HW1

Yifei Chen

October 14, 2019

Honor Code:"The codes and results derived by using these codes constitutemy own work. I have consulted the following resources regarding this assignment: https://stackoverflow.com/questions/21502332/generating-random-dates"

1

a.

(i)

```r
city = c("A","B","C","D","E","F","G","H","I","J")
temp = runif(10, min = 40, max = 100)
precip = runif(10, min = 0, max = 100)
PM10 = runif(10, min = 0, max = 50)
death = runif(10, min = 0, max = 500)
#set up the dataframe
health_effect = data.frame(temp, precip, PM10, death, row.names = city)
health_effect
```

```
##       temp   precip     PM10     death
## A 79.55539 61.05354 48.81940 167.88907
## B 53.25688 19.56528 42.03017 223.19210
## C 93.09359 26.19003 15.75808 431.05911
## D 85.08030 95.55628 47.51082  39.41251
## E 65.23962 69.69376 27.68785 130.74124
## F 97.50896 59.94260 11.02458 369.65818
## G 94.62735 39.36248 27.06219 213.56515
## H 69.60308 65.95475 30.27474 297.80878
## I 63.62809 56.14343 37.97493 185.13710
## J 50.86398 57.56128 12.04689 246.48367
```

```r
# The average number of deaths for each of the citieson days
# where thePM10 concentration is greater than 20

# because death is on the fourth column,
# the column number is "4"
health_effect[health_effect$PM10 > 20, 4]
```

```
## [1] 167.88907 223.19210  39.41251 130.74124 213.56515 297.80878 185.13710
```

(ii)

```r
# The average PM10 concentration for each of the cities on days with noprecipitation
# and average temperature above 80 degrees F

# because PM10 is on the third column,
# the column number is "3"
health_effect[health_effect$precip == 0 & health_effect$temp > 80, 3]
```

```
## numeric(0)
```

b.

(i)

```r
patients = c("A","B","C","D","E","F")
# patient 1
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_1 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 2
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_2 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 3
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_3 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 4
counts = sample(1:10, 1)
```

```r
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_4 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 5
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_5 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 6
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_6 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# list of patients
patients_list = list(patient_1, patient_2, patient_3, patient_4, patient_5, patient_6)
patients_list
```

```
## [[1]]
##             age gender    weight    BP_sys    BP_dia  glucose
## 1995-01-19   28 Female  66.33462 332.55555  82.32501 546.0019
## 2009-01-18   42 Female  56.80991 186.16598 253.71622 565.4647
## 1982-01-05   15 Female  77.74298 426.44867  12.51068 237.7573
## 1993-08-30   26 Female 119.41258 547.86052 409.19460 540.1165
## 1978-11-12   11 Female 197.24607  28.91482 152.55234 579.8088
##
## [[2]]
##             age gender    weight     BP_sys    BP_dia  glucose
## 2016-09-01   16 Female 172.80527 186.684168 512.08718 536.7981
## 2001-07-23    1 Female  59.46689   1.164126 310.99318 391.6528
## 2012-04-09   12 Female 134.95501  87.051682 247.88040 162.4358
## 2004-09-11    4 Female 172.37590 161.399425  88.12748 202.5857
## 2018-07-07   18 Female  91.73767 495.417467 314.81760 217.6546
```

```
##
## [[3]]
##            age gender    weight   BP_sys   BP_dia   glucose
## 1992-05-07  40 Female 107.84975 572.7287 330.3021 428.11348
## 2016-04-07  64 Female 195.28281 313.9041 316.5424 523.80595
## 1967-09-15  15 Female 164.92505 469.6541 554.7218 258.35185
## 1999-02-08  47 Female 177.52713 515.4834 541.4241 578.30985
## 1957-05-01   5 Female 192.26682 130.1640 594.3994  21.36858
## 1993-01-24  40 Female 155.26292 266.8041 142.0198 284.18954
## 1969-04-07  17 Female  58.51832 201.7356 315.2173  43.82639
##
## [[4]]
##            age gender    weight   BP_sys   BP_dia  glucose
## 2017-02-23  58 Female  81.09960 244.8843 116.6477 309.4101
## 1962-11-11   4 Female 199.15728 516.6612 485.7858 102.8402
## 2005-11-08  47 Female 172.14806 172.3619 312.1337 504.8305
## 1985-07-03  26 Female  64.83307 104.9629 588.5159 150.7419
##
## [[5]]
##            age gender    weight   BP_sys    BP_dia   glucose
## 1994-02-15   9   Male  51.11359 302.1422 290.41963 227.03292
## 2006-03-04  21   Male 153.23877 386.6074  93.89065 201.47596
## 1996-01-28  11   Male 140.52539 282.9282  23.20524 316.18547
## 2017-11-07  33   Male 153.90156 358.0181 210.28024 177.56765
## 2010-05-20  25   Male 147.61717 289.9885 244.56465 475.90755
## 1997-10-08  13   Male  75.39105 477.7065 393.11215  29.62577
## 2004-08-09  20   Male 131.11957 229.3513 383.05345 563.30480
## 1986-02-09   1   Male  74.04459 459.3969 346.98292 436.61736
## 2009-05-01  24   Male 114.65056 387.6919 528.53005  45.17407
## 1996-10-24  12   Male 103.63938 500.3829 104.64489 457.18782
##
## [[6]]
##            age gender    weight   BP_sys   BP_dia    glucose
## 2012-09-04   5 Female  81.11843 142.9275 260.6532 257.109900
## 2007-08-02   0 Female 166.44386 541.6575 208.5945   3.847559
## 2014-05-27   7 Female  69.77276 256.9056 198.3254  18.415545
```

```r
# Times of each patient visiting the clinic
visit_times = sapply(patients_list,NROW)
visit_times
```

```
## [1]  5  5  7  4 10  3
```

