Problem 3

HW2

Hanyuan Chi(chixx105), Zhi Shen(shenx704)

April 21, 2017

```
suppressPackageStartupMessages({
  library(purrr)
  library(tidyr)
  library(ggplot2)
  library(dplyr)
})

options(scipen = 99) # Penalize scientific notation

set.seed(123456) # PLEASE DO NOT CHANGE THE SEED
```

Advanced Monte-Carlo for Poisson Process

In the context of Problem 2, please modify your solution to include that:

1. Car salesmen work 3 separate shifts: 12am-8am, 8am-4pm, 4pm-12am rather than hourly
2. Every hour of the day has its own average arrival rate
3. Car salesmen cannot work two shifts in one day.

You are working for Poisson Car Dealerships Inc. and your task is to optimize the employment in a particular popular car dealership location.

The customers arrive to that dealership according to a Poisson arrival process (meaning that the number of customers that will arrive to the dealership on a particular hour is distributed with Poisson distribution and average arrival rate λ)

Salesmen are assigned to customers on 1-on-1 basis for the entire hour. Say, if 1 customer comes in at 3pm, then he occupes 1 salesman for that entire hour. The occupied salesman will be free again at 4pm to work with another customer.

Assume your dealership is open 24 hours a day / 7 days a week.

Also, assume that car salesmen work not on hourly basis but on a shift-basis. More precisely, they have 3 separate shifts: 12am-8am, 8am-4pm, 4pm-12am. Also note that current laws prohibit the person from working more than 1 shift in a day. If a salesman works in the morning shift then he is done for the day. (The laws do permit to work adjacent shift across different days: say, it is allowed to have 4pm-12am shift one day followed immediately by 12am-8am shift the next day.)

Question 1

- Please simulate one possible future for the next month (30 days = 720 hours).
- Output:

    - Please create data.frame `df1` with N rows that contains your sample values in column `df1$X`, current hour (in 0-23 format) in `df1$hour` and current day (in 1-30 format) `df1$day`.

```
N <- 720L

# Specify a given average arrival rate for every hour
AVG_ARRIVAL_RATE <- 10*sin(seq(0,3*pi/4,length.out = 24))
names(AVG_ARRIVAL_RATE) <- 0:23 # 24-hour based time

# Print the (rounded) arrival rate for every hour of the day
print(round(AVG_ARRIVAL_RATE,2))
```

```
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
## 0.00 1.02 2.03 3.03 3.98 4.90 5.77 6.57 7.31 7.97 8.54 9.03 9.42 9.72 9.91
##   15   16   17   18   19   20   21   22   23
## 9.99 9.98 9.85 9.63 9.30 8.88 8.36 7.76 7.07
```

```
# Please write your code below

df1 <- data.frame(hour = rep(0:23,N/24),
                  day = rep(1:30, each =24))%>%
  group_by(hour)%>%
  mutate(X = rpois(30,AVG_ARRIVAL_RATE[hour+1]))

head(df1)
```

```
## Source: local data frame [6 x 3]
## Groups: hour [6]
##
##    hour   day     X
##   <int> <int> <int>
## 1     0     1     0
## 2     1     1     2
## 3     2     1     2
## 4     3     1     2
## 5     4     1     3
## 6     5     1     3
```

Question 2

- Please simulate $R = 2000$ possible ways your future may look for the next month (30 days = 720 hours)
- Output:
    - Please create data.frame `df2` with $N \times R$ rows that contains your sample values in column `df2$X`, current hour (in 0-23 format) in `df2$hour`, current day (in 1-30 format) `df2$day` and sample id in column `df2$id`.

```
R <- 2000L

# Please write your code below

df2 <- data.frame(id = rep(1:R, each=N),
                  hour = rep(0:23, R*N/24),
                  day = rep(rep(1:30, each=24),R))%>%
  group_by(hour)%>%
```

```
  mutate(X = rpois(30*R,AVG_ARRIVAL_RATE[hour+1]))

head(df2)
```

```
## Source: local data frame [6 x 4]
## Groups: hour [6]
##
##        id  hour   day     X
##     <int> <int> <int> <int>
## 1       1     0     1     0
## 2       1     1     1     1
## 3       1     2     1     2
## 4       1     3     1     2
## 5       1     4     1     3
## 6       1     5     1     6
```

Question 3

- Please compute how much workforce you should deploy every day to make sure that in **99% of the days** there are enough salesmen present for every customer (without waiting).
- Hints:
  - you may want to look at `quantile()` function
- Output:
  - Please save the value into integer variable `q3`

```
# Please write your code below

df3 <- df2 %>%
  mutate(shift = ifelse(hour %in% c(0:7),1,ifelse(hour %in% c(8:15),2,3))) %>%
  group_by(id,day,shift) %>%
  summarise(max_shift_X = max(X))%>%
  group_by(id,day) %>%
  summarise(day_X = sum(max_shift_X))

q3 <- as.integer(quantile(df3$day_X,0.99))
q3
```

```
## [1] 45
```