Assignment 1.b for **STATGR6103**

submitted to Professor Andrew Gelman

Advait Rajagopal

11 September 2016

1 Solution to Exercise 1.9 from BDA

1.1 Part A

Table 1: Summary of answers to **Part A**

Number of patients	People who waited	Average wait time	Closing time
54	17	3.26 min	4:15 pm

1.2 Part B

Table 2: Summary of answers to **Part B**

Summary	Number of patients	People who waited	Average wait time	Closing time
Quartile 1	39	3	2.21 min	4:00pm
Median	43	5	3.01 min	4:06pm
Quartile 3	48	9	3.85 min	4:11pm

2 Solution to Question 2 in Assignment 1.b

If I was to use this survey data to appropriately capture the prevalence of drug use I would have to keep several things in mind. The first is the goal of identifying the prevalence of drug use in the sample. This would involve developing some metric of 'drug use' like asking the question "How many people do you know who do drugs?" or "Rate the vulnerability of Native Americans to drug use on a scale of 1-10", or simply "do you do drugs?". The development of this metric is important to understand whether it is a quantitative variable or perhaps an indicator.

To get an overview of the Native American population in NYC we should use publicly available census data to identify relevant demographic characteristics, and other strata in the population. Then the next thing to do is to make sure that the sample is as representative of the population as possible this we can use the concepts of 'poststratification' and 'weighting.

Poststratification will help me resolve and account for known differences in the sample that make it unrepresentative of the population. However as Gelman [2007][1] puts it, poststratification for "deep interactions" can be difficult. Now given that we have used a snowball sampling technique, each successive group of 100 is generated from a previous known group so we should consider weighing the survey data according to publicly available data. We would use census data to get a picture of the true population and include sample units in the final calculation based on the original population. We can use hierarchical regression models along with the poststratification[2] methods to reconcile differences between the population and sample. This will help account for bias and dependence arising from the sampling method.

3 R Code for Exercise 1.9 from BDA

```
############################
Answer a)
############################
#Creating outcome variables
```

[1] http://www.stat.columbia.edu/ gelman/research/published/STS226.pdf

[2] http://www.stat.columbia.edu/ gelman/research/published/parkgelmanbafumi.pdf

```
patients <- NULL
number_waited <- NULL
mean_wait_time <-NULL
close_late <- NULL

#Creating intermediate variables
patient_arrival_dist <- NULL
pats <- NULL
arrival_time <- NULL
time_between <- NULL
leave_time <-NULL
doc_time <- NULL
wait_time <- NULL

#Number of patients in one day
patient_arrival_dist <- rexp(1000, 1/10)
for (i in 1:1000){
        pats[i] <- sum(patient_arrival_dist[1:i])
        }

patients <- length(pats[pats < 420])

#Time between the arrival of each patient
time_between <- patient_arrival_dist[1:patients]

#Arrival time of a patient in minutes after 9am
for (i in 1:patients){
        arrival_time[i] <- sum(time_between[1:i])
}

#Time spent by each patient with a doctor
doc_time <- runif(patients, 5, 20)

#Waiting time for a typical patient
wait_time[1:3] <- c(0,0,0)
```

```
leave_time[1:2] <- arrival_time[1:2] + doc_time[1:2]

for (i in 4: patients){
        leave_time[i-1] <- arrival_time[i-1] + wait_time[i-1] + doc_time [i-1]
        wait_time[i] <- ifelse(
        length((leave_time[1:(i-1)])[leave_time[1:(i-1)] > arrival_time[i]])
        < 3, 0,
        min((leave_time[1:(i-1)])[leave_time[1:(i-1)] >
        arrival_time[i]]) - arrival_time[i])
        }
leave_time <- arrival_time + doc_time + wait_time

#Answers to part A
patients
a <- table(wait_time)
length(a[names(a) != 0])
number_waited <- length(a[names(a) != 0])
mean_wait_time <- ifelse(
  length(wait_time[wait_time > 0])==0,0, mean(wait_time[wait_time > 0]))
close_time <- ifelse(max(leave_time > 420),max(leave_time) - 420,0)


############################
Answer b)
############################

#Creating outcome variables
patients <- NULL
number_waited <- NULL
mean_wait_time <-NULL
close_late <- NULL

for(n in 1:100){

#Creating intermediate variables
```

```
patient_arrival_dist <- NULL
pats <- NULL
arrival_time <- NULL
time_between <- NULL
leave_time <-NULL
doc_time <- NULL
wait_time <- NULL

#Number of patients in one day

patient_arrival_dist <- rexp(1000, 1/10)
for (i in 1:1000){
        pats[i] <- sum(patient_arrival_dist[1:i])
        }

patients[n] <- length(pats[pats < 420])

#Time between the arrival of each patient
time_between <- patient_arrival_dist[1:patients[n]]

#Arrival time of a patient in minutes after 9am
for (i in 1:patients[n]){
        arrival_time[i] <- sum(time_between[1:i])
}

#Time spent by each patient with a doctor
doc_time <- runif(patients[n], 5, 20)

#Waiting time for a typical patient
wait_time[1:3] <- c(0,0,0)
leave_time[1:2] <- arrival_time[1:2] + doc_time[1:2]
for (i in 4: patients){
        leave_time[i-1] <- arrival_time[i-1] + wait_time[i-1] + doc_time [i-1]
        wait_time[i] <- ifelse(
        length((leave_time[1:(i-1)])[leave_time[1:(i-1)] > arrival_time[i]])
```

```
        < 3, 0,
        min((leave_time[1:(i-1)])[leave_time[1:(i-1)] >
        arrival_time[i]]) - arrival_time[i])
        }

leave_time <- arrival_time + doc_time + wait_time


a <- table(wait_time)
length(a[names(a) != 0])
number_waited[n] <- length(a[names(a) != 0])
mean_wait_time[n] <- ifelse(
  length(wait_time[wait_time > 0])==0,0, mean(wait_time[wait_time > 0]))
close_time[n] <- ifelse(max(leave_time > 420),max(leave_time) - 420,0)
}

summary(patients)
summary(number_waited)
summary(mean_wait_time)
summary(close_time)
```