(ii)

```r
# The average systolic blood pressure level for each of
# the patients withmaximum weight (during the study period) greater than 180 lb
avg_BP_sys <- lapply(patients_list, function(x){
  if(max(x$"weight", 0)>180){
    y = mean(x$"BP_sys")
  }else{
    y = NaN
  }
```

```r
  return(y)
})
avg_BP_sys
```

```
## [[1]]
## [1] 304.3891
##
## [[2]]
## [1] NaN
##
## [[3]]
## [1] 352.9249
##
## [[4]]
## [1] 259.7176
##
## [[5]]
## [1] NaN
##
## [[6]]
## [1] NaN
```

(iii)

```r
# the average blood glucose level for each of the
# patients with age at least 40 years at the first visit
avg_glucose <- lapply(patients_list, function(x){
  if(min(x$"age") >= 40){
    y = mean(x$"glucose")
  }else{
    y = NaN
  }
})
avg_glucose
```

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

2

(a)

```
animal = c("cat", "dog", "cow", "squirrel")
color = c("white", "black", "brown", "red")
attribute = c("big", "small", "angry", "cute", "finicky")
# Generate random samples, with replacement
Animal = sample(animal, size = 100, replace = TRUE)
Color = sample(color, size = 100, replace = TRUE)
Attribute = sample(attribute, size = 100, replace = TRUE)
Animal
```

```
##   [1] "squirrel" "squirrel" "cow"      "squirrel" "squirrel" "squirrel"
##   [7] "dog"      "dog"      "dog"      "cat"      "squirrel" "dog"
##  [13] "cat"      "cat"      "cow"      "cat"      "cow"      "dog"
##  [19] "squirrel" "dog"      "squirrel" "cow"      "cat"      "cow"
##  [25] "cow"      "cat"      "dog"      "cow"      "dog"      "dog"
##  [31] "dog"      "squirrel" "dog"      "cat"      "cow"      "cow"
##  [37] "dog"      "squirrel" "cat"      "dog"      "squirrel" "cow"
##  [43] "squirrel" "dog"      "cat"      "cat"      "squirrel" "cow"
##  [49] "dog"      "dog"      "squirrel" "cat"      "squirrel" "squirrel"
##  [55] "cat"      "squirrel" "cow"      "squirrel" "cow"      "dog"
##  [61] "dog"      "cat"      "squirrel" "cat"      "cow"      "dog"
##  [67] "squirrel" "cat"      "cow"      "cat"      "dog"      "dog"
##  [73] "cat"      "squirrel" "cow"      "cow"      "cow"      "dog"
##  [79] "cow"      "cat"      "dog"      "cow"      "squirrel" "squirrel"
##  [85] "squirrel" "cow"      "cat"      "dog"      "dog"      "cow"
##  [91] "cow"      "squirrel" "cow"      "squirrel" "cow"      "cat"
##  [97] "cow"      "dog"      "cow"      "squirrel"
```

```
Color
```

```
##   [1] "black" "brown" "white" "white" "brown" "red"   "brown" "black"
##   [9] "brown" "black" "brown" "black" "black" "white" "black" "red"
##  [17] "black" "brown" "brown" "brown" "white" "white" "black" "brown"
##  [25] "white" "brown" "brown" "black" "red"   "red"   "brown" "brown"
##  [33] "black" "brown" "brown" "white" "red"   "brown" "red"   "red"
##  [41] "white" "white" "black" "red"   "black" "black" "red"   "white"
##  [49] "black" "red"   "white" "red"   "red"   "white" "red"   "white"
##  [57] "white" "brown" "red"   "black" "brown" "white" "white" "brown"
##  [65] "black" "brown" "white" "red"   "brown" "white" "white" "brown"
##  [73] "white" "brown" "white" "brown" "red"   "brown" "white" "brown"
##  [81] "brown" "white" "white" "red"   "red"   "red"   "brown" "white"
##  [89] "red"   "red"   "white" "red"   "red"   "black" "black" "brown"
##  [97] "black" "red"   "black" "black"
```

```
Attribute
```

```
##   [1] "cute"    "finicky" "angry"   "big"     "big"     "small"   "cute"
##   [8] "finicky" "small"   "small"   "small"   "finicky" "finicky" "small"
##  [15] "angry"   "big"     "angry"   "small"   "small"   "big"     "small"
##  [22] "small"   "cute"    "finicky" "big"     "cute"    "angry"   "big"
```

