End Semester Project

33506: Statistical analysis simulation and modeling 2

Cho Keunhee
26001904131

I. Get four data samples from the files named “datafile1.csv” - “datafile4.csv” (attached to the same email, with which you received this document), and conduct a visual analysis of the data (plot histograms, etc.) Based on the analysis, select one data sample that, in your opinion, would describe water temperature measured on a daily basis in a river.

In order to effectively conduct a visual analysis of the given data, this paper have used histograms. The following are the histograms of each data given.

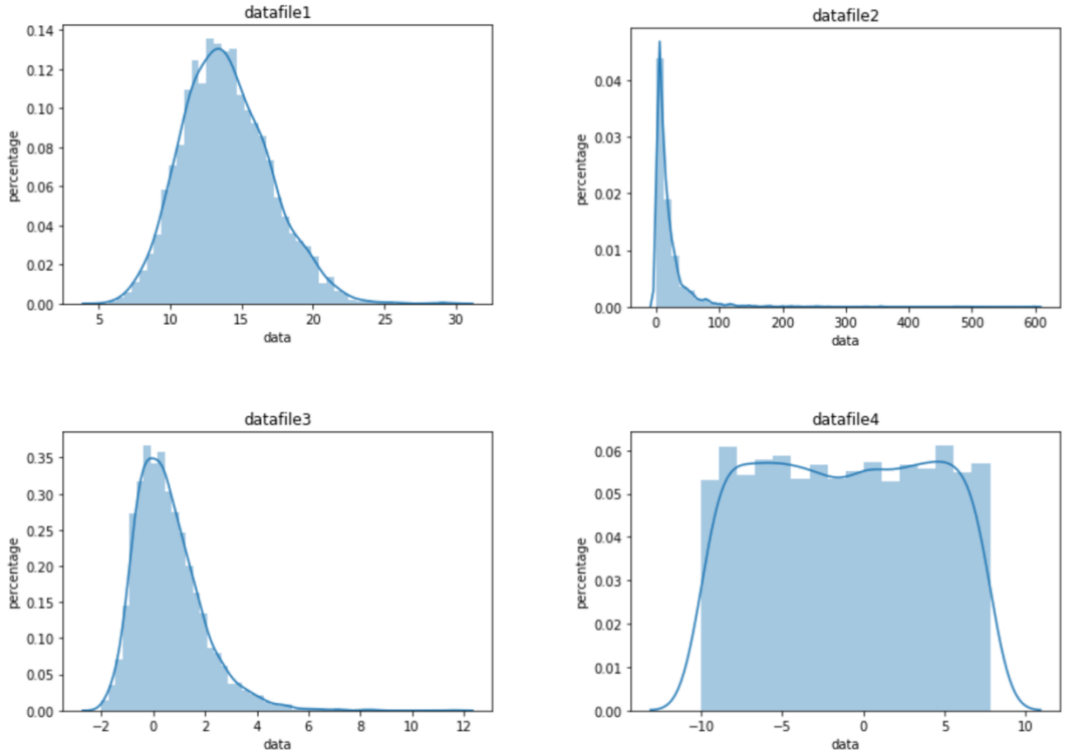

Figure 1. Histogram and pdf of datafile1, datafile2, datafile3 and datafile

Based on the data analysis the following is the table of mean, min and max of the four datasets.

	datafile1	datafile2	datafile3	datafile4
mean	13.9346	20.9615	0.6060	-1.0911
max	29.3315	599.4140	11.6170	7.7965
min	5.7206	0.1923	-1.9985	-9.9990

Table 1 mean, max, and min value of given datasets

Based on the conducted visual analysis and the values from the chart. The dataset which best describes the water temperature measured on a daily basis in a river is the first datafile. Scientifically, water can only hold a temperature between 0 and 100. Based on this fact, the only dataset fit is the first dataset. Moreover, based on the instruction that it is a daily measured data. Therefore, to check my estimation I have plot the 30 data points of the datafile1. The following is the result. The first graph is datapoint 1 to 30 and the second graph consist of datapoint 181 to 210. The following number were chosen because this paper wanted to conduct two different seasons to see the difference.

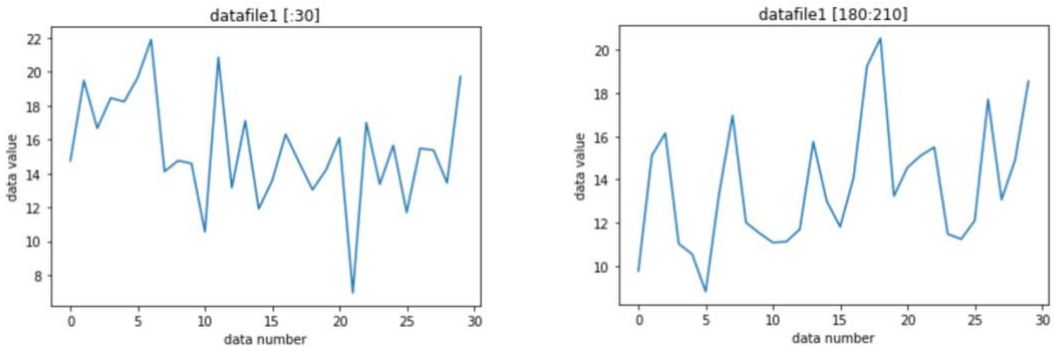

Figure 2. datafile1 data point from 1 to 30(left), datafile1 data point from 181 to 210(right)

