Characterizing Measurement Data

Assignment 1

Mina Mousavifar - 11279515 - sem311

2019-10-01 10:11:18

Question 1.

a)

$$\lambda = 5\frac{session}{minute} \rightarrow mean = \frac{1}{\lambda} = \frac{1minute}{5session} \times \frac{60seconds}{1minute} = 12seconds$$

b)

Exponential distribution has *memoryless property*. It means that no matter how much time has elapsed since the previous arrival, the mean waiting time at time T for the next entry will still be 12 seconds.

c)

$$arrivals = period * lambda = 2minutes * 5\frac{arrival}{minute} = 10arrivals$$

d)

Because interarrival times are IID, the first arrival doesn't change the mean.

$$time = 3 * 60 = 180seconds \rightarrow arrivals = \left\lfloor \frac{180 - 18}{12\frac{arrival}{seconds}} \right\rfloor = 13 + 1(firstarrival) = 14$$

Question 2.

a)

W:

$$mean = \frac{1}{\lambda} = 12 \rightarrow \lambda = \frac{1}{12} = 0.08\overline{3}$$

X:

$$mean = n\gamma \rightarrow \gamma = \frac{12}{3} = 4$$

Y:

$$mean = e^{(\mu+\frac{\sigma^2}{2})} \rightarrow e^{(1+\frac{\sigma^2}{2})} = 12 \rightarrow ln(e^{(1+\frac{\sigma^2}{2})}) = ln(12) \rightarrow 1+\frac{\sigma^2}{2} = ln(12) \rightarrow \frac{\sigma^2}{2} = ln(12)-1 \rightarrow \sigma^2 = 2ln(12)-2$$

$$\sigma = \pm(2ln(12) - 2)$$

```r
sigma=sqrt(2*log(12) - 2)
cat("sigma is: +-",sigma)
```

```
## sigma is: +- 1.723315
```

Z:

$$mean = \begin{cases} \infty & for \quad \alpha \leq 1 \\ \frac{\alpha * k}{\alpha - 1} & for \quad \alpha > 1 \end{cases}$$

$$\alpha = 1.25 \rightarrow mean = \frac{\alpha * k}{\alpha - 1} \rightarrow 12 = \frac{1.25k}{1.25 - 1} \rightarrow 12 = \frac{1.25k}{0.25} \rightarrow 12 = 5k \rightarrow k = \frac{12}{5} = 2.4$$

b)

W:

$$F(x) = P(x < 12) = 1 - e^{-\lambda x} = 1 - e^{-\frac{1}{12}*12} = 1 - \frac{1}{e}$$

```r
p=pexp(12, rate=1/12)
cat("Probability for exponential distribution is: ",p)
```

```
## Probability for exponential distribution is:  0.6321206
```

X:

$$F(x) = P(x < 12) = 1 - \sum_{n=0}^{2} \frac{e^{-\frac{x}{\gamma}} x^n}{\gamma^n n!} = 1 - \sum_{i=0}^{2} \frac{e^{-\frac{x}{4}} x^n}{4^n n!}$$

```r
p=pgamma(12, shape= 3, scale= 4)
cat("Probability for erlang distribution is: ",p)
```

```
## Probability for erlang distribution is:  0.5768099
```

Y:

$$F(x) = P(x < 12) = \phi(\frac{(lnx) - \mu}{\sigma})$$

where ϕ is the cumulative distribution function of the standard normal distribution.

```r
p=plnorm(12, meanlog = 1, sdlog = sigma)
cat("Probability for lognormal distribution is: ",p)
```

```
## Probability for lognormal distribution is:  0.8055619
```

Z:

$$F(x) = P(x < 12) = 1 - \frac{k}{x}^{\alpha}$$

```r
library(EnvStats)
p=ppareto(12, location=2.4, shape=1.25)
cat("Probability for pareto distribution is: ",p)
```

```
## Probability for pareto distribution is:  0.8662519
```

c)

