Exploring QQ-Plots with Confidence Intervals

STAT 061 Final Project

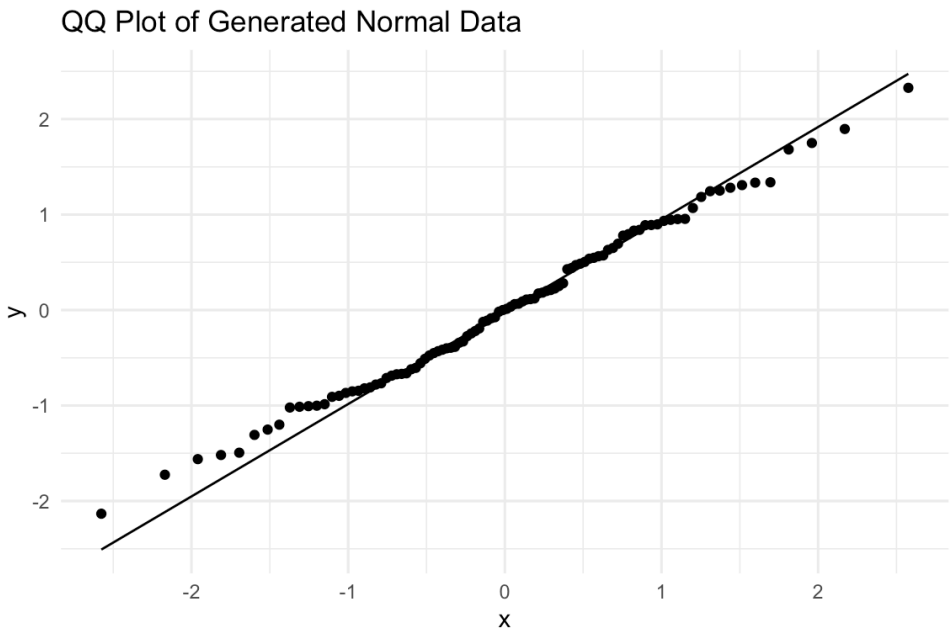
AUTHOR
Anna Jing and Alicia Liu

Introduction

Today we will be exploring making confidence intervals for QQ plots. QQ plots, or quantile-quantile plots, are scatter plots that plot two sets of quantiles against each other. One set of quantiles comes from an assumed distribution, and the other comes from the sample provided. If the sample data fits the assumed distribution it is plotted against, there should be a straight line.

In the code below, we'll generate normal data and then create a QQ plot.

```
# Create a dataset that contains 100 N(0,1) draws
set.seed(52)
simulated_data = tibble(
  x = rnorm(100)
)

#Plot the quantiles of the simulated_data against the quar
qq1 <- ggplot(simulated_data, aes(sample = x)) +
  geom_qq() + # Add qq points
  geom_qq_line() + # Add qq line
  theme_minimal() + ggtitle("QQ Plot of Generated Normal D
qq1
```