As shown from two graphs above, it is shown that among 30 days the values of the data have a very large scale from 8 to 22. Moreover, the datapoint does not seem to have a great difference between two seasons. Concluding that it is hard to determine that this dataset holds characteristic of a normal temperature data of a river. This assumption could be rebutted if the data is collected more than once a day (day and night). If so the rapid change between the data might be explained. However, still considering the fact that temperature between two given figures are similar could imply us that there is not much of climate change throughout the season. Therefore, based on scientific fact, if I have to choose one datafile as a river temperature it would be datafile1. However, based on the analyzation it is hard to conclude with the decision, with only the information given.

II. Plot PDFs of the three distributions listed below, and explore the shapes of the PDFs by varying (assigning different values to) their parameters:

i) Lognormal distribution (two parameters, μ and σ):

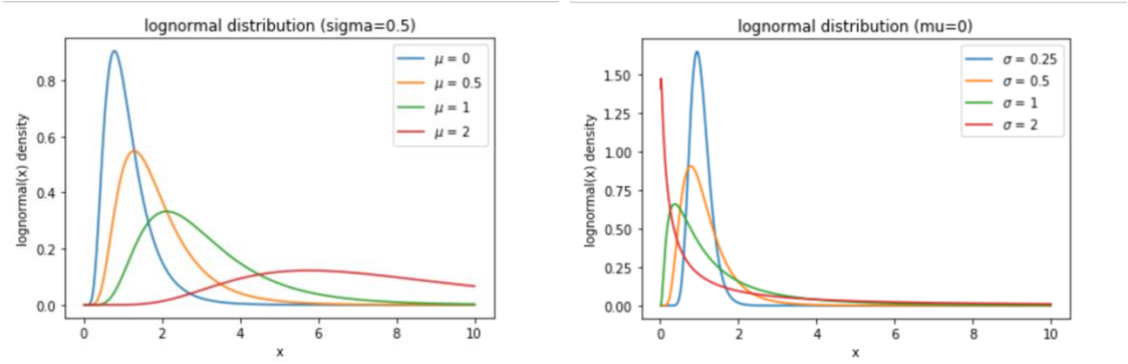

Figure 3. lognormal distribution with changing parameters μ(left), σ(right)

Based on figure3, parameter μ is the scale parameter, while σ is the shape parameter. This is because as value of μ is bigger the more spread out the distribution is. Moreover, for σ, the shape of the distribution has changed as the value changed.

ii) Weibull distribution (two parameters, c and b):

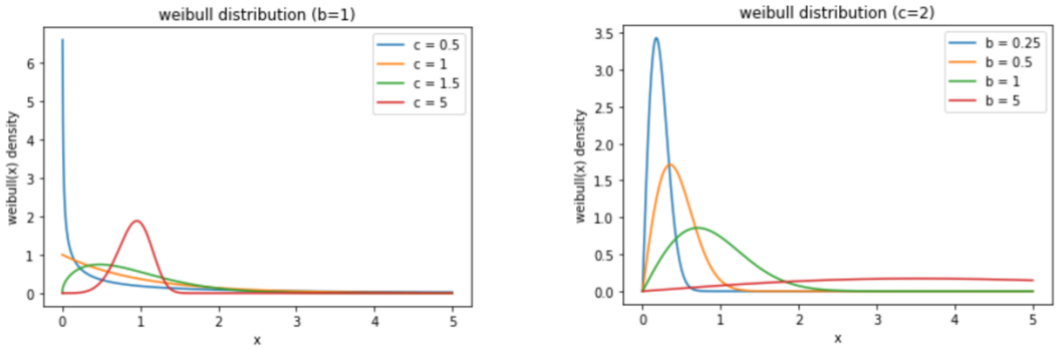

Figure 4. Weibull distribution with changing parameters c(left), b(right)

Based on figure4, parameter c is the shape parameter, while b is the scale parameter. This is because as value of b is bigger the more spread out the distribution is. Moreover, for c, the shape of the distribution has changed as the value changed.

iii) Wald distribution (two parameters, η and λ):

