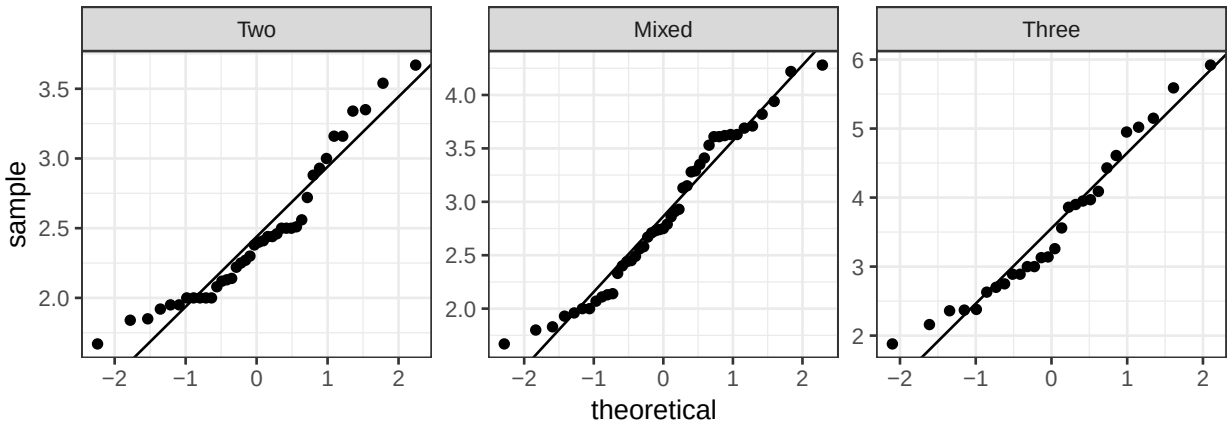
EDA Homework 2 Solutions

Question 1

Let's make the Q-normal plot:

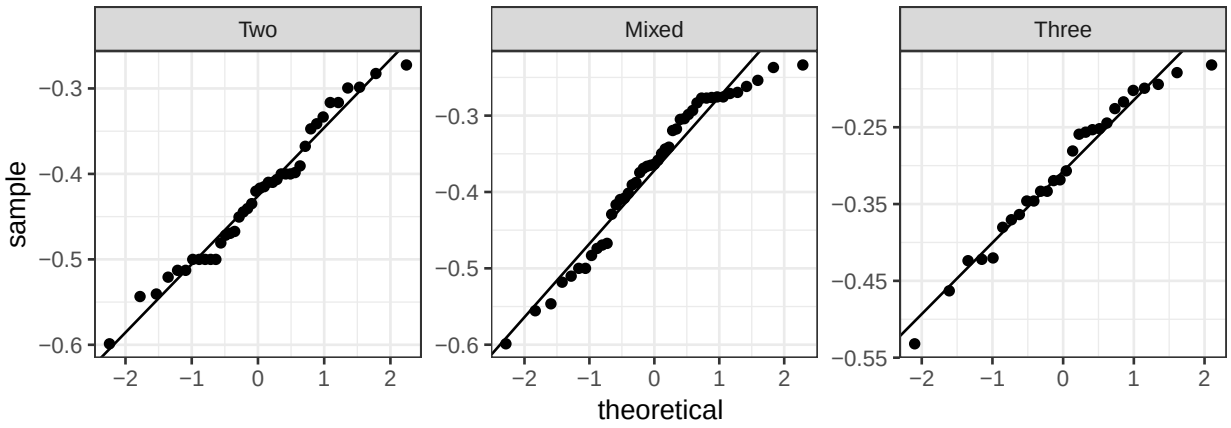
```
library(tidyverse)

## - Attaching packages ---------------------- tidyverse 1.2.1 -

##  tibble  1.4.2     purrr  0.2.5
##  tidyr  0.8.1     dplyr  0.7.6
##  readr  1.1.1     stringr 1.3.1
##  tibble  1.4.2     forcats 0.3.0

## - Conflicts ------------------------ tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

load("lattice.RData")
## This data frame allows us to plot lines with the correct slope
## and intercept on each QQ plot. It's just for demonstration
## though, not required by the question.
mean_and_sd = food.web %>%
    group_by(dimension) %>%
    summarise(mean = mean(mean.length), sd = sd(mean.length))
ggplot(food.web) +
    stat_qq(aes(sample = mean.length)) +
    geom_abline(aes(intercept = mean, slope = sd), data = mean_and_sd) +
    facet_wrap(~ dimension, scales = "free")
```



Question 2

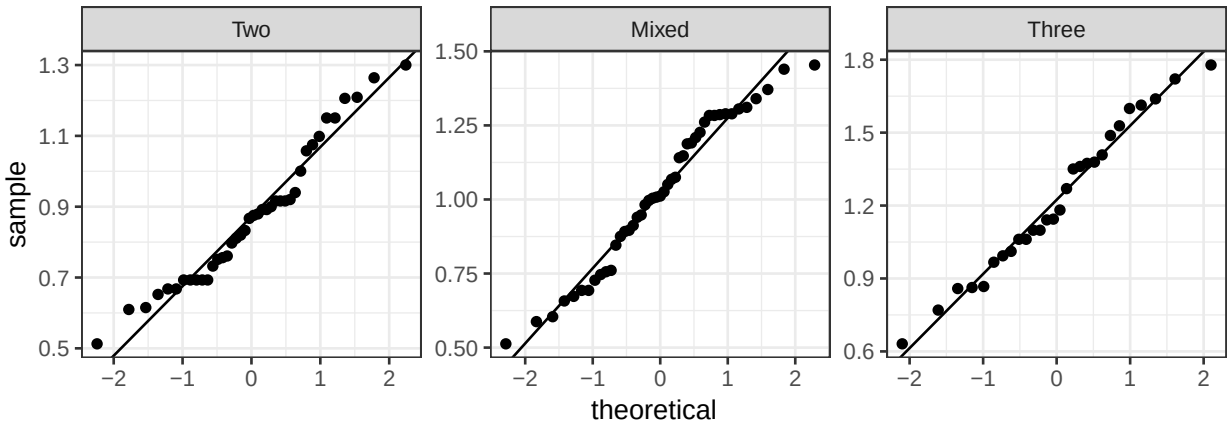
Here we have a plot of the inverse-transformed values:

```
mean_and_sd_inv = food.web %>%
    group_by(dimension) %>%
    summarise(mean = mean(-mean.length^(-1)), sd = sd(-mean.length^(-1)))
ggplot(food.web) +
    stat_qq(aes(sample = -mean.length^(-1))) +
    geom_abline(aes(intercept = mean, slope = sd), data = mean_and_sd_inv) +
    facet_wrap(~ dimension, scales = "free")
```



