

```
load("lattice.RData")
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.1.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
head(food.web)
```

```
##   mean.length dimension
## 1         3.13    Mixed
## 2         2.71    Mixed
## 3         2.30      Two
## 4         2.74    Mixed
## 5         2.00    Mixed
## 6         3.82    Mixed
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

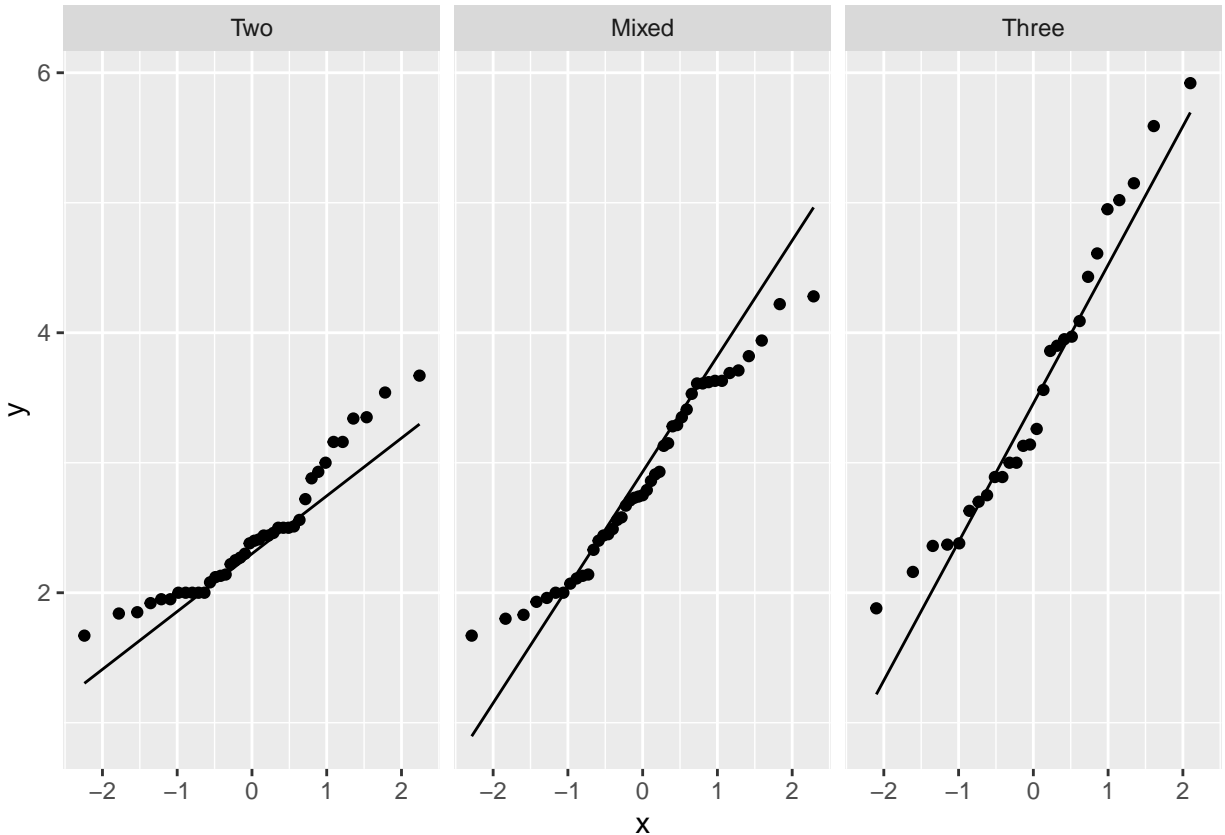
```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unable to identify current timezone 'H':
## please set environment variable 'TZ'
```

Q.1)

Answer:

1. We first plot the mean.length attribute on Q-normal plot faceted by Dimension

```
ggplot(food.web,aes(sample=mean.length)) + stat_qq() + facet_wrap(~ dimension) + stat_qq_line()
```



2. For the dimension Two, we can see that the quantiles for the mean.length do not lie in a straight line and the data is too peaked in the middle. 3. This shows that data for dimension Two is not described by a normal distribution 4. For the dimension Mixed and dimension Three, we can see that the quantiles do not lie in a straight line for either of the two. 5. So we can say that the data does not follow the Normal Distribution for any of the given dimensions. 6. To normalize the data, we can use transformations like Log and Power Transformations.

Q.2)

Answer:

1. We first created a function for Box-Cox Transformation. Using this function we can use both the Log Transformation(in case Tau is 0) and Power Transformation(in case Tau is non-zero value)

```
bc_trans = function(x, tau) {
  if(tau == 0){
    return(log(x))
  }
  return((x^tau - 1) / tau)
}
```

2. Then we apply the function on our data to get Power transformed values of mean.length attribute