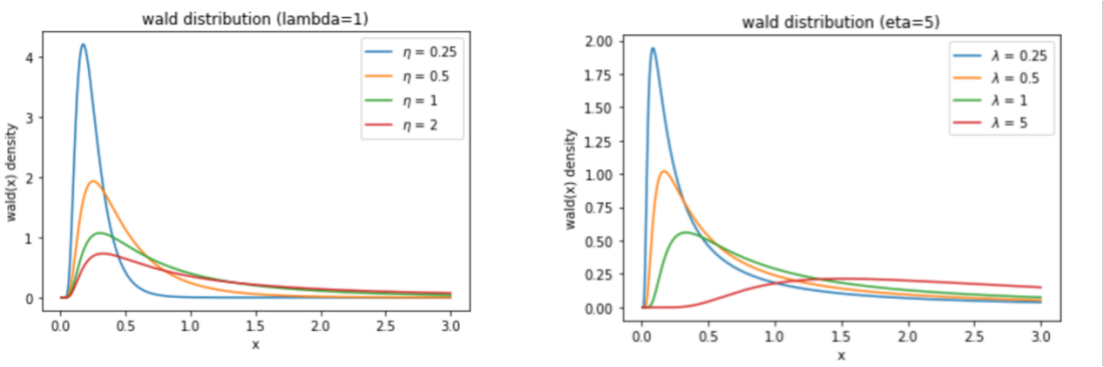

Figure 5. Wald distribution with changing parameters η(left), λ(right)

Based on figure5, both the parameter η and λ seems like a scale parameter. This is because for both the parameter, as the value grow the wider the distribution spread. Which is the characteristic of a scale parameter.

III. Use the data sample of “datafile2.csv” and obtain parameter estimates for the three models of (II). For each of the three models, assess and write down the accuracy of the obtained parameter estimates by constructing (e.g. via Bootstrap) the corresponding 99% CIs.

Using the library, we can compute the estimate the parameter of each models. This paper is using R for this part of the question. Using package fitdistrplus and package actuar to compute the following parameter estimates. Moreover, this report has used package boot, package fitdistrplus and package extraDistr to compute the bootstrap for each of the parameters and the confidence interval for each of the computed results.

i) Lognormal distribution

Based on the computation the following is the estimated value of parameter μ and σ.

```
> fln$estimate[1]
 meanlog
2.416327
> fln$estimate[2]
   sdlog
1.121919
```

μ (mean estimate) equals to 2.4163 and σ (standard deviation estimate) equals to 1.1220.

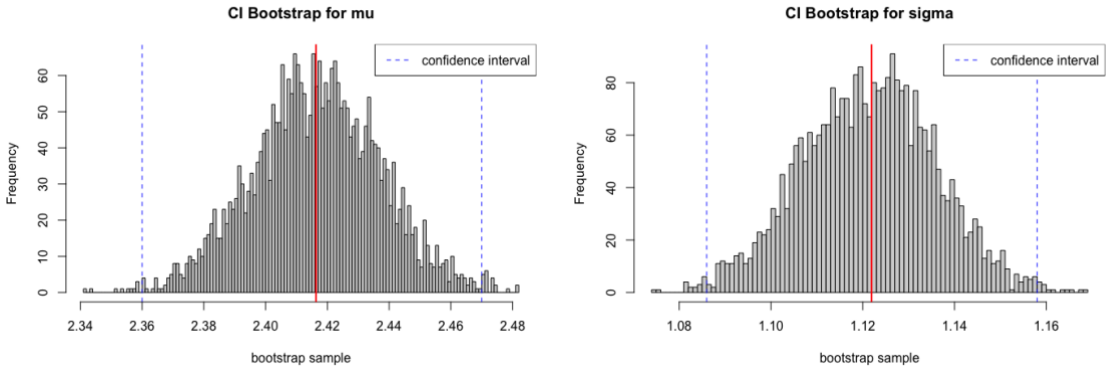

Figure 6. parameter estimation confidence interval bootstrap of lognormal distribution

Figure 6 show the confidence interval bootstrap for mean and standard deviation parameter with 99% of significance level. The CI computed is the following.

```
CALL :
boot.ci(boot.out = bs11, conf = 0.99, type = "basic")

Intervals :
Level      Basic
99%   ( 2.362,  2.471 )
Calculations and Intervals on Original Scale
```

```
CALL :
boot.ci(boot.out = bs12, conf = 0.99, type = "basic")

Intervals :
Level      Basic
99%   ( 1.086,  1.158 )
Calculations and Intervals on Original Scale
```

Confidence interval for mean estimate (μ) = [2.362, 2.471]
Confidence interval for standard deviation estimate (σ) = [1.086,1.158]

ii) Weibull distribution

Based on the computation the following is the estimated value of parameter c and b.

```
> fw$estimate[1]
    shape
0.8944451
> fw$estimate[2]
   scale
19.63096
```

c(shape estimate) equals to 0.8944 and b(scale estimate) equals to 19.6310.