W:

$$1 - P(x < 240) = 1 - (1 - e^{-\lambda x}) = e^{-\frac{1}{12}*240} = \frac{1}{e^{20}}$$

```r
p= 1 - pexp(240, rate=1/12)
cat("Probability for exponential distribution is: ",p)
```

```
## Probability for exponential distribution is:  2.061154e-09
```

X:

$$1 - P(x < 240) = 1 - (1 - \sum_{n=0}^{2} \frac{e^{-\frac{x}{\gamma}} x^n}{\gamma^n n!}) = \sum_{i=0}^{2} \frac{e^{-\frac{x}{4}} x^n}{4^n n!}$$

```r
p= 1 - pgamma(240, shape= 3, scale= 4)
cat("Probability for erlang distribution is: ",p)
```

```
## Probability for erlang distribution is:  0
```

Y:

$$1 - P(x < 240) = 1 - \phi(\frac{(lnx) - \mu}{\sigma})$$

where ϕ is the cumulative distribution function of the standard normal distribution.

```r
p= 1 - plnorm(240, meanlog = 1, sdlog = sigma)
cat("Probability for lognormal distribution is: ",p)
```

```
## Probability for lognormal distribution is:  0.004661023
```

Z:

$$1 - P(x < 240) = 1 - (1 - \frac{k}{x}^{\alpha}) = \frac{k}{x}^{\alpha}$$

```r
library(EnvStats)
p= 1 - ppareto(240, location=2.4, shape=1.25)
cat("Probability for pareto distribution is: ",p)
```

```
## Probability for pareto distribution is:  0.003162278
```

Question 3.

$$P_R(n) = \frac{\frac{1}{n^{\alpha}}}{\sum_{m=1}^{k} \frac{1}{m^{\alpha}}} \rightarrow i^{-\alpha} < \frac{1}{day} = \frac{1}{60 * 60 * 24 seconds} = \frac{1}{86400} \rightarrow i^{\alpha} > 86400 \rightarrow i^{1.25} > 86400$$

```r
library(dplyr)
library(sads)

powers <- data.frame("i"=0:10000000, "pow"=rep(0,10000001), stringsAsFactors = FALSE)
powers$pow <- powers$i^1.25
is <- powers %>% filter(pow > 86400)
cat("i: ", is$i[1])
```

```
## i:  8897
```

```r
p = 1 - pzipf(is$i[1], 10000000, 1.25, lower.tail=TRUE, log.p=FALSE)
cat("CDF for cold items of Zipf(1.25) is: ", p)
```

```
## CDF for cold items of Zipf(1.25) is:  0.07531477
```

```r
powers$pow <- powers$i^0.85
is <- powers %>% filter(pow > 86400)
cat("i: ", is$i[1])
```

