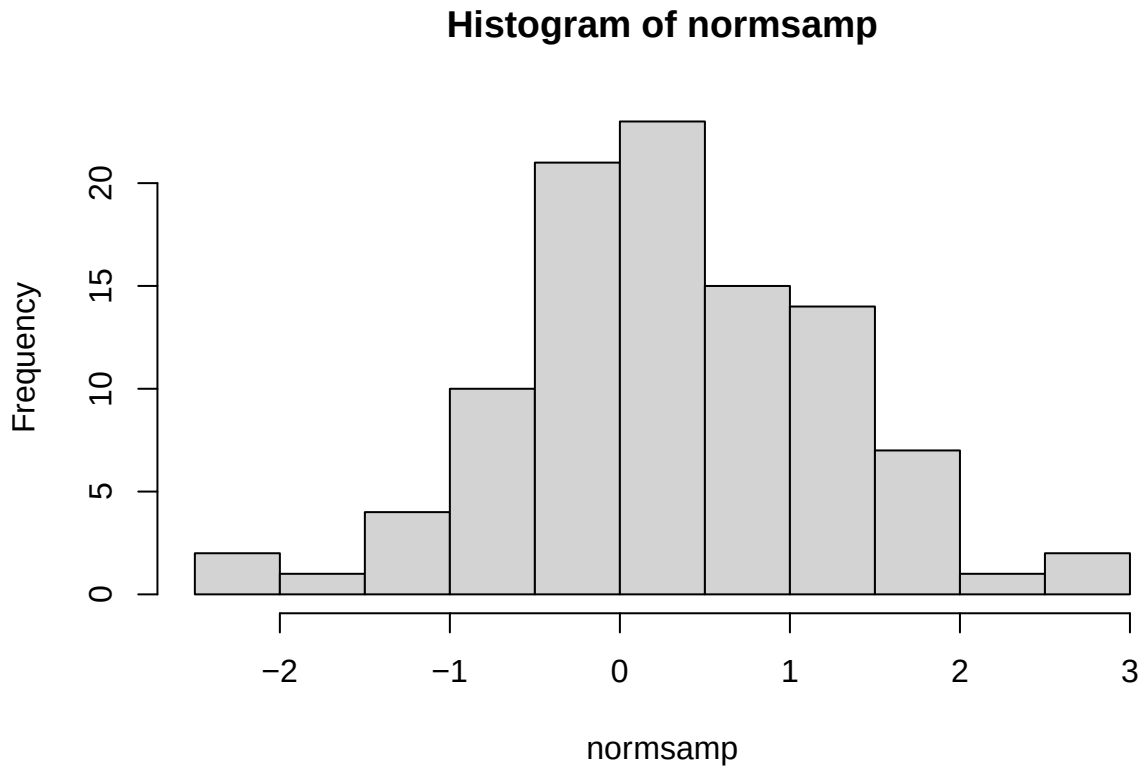
Simulating P-P and Q-Q Plots

2025-09-06

1.

a.

```
set.seed(1230)
x <- rnorm(100,mean= 0, sd=1)
normsamp <- rnorm(100)
hist(normsamp)
```



b.

```
set.seed(1073)
x <- rnorm(100,mean=0,sd=1)
normsamp <- rnorm(100)
bins <- seq(min(normsamp),max(normsamp),by= 0.1)
count_bins <- table(cut(normsamp, breaks= bins, right = FALSE))
```

