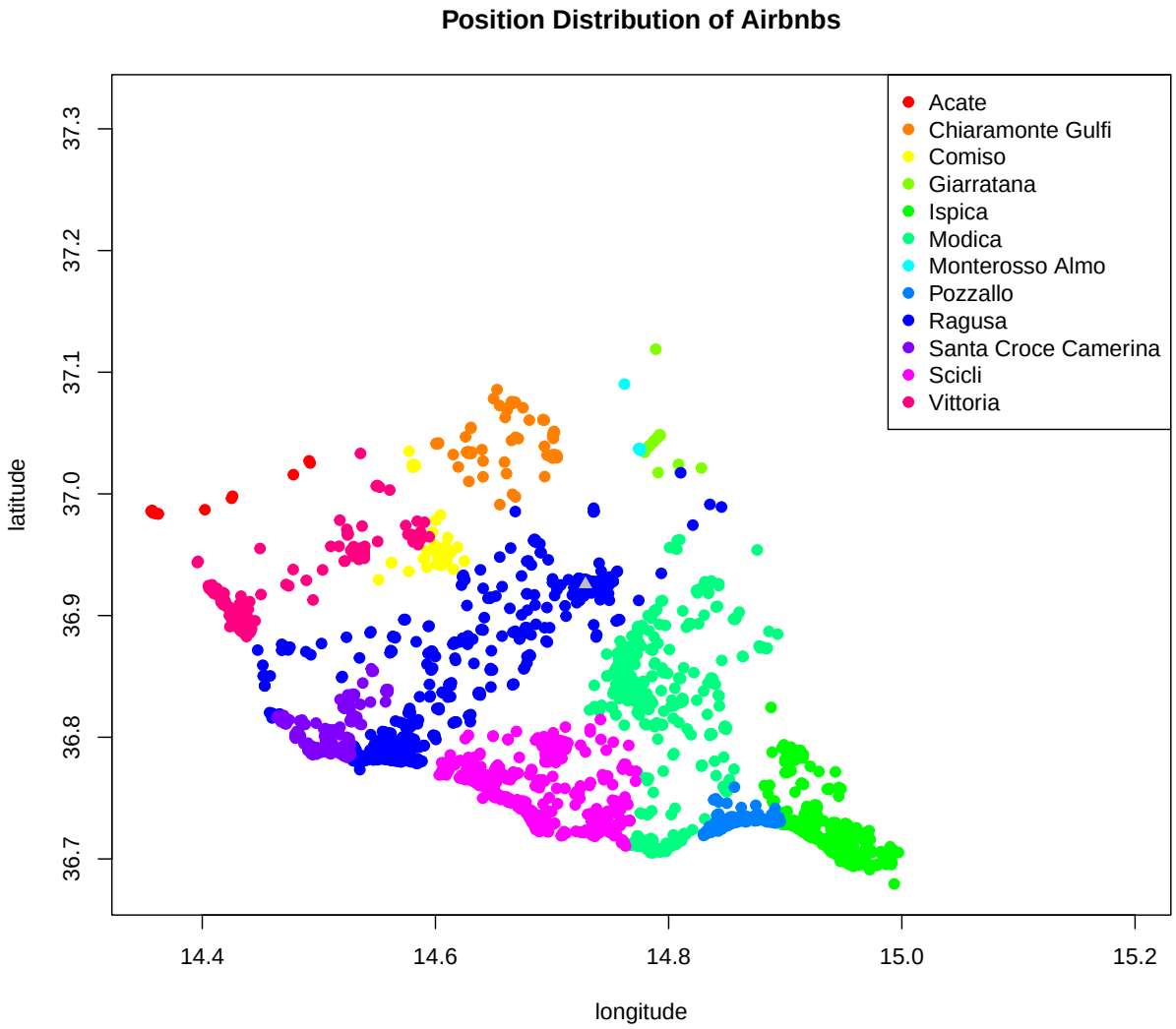


9

Yes, the two maps are similar.