```
## [29] "finicky" "big"     "cute"    "big"     "finicky" "cute"    "big"
## [36] "finicky" "finicky" "big"     "cute"    "angry"   "finicky" "cute"
## [43] "small"   "cute"    "cute"    "small"   "big"     "cute"    "small"
## [50] "small"   "small"   "angry"   "finicky" "cute"    "finicky" "cute"
## [57] "small"   "big"     "big"     "small"   "small"   "big"     "small"
## [64] "cute"    "finicky" "angry"   "finicky" "cute"    "big"     "angry"
## [71] "angry"   "big"     "angry"   "finicky" "small"   "angry"   "finicky"
## [78] "angry"   "angry"   "angry"   "finicky" "big"     "small"   "small"
## [85] "angry"   "big"     "small"   "big"     "big"     "small"   "cute"
## [92] "angry"   "finicky" "small"   "angry"   "big"     "cute"    "small"
## [99] "finicky" "cute"
```

(b)

```
# Combine the results to produce phrases
phrases = paste(Attribute,Color,Animal)
phrases
```

```
##   [1] "cute black squirrel"    "finicky brown squirrel"
##   [3] "angry white cow"        "big white squirrel"
##   [5] "big brown squirrel"     "small red squirrel"
##   [7] "cute brown dog"         "finicky black dog"
##   [9] "small brown dog"        "small black cat"
##  [11] "small brown squirrel"   "finicky black dog"
##  [13] "finicky black cat"      "small white cat"
##  [15] "angry black cow"        "big red cat"
##  [17] "angry black cow"        "small brown dog"
##  [19] "small brown squirrel"   "big brown dog"
##  [21] "small white squirrel"   "small white cow"
##  [23] "cute black cat"         "finicky brown cow"
##  [25] "big white cow"          "cute brown cat"
##  [27] "angry brown dog"        "big black cow"
##  [29] "finicky red dog"        "big red dog"
##  [31] "cute brown dog"         "big brown squirrel"
##  [33] "finicky black dog"      "cute brown cat"
##  [35] "big brown cow"          "finicky white cow"
##  [37] "finicky red dog"        "big brown squirrel"
##  [39] "cute red cat"           "angry red dog"
##  [41] "finicky white squirrel" "cute white cow"
##  [43] "small black squirrel"   "cute red dog"
##  [45] "cute black cat"         "small black cat"
##  [47] "big red squirrel"       "cute white cow"
##  [49] "small black dog"        "small red dog"
##  [51] "small white squirrel"   "angry red cat"
##  [53] "finicky red squirrel"   "cute white squirrel"
##  [55] "finicky red cat"        "cute white squirrel"
##  [57] "small white cow"        "big brown squirrel"
##  [59] "big red cow"            "small black dog"
##  [61] "small brown dog"        "big white cat"
##  [63] "small white squirrel"   "cute brown cat"
##  [65] "finicky black cow"      "angry brown dog"
##  [67] "finicky white squirrel" "cute red cat"
##  [69] "big brown cow"          "angry white cat"
```

```
##  [71] "angry white dog"        "big brown dog"
##  [73] "angry white cat"        "finicky brown squirrel"
##  [75] "small white cow"        "angry brown cow"
##  [77] "finicky red cow"        "angry brown dog"
##  [79] "angry white cow"        "angry brown cat"
##  [81] "finicky brown dog"      "big white cow"
##  [83] "small white squirrel"   "small red squirrel"
##  [85] "angry red squirrel"     "big red cow"
##  [87] "small brown cat"        "big white dog"
##  [89] "big red dog"            "small red cow"
##  [91] "cute white cow"         "angry red squirrel"
##  [93] "finicky red cow"        "small black squirrel"
##  [95] "angry black cow"        "big brown cat"
##  [97] "cute black cow"         "small red dog"
##  [99] "finicky black cow"      "cute black squirrel"
```