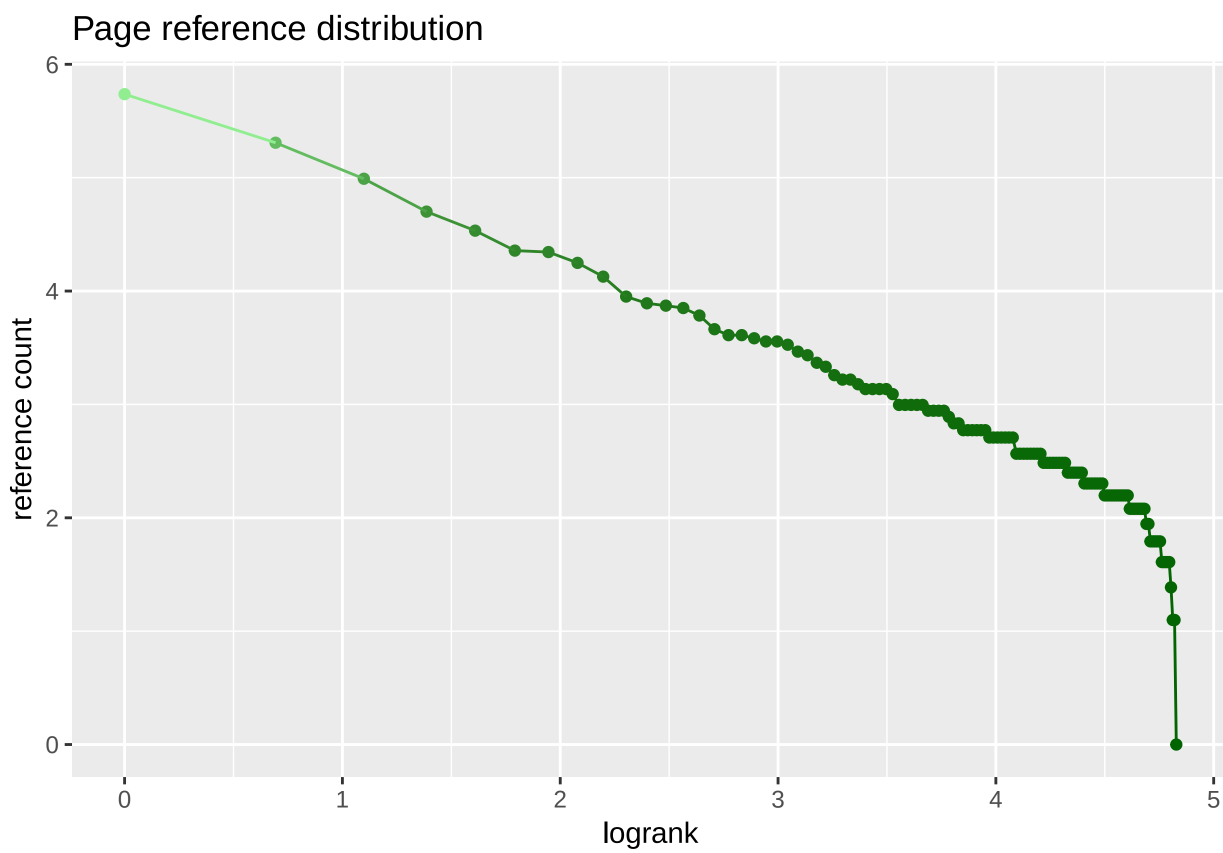
```
## i:  642190
```

```r
p = 1 - pzipf(is$i[1], 10000000, 0.85, lower.tail=TRUE, log.p=FALSE)
cat("CDF for cold items of Zipf(0.85) is: ", p)
```

```
## CDF for cold items of Zipf(0.85) is:  0.3675296
```

Question 4.

```r
library(ggplot2)

data <- read.delim("data/refcounts.txt", header = FALSE, sep = "\t", stringsAsFactor = FALSE)
names(data) <- c("index", "refcount")
data <- data %>% mutate(rank=row_number(-refcount)) %>% mutate(logcount=log(refcount), logrank=log(rank

p <- ggplot(data = data, mapping = aes(x = logrank, y=logcount, color = refcount)) +  geom_point() + gg
p
```




```r
xm <- min(data[,2])
#pareto<- data %>% mutate()

lzipf <- function(s,N) -s*log(1:N)-log(sum(1/(1:N)^s))
opt.f <- function(s) sum((data$logcount-lzipf(s,length(data$logcount)))^2)
opt <- optimize(opt.f,c(0.5,125))
opt
```

```
## $minimum
## [1] 0.5000669
##
## $objective
## [1] 7420.24
```

Question 5.