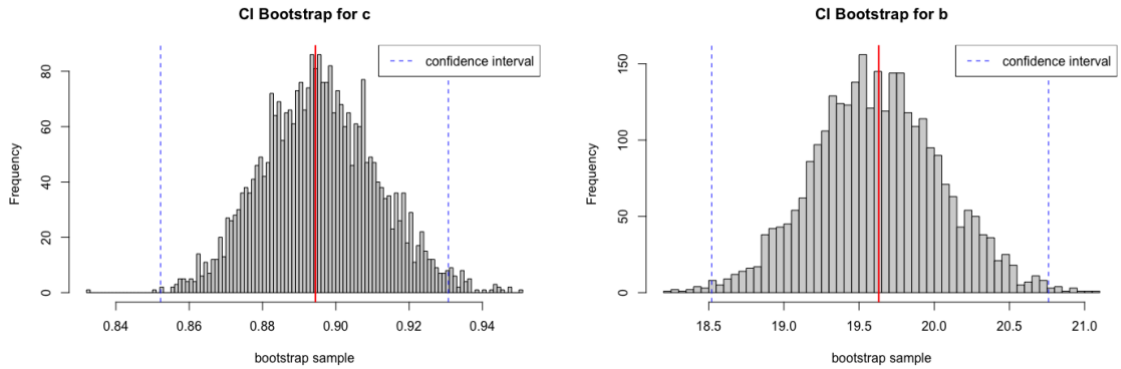

Figure 7. parameter estimation confidence interval bootstrap of weibull distribution

Figure 7 show the confidence interval bootstrap for shape and scale parameter with 99% of significance level. The CI computed is the following.

```
CALL :
boot.ci(boot.out = bs21, conf = 0.99, type = "basic")

Intervals :
Level      Basic
99%   ( 0.8522,  0.9307 )
Calculations and Intervals on Original Scale
```

```
CALL :
boot.ci(boot.out = bs22, conf = 0.99, type = "basic")

Intervals :
Level      Basic
99%   (18.52, 20.76 )
Calculations and Intervals on Original Scale
```

Confidence interval for shape estimate (c) = [0.8522, 0.9307]
Confidence interval for standard deviation estimate (b) = [18.52, 20.76]

iii) Wald distribution

Based on the computation the following is the estimated value of parameter η and λ.

```
> fi$estimate[1]
    mean
20.96648
> fi$estimate[2]
   shape
8.319866
```

η (mean estimate) equals to 20.9665 and λ (shape estimate) equals to 8.3199.

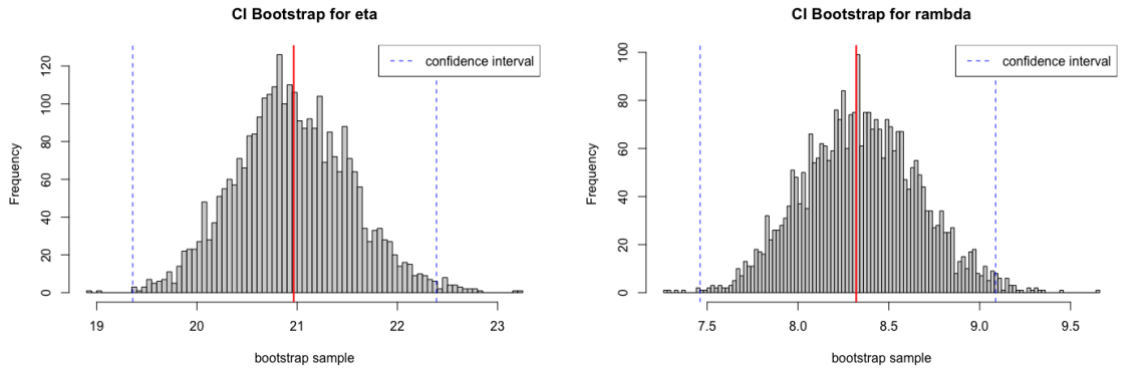

Figure 8. parameter estimation confidence interval bootstrap of waldl distribution

Figure 8 show the confidence interval bootstrap for mean and shape parameter with 99% of significance level. The CI computed is the following.

```
CALL :
boot.ci(boot.out = bs31, conf = 0.99, type = "basic")

Intervals :
Level      Basic
99%   (19.36, 22.39 )
Calculations and Intervals on Original Scale
```

```
CALL :
boot.ci(boot.out = bs32, conf = 0.99, type = "basic")

Intervals :
Level      Basic
99%   ( 7.461,  9.087 )
Calculations and Intervals on Original Scale
```

Confidence interval for mean estimate (η) = [19.36,22.39]
Confidence interval for shape estimate (λ) = [7.461,9.087]

IV. Consider the three models from (III) and conduct a model selection analysis. Select “the best model” for the data (i.e. the “best” distribution from (II) to describe the “datafile2.csv” sample). Justify your choice of “the best model” both qualitatively (e.g. with Q-Q plots) and quantitatively (e.g. with BIC, AIC, etc.)