(c)

```
# frequency distribution of the different types of
# animals together with colors and attributes based on the sampled data.
table(data.frame(Animal, Color, Attribute))
```

```
## , , Attribute = angry
##
##           Color
## Animal     black brown red white
##   cat          0     1   1     2
##   cow          3     1   0     2
##   dog          0     3   1     1
##   squirrel     0     0   2     0
##
## , , Attribute = big
##
##           Color
## Animal     black brown red white
##   cat          0     1   1     1
##   cow          1     2   2     2
##   dog          0     2   2     1
##   squirrel     0     4   1     1
##
## , , Attribute = cute
##
##           Color
## Animal     black brown red white
##   cat          2     3   2     0
##   cow          1     0   0     3
##   dog          0     2   1     0
##   squirrel     2     0   0     2
##
## , , Attribute = finicky
##
##           Color
## Animal     black brown red white
```

```
##    cat           1     0   1     0
##    cow           2     1   2     1
##    dog           3     1   2     0
##    squirrel      0     2   1     2
##
## , , Attribute = small
##
##           Color
## Animal     black brown red white
##    cat           2     1   0     1
##    cow           0     0   1     3
##    dog           2     3   2     0
##    squirrel      2     2   2     4
```

(d)

```r
# Animal vs.Color
table(data.frame(Animal, Color))
```

```
##           Color
## Animal     black brown red white
##    cat           5     6   5     4
##    cow           7     4   5    11
##    dog           5    11   8     2
##    squirrel      4     8   6     9
```

```r
# Animal vs. Attribute
table(data.frame(Animal, Attribute))
```

```
##           Attribute
## Animal     angry big cute finicky small
##    cat         4   3    7       2     4
##    cow         6   7    4       6     4
##    dog         5   5    3       6     7
##    squirrel    2   6    4       5    10
```

