

Characterizing Measurement Data

Assignment 1

Mina Mousavifar - 11279515 - sem311

Question 1.

a)

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

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)

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

d)

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

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

Question 2.

a)

W:

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

X:

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

Y:

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

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

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

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

Z:

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

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

b)

W:

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

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

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

X:

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

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

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

Y:

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

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

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

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

Z:

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

```
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:

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

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

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

X:

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

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

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

Y:

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

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

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

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

Z:

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

```
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.

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

```
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
```

```
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
```

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

```
## i: 642190
```

```
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)
```

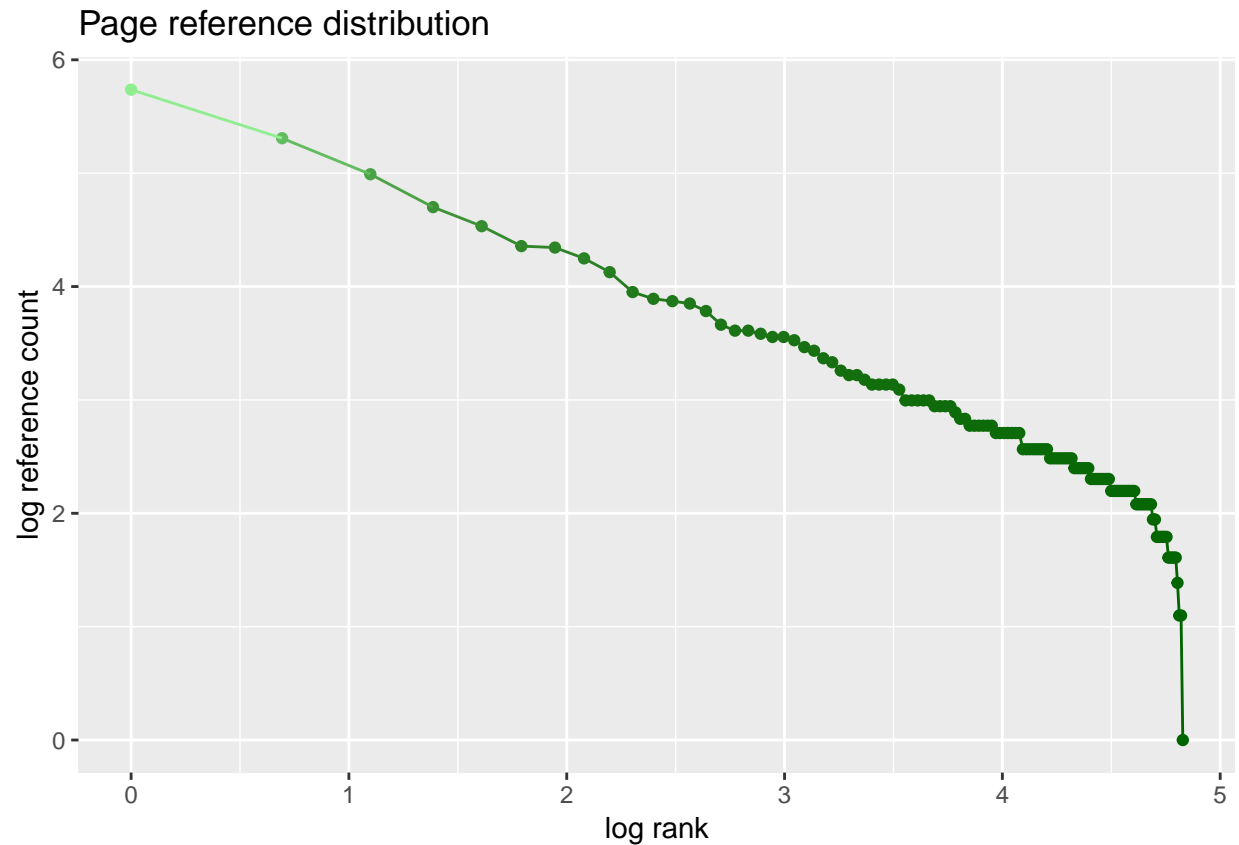
```
## CDF for cold items of Zipf(0.85) is: 0.3675296
```

Question 4.

```
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() + ggtitle("Page reference distribution") +
  ylab("log reference count") + xlab("log rank") +
  scale_color_gradient(low="darkgreen", high="lightgreen") + geom_line() +
  theme(legend.position = "none")
p
```



```
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
```

```
cat("alpha is: ", opt$minimum)
```

```
## alpha is: 0.5000669
```