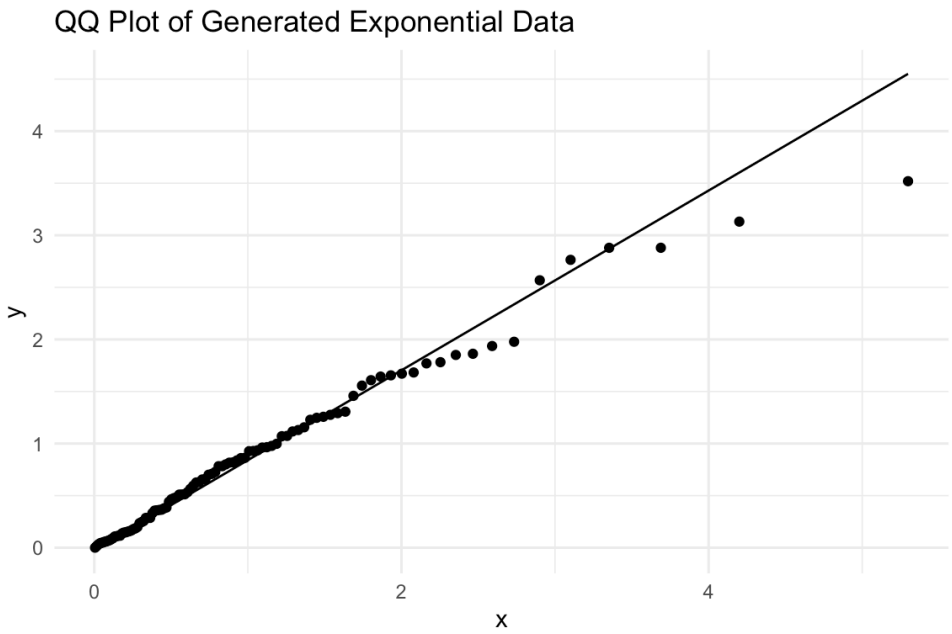


So, the QQ plot indicates that the data follows a normal distribution well.

Let's see what this will look with non-normal data. We will use exponential data in this next examples.

The code below generates some random exponential distribution data and creates a QQ plot.

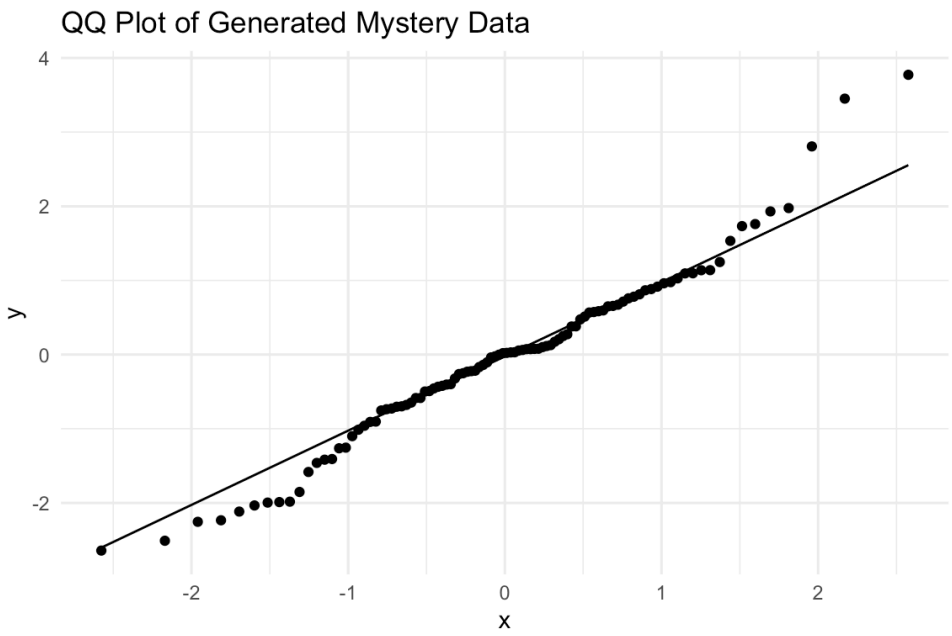
```
# Create a dataset that contains 100 Expo draws
set.seed(52)
simulated_data2 = tibble(
  x = rexp(100, rate = 1)
)

qq2 <- ggplot(simulated_data2, aes(sample = x)) +
  geom_qq(distribution = qexp) + # Add qq points, specify
  geom_qq_line(distribution = qexp) + # Add qq line, spec
  theme_minimal() + ggtitle("QQ Plot of Generated Exponen
qq2
```



Seems to be a bit inconclusive…

I've generated some MYSTERY data and plotted its QQ plot, assuming a normal distribution.

```
#Upload mystery data
simulated_data3 <- read_csv("mystery_data.csv")

qqm <- ggplot(simulated_data3, aes(sample = x)) +
  geom_qq(distribution = qnorm) +
  geom_qq_line(distribution = qnorm) +
  theme_minimal() + ggtitle("QQ Plot of Generated Mystery
qqm
```