a)

for calculating interarrival times in this question, we calculate exact second for each data and find the differrence between each two consequtive row. Although we should manage the difference between 23 and 00 hour.

```r
library(lubridate)

fulldata <- read.delim("data/sep27.txt", header = FALSE, sep = "\t", stringsAsFactor = FALSE)
halfdata <- read.delim("data/sep27h12-16.txt", header = FALSE, sep = "\t", stringsAsFactor = FALSE)

names(fulldata) = c('time')
names(halfdata) = c('time')

fulldata <- fulldata %>% mutate(time = parse_date_time(time, orders="HMS")) %>%
  mutate(hour = hour(time), minute = minute(time), second = second(time)) %>%
  mutate(tot = hour*3600 + minute*60 + second) %>%
  mutate(diff = tot - lag(tot), diff = ifelse(diff < 0, diff + 86400, diff))
# delete first row for removing one data without any prior to calculate
full_data <- fulldata[-1,]

halfdata <- halfdata %>% mutate(time = parse_date_time(time, orders="HMS")) %>%
  mutate(hour = hour(time), minute = minute(time), second = second(time)) %>%
  mutate(tot = hour*3600 + minute*60 + second) %>%
  mutate(diff = tot - lag(tot))
half_data <- halfdata[-1,]

fullmean = mean(full_data$diff)
fullsc = var(full_data$diff)/(fullmean^2)

halfmean = mean(half_data$diff)
halfsc = var(half_data$diff)/(halfmean^2)
cat("mean interarrival times for sep 27 trace is: ", fullmean, " seconds and variance is: ", fullsc, " s
```

```
## mean interarrival times for sep 27 trace is:  42.43435  seconds and variance is:  9.557852  seconds
```

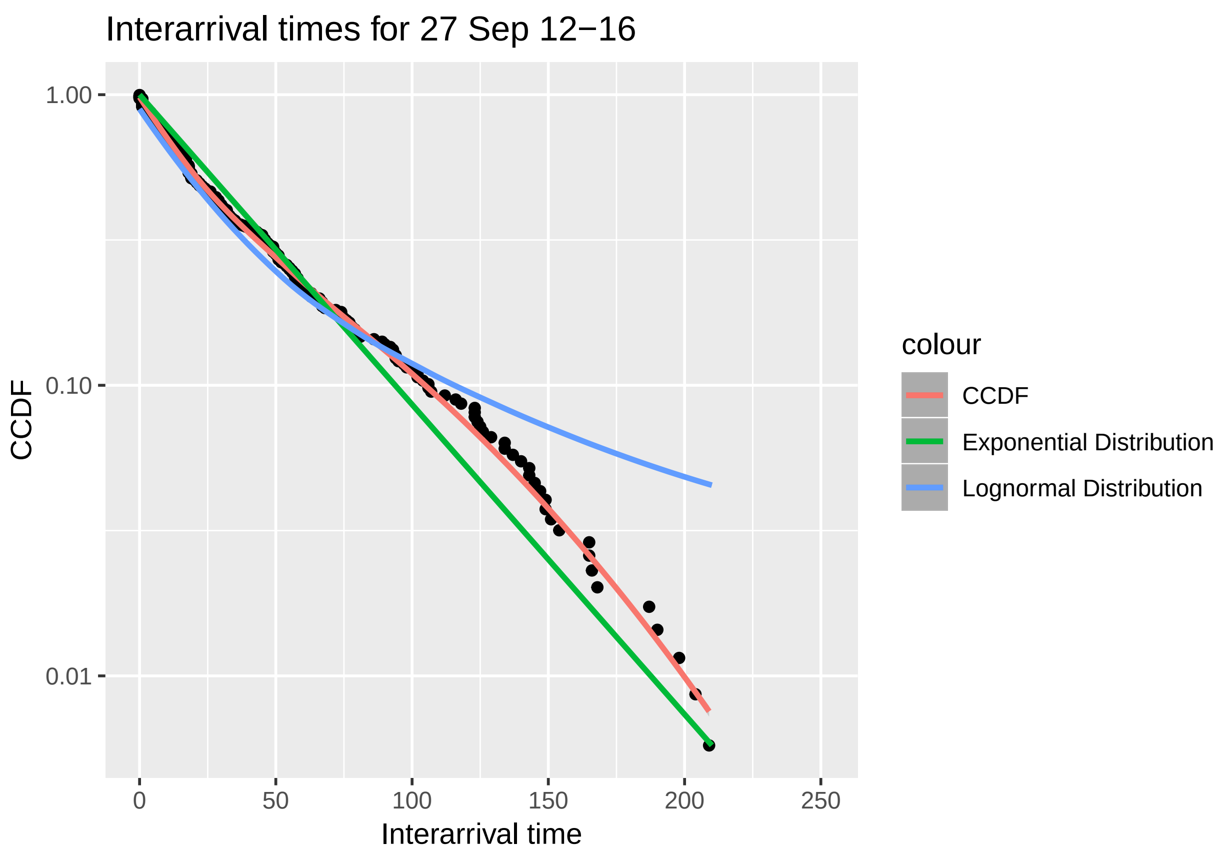
```r
cat("mean interarrival times for sep 27 12 - 16 trace is: ", halfmean, " seconds variance is: ", halfsc
```

```
## mean interarrival times for sep 27 12 - 16 trace is:  40.73487  seconds variance is:  1.383628  secon
```