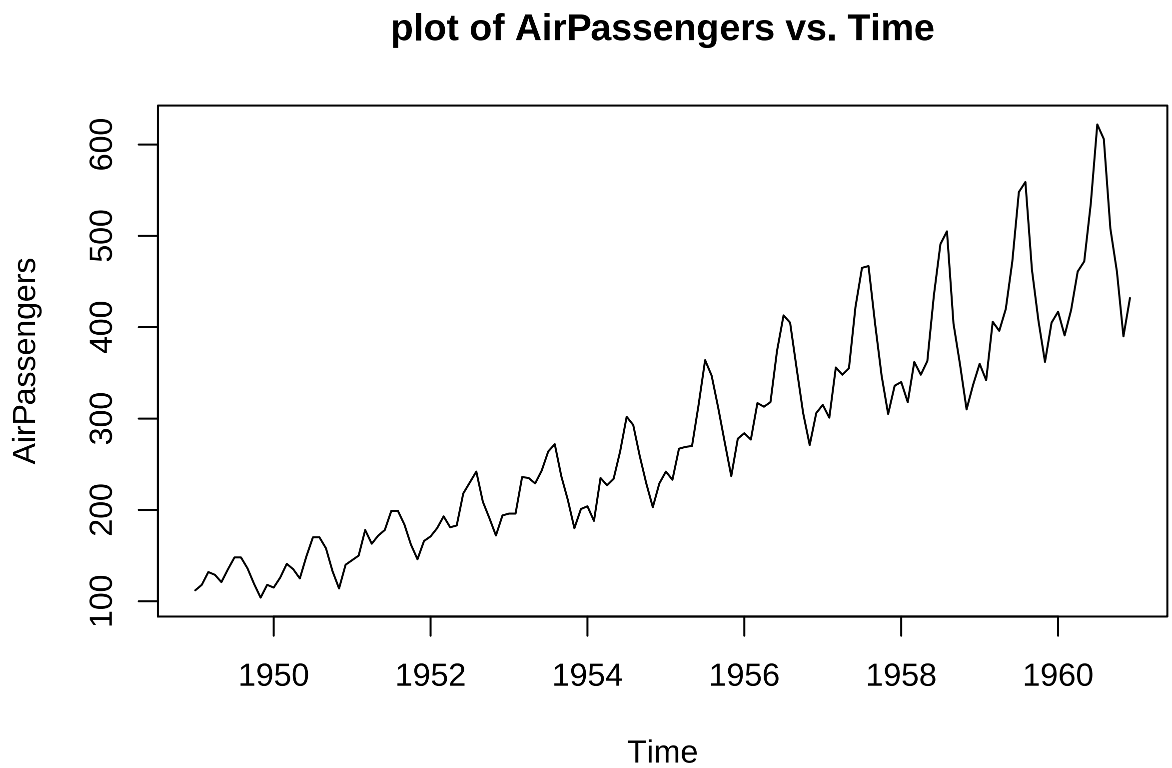
```r
# Animal
table(Animal)
```

```
## Animal
##      cat      cow      dog squirrel
##       20       27       26       27
```

3

(a)

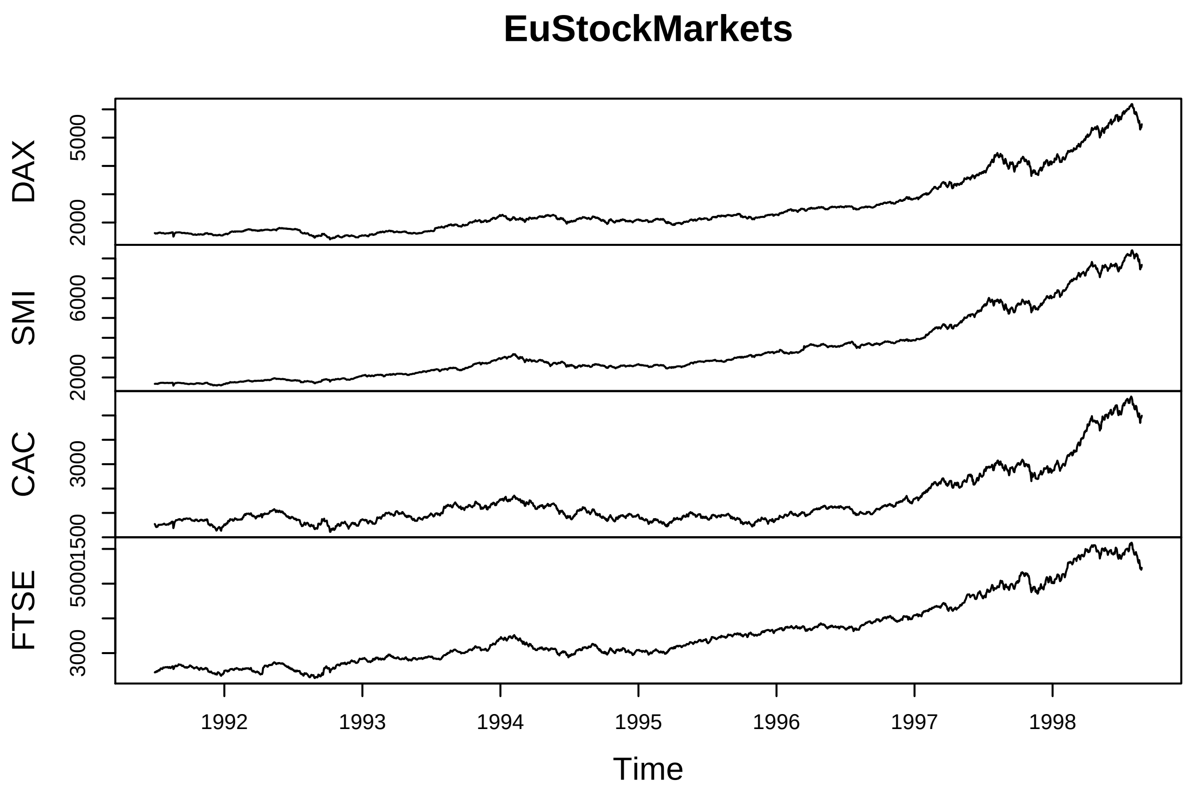
```r
plot(AirPassengers, main = "plot of AirPassengers vs. Time")
```




```
# answer: This package only has two variables: number of
# air passengers and years.
# According to the plot, we can see that as the time
# increases, the number of air passengers is also increasing.
# Therefore, they have positive correlation and are
# dependent. As we can see from that graph that the frequency
# for each period are very similar. It always increases then decreases
# a little bit for every two years
```