Question 5.

a)

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

```

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")

## mean interarrival times for sep 27 trace is: 42.43435 seconds

cat("variance for sep 27 trace is: ", fullsc, " seconds")

## variance for sep 27 trace is: 9.557852 seconds

cat("mean interarrival times for sep 27 12 - 16 trace is: ", halfmean, " seconds")

## mean interarrival times for sep 27 12 - 16 trace is: 40.73487 seconds

cat("variance for sep 27 12 - 16 trace is: ", halfsc, " seconds")

## variance for sep 27 12 - 16 trace is: 1.383628 seconds

b)

# i)
n = nrow(half_data)

plot_half_data <- half_data %>% filter(diff < 251)

n = nrow(plot_half_data)

```

```

plot_half_data <- plot_half_data %>% 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
cat("mean for 27 Sep 12-16", mu)

```

```
## mean for 27 Sep 12-16 2.92046
```

```

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))
cat("sigma for 27 Sep 12-16", sigma)

```

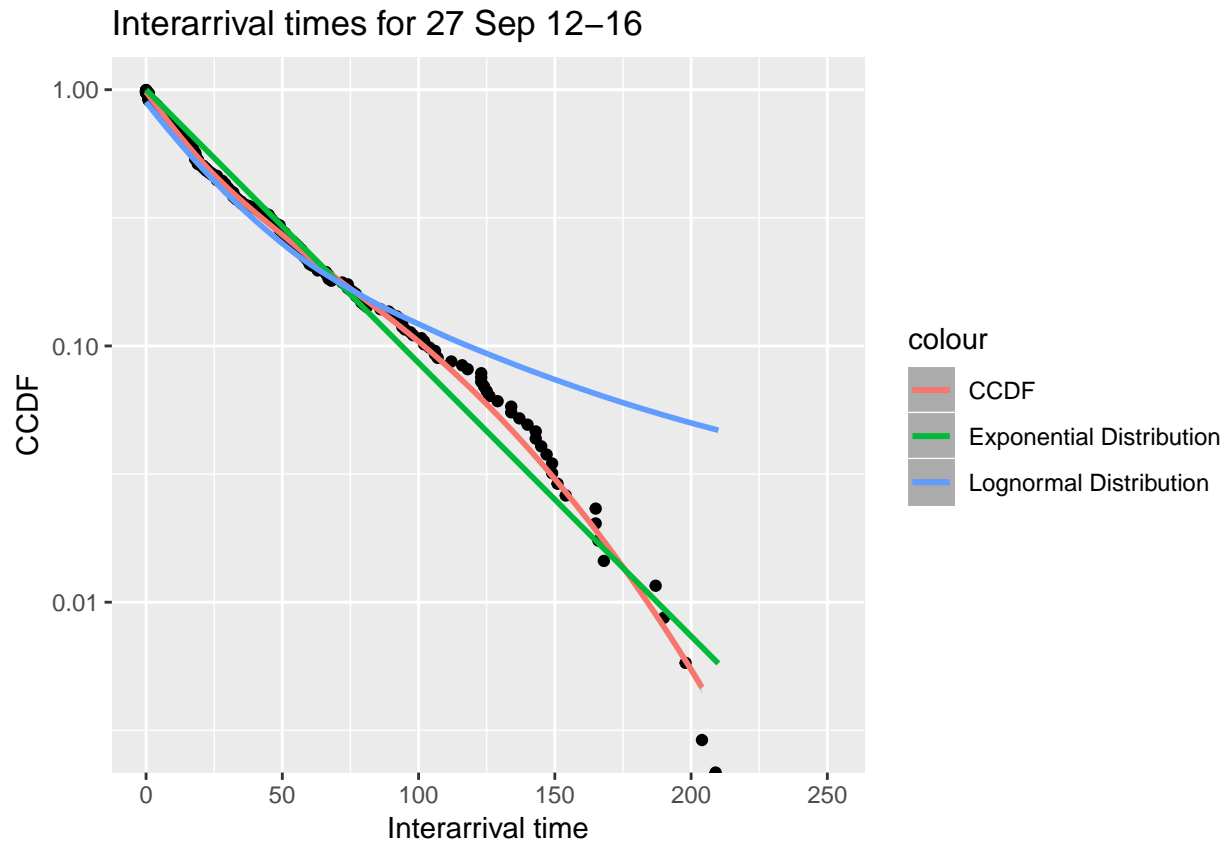
```
## sigma for 27 Sep 12-16 1.445565
```

```

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 Sep 12-16") +
  xlab("Interarrival time") + ylab("CCDF") +
  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 %>% arrange(diff) %>%
  mutate(ccdf = (n - row_number())/n)

# ii)
x = seq(min(plot_full_data$diff), 250, 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
cat("mean for 27 Sep", mu)
```

```
## mean for 27 Sep 2.801741
```

```
plot_full_data <- plot_full_data %>% mutate(minus_mu_diff = (zero_diff - mu)^2)
sigma <- sqrt(sum(plot_full_data$minus_mu_diff)/n)
```