i) Qualitative analysis

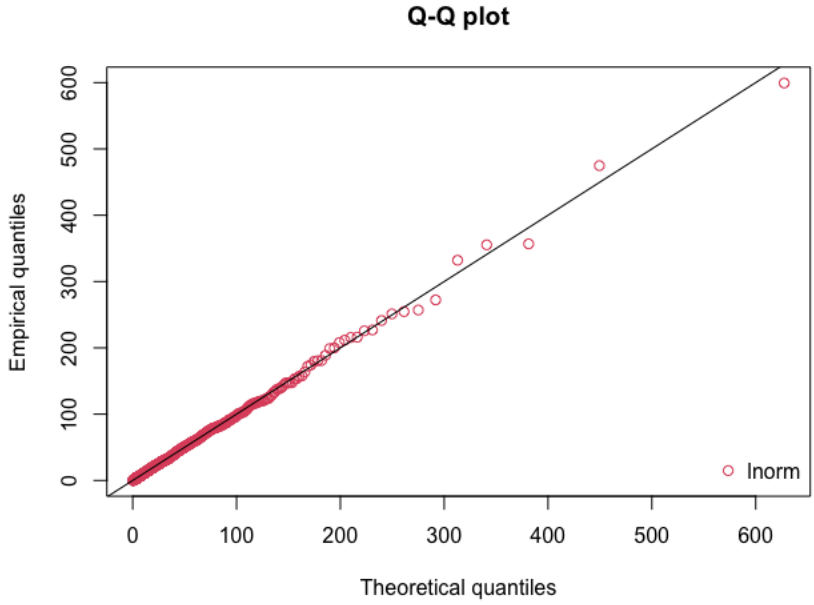

Figure 9. lognormal distribution QQplot

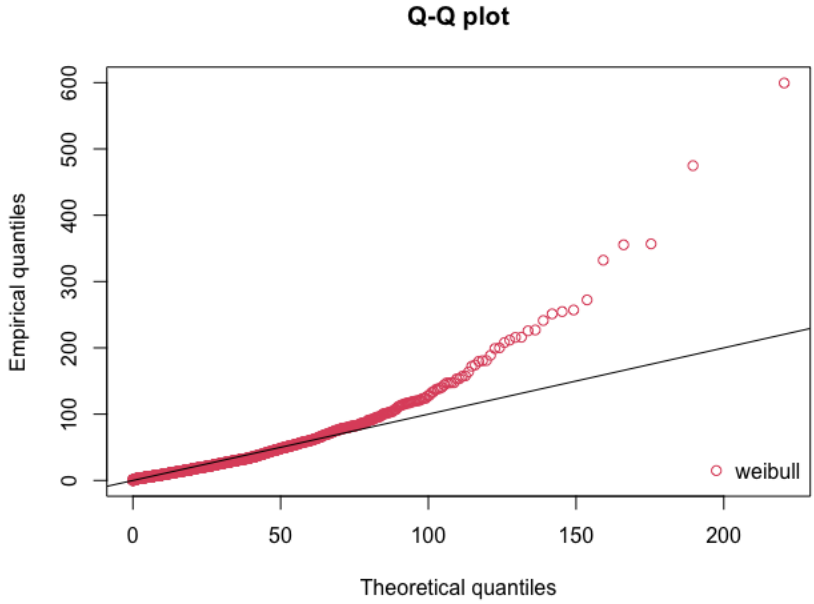

Figure 10. weibull distribution QQplot

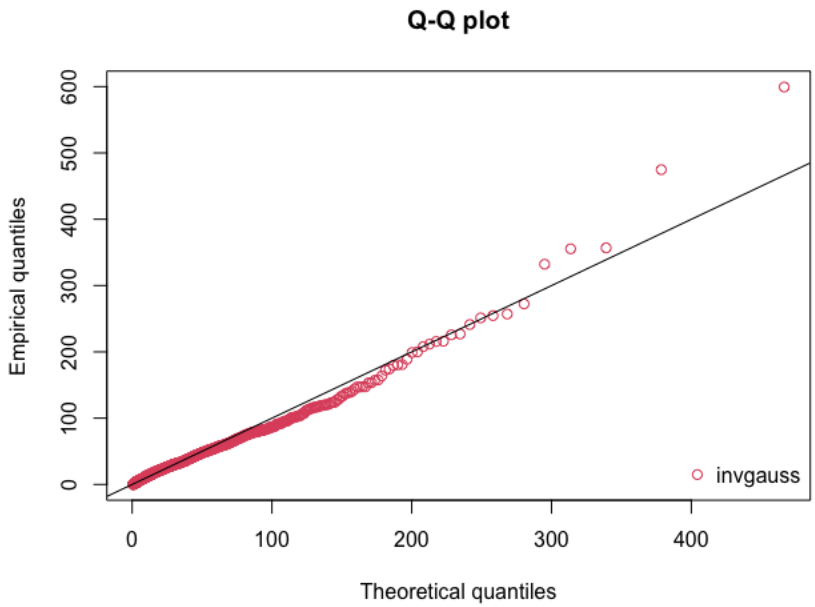

Figure 11. wald distribution QQplot

Based on the three QQplot (Figure 9 ,10 and 11), qualitatively, QQplot of lognormal distribution fits the 45degree line the best. Informing that it is the most accurate model among the three. Therefore, lognormal distribution could be considered as the "the best model" in qualitative view.

ii) Quantitative analysis

To compute the AIC and BIC of each three model this paper has implemented two functions from source code provided on Statsmodels documentation [1]. There are three parameters (llf, nobs and df_modelwc). llf is the value of loglikelihood, nobs are the number of observation and df_modelwc is the number of the parameters including the constant.

```
def aicc(llf, nobs, df_modelwc):
    return -2.0 * llf + 2.0 * df_modelwc * nobs / (nobs - df_modelwc - 1.0)
```

```
def bic(llf, nobs, df_modelwc):
    return -2.0 * llf + np.log(nobs) * df_modelwc
```

Using the given functions, the following is the AIC and BIC value for each three models.

```
AIC of lognormal distribution : [23705.88945929]
BIC of lognormal distribution : [23717.89819042]
AIC of weibull distribution   : [24179.92947088]
BIC of weibull distribution   : [24191.93820201]
AIC of wald distribution      : [23906.59236589]
BIC of wald distribution      : [23918.60109702]
```

From the given values it is shown that lognormal distribution holds the smallest AIC and BIC. Moreover, both Delta AIC and Delta BIC is larger than 10. Therefore, quantitatively lognormal distribution can be considered as ‘the best model’.