(b)

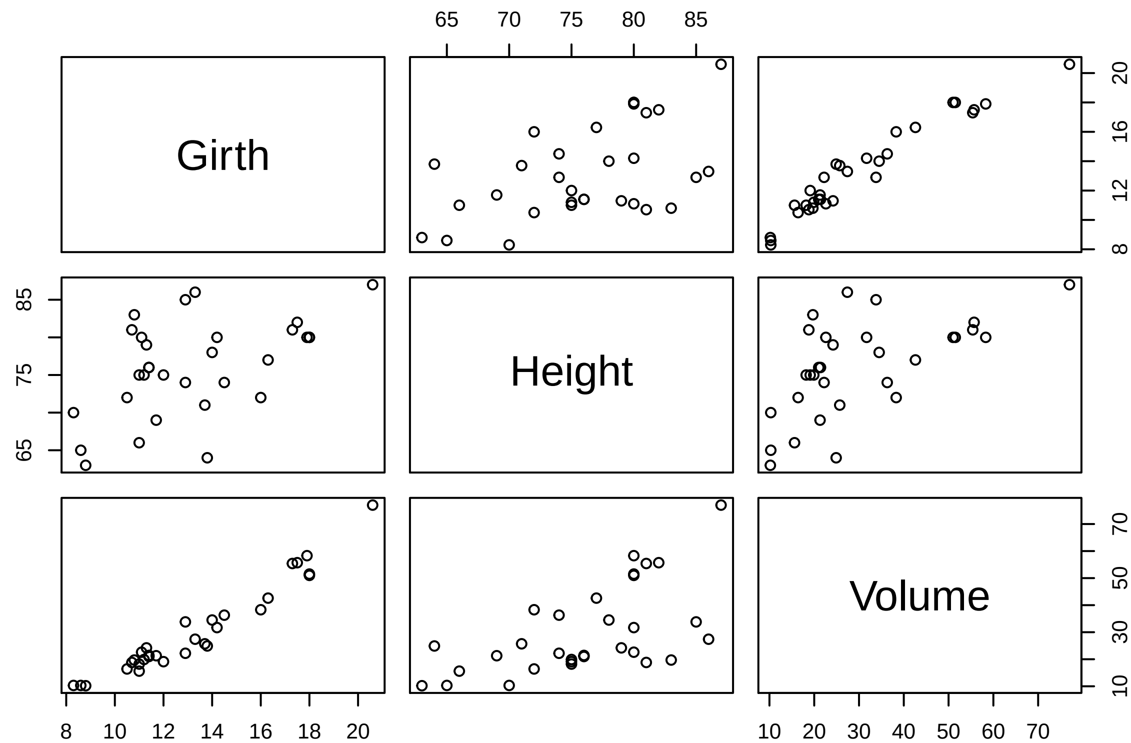
```r
plot(EuStockMarkets)
```




```
# answer: This package is about the daily closing prices of
# major European stock indices during 1991-1998. As we can see
# from the plot, the price of FTSE is almost 1.5 times higher
# than the others. These four stock indices all have positive
# correlation with the time(years). As times increase, the
# closing prices will also increase.
```

(c)

```
plot(trees)
```




```
# answer: This package has three variables: girth, height, volume
# By looking at the graph, we can see that girth and volumn have
# very strong positive relationship. Hight and volumn have a weak
# positive relationship. It's hard to see the relationship between
# girth and height.
```

