ETC 2420/5242 Lab 4 2016

Di Cook

SOLUTION

Question 1

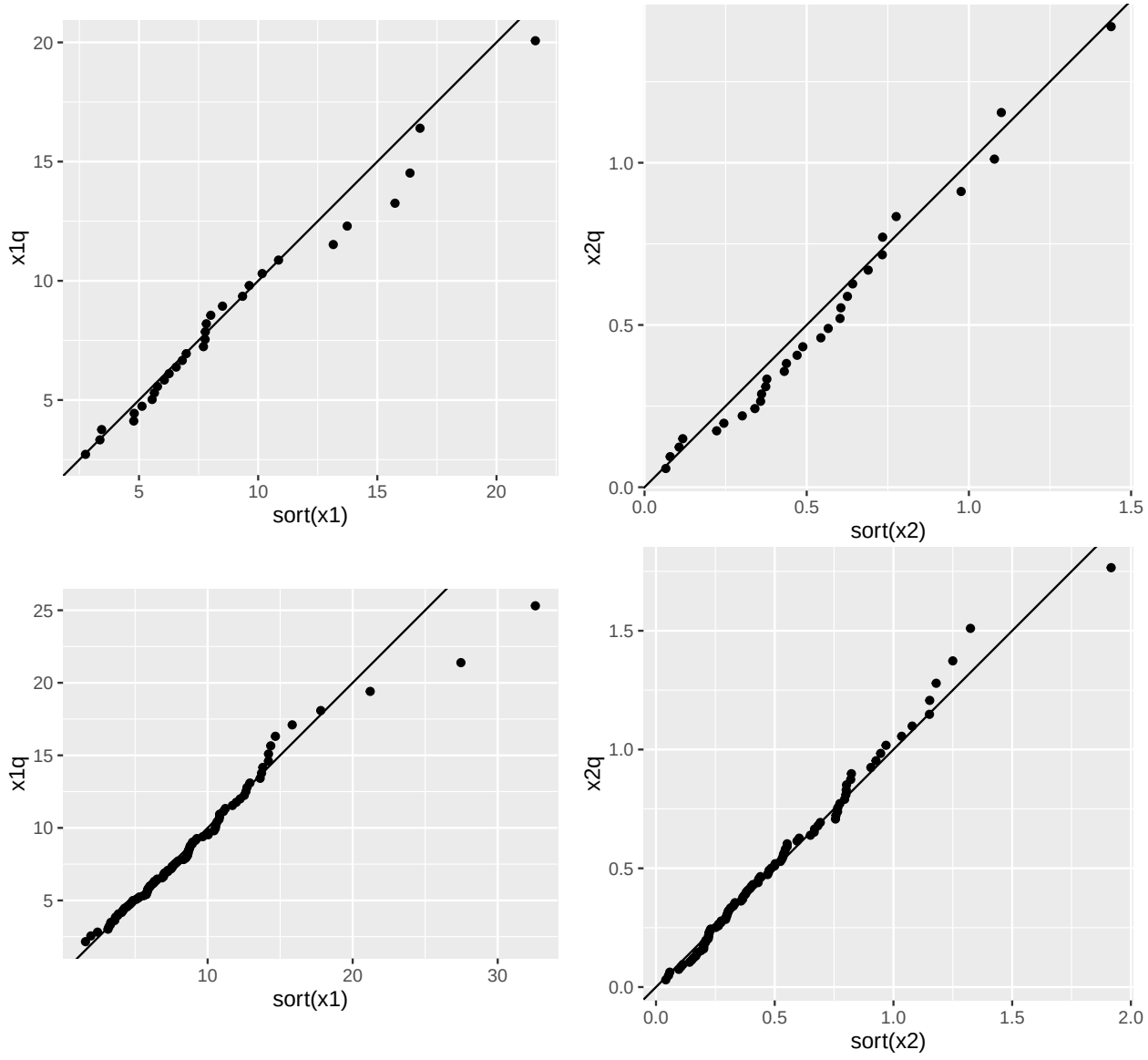
1. Simulate samples of size $n = 30, 100, 500$ from these distributions
    a. Lognormal(2, 0.5)
    b. Gamma(2, 4)