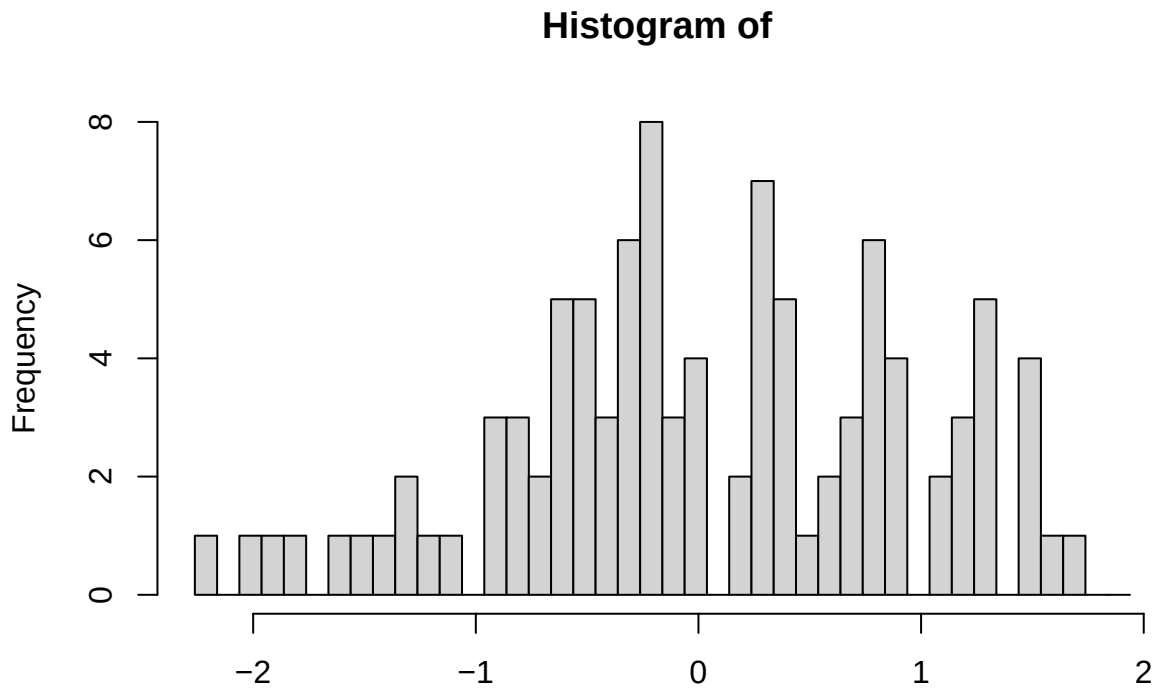
```
count <- count_bins

length(bins)
```

```
## [1] 43
```

```
length(count)
```

```
## [1] 42
```

```
histo <- list(breaks= bins, counts = count)
class(histo) <- "histogram"

plot(histo)
```



2.

a. n = 30 days X = 31 mean = 30 sd = 1.5 Z = (X - mean)/sd = (31-30)/1.5 = 0.67 standard deviations above mean P(X > 31) = 1 - P(X <= 31 or Z) Get CDF of P(X <= 31) to find probability the value is less than or equal to 31, then subtract to get the probability of getting a value that is greater than 31.

P(X > 31) = 1 - P(0.67)

Mean = n*p Variance = np(1-p) Mean = 7.542867 Variance = 5.646372

The distribution is Binomial because it meets the requirements: 1. Independent samples 2. Fixed n 3. 0 or 1 (either under or equal to 30 min OR over 30 min) 4. Same probability of success and failure for each sample

Y ~ Binomial(mean = 30, probability = 0.2514289)

```
pnorm(0.67)
```

```
## [1] 0.7485711
```

```
1 - pnorm(0.67)
```

```
## [1] 0.2514289
```

```
walk_mean <- 30*(1 - pnorm(0.67))
walk_var <- (30*(1 - pnorm(0.67)))*(1-(1 - pnorm(0.67)))
walk_mean
```