```
tau_vec = seq(-1,-1)
transforms = sapply(tau_vec, function(tau) bc_trans(food.web$mean.length, tau))
transforms_melted = melt(transforms)
transforms_for_plotting = transforms_melted %>% mutate(Original_value=food.web$mean.length[Var1],tau =
```

```
head(transforms_melted)
```

```
##   Var1 Var2   value
## 1    1    1 0.6805112
## 2    2    1 0.6309963
## 3    3    1 0.5652174
## 4    4    1 0.6350365
## 5    5    1 0.5000000
## 6    6    1 0.7382199
```

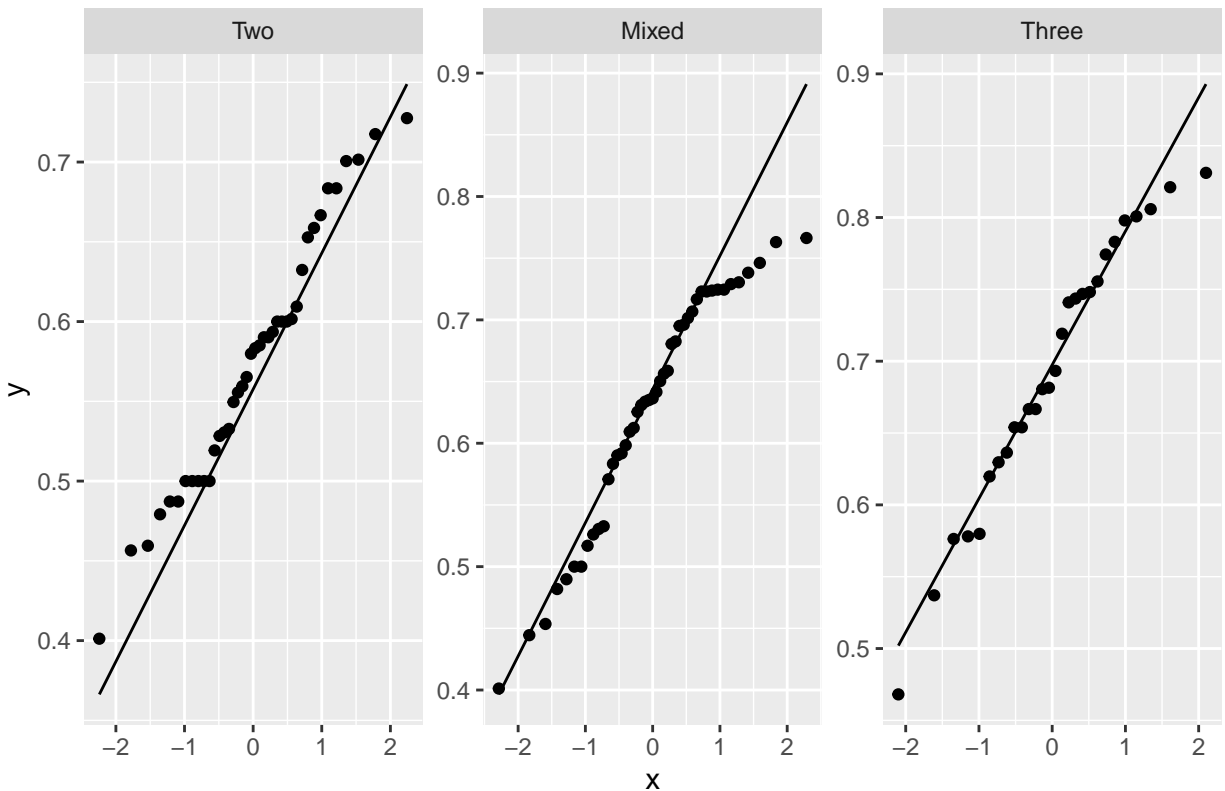
```
head(transforms_for_plotting)
```

```
##   Var1 Var2   value Original_value tau dimension
## 1    1    1 0.6805112          3.13  -1      Mixed
## 2    2    1 0.6309963          2.71  -1      Mixed
## 3    3    1 0.5652174          2.30  -1        Two
## 4    4    1 0.6350365          2.74  -1      Mixed
## 5    5    1 0.5000000          2.00  -1      Mixed
## 6    6    1 0.7382199          3.82  -1      Mixed
```

3. We can see our original value and the value we get after applying Power Transformation
4. Then we plot our data on the Q-normal plots according to each dimension

```
ggplot(transforms_for_plotting) + stat_qq(aes(sample = value)) +facet_wrap(~ dimension, scales = "free")
ggtitle("Q-normal plots for Inverse mean.length (Power Transformation) for each dimension")
```

Q-normal plots for Inverse mean.length (Power Transformation) for each di



5. From the above plots, we see that for dimension Two there is significant difference in the distributions as compared to before applying transformation. However the points still do not lie in a straight line and hence we can say that it does not follow normal distribution. 6. For the dimension Mixed and Three, we see though the transformation has made a difference in distribution but the data for both seems to be right-skewed.