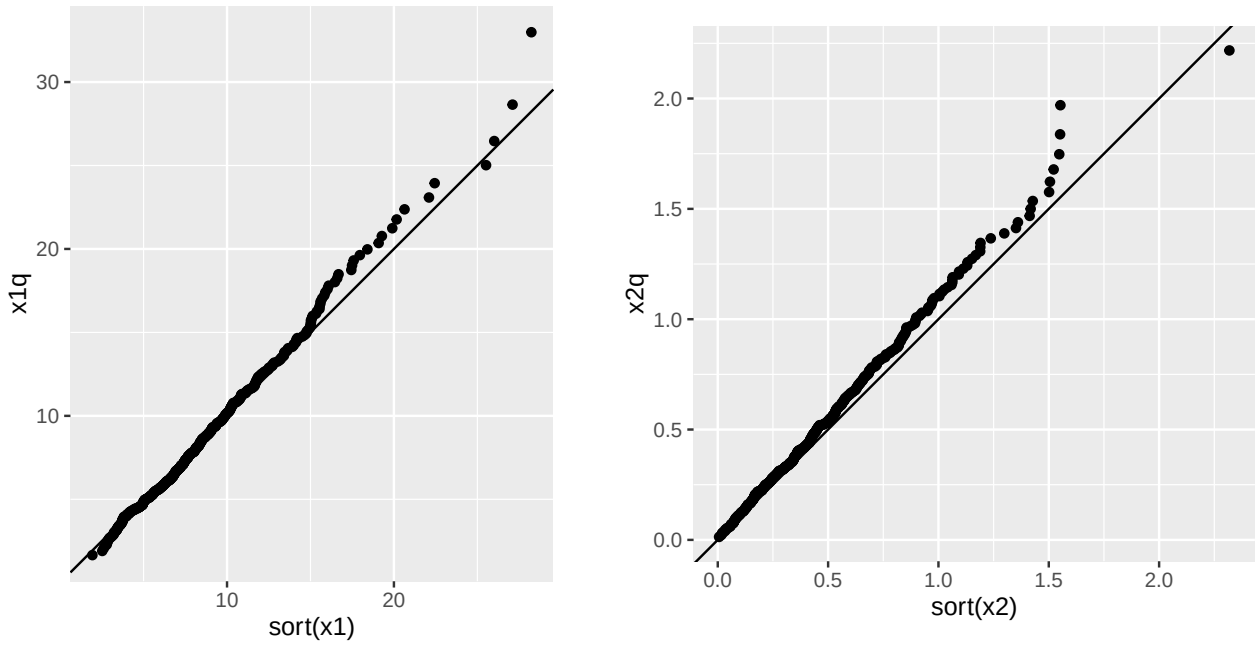
```
library(ggplot2)
library(gridExtra)
df_30 <- data.frame(x1=rlnorm(30, 2, 0.5), x2=rgamma(30, 2, 4))
ggplot(df_30, aes(x=x1)) + geom_histogram()
ggplot(df_30, aes(x=x2)) + geom_histogram()
df_100 <- data.frame(x1=rlnorm(100, 2, 0.5), x2=rgamma(100, 2, 4))
ggplot(df_100, aes(x=x1)) + geom_histogram()
ggplot(df_100, aes(x=x2)) + geom_histogram()
df_500 <- data.frame(x1=rlnorm(500, 2, 0.5), x2=rgamma(500, 2, 4))
ggplot(df_500, aes(x=x1)) + geom_histogram()
ggplot(df_500, aes(x=x2)) + geom_histogram()
```

2. Make a QQ-plot of each these samples.

```
n <- 30
df_30$x1q = qlnorm(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                       (n + 0.365),0.5^(1/n)), 2, 0.5)
df_30$x2q = qgamma(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                       (n + 0.365),0.5^(1/n)), 2, 4)
p1 <- ggplot(df_30, aes(x=sort(x1), y=x1q)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + coord_equal()
p2 <- ggplot(df_30, aes(x=sort(x2), y=x2q)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + coord_equal()
grid.arrange(p1, p2, ncol=2)
n <- 100
df_100$x1q = qlnorm(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                       (n + 0.365),0.5^(1/n)), 2, 0.5)
df_100$x2q = qgamma(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                       (n + 0.365),0.5^(1/n)), 2, 4)
p1 <- ggplot(df_100, aes(x=sort(x1), y=x1q)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + coord_equal()
p2 <- ggplot(df_100, aes(x=sort(x2), y=x2q)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + coord_equal()
grid.arrange(p1, p2, ncol=2)
n <- 500
```

```
df_500$x1q = qlnorm(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                      (n + 0.365),0.5^(1/n)), 2, 0.5)
df_500$x2q = qgamma(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                      (n + 0.365),0.5^(1/n)), 2, 4)
p1 <- ggplot(df_500, aes(x=sort(x1), y=x1q)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + coord_equal()
p2 <- ggplot(df_500, aes(x=sort(x2), y=x2q)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + coord_equal()
grid.arrange(p1, p2, ncol=2)
```