Is this mystery data normal? Maybe…

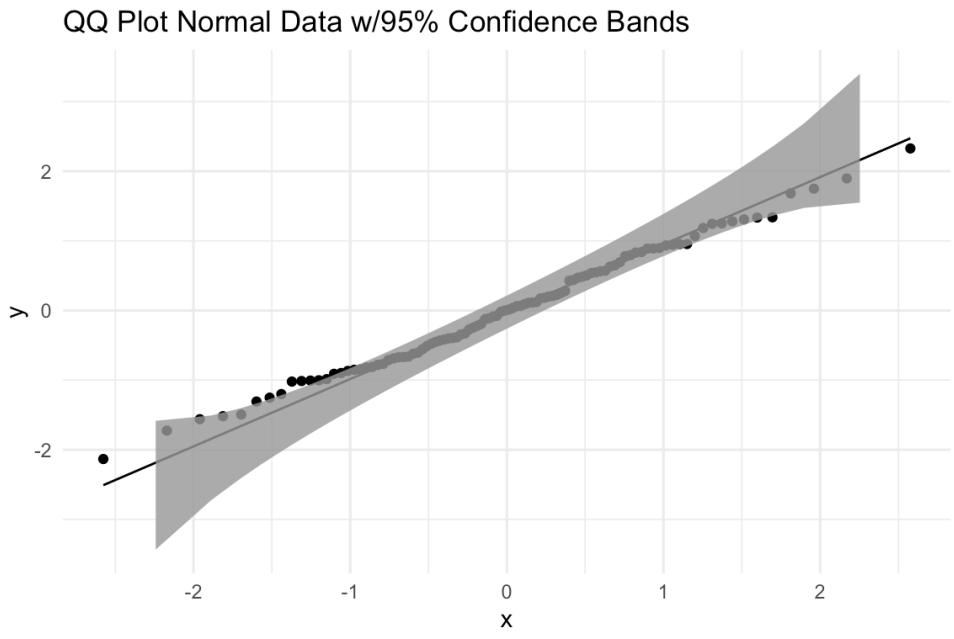
For all of these plots, how can we tell if the distribution fits a QQ plot "well enough"? Is there a way to somehow quantify our certainty?

Enter confidence intervals for QQ plots. Rather than just "eyeballing it," a confidence interval for a QQ plot provides us bands of confidence to help us quantify how "well enough" the data fits the distribution.

QQ plots with confidence intervals, along with histograms and distribution tests, are useful for figuring out if your data fits some underlying distribution. We'll compare the methods later.

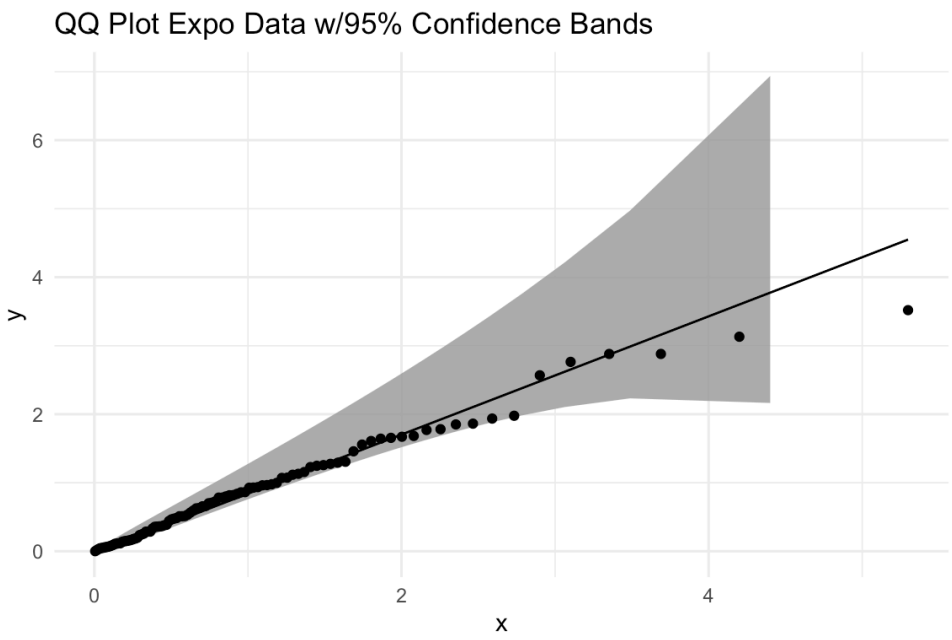
But let's first see the QQ plots with confidence bands to see if our previous data fit "well enough."

```
#Normal QQ plot
qq3 <- ggplot(simulated_data, aes(sample = x)) +
  geom_qq() + # Add qq points
  geom_qq_line() + # Add qq line
  stat_qq_band() + #Add confidence bands
  theme_minimal() + ggtitle("QQ Plot Normal Data w/95% Co
qq3
```