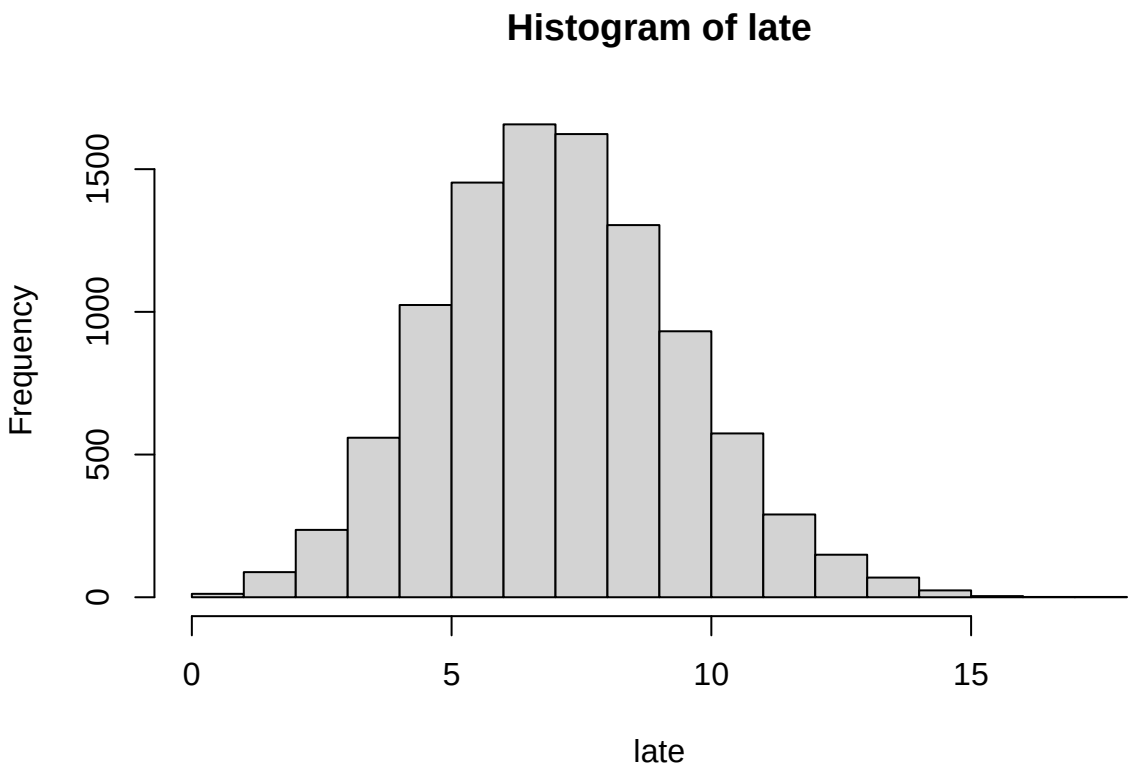
```
## [1] 7.542867
```

```
walk_var
```

```
## [1] 5.646372
```

b.

```
set.seed(3701)
simnum <- 10000
days <- 30
countwalk <- vector()
mean <- 30
sd <- 1.5

late <- replicate(10000, {
  walktime <- rnorm(30,mean,sd)
  sum(walktime> 31)
})

hist(late)
```




```
mean(late)
```