R Appendix

```
knitr::opts_chunk$set(echo = TRUE)
city = c("A","B","C","D","E","F","G","H","I","J")
temp = runif(10, min = 40, max = 100)
precip = runif(10, min = 0, max = 100)
PM10 = runif(10, min = 0, max = 50)
death = runif(10, min = 0, max = 500)
#set up the dataframe
health_effect = data.frame(temp, precip, PM10, death, row.names = city)
health_effect
# The average number of deaths for each of the citieson days
# where thePM10 concentration is greater than 20

# because death is on the fourth column,
# the column number is "4"
health_effect[health_effect$PM10 > 20, 4]
# The average PM10 concentration for each of the cities on days with noprecipitation
```

```r
# and average temperature above 80 degrees F

# because PM10 is on the third column,
# the column number is "3"
health_effect[health_effect$precip == 0 & health_effect$temp > 80, 3]
patients = c("A","B","C","D","E","F")
# patient 1
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_1 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 2
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_2 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 3
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_3 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 4
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
```

```r
patient_4 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 5
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_5 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# patient 6
counts = sample(1:10, 1)
date_of_birth = sample(seq(as.Date('1950/01/01'), as.Date('2010/01/01'), by="day"), 1)
visit_date = sample(seq(date_of_birth, as.Date('2019/01/01'), by="day"), counts)
age = as.integer((visit_date- date_of_birth)/365)
gender= sample(c("Male","Female"), size = 1)
gender = rep(gender, times = counts)
weight= runif(counts, min = 50, max = 200)
BP_sys= runif(counts, min = 0, max = 600)
BP_dia= runif(counts, min = 0, max = 600)
glucose= runif(counts, min = 0, max = 600)
patient_6 = data.frame(age, gender, weight, BP_sys, BP_dia, glucose, row.names = visit_date)
# list of patients
patients_list = list(patient_1, patient_2, patient_3, patient_4, patient_5, patient_6)
patients_list
# Times of each patient visiting the clinic
visit_times = sapply(patients_list,NROW)
visit_times
# The average systolic blood pressure level for each of
# the patients withmaximum weight (during the study period) greater than 180 lb
avg_BP_sys <- lapply(patients_list, function(x){
  if(max(x$"weight", 0)>180){
    y = mean(x$"BP_sys")
  }else{
    y = NaN
  }
  return(y)
})
avg_BP_sys
# the average blood glucose level for each of the
# patients with age at least 40 years at the first visit
avg_glucose <- lapply(patients_list, function(x){
  if(min(x$"age") >= 40){
    y = mean(x$"glucose")
  }else{
    y = NaN
  }
})
avg_glucose
animal = c("cat", "dog", "cow", "squirrel")
```

```r
color = c("white", "black", "brown", "red")
attribute = c("big", "small", "angry", "cute", "finicky")
# Generate random samples, with replacement
Animal = sample(animal, size = 100, replace = TRUE)
Color = sample(color, size = 100, replace = TRUE)
Attribute = sample(attribute, size = 100, replace = TRUE)
Animal
Color
Attribute
# Combine the results to produce phrases
phrases = paste(Attribute,Color,Animal)
phrases
# frequency distribution of the different types of
# animals together with colors and attributes based on the sampled data.
table(data.frame(Animal, Color, Attribute))
# Animal vs.Color
table(data.frame(Animal, Color))
# Animal vs. Attribute
table(data.frame(Animal, Attribute))
# Animal
table(Animal)
plot(AirPassengers, main = "plot of AirPassengers vs. Time")
# answer: This package only has two variables: number of
# air passengers and years.
# According to the plot, we can see that as the time
# increases, the number of air passengers is also increasing.
# Therefore, they have positive correlation and are
# dependent. As we can see from that graph that the frequency
# for each period are very similar. It always increases then decreases
# a little bit for every two years
plot(EuStockMarkets)
# answer: This package is about the daily closing prices of
# major European stock indices during 1991-1998. As we can see
# from the plot, the price of FTSE is almost 1.5 times higher
# than the others. These four stock indices all have positive
# correlation with the time(years). As times increase, the
# closing prices will also increase.
plot(trees)
# answer: This package has three variables: girth, height, volume
# By looking at the graph, we can see that girth and volumn have
# very strong positive relationship. Hight and volumn have a weak
# positive relationship. It's hard to see the relationship between
# girth and height.
```