In conclusion, lognormal distribution is chosen as the true model considering both qualitatively and quantitatively.

Appendix

Question #1

```
# # Question1
# In[ ]:
import pandas as pd
import numpy as np
import seaborn as sns
import pylab as py
import matplotlib.pyplot as plt
# In[ ]:
df1 = pd.read_csv (r'datafile1.csv')
df2 = pd.read_csv (r'datafile2.csv')
df3 = pd.read_csv (r'datafile3.csv')
df4 = pd.read_csv (r'datafile4.csv')
X1 = np.array(df1)
X2 = np.array(df2)
X3 = np.array(df3)
X4 = np.array(df4)
# In[ ]:
py.figure(1)
sns.distplot(X1,hist=True)
plt.title("datafile1")
plt.xlabel("data")
plt.ylabel("percentage")
py.show()
print("Datafile1.csv")
print ("Max  :",np.nanmax(X1))
print ("Min  :",np.nanmin(X1))
print ("Mean :",np.nanmean(X1))

py.figure(2)
sns.distplot(X2,hist=True)
plt.title("datafile2")
plt.xlabel("data")
plt.ylabel("percentage")
py.show()
print("Datafile2.csv")
print ("Max  :",np.nanmax(X2))
print ("Min  :",np.nanmin(X2))
print ("Mean :",np.nanmean(X2))
py.figure(3)
sns.distplot(X3,hist=True)
plt.title("datafile3")
plt.xlabel("data")
plt.ylabel("percentage")
py.show()
print("Datafile3.csv")
print ("Max  :",np.nanmax(X3))
print ("Min  :",np.nanmin(X3))
print ("Mean :",np.nanmean(X3))

py.figure(4)
sns.distplot(X4,hist=True)
plt.title("datafile4")
plt.xlabel("data")
plt.ylabel("percentage")
py.show()
print("Datafile4.csv")
print ("Max  :",np.nanmax(X4))
print ("Min  :",np.nanmin(X4))
print ("Mean :",np.nanmean(X4))
X11 = X1[0:30]
X12 = X1[180:210]

plt.plot(X11)
plt.xlabel("data number")
plt.ylabel("data value")
plt.title("datafile1 [:30]")
plt.show()
plt.plot(X12)
plt.xlabel("data number")
plt.ylabel("data value")
plt.title("datafile1 [180:210]")
plt.show()
```