b)

```r
# i)
n = nrow(half_data)

plot_half_data <- half_data %>% filter(diff < 251) %>%
  arrange(diff) %>%
  mutate(ccdf = (n - row_number())/n)

# ii)
x = seq(min(plot_half_data$diff), max(plot_half_data$diff) + 1, length.out=n)
exp = data.frame(x=x, px=1-pexp(x, rate=1/halfmean))

# iii)
plot_half_data <- plot_half_data %>% mutate(zero_diff = ifelse(diff == 0, log(0.5), log(diff)))
mu <- sum(plot_half_data$zero_diff)/n

plot_half_data <- plot_half_data %>% mutate(minus_mu_diff = (zero_diff - mu)^2)
```

```
sigma <- sqrt(sum(plot_half_data$minus_mu_diff)/n)
lognorm = data.frame(x=x, px=1-plnorm(x, meanlog = mu, sdlog = sigma))


p <- ggplot(data = plot_half_data, mapping = aes(x = diff, y = ccdf)) +
  xlim(0,251) + ylim(NA, 1) + scale_y_continuous(trans = 'log10') + ggtitle("Interarrival times for 27 S
  geom_point() +
  geom_smooth(data = plot_half_data, aes(x = diff, y = ccdf, color = "CCDF")) +
  geom_smooth(data = exp, aes(x = x, y = px, color = "Exponential Distribution")) +
  geom_smooth(data = lognorm, aes(x = x, y = px, color = "Lognormal Distribution"))

p
```