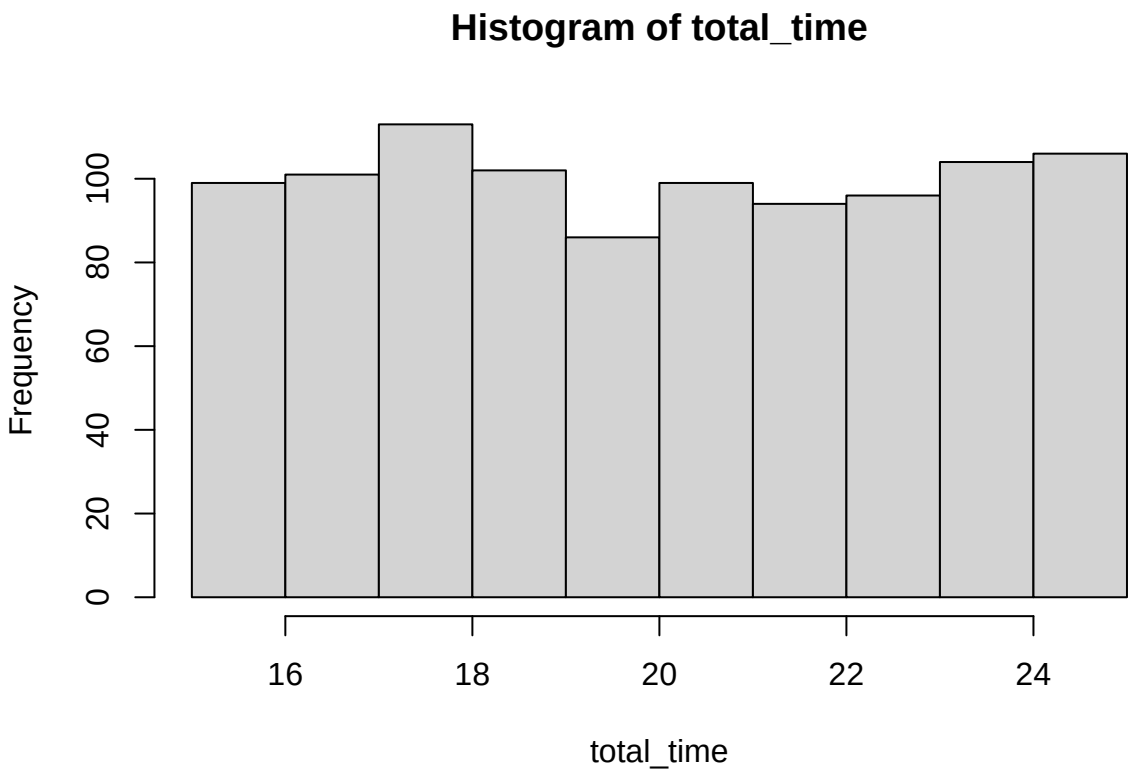
```
## [1] 7.5764
```

```
var(late)
```

```
## [1] 5.617125
```

Yes they are close to question a.

```
set.seed(3701)
simnum <- 1000
train_wait_times <- runif(simnum, min = 0, max = 10)
trainwalktime <- 15
total_time <- trainwalktime + train_wait_times

hist(total_time)
```




```
mean(total_time)
```

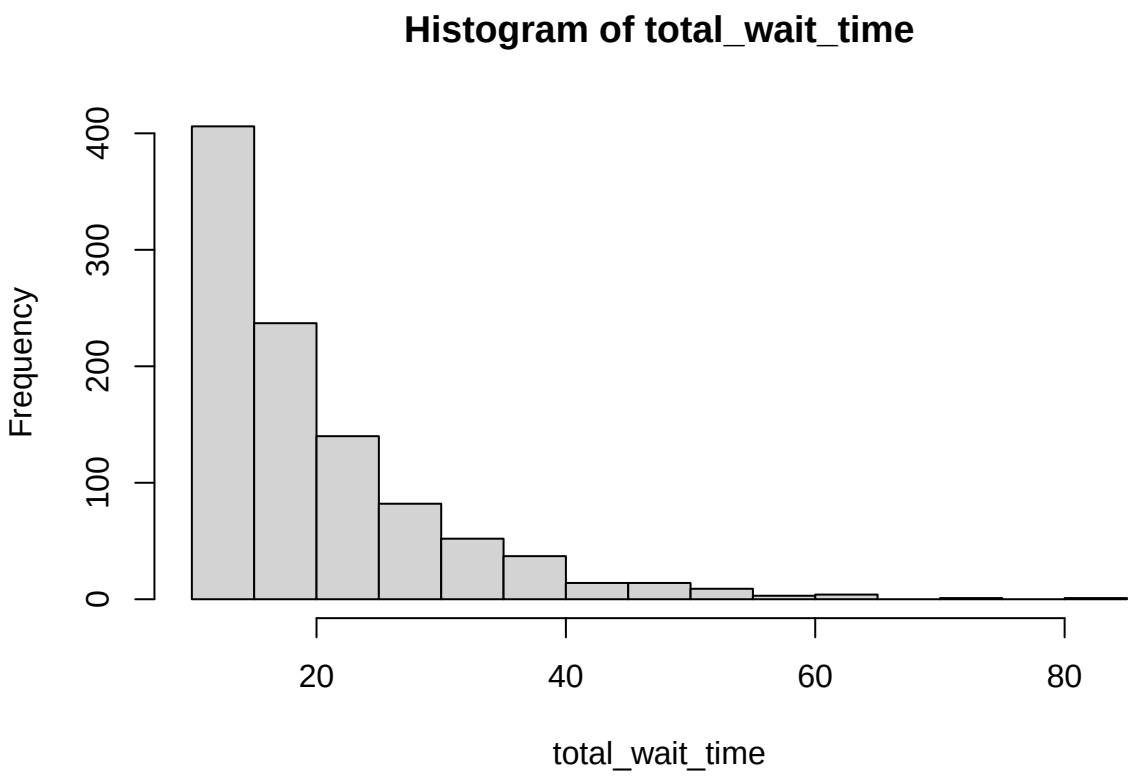
```
## [1] 19.99089
```

```
var(total_time)
```

```
## [1] 8.555915
```