Question #2

```
# # Question2
# In[ ]:
import matplotlib.pyplot as plt
import numpy as np
# In[ ]:
x1 = np.linspace(0,10,1000)
x2 = np.linspace(0,5,1000)
x3 = np.linspace(0,3,1000)
# lognormal distribution

# In[ ]:
def lognormal(mu,sig):
    return  (1 / (x1 * sig * ((2 * np.pi) ** 0.5))) * np.exp(-(((np.log(x1) - mu) ** 2) / (2 * (sig ** 2))))
# weibull distribution
# In[ ]:
def weibull(c,b):
    return (c * (x2 ** (c - 1))) / ((b ** c) * np.exp((x2 / b) ** c))
# wald distribution
# In[ ]:
def wald(eta,lam):
    return (((lam) / (2 * np.pi * (x3 ** 3))) ** 0.5) * np.exp((-lam) * (((x3 - eta) ** 2)) / (2 * (eta ** 2) * x3))
# plot lognormal distribution
# In[ ]:
Log11 = lognormal(0,0.25)
Log12 = lognormal(0,0.5)
Log13 = lognormal(0,1)
Log14 = lognormal(0,2)
Log21 = lognormal(0,0.5)
Log22 = lognormal(0.5,0.5)
Log23 = lognormal(1,0.5)
Log24 = lognormal(2,0.5)
# In[ ]:
plt.figure(1)
plt.plot(x1,Log11,label=r'$\sigma$ = 0.25')
plt.plot(x1,Log12,label=r'$\sigma$ = 0.5')
plt.plot(x1,Log13,label=r'$\sigma$ = 1')
plt.plot(x1,Log14,label=r'$\sigma$ = 2')
plt.title('lognormal distribution (mu=0)')
plt.xlabel('x')
plt.ylabel('lognormal(x) density')
plt.legend()
plt.show()
# In[ ]:
plt.figure(2)
plt.plot(x1,Log21,label=r'$\mu$ = 0')
plt.plot(x1,Log22,label=r'$\mu$ = 0.5')
plt.plot(x1,Log23,label=r'$\mu$ = 1')
plt.plot(x1,Log24,label=r'$\mu$ = 2')
plt.title('lognormal distribution (sigma=0.5)')
plt.xlabel('x')
plt.ylabel('lognormal(x) density')
plt.legend()
plt.show()
# plot weibull distribution
# In[ ]:
Wei11 = weibull(0.5,1)
Wei12 = weibull(1,1)
Wei13 = weibull(1.5,1)
Wei14 = weibull(5,1)
Wei21 = weibull(2,0.25)
Wei22 = weibull(2,0.5)
Wei23 = weibull(2,1)
Wei24 = weibull(2,5)
# In[ ]:
plt.figure(3)
plt.plot(x2,Wei11,label=r'c = 0.5')
plt.plot(x2,Wei12,label=r'c = 1')
plt.plot(x2,Wei13,label=r'c = 1.5')
plt.plot(x2,Wei14,label=r'c = 5')
plt.title('weibull distribution (b=1)')
plt.xlabel('x')
plt.ylabel('weibull(x) density')
plt.legend()
plt.show()
plt.figure(4)
plt.plot(x2,Wei21,label=r'b = 0.25')
plt.plot(x2,Wei22,label=r'b = 0.5')
plt.plot(x2,Wei23,label=r'b = 1')
plt.plot(x2,Wei24,label=r'b = 5')
plt.title('weibull distribution (c=2)')
plt.xlabel('x')
plt.ylabel('weibull(x) density')
plt.legend()
plt.show()
```

```
# plot wald distribution
# In[ ]:
Wal11 = wald(0.25,1)
Wal12 = wald(0.5,1)
Wal13 = wald(1,1)
Wal14 = wald(2,1)
Wal21 = wald(5,0.25)
Wal22 = wald(5,0.5)
Wal23 = wald(5,1)
Wal24 = wald(5,5)
# In[ ]:
plt.figure(5)
plt.plot(x3,Wal11,label=r'$\eta$ = 0.25')
plt.plot(x3,Wal12,label=r'$\eta$ = 0.5')
plt.plot(x3,Wal13,label=r'$\eta$ = 1')
plt.plot(x3,Wal14,label=r'$\eta$ = 2')
plt.title('wald distribution (lambda=1)')
plt.xlabel('x')
plt.ylabel('wald(x) density')
plt.legend()
plt.show()
# In[ ]:
plt.figure(5)
plt.plot(x3,Wal21,label=r'$\lambda$ = 0.25')
plt.plot(x3,Wal22,label=r'$\lambda$ = 0.5')
plt.plot(x3,Wal23,label=r'$\lambda$ = 1')
plt.plot(x3,Wal24,label=r'$\lambda$ = 5')
plt.title('wald distribution (eta=5)')
plt.xlabel('x')
plt.ylabel('wald(x) density')
plt.legend()
plt.show()
```

Question #3 (estimates), Question#4 (QQplot)

```
library(fitdistrplus)
library(actuar)
datafile2 <- read.csv("~/Desktop/Stat_final/final/datafile2.csv", header=FALSE)
datafile2 <- datafile2[[1]]
fw <- fitdist(datafile2, "weibull")
fln <- fitdist(datafile2, "lnorm")
fi <- fitdist(datafile2, "invgauss", start = list(mean = 1, shape = 1))

#estimates
fln$estimate[1]
fln$estimate[2]

fw$estimate[1]
fw$estimate[2]

fi$estimate[1]
fi$estimate[2]

#QQplot
qqcomp(fln)
qqcomp(fw)
qqcomp(fi)
```