```
tau_vec2 = seq(0,0)
transforms2 = sapply(tau_vec2, function(tau) bc_trans(food.web$mean.length, tau))
transforms_melted2 = melt(transforms2)
transforms_for_plotting2 = transforms_melted2 %>% mutate(Original_value=food.web$mean.length[Var1],tau = tau)
```

7. Here we can see the original value with the value obtained after applying Log Transformation to the mean.length attribute.

```
head(transforms_melted2)
```

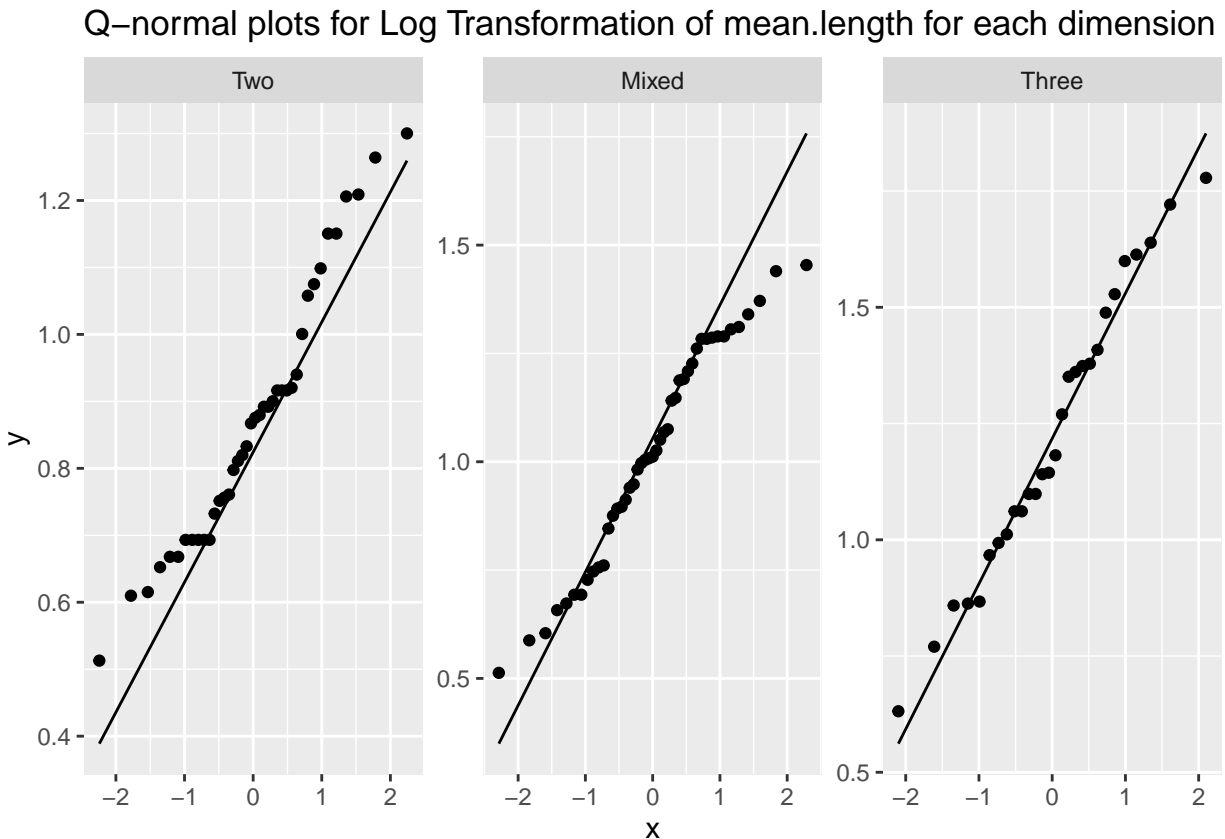
```
##   Var1 Var2    value
## 1    1    1 1.1410330
## 2    2    1 0.9969486
## 3    3    1 0.8329091
## 4    4    1 1.0079579
## 5    5    1 0.6931472
## 6    6    1 1.3402504
```

```
head(transforms_for_plotting2)
```

```
##   Var1 Var2   value Original_value tau dimension
## 1    1    1 1.1410330          3.13    0   Mixed
## 2    2    1 0.9969486          2.71    0   Mixed
## 3    3    1 0.8329091          2.30    0    Two
## 4    4    1 1.0079579          2.74    0   Mixed
## 5    5    1 0.6931472          2.00    0   Mixed
## 6    6    1 1.3402504          3.82    0   Mixed
```

8. Then we plot our data on the Q-normal plots according to each dimension