d.

```
set.seed(1111)
simnum <- 1000
x<-runif(simnum)
bus_wait_time <- -10*log(x)
total_wait_time <- bus_wait_time + 10

hist(total_wait_time)
```




```r
mean(total_wait_time)
```

```
## [1] 19.71079
```

```r
var(total_wait_time)
```

```
## [1] 98.171
```

e. Probability of each method is more than 20 min(aka late)

```r
#walking
walkprob <- pnorm(20,30,1.5, lower.tail = FALSE)
walkprob
```

```
## [1] 1
```

```r
#Train
trainprob <- punif(20, min=15,max=25, lower.tail = FALSE)
trainprob
```

```
## [1] 0.5
```

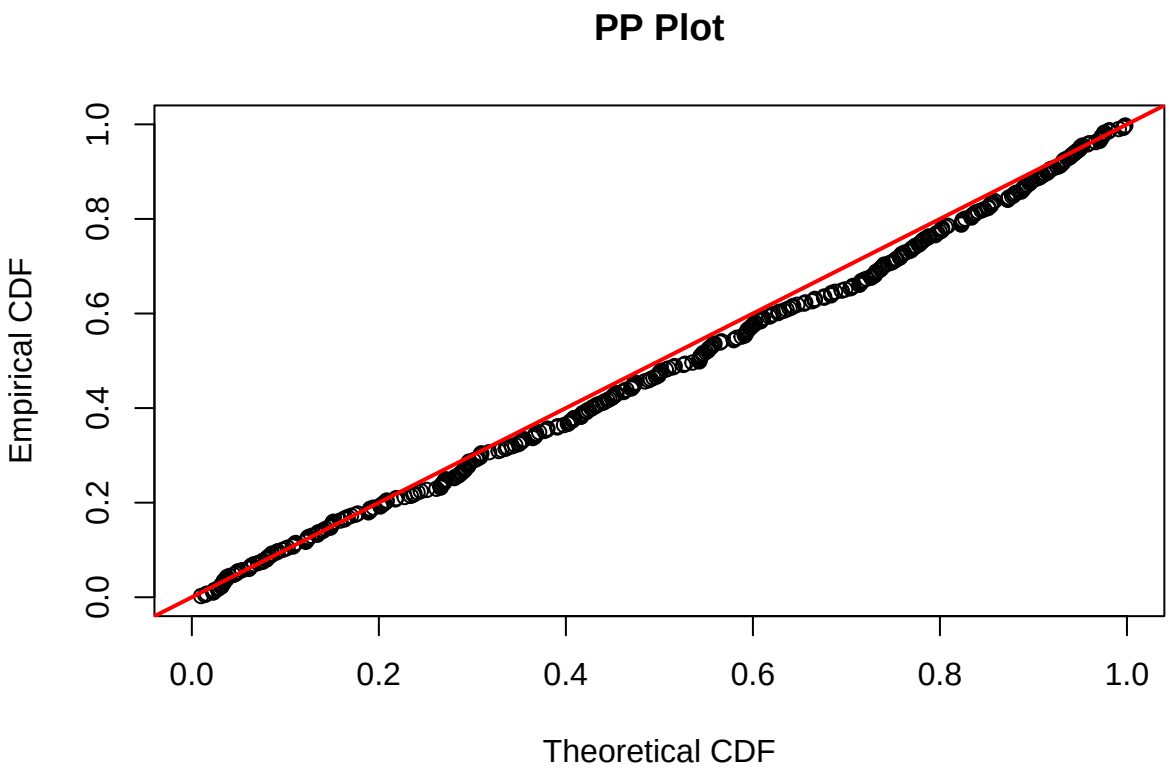
```r
#Bus
busprob <- pexp(20, rate = (1/10), lower.tail = FALSE)
busprob
```

```
## [1] 0.1353353
```