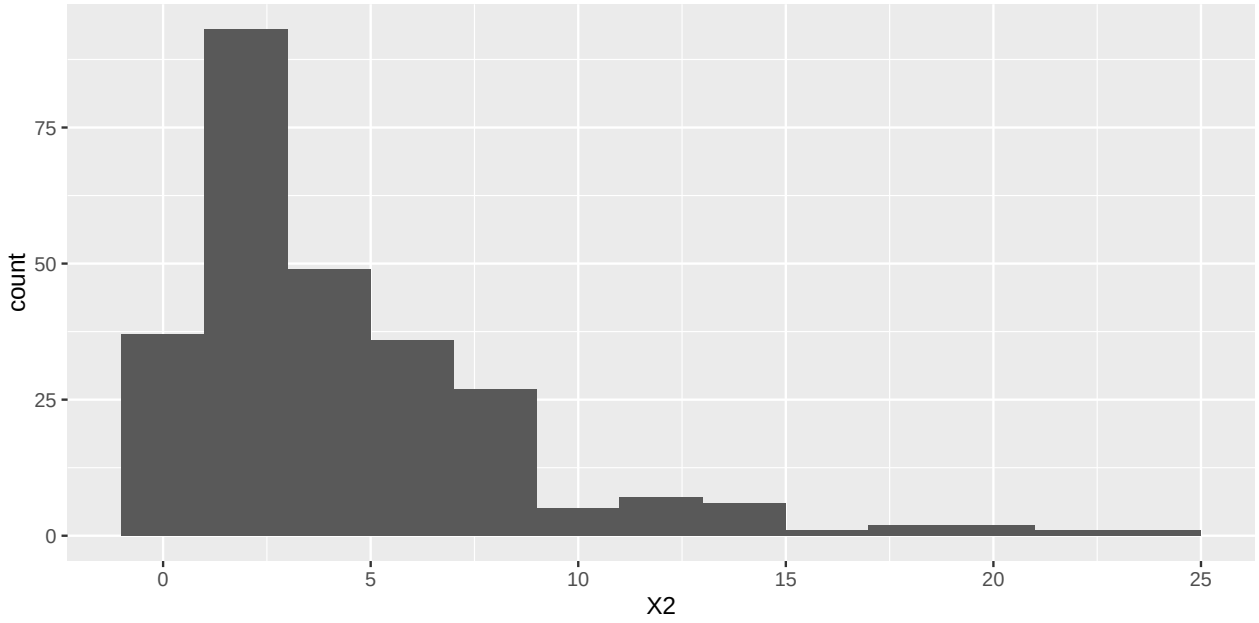
Question 2

Using this code, generate a sample of size $n = 267$ from a $Gamma(1.2, 0.25)$ distribution.

```
set.seed(123)
X2 <- rgamma(n=267, 1.2, 0.25)
```

a. Plot the sample, using a histogram, describe the shape of the distribution.

```
ggplot(data.frame(X2), aes(x=X2)) + geom_histogram(binwidth=2)
```



b. What parameters of the gamma distribution were used to simulate the sample? (α, β) `1.2, 0.25`