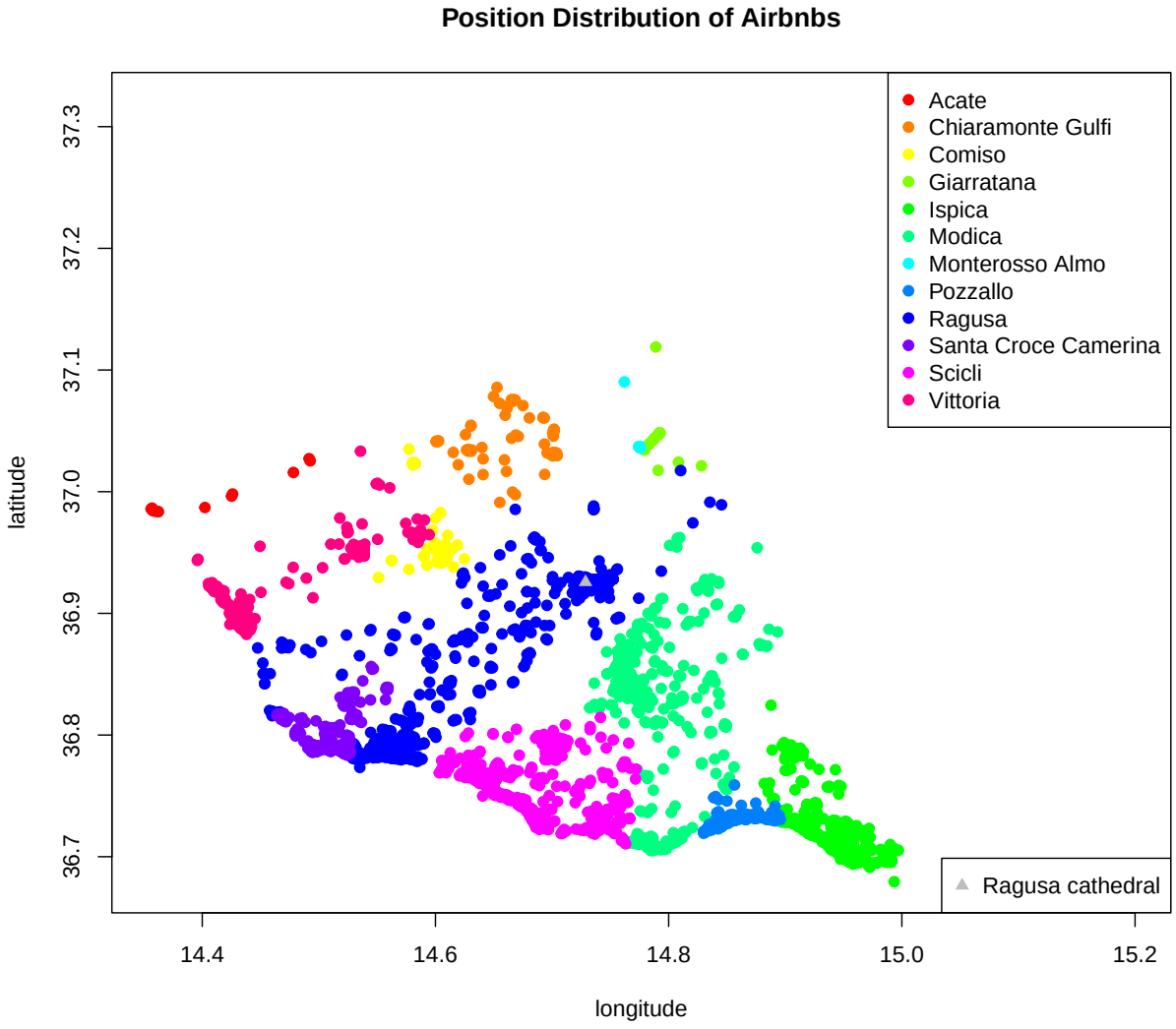
10

```
plot(valid_df$longitude, valid_df$latitude, type = "p",
    col = valid_df$color, ylim = c(min(valid_df$latitude),
        max(valid_df$latitude) + 0.2), xlim = c(min(valid_df$longitude),
        max(valid_df$longitude) + 0.2), ylab = "latitude",
    xlab = "longitude", main = "Position Distribution of Airbnbs",
    pch = 19)
legend(x = "topright", legend = ave_price_df$neighbourhood_cleansed,
    pch = 19, col = ave_price_df$color)
points(x = 14.7287, y = 36.9257, pch = 17, col = "grey")
```