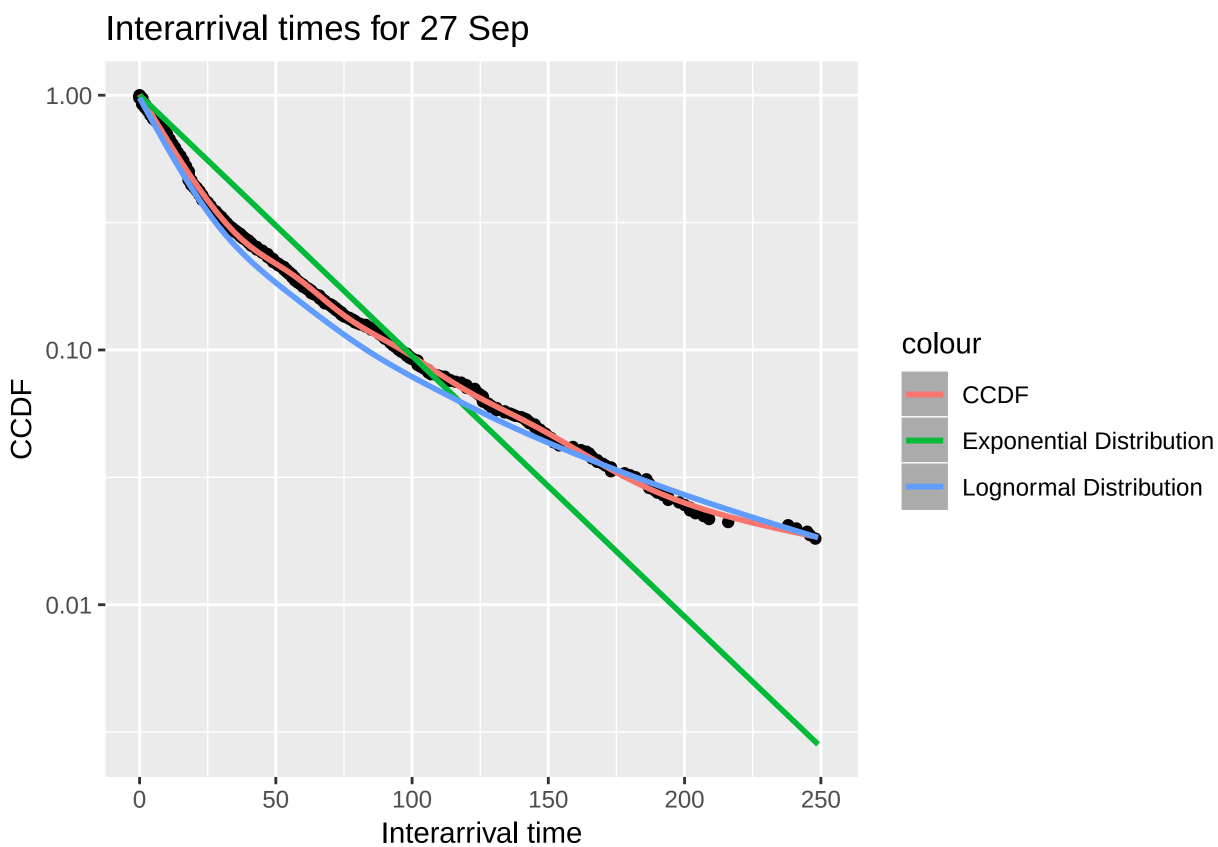


As we can see here, at the beginning, lognormal distribution is a better fit than exponential distribution. However, as we proceed to bigger interarrival times, both distributions are taking distance from the original distribution.

c)

```
# i)
n = nrow(full_data)

plot_full_data <- full_data %>% filter(diff < 251) %>%
  arrange(diff) %>%
  mutate(ccdf = (n - row_number())/n)
```

```r
# ii)
x = seq(min(plot_full_data$diff), max(plot_full_data$diff) + 1, length.out=n)
exp = data.frame(x=x, px=1-pexp(x, rate=1/fullmean))

# iii)
plot_full_data <- plot_full_data %>% mutate(zero_diff = ifelse(diff == 0, log(0.5), log(diff)))
mu <- sum(plot_full_data$zero_diff)/n

plot_full_data <- plot_full_data %>% mutate(minus_mu_diff = (zero_diff - mu)^2)
sigma <- sqrt(sum(plot_full_data$minus_mu_diff)/n)
lognorm = data.frame(x=x, px=1-plnorm(x, meanlog = mu, sdlog = sigma))


p <- ggplot(data = plot_full_data, mapping = aes(x = diff, y = ccdf)) +
  xlim(0,251) + ylim(NA, 1) + scale_y_continuous(trans = 'log10') + ggtitle("Interarrival times for 27 S
  geom_point() +
  geom_smooth(data = plot_full_data, aes(x = diff, y = ccdf, color = "CCDF")) +
  geom_smooth(data = exp, aes(x = x, y = px, color = "Exponential Distribution")) +
  geom_smooth(data = lognorm, aes(x = x, y = px, color = "Lognormal Distribution"))

p
```



As we can see here, lognormal distribution is completely a better fit than exponential distribution and is acting close to the original distribution.