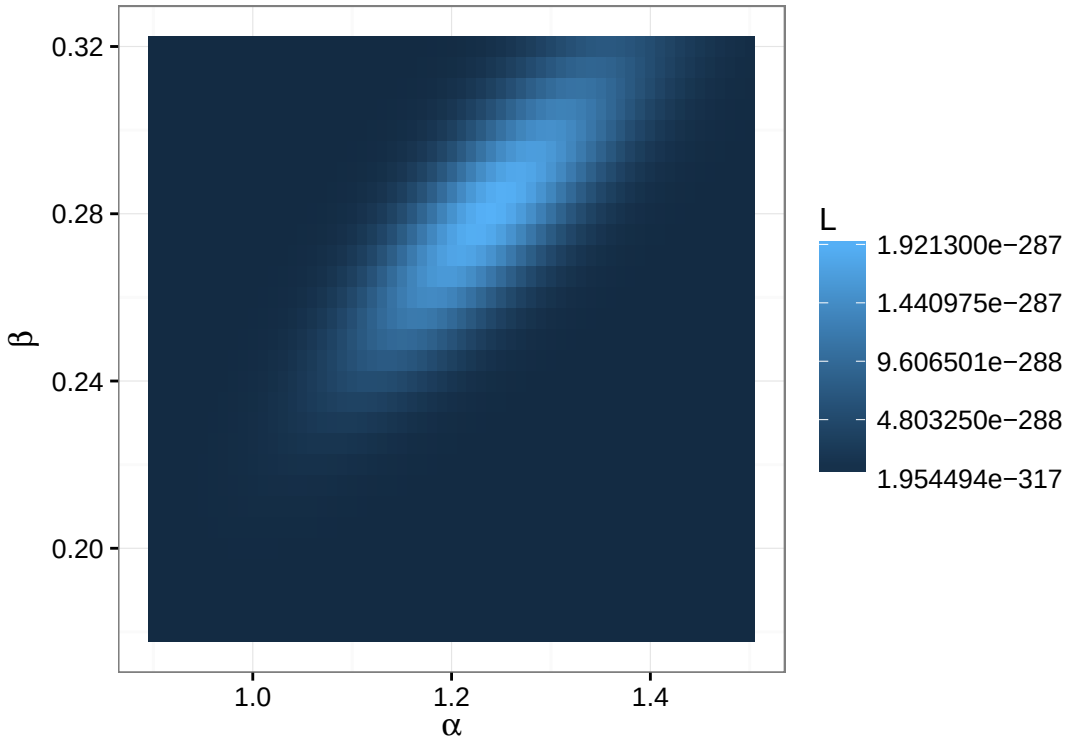
c. If we are to use maximum likelihood distribution what values would we expect to get as the parameter estimates? `Close to 1.2, 0.25`

d. Write a function to compute the likelihood function.

```
nmle <- function(x, a, b) {
  f <- prod(dgamma(x, a, b))
  return(f)
}
```

e. Plot the likelihood function for a range of values of α, β that shows the maximum likelihood estimates for each parameter.

```
a <- seq(0.9, 1.5, 0.01)
b <- seq(0.18, 0.32, 0.005)
g <- expand.grid(x=a, y=b)
g$f <- 0
for (i in 1:nrow(g)) {
  g$f[i] <- nmle(X2, g$x[i], g$y[i])
}
ggplot(g, aes(x=x, y=y, fill=f)) + geom_tile() + xlab(expression(alpha)) + ylab(expression(beta)) + ther
  scale_fill_continuous("L") +
  theme(aspect.ratio=1)
```



f. Look up the function `fitdistr` from the `MASS` library. Explain what this does. Use it to find the MLE estimates for α, β. How do these compare with the values you read off your plot?

```
library(MASS)
fitdistr(X2, "gamma")
#      shape          rate
#   1.23994762   0.28051981
#  (0.09624333) (0.02667898)
```