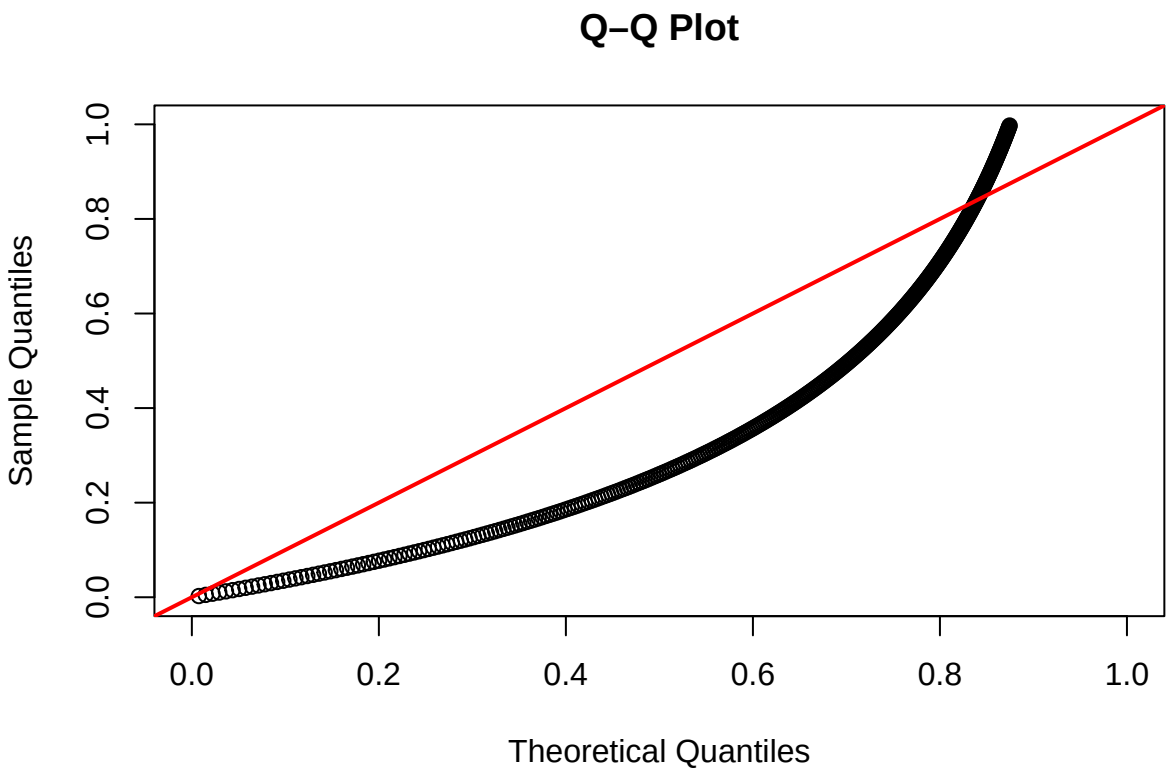
Taking the bus is the best option, since the probability of it taking over 20 minutes is the lowest of all the transportation methods.

3.