Question #3 (bootstrap, CI)

```r
#Question3

library(boot)
library(extraDistr)
library(fitdistrplus)
datafile2 <- read.csv("~/Desktop/Stat_final/final/datafile2.csv", header = FALSE)
datafile2 <- data.frame(datafile2)
set.seed(0)
#lognormal
dist11 <- function(data, i){
  d <- data[i,]
  return(fitdist(d, "lnorm", method = "mle")$estimate[1])
}

dist12 <- function(data, i){
  d <- data[i,]
  return(fitdist(d, "lnorm", method = "mle")$estimate[2])
}
#weibull
dist21 <- function(data, i){
  d <- data[i,]
  return(fitdist(d, "weibull", method = "mle")$estimate[1])
}

dist22 <- function(data, i){
  d <- data[i,]
  return(fitdist(d, "weibull", method = "mle")$estimate[2])
}
#wald
dist31 <- function(data, i){
  d <- data[i,]
  return(fitdist(d, "wald", start=list(mu=1,lambda=1))$estimate[1])
}

dist32 <- function(data, i){
  d <- data[i,]
  return(fitdist(d, "wald", start=list(mu=1,lambda=1))$estimate[2])
}

bs11 <- boot(data = datafile2, dist11, R = 3000)
bs12 <- boot(data = datafile2, dist12, R = 3000)
bs21 <- boot(data = datafile2, dist21, R = 3000)
bs22 <- boot(data = datafile2, dist22, R = 3000)
bs31 <- boot(data = datafile2, dist31, R = 3000)
bs32 <- boot(data = datafile2, dist32, R = 3000)
bs11
bs12
bs21
bs22
bs31
bs32
```

```r
# log mu
hist(bs11$t, main = "CI Bootstrap for mu", xlab="bootstrap sample",breaks = 100)
abline(v=bs11$t0,col="red",lty=1,lwd=2)
abline(v=c(2.36,2.47),col="blue",lty=2,lwd=1)
legend("topright",
       legend = c(expression(paste("confidence interval"))),col="blue",lty =2, lwd=1)
#log sig
hist(bs12$t, main = "CI Bootstrap for sigma", xlab="bootstrap sample",breaks = 100)
abline(v=bs12$t0,col="red",lty=1,lwd=2)
abline(v=c(1.086,1.158),col="blue",lty=2,lwd=1)
legend("topright",
       legend = c(expression(paste("confidence interval"))),col="blue",lty =2, lwd=1)
#wei c
hist(bs21$t, main = "CI Bootstrap for c", xlab="bootstrap sample",breaks = 100)
abline(v=bs21$t0,col="red",lty=1,lwd=2)
abline(v=c(0.8522,0.9307),col="blue",lty=2,lwd=1)
legend("topright",
       legend = c(expression(paste("confidence interval"))),col="blue",lty =2, lwd=1)
#wei b
hist(bs22$t, main = "CI Bootstrap for b", xlab="bootstrap sample",breaks = 100)
abline(v=bs22$t0,col="red",lty=1,lwd=2)
abline(v=c(18.52,20.76),col="blue",lty=2,lwd=1)
legend("topright",
       legend = c(expression(paste("confidence interval"))),col="blue",lty =2, lwd=1)
#wal eta
hist(bs31$t, main = "CI Bootstrap for eta", xlab="bootstrap sample",breaks = 100)
abline(v=bs31$t0,col="red",lty=1,lwd=2)
abline(v=c(19.36,22.39),col="blue",lty=2,lwd=1)
legend("topright",
       legend = c(expression(paste("confidence interval"))),col="blue",lty =2, lwd=1)
#wal rambda
hist(bs32$t, main = "CI Bootstrap for rambda", xlab="bootstrap sample",breaks = 100)
abline(v=bs32$t0,col="red",lty=1,lwd=2)
abline(v=c(7.461,9.087),col="blue",lty=2,lwd=1)
legend("topright",
       legend = c(expression(paste("confidence interval"))),col="blue",lty =2, lwd=1)


#CI-99%
boot.ci(boot.out = bs11, conf = 0.99, type = "basic")
boot.ci(boot.out = bs12, conf = 0.99, type = "basic")
boot.ci(boot.out = bs21, conf = 0.99, type = "basic")
boot.ci(boot.out = bs22, conf = 0.99, type = "basic")
boot.ci(boot.out = bs31, conf = 0.99, type = "basic")
boot.ci(boot.out = bs32, conf = 0.99, type = "basic")
```

Question#4 AIC,BIC

```python
import pandas as pd
import numpy as np

df = pd.read_csv('datafile2.csv',header=None)
df2 = df.values
llf_log = 0
llf_weibull = 0
llf_wald = 0


def aicc(llf, nobs, df_modelwc):
    return -2.0 * llf + 2.0 * df_modelwc * nobs / (nobs - df_modelwc - 1.0)

#https://www.statsmodels.org/stable/_modules/statsmodels/tools/eval_measures.html#aic

def bic(llf, nobs, df_modelwc):
    return -2.0 * llf + np.log(nobs) * df_modelwc

#https://www.statsmodels.org/stable/_modules/statsmodels/tools/eval_measures.html#aic

def loglikelihood_log(x, mu, sigma):
    return np.log((1 / (x * sigma * ((2 * np.pi) ** 0.5))) * np.exp(-(((np.log(x) - mu) ** 2) / (2 * (sigma ** 2

def loglikelihood_wei(x, c, b):
    return np.log((c * (x ** (c - 1))) / ((b ** c) * np.exp((x / b) ** c)))

def loglikelihood_wal(x, eta, lamda):
    return np.log((((lamda) / (2 * np.pi * (x ** 3))) ** 0.5) * np.exp((-lamda) * (((x - eta) ** 2)) / (2 * (eta


for x in df2:
    llf_log += loglikelihood_log(x, 2.42, 1.12)
    llf_weibull += loglikelihood_wei(x, 0.894, 19.631)
    llf_wald += loglikelihood_wal(x, 20.97, 8.32)

print("AIC of lognormal distribution :",aicc(llf_log,len(df2),2))
print("BIC of lognormal distribution :",bic(llf_log,len(df2),2))

print("AIC of weibull distribution   :",aicc(llf_weibull,len(df2),2))
print("BIC of weibull distribution   :",bic(llf_weibull,len(df2),2))

print("AIC of wald distribution      :",aicc(llf_wald,len(df2),2))
print("BIC of wald distribution      :",bic(llf_wald,len(df2),2))
```