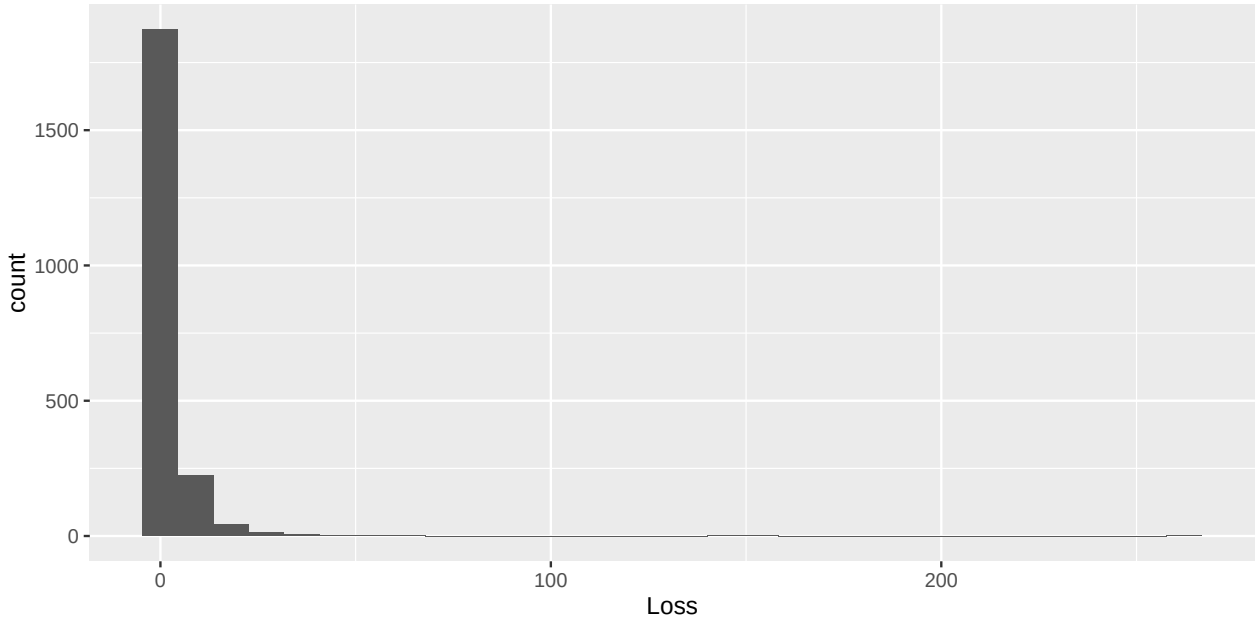
Question 3

Take a look at the data set `danishuni` from the `CASdatasets` library.

```
library(CASdatasets)
data(danishuni)
```

a. Make a histogram of the `Loss`. Describe the shape.

```
ggplot(danishuni, aes(x=Loss)) + geom_histogram()
```

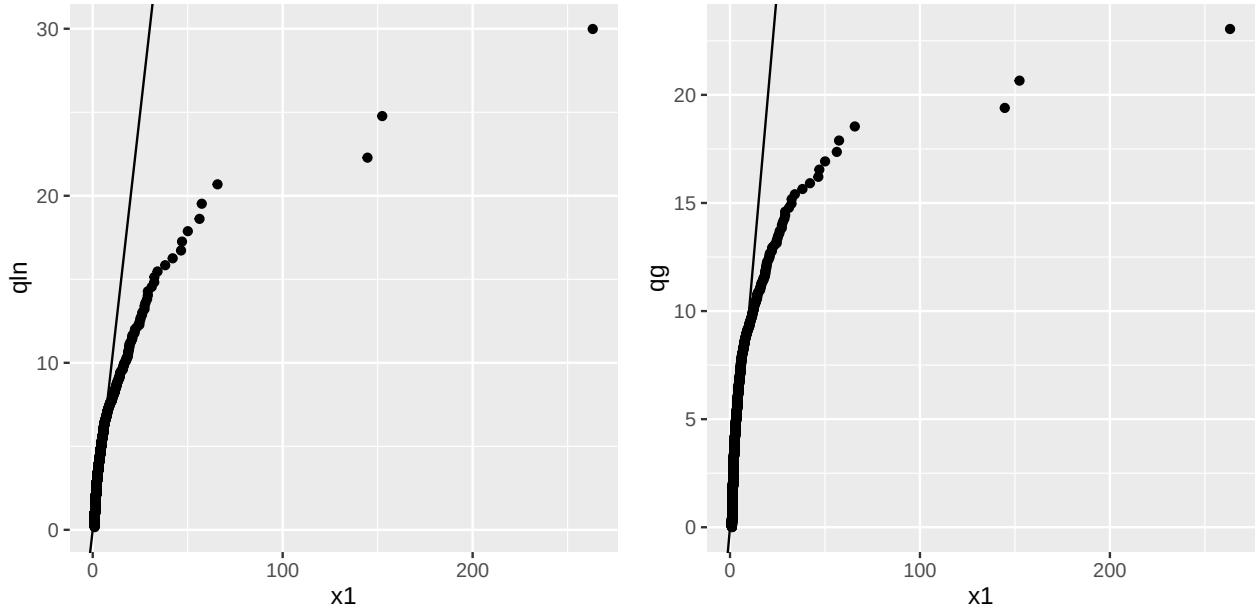


b. Fit both a gamma and lognormal distribution to the sample, i.e. find the MLEs.

```
fitdistr(danishuni$Loss, "gamma")
#      shape          rate
#   1.29760890   0.38333093
#  (0.03548495) (0.01273354)
fitdistr(danishuni$Loss, "lognormal")
#     meanlog       sdlog
#   0.78695008   0.71655451
#  (0.01539288) (0.01088441)
```