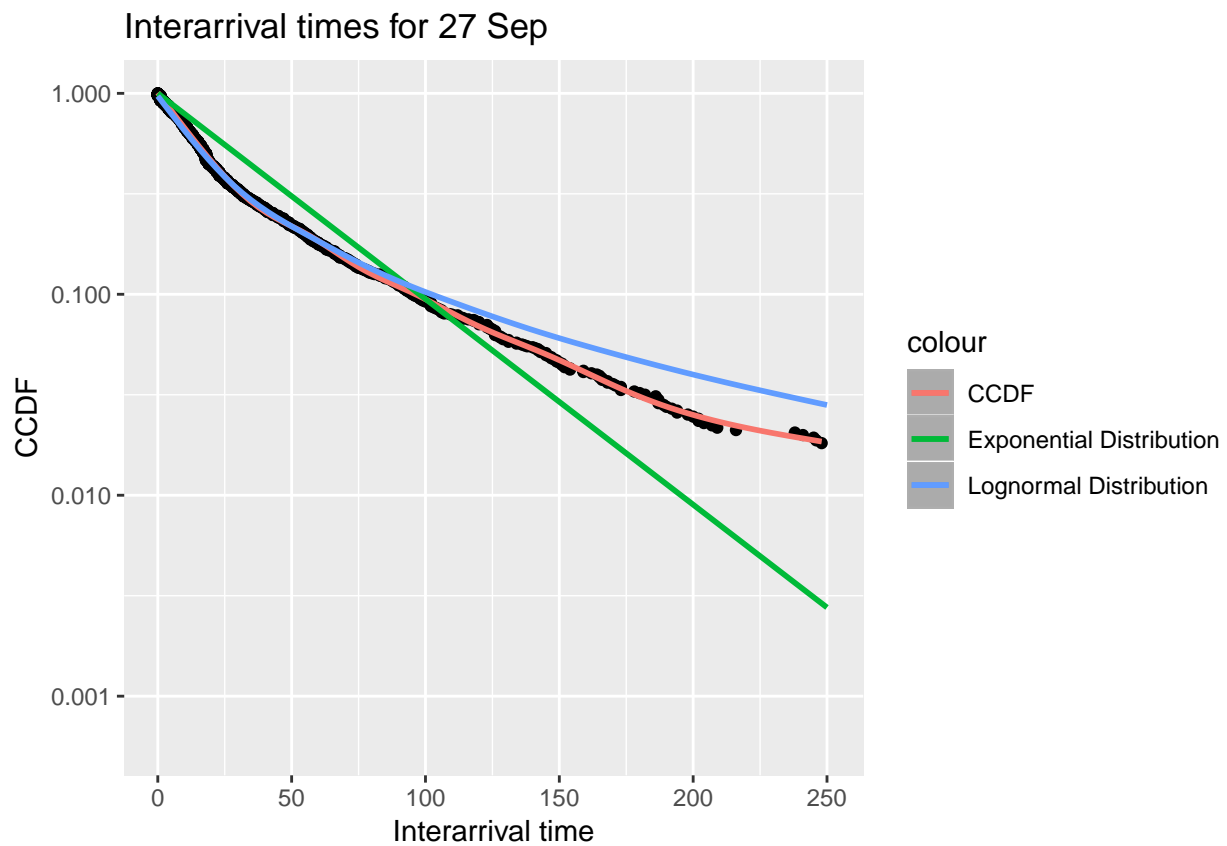


```
lognorm = data.frame(x=x, px=1-plnorm(x, meanlog = mu, sdlog = sigma))
cat("sigma for 27 Sep", sigma)
```

```
## sigma for 27 Sep 1.424872
```

```
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 Sep") +
  xlab("Interarrival time") + ylab("CCDF") +
  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.