And a Q-normal plot of the log-transformed values.

```
mean_and_sd_log = food.web %>%
    group_by(dimension) %>%
    summarise(mean = mean(log(mean.length)), sd = sd(log(mean.length)))
ggplot(food.web) +
    stat_qq(aes(sample = log(mean.length))) +
    geom_abline(aes(intercept = mean, slope = sd), data = mean_and_sd_log) +
    facet_wrap(~ dimension, scales = "free")
```



In both cases I have added the lines that the points should fall along if they were normally distributed.

We see that both the log and the inverse transformation both make the distributions closer to normal, but in neither case is it a perfect match.

Question 3

Here we make a linear model to describe inverse mean length as a function of dimension:

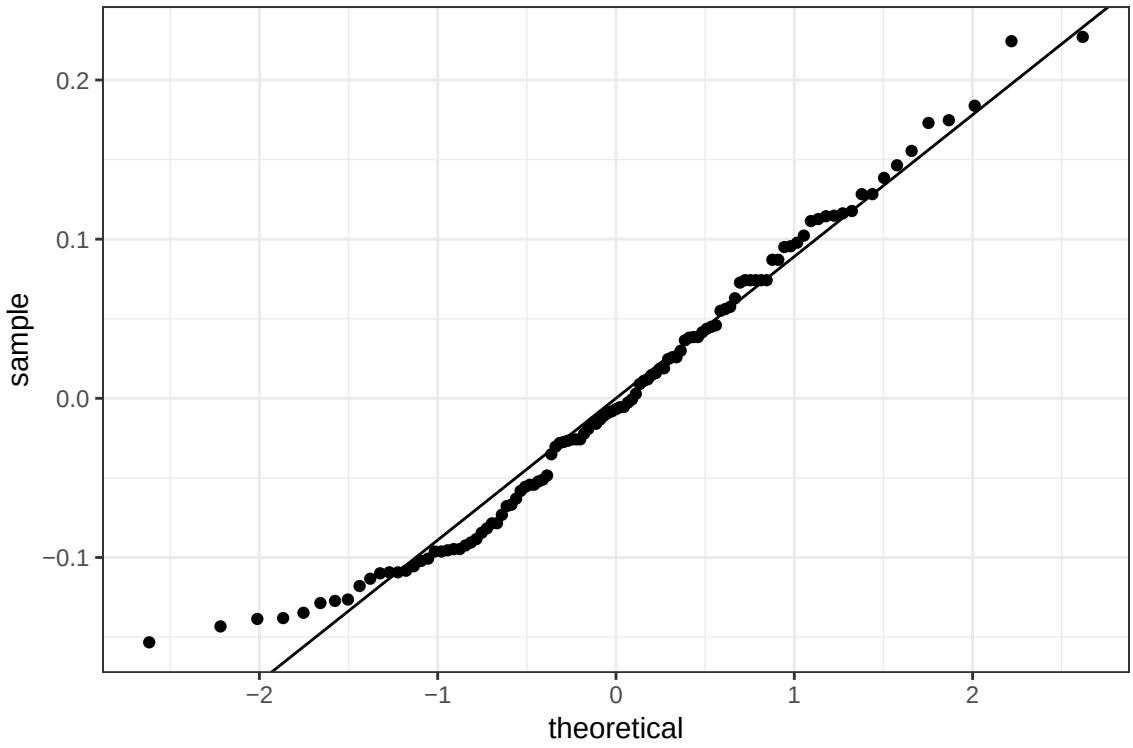
```
library(broom)
lminv = lm(mean.length^(-1) ~ 0 + dimension, data = food.web)
tidy(lminv)
```

```
## # A tibble: 3 x 5
##   term           estimate std.error statistic  p.value
##   <chr>             <dbl>     <dbl>     <dbl>    <dbl>
## 1 dimensionTwo      0.426    0.0142      30.0 9.93e-55
## 2 dimensionMixed    0.372    0.0134      27.7 1.76e-51
## 3 dimensionThree    0.308    0.0170      18.1 8.88e-35
```

We see that the inverse mean length decreases with dimension: two-dimensional webs have the highest expected inverse mean length, at .426, three-dimensional webs have the smallest, at .308, and mixed-dimension webs are intemediate at .372.

We can plot the pooled residuals for this model, and we see that they look reasonably normal, as they fall roughly along a straight line:

```
ggplot(augment(lminv)) +
    stat_qq(aes(sample = .resid)) +
    geom_abline(aes(intercept = 0, slope = sd(.resid)))
```


0.2
0.1
0.0
−0.1
sample
−2
−1
0
1
2
theoretical