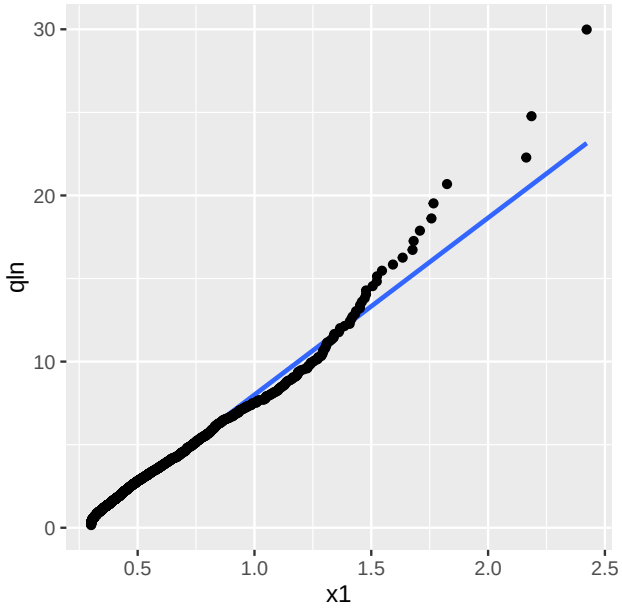
c. Produce a QQ-plot for each of the distributions.

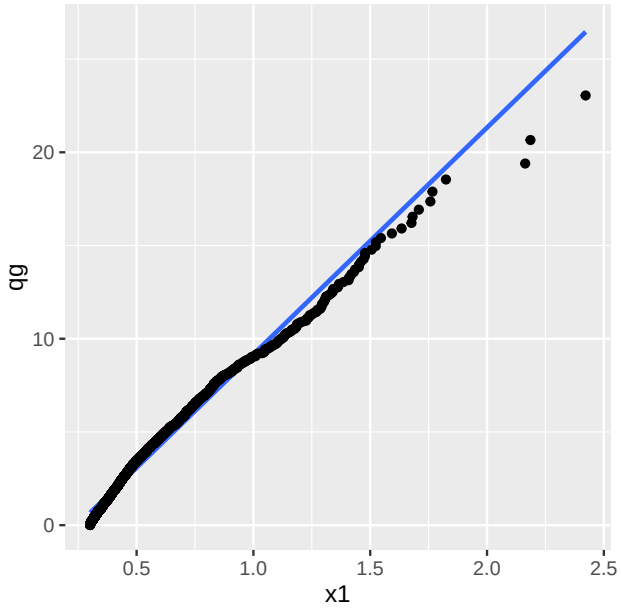
```
n <- nrow(danishuni)
df <- data.frame(x1=sort(danishuni$Loss))
df$qln = qlnorm(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                        (n + 0.365),0.5^(1/n)), 0.78695, 0.7655451)
df$qg = qgamma(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                        (n + 0.365),0.5^(1/n)), 1.2976, 0.38333)
p1 <- ggplot(df, aes(x=x1, y=qln)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + theme(aspect.ratio=1)
p2 <- ggplot(df, aes(x=x1, y=qg)) +
  geom_abline(intercept=0, slope=1) +
  geom_point() + theme(aspect.ratio=1)
grid.arrange(p1, p2, ncol=2)
```



d. Which is the better fit to the sample? `Neither!! Maybe we should try a pareto?`
e. Re-do steps a-d after log-transforming the `Loss`.

```
n <- nrow(danishuni)
df <- data.frame(x1=log10(sort(danishuni$Loss+1)))
df$qln = qlnorm(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                        (n + 0.365),0.5^(1/n)), 0.78695, 0.7655451)
df$qg = qgamma(c(1 - 0.5^(1/n), (2:(n-1) - 0.3175) /
                        (n + 0.365),0.5^(1/n)), 1.2976, 0.38333)
p1 <- ggplot(df, aes(x=x1, y=qln)) +
  geom_smooth(method="lm") +
  geom_point() + theme(aspect.ratio=1)
p2 <- ggplot(df, aes(x=x1, y=qg)) +
  geom_smooth(method="lm") +
  geom_point() + theme(aspect.ratio=1)
grid.arrange(p1, p2, ncol=2)
```





TURN IN

- Your `.Rmd` file
- Your Word (or pdf) file that results from knitting the Rmd.
- Make sure your group members are listed as authors, one person per group will turn in the report
- DUE: Wednesday after the lab, by 7am, loaded into moodle

Resources

- PSU lecture notes on MLE
- CASdatasets