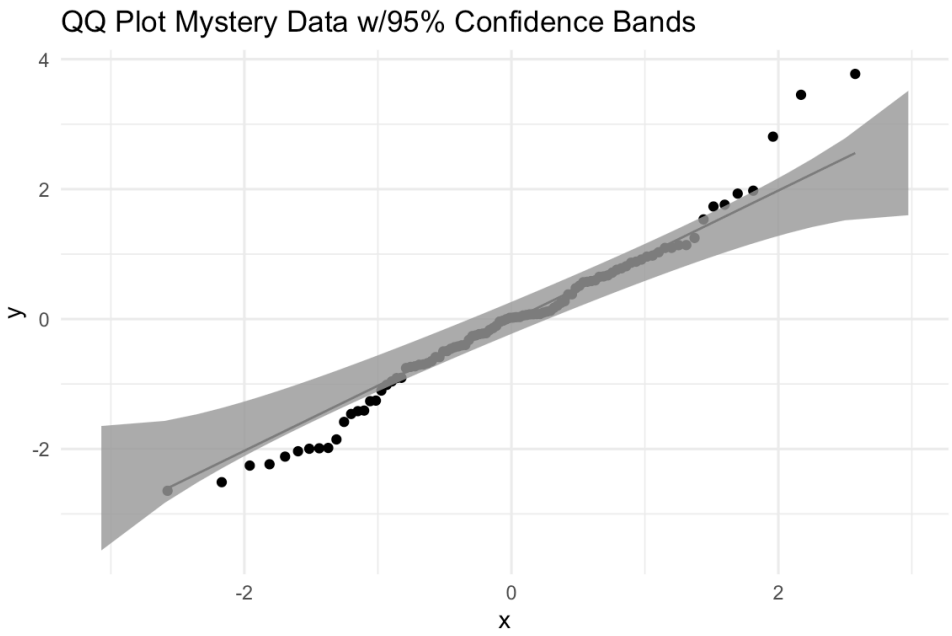



```
#Expo QQ plot
qq4 <- ggplot(simulated_data2, aes(sample = x)) +
  stat_qq_band(distribution = "exp") +
  geom_qq(distribution = qexp) + # Add qq points, specify
  geom_qq_line(distribution = qexp) + # Add qq line, spec
```

```
  theme_minimal() + ggtitle("QQ Plot Expo Data w/95% Confi
qq4
```




```
#Mystery QQ plot
qq5 <- ggplot(simulated_data3, aes(sample = x)) +
  geom_qq(distribution = qnorm) +
  geom_qq_line(distribution = qnorm) +
  stat_qq_band() + #Add confidence bands
  theme_minimal() + ggtitle("QQ Plot Mystery Data w/95% Co
qq5
```



Much better! So for the two with known distributions, the confidence bands indicated the proper distribution, with around 95% of the points

within the bands.

For the mystery data, this data deviated significantly from the normal distribution in the tails.

> **Note**
>
> Question 1: Any guesses for a distribution that looks a lot like a normal data but differs from it in the tails?

Hint: Dark Irish stout.

T-riffic! (Spelling intended: another hint!) Now, we've discovered a helpful tool to help us determine how well sample data fits an assumed distribution. Let's see how to calculate these bands.