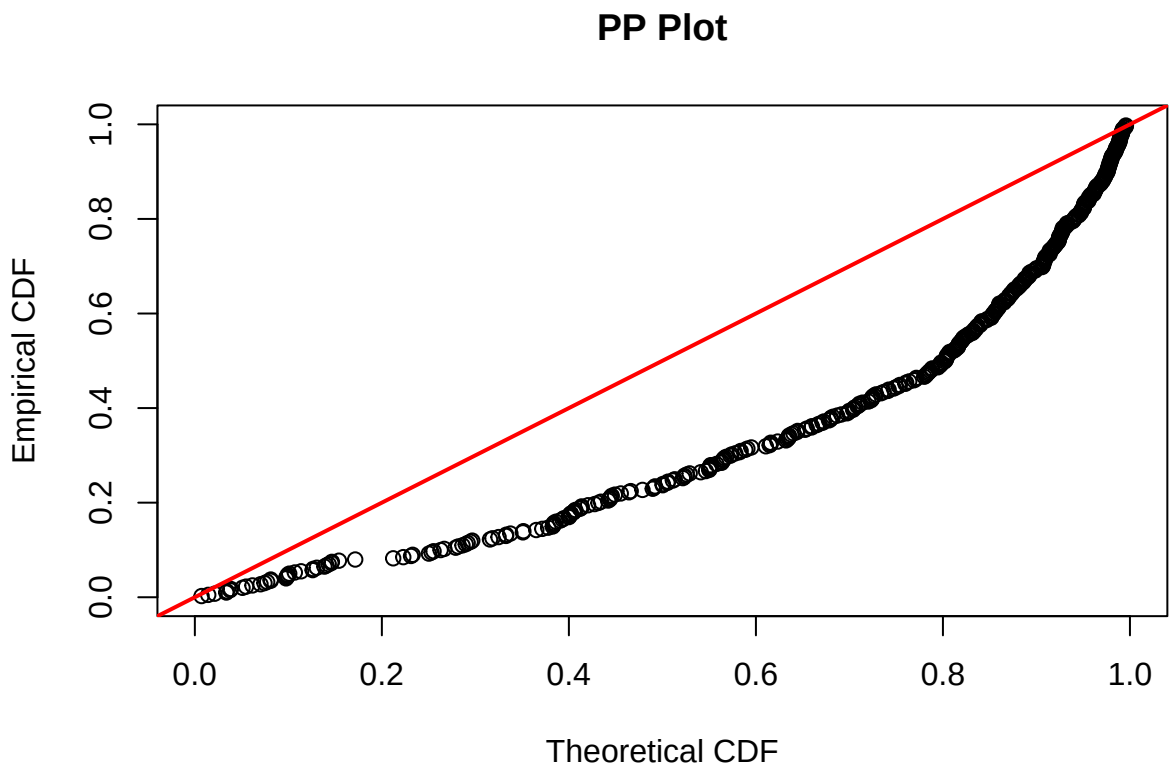
```r
simnum <- 400
theta <- 3
mypareto <- function(simnum,theta){
  dist <- runif(simnum, 0, 1)
  x <- (dist^(-1/theta)-1)
  return(x)
}
cdf <- function(x, theta){
  return(1-(x+1)^(-theta))
}


#PP Plot Pareto
pp_plot <- function(data, p, ..., main = "PP Plot") {
empprob <- (1:simnum) / (simnum + 1)
theoryprob <- p(sort(data),...)

plot(theoryprob, empprob, main = main,
xlab = "Theoretical CDF",
ylab = "Empirical CDF",
xlim = c(0,1), ylim = c(0,1))
abline(0, 1, col = "red", lwd = 2)

}

data <- mypareto(simnum, theta)

pp_plot(data, cdf, theta=theta)
```