11

```
plot(valid_df$longitude, valid_df$latitude, type = "p",
    col = valid_df$color, ylim = c(min(valid_df$latitude),
        max(valid_df$latitude) + 0.2), xlim = c(min(valid_df$longitude),
        max(valid_df$longitude) + 0.2), ylab = "latitude",
    xlab = "longitude", main = "Position Distribution of Airbnbs",
    pch = 19)
legend(x = "topright", legend = ave_price_df$neighbourhood_cleansed,
    pch = 19, col = ave_price_df$color)
points(x = 14.7287, y = 36.9257, pch = 17, col = "grey")
legend(x = "bottomright", legend = "Ragusa cathedral", pch = 17,
    col = "grey")
```



12

```
jpeg(file = "plot/Assignment2_Exercise1_12.jpg", width = 10,
    height = 9, units = "in", res = 300, quality = 100)
plot(valid_df$longitude, valid_df$latitude, type = "p",
    col = valid_df$color, ylim = c(min(valid_df$latitude),
        max(valid_df$latitude) + 0.2), xlim = c(min(valid_df$longitude),
        max(valid_df$longitude) + 0.2), ylab = "latitude",
    xlab = "longitude", main = "Position Distribution of Airbnbs",
    pch = 19)
legend(x = "topright", legend = ave_price_df$neighbourhood_cleansed,
    pch = 19, col = ave_price_df$color)
points(x = 14.7287, y = 36.9257, pch = 17, col = "grey")
legend(x = "bottomright", legend = "Ragusa cathedral", pch = 17,
    col = "grey")
dev.off()
```

```
## pdf
##   2
```

Exercise 2

Task 1

```
stock_price_df = read.csv("data/AAPL_complete.csv")
stock_price_df$Open = as.numeric(stock_price_df$Open)
stock_price_df$Date = as.Date(stock_price_df$Date, "%Y-%m-%d")
stock_price_df = stock_price_df[order(stock_price_df$Date),
    ]
head(stock_price_df, 5)
```

```
##         Date Close.Last    Volume    Open    High     Low
## 1 2010-03-01    29.8557 137312041 29.3928 29.9286 29.3500
## 2 2010-03-02    29.8357 141486282 29.9900 30.1186 29.6771
## 3 2010-03-03    29.9043  92846488 29.8486 29.9814 29.7057
## 4 2010-03-04    30.1014  89591907 29.8971 30.1314 29.8043
## 5 2010-03-05    31.2786 224647427 30.7057 31.3857 30.6614
```

Task 2

```
simulate_investment = function(start_date, end_date, investment_USD) {
    if (!is.character(start_date) | !is.character(end_date) |
        !is.numeric(investment_USD)) {
        warning("The input is not available!!!")
    }
    start_date_d = as.Date(start_date, "%Y-%m-%d")
    end_date_d = as.Date(end_date, "%Y-%m-%d")
    if (is.na(start_date_d) | is.na(end_date_d)) {
        warning("The dates entered do not exit or are not correctly formatted!!!")
    }
    min_date = min(stock_price_df$Date)
    max_date = max(stock_price_df$Date)
    if (start_date_d < min_date | end_date_d > max_date |
        start_date_d > end_date_d) {
        warning("The dates entered exceed the limit or are in reverse order!!!")
    }
    if (investment_USD <= 0) {
        warning("The initial investment entered is not correct!!!")
    }
    hold_price_df = stock_price_df[(stock_price_df$Date >=
        start_date_d) & (stock_price_df$Date <= end_date_d),
        ]
    buy_price = hold_price_df$Open[1]
    stock = floor(investment_USD/buy_price)
    cash = investment_USD - buy_price * stock
    hold_price_df$Value = hold_price_df$Open * stock + cash
```

```
    hold_price_df$Profit = round(hold_price_df$Value - investment_USD,
        2)
    result_df = hold_price_df[, c("Date", "Value", "Profit")]
    return(result_df)
}
```