```
ggplot(transforms_for_plotting2) +
  stat_qq(aes(sample = value)) +
  facet_wrap(~ dimension, scales = "free", ncol = 3) + stat_qq_line(aes(sample = value))+
  ggtitle("Q-normal plots for Log Transformation of mean.length for each dimension")
```



9. From the above plots, we can see for dimension Two that though Log transformation has changed the data distribution it is still less normally distributed in comparison with Power Transformation and does not follow normal distribution 10. The dimension Mixed seems to have right-skewed distribution but it is still not normally distributed. 11. The dimension Three is better and seems that points almost lie in a straight line. 12. So we can conclude that applying these transformations has surely made data more normalized.

Q3)

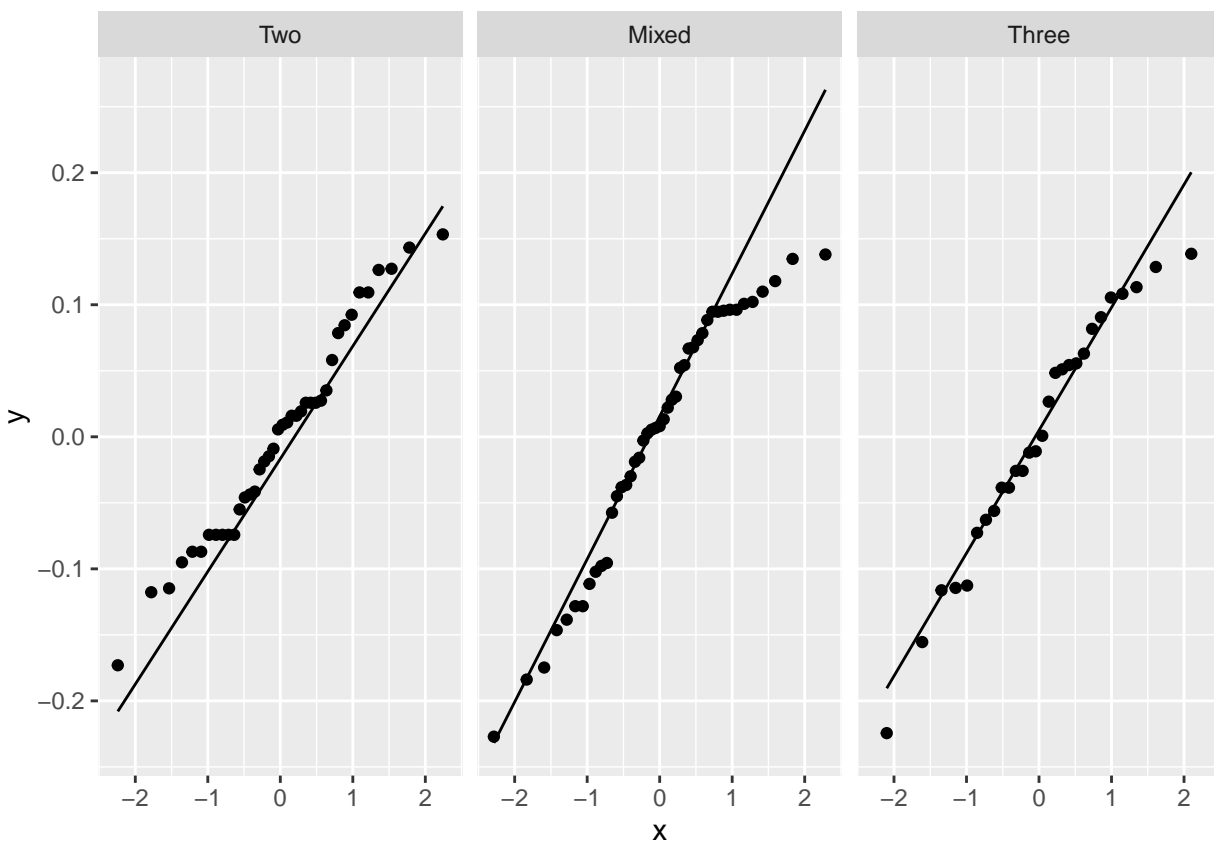
```
food_lm = lm(value ~ dimension, data=transforms_for_plotting)
```

```
food_res = data.frame(dimension = food.web$dimension, residual = residuals(food_lm))
```

```
head(food_res)
```

```
##   dimension   residual
## 1    Mixed  0.052219317
## 2    Mixed  0.002704439
## 3     Two -0.009007435
## 4    Mixed  0.006744621
## 5    Mixed -0.128291877
## 6    Mixed  0.109928014
```

```
ggplot(food_res, aes(sample = residual)) +  
  stat_qq() + facet_wrap(~ dimension, ncol=3) + stat_qq_line()
```



```
ggplot(food_res, aes(sample = residual)) +  
  stat_qq() + geom_abline(intercept = 0, slope = sd(food_res$residual))
```