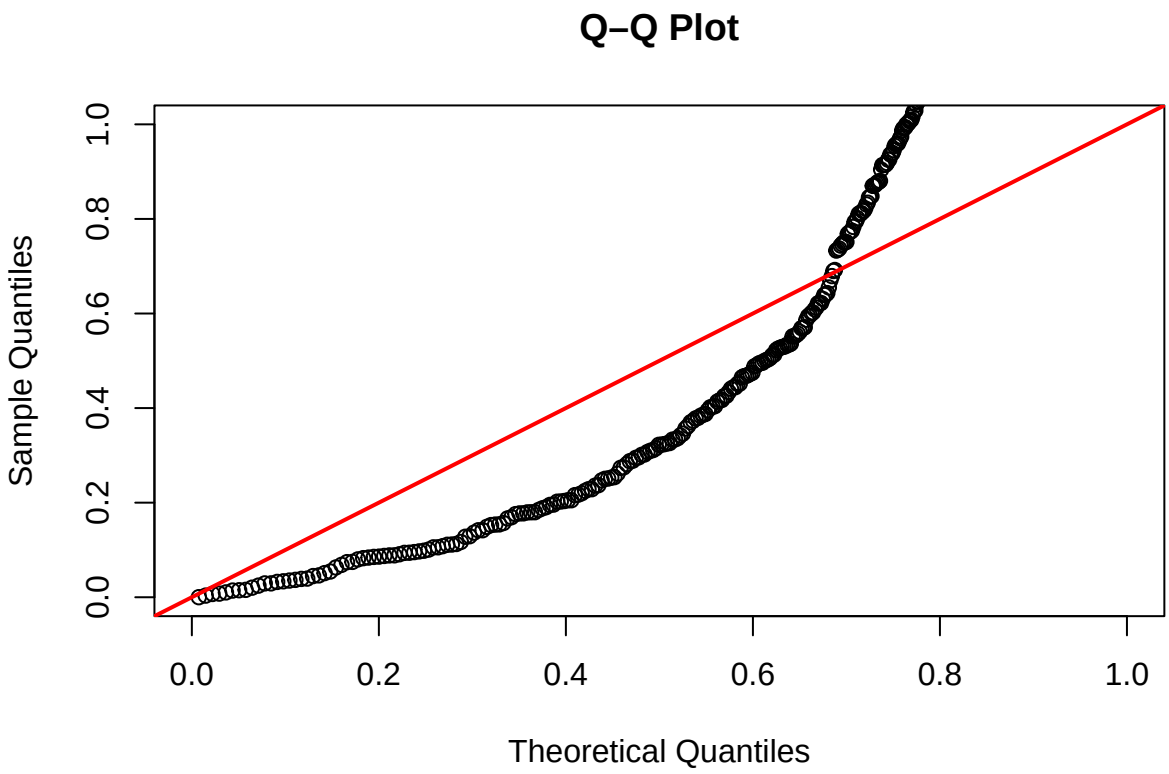

```r
#qq plot pareto
simnum <- 400
theta <- 3
qq_plot <- function(data, q,..., main="Q-Q Plot") {
simnum <- length(data)
empprob <- (1:simnum) / (simnum + 1)
theo_q <- q(empprob,...)

plot(theo_q, empprob, main = main,
xlab = "Theoretical Quantiles",
ylab = "Sample Quantiles",
xlim = c(0,1), ylim = c(0,1))
abline(0, 1, col = "red", lwd = 2)
}


data <- mypareto(simnum, theta)

qq_plot(data, cdf, theta=theta)
```




```r
#pp plot exp
simnum = 400
theta = 3
x_exp <- rexp(simnum, 1)
pp_plot_exp <- function(x_exp, p,..., main="P-P Plot EXP") {
empprob <- (1:simnum) / (simnum + 1)
theoryprob <- p(x_exp,...)
plot(theo_probs, empprob, main = main,
xlab = "Theoretical CDF", ylab = "Empirical CDF")
abline(0, 1, col = "red")
}

pp_plot(x_exp, cdf, theta=theta)
```




```r
#qq plot exp
simnum = 400
theta = 3
x_exp <- rexp(simnum, 1)
qq_plot <- function(x_exp, q,..., main="Q-Q Plot") {
simnum <- length(data)
empprob <- (1:simnum) / (simnum + 1)
theo_q <- q(empprob,...)

plot(theo_q, sort(x_exp), main = main,
xlab = "Theoretical Quantiles",
ylab = "Sample Quantiles",
xlim = c(0,1), ylim = c(0,1))
abline(0, 1, col = "red", lwd = 2)
}

qq_plot(x_exp, cdf, theta=theta)
```



It looks like the PP/QQ plots vs the Pareto distribution fit better. Overall, I think that the PP plots are more intuitive since they compare the empirical and theoretical cdfs directly.