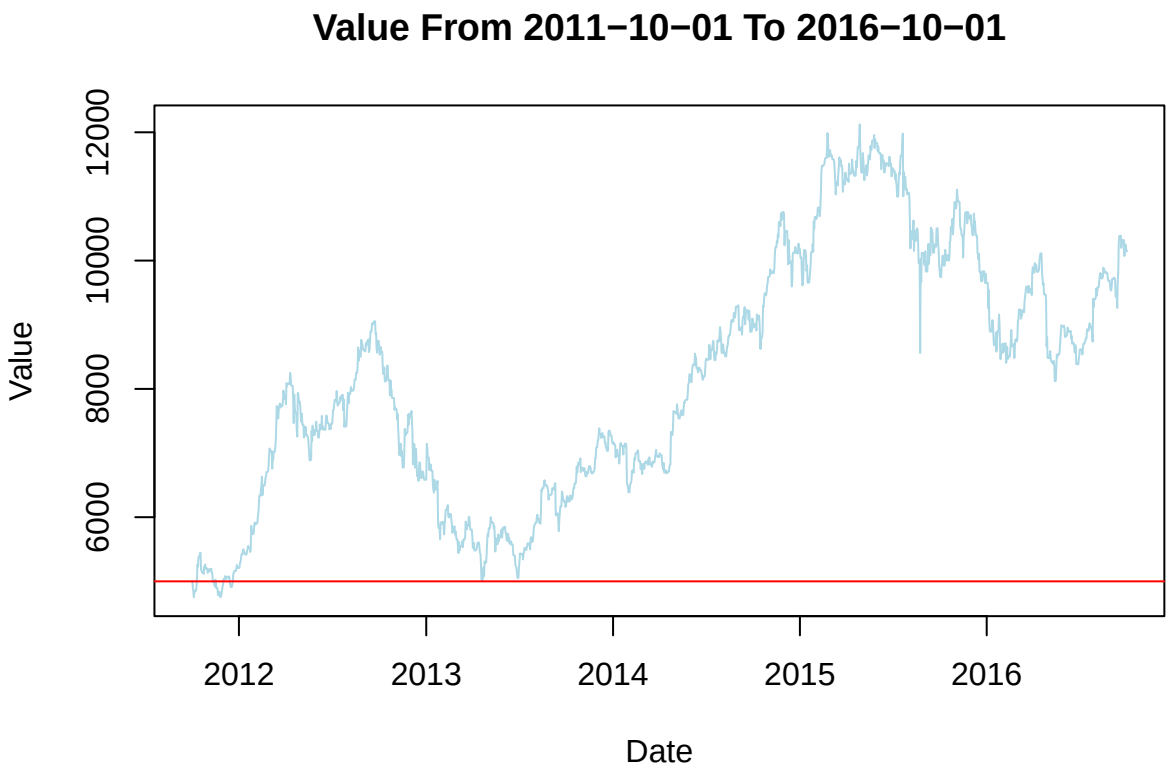
Task 3

```
simulate_investment("2011-10-01", "2011-10-11", 10000)
```

```
##           Date     Value  Profit
## 580 2011-10-01 10000.000    0.00
## 581 2011-10-02 10000.000    0.00
## 582 2011-10-03  9826.444 -173.56
## 583 2011-10-04  9677.296 -322.70
## 584 2011-10-05  9504.748 -495.25
## 585 2011-10-06  9645.400 -354.60
## 586 2011-10-07  9708.400 -291.60
## 587 2011-10-08  9708.400 -291.60
## 588 2011-10-09  9708.400 -291.60
## 589 2011-10-10  9793.522 -206.48
## 590 2011-10-11 10140.148  140.15
```

Task 4

```
df_summary = simulate_investment("2011-10-01", "2016-10-01",
    5000)
plot(df_summary$Date, df_summary$Value, type = "l", xlab = "Date",
    ylab = "Value", main = "Value From 2011-10-01 To 2016-10-01",
    col = "lightblue")
abline(h = 5000, col = "red")
```



The Value takes the highest value in April 2015, so it would be best to sell the stock in April 2015.

Task 5

```
ind = which.max(df_summary$Value)
list(Date = df_summary$Date[ind], Value = df_summary$Value[ind])
```

```
## $Date
## [1] "2015-04-28"
##
## $Value
## [1] 12123.7
```