How Are QQ plot Intervals Calculated?

To find intervals for QQ plots, there are three steps. First, we need to create an "empirical CDF" of the sample data. Then, we need to compute the distance between this empirical CDF to the CDF of the assumed distribution, and then lastly, we need to determine, using the assumed underlying distribution, how far the confidence bands can extend.

For the random, i.i.d sample $X_1, X_2, \ldots, X_n \sim F$, an empirical CDF is a a step-wise function with step size $1/n$ defined as below.

$$\hat{F_n}(t) = \frac{\text{number of } X_i \text{ in sample} \leq t}{n} = \frac{1}{n} \sum_{i=1}^{t} \mathbb{1}_{X_i \leq t}$$

where $\mathbb{1}_{X_i \leq t}$ is an indicator function that equals 1 if $X_i \leq t$.

This equation is an unbiased estimator of F_n.

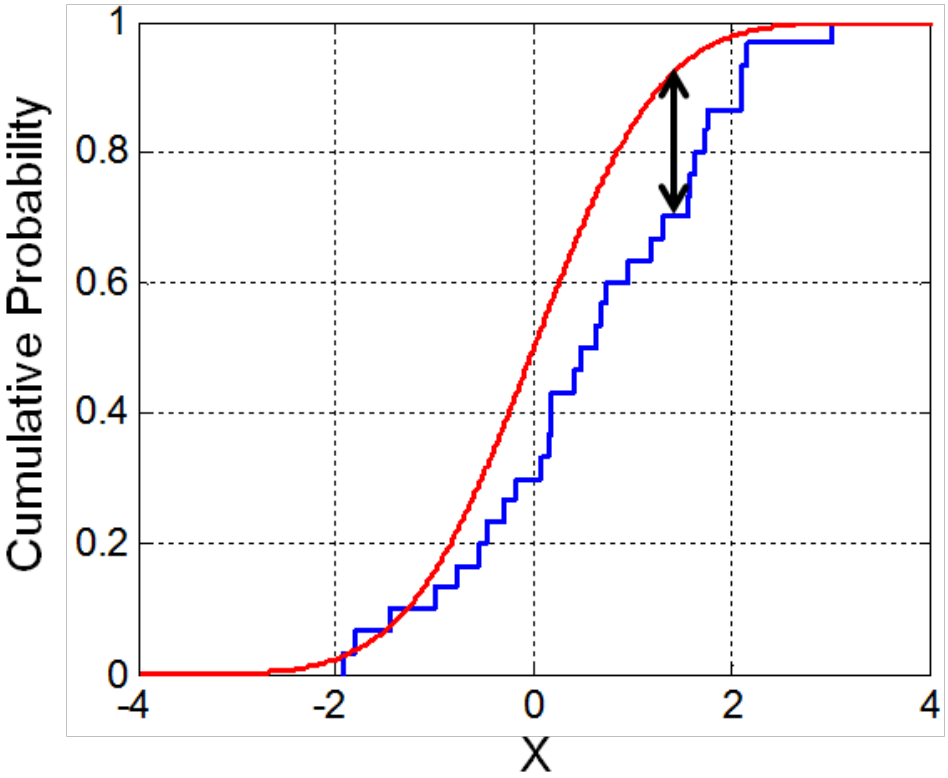
Next, we need to compute the distance between this empirical CDF and the CDF of the assumed distribution.

The one sample Kolmogorov-Smirnov test tests the null hypothesis that for the random, i.i.d sample $X_1, X_2, \ldots, X_n \sim F, H_0 : F = F_0$, where F_0 is the assumed underlying distribution.

It is computed as $S = \sup |F_n(x) - F(x)|$, where $\sup$ denotes the supremum, or the least upper bound, and where $F(x)$ is the empirical CDF. We would reject the null if S is too large.

Intuitively, you can view the absolute value function as measuring the distance between the empirical CDF and the assumed CDF. Since the supremum is an upper bound, taking the supremum means that the S test statistic is taking the $maximum$ distance between these two functions across all values of x. To summarize, S measures the greatest distance between the empirical CDF and the assumed underlying CDF for the entire function.

Below is a diagram showing the Kolmogorov-Smirnov test. The red is the CDF of the assumed distribution; the blue is the empirical CDF, and the black arrow is the S statistic which is the distance between the empirical CDF and the assumed distribution. Notice that the blue (empirical CDF) is a step-wise function!



"Diagram of a K-S Test."

Source: By Bscan - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=25222928

The S test statistic does have a corresponding distribution, and we'll denote the critical value of the test as k. With an α level of significance, we know that $P[S \leq k] = 1 - \alpha$. We can plug in the $|F_n(x) - F(x)|$ for S to create a confidence band.

$$|F_n(x) - F(x)| \leq k$$

$$-k \leq F_n(x) - F(x) \leq k$$

$$F_n(x) - k \leq F(x) \leq F_n(x) + k$$

So for $F(x)$, we have the bounds $[F_n(x) - k, F_n(x) + k]$.

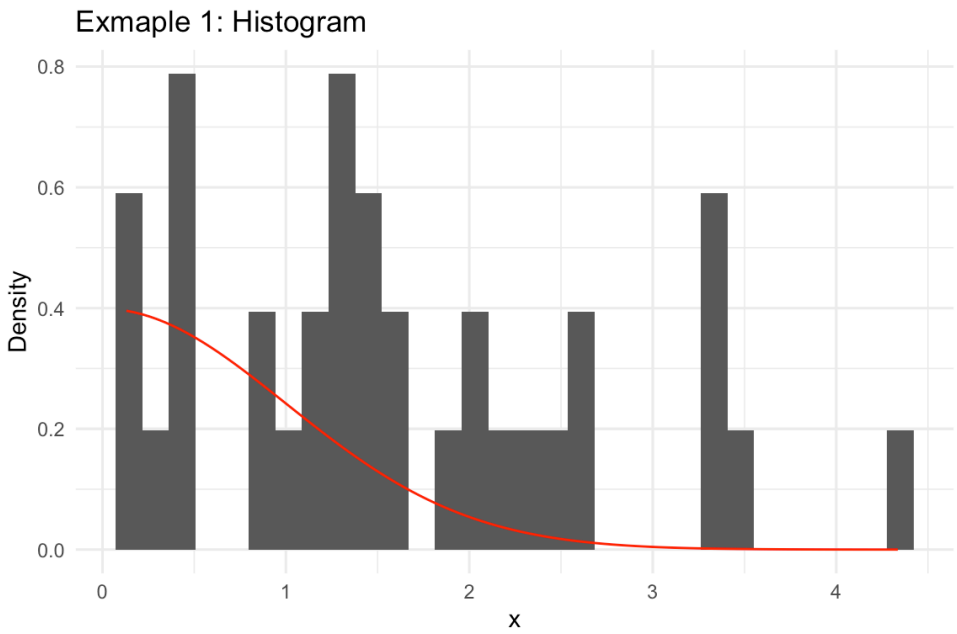
If we believe the underlying distribution F_0 is normal, we can replace F with $\Phi(\frac{x-\mu}{\sigma})$ to get $F_n(x) - k \leq \Phi(\frac{x-\mu}{\sigma}) \leq F_n(x) + k$. We can take the inverse of the normal distribution to get the bounds, which results in $\Phi^{-1}(F_n(x) - k) \leq \frac{x-\mu}{\sigma} \leq \Phi^{-1}(F_n(x) + k)$.

QQ plot vs. Histogram

One visual tool that provides statisticians insights into the underlying pattern or shape of the data distribution is the histogram, a graph that displays the frequencies across different values.

Looking at the histogram generated below, do you think it follows a normal distribution?

```
# Set seed for reproducibility
set.seed(123)

# Generate data
data <- rgamma(35, shape = 2)

# Create a data frame
df <- data.frame(values = data)

# Plot histogram using ggplot2
histogram <- ggplot(df, aes(x = values)) +
  geom_histogram(aes(y = after_stat(density))) +
  stat_function(
    fun = dnorm,
    args = list(mean = 0, sd = 1),
    col="red"
  )+
  theme_minimal() +
  labs(title = "Exmaple 1: Histogram", x = "x", y = "Dens

histogram
```



It exhibits a right-skewed distribution. However, if you check its QQ plot:

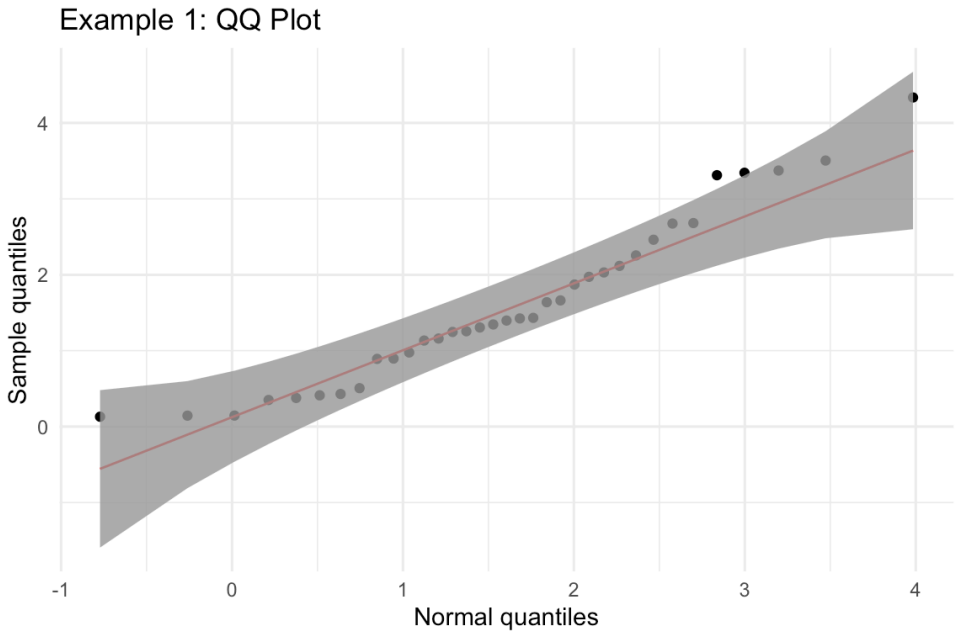
```
mean <- mean(df$values)
sd <- sd(df$values)
mean
```

```
[1] 1.605856
```

```
sd
```

```
[1] 1.086201
```

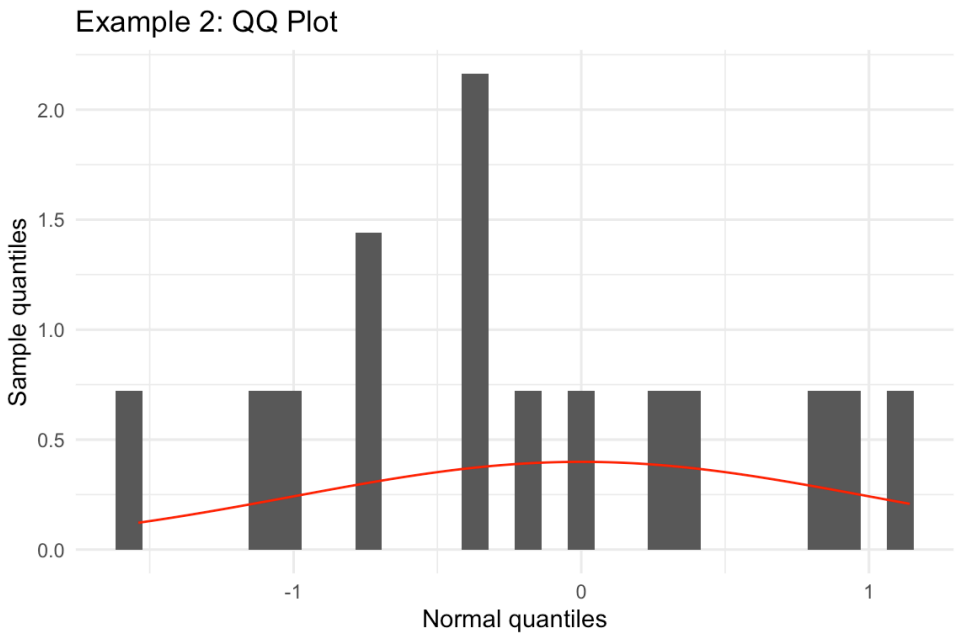
```
# Plot QQ plot
qqplot <- ggplot(df, aes(sample = values)) +
  geom_qq(distribution = qnorm, dparams = list(mean = mean
  geom_qq_line(distribution = qnorm, dparams = list(mean =
  stat_qq_band(distribution = "norm", dparams = list(mean
  theme_minimal() +
  labs(title = "Example 1: QQ Plot",  x = "Normal quantile

qqplot
```



It shows normality in the QQ plot but with heavier tails at two ends.

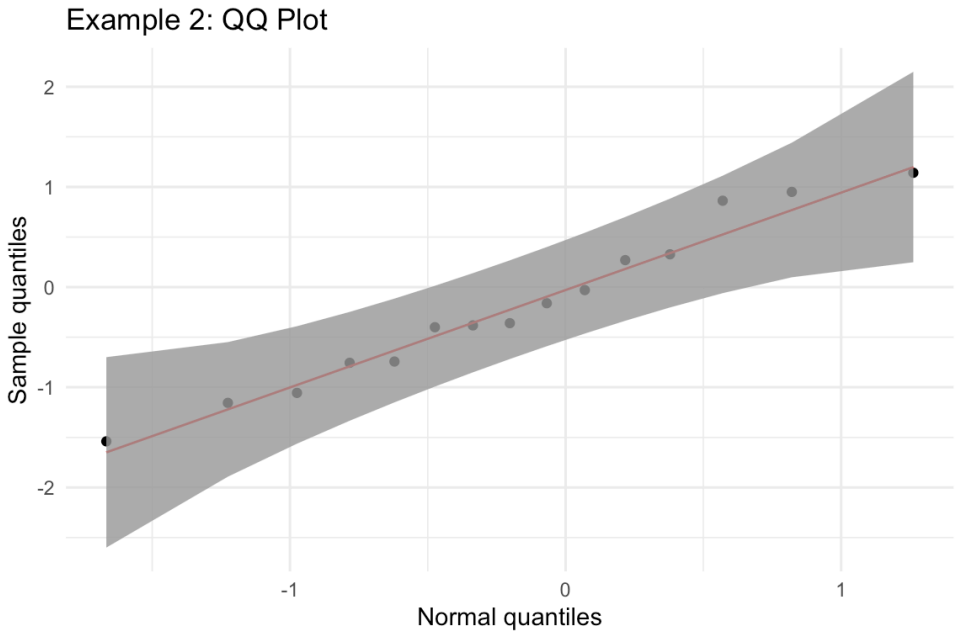
Now we'll look at another example. Looking at the histogram below, can you tell if it's normal?

```
# read in example 2 data
example_data <- read_csv("data_examples.csv",show_col_type

# plot histogram
histogram2 <- ggplot(example_data, aes(x = C4)) +
  geom_histogram(aes(y = after_stat(density)))+
  stat_function(
    fun = dnorm,
    args = list(mean = 0, sd = 1),
    col="red"
  )+
  theme_minimal() +
  labs(title = "Example 2: QQ Plot",  x = "Normal quantile

histogram2
```



It's hard to tell if it's a normal distribution, right? But if we look at the QQ plot with bands, the answer is clearer.

```
# Plot QQ plot
mu2 = mean(example_data$C4, na.rm = TRUE)
sd2 = sd(example_data$C4, na.rm = TRUE)

qqplot2 <- ggplot(example_data, aes(sample = C4)) +
  geom_qq(distribution = qnorm, dparams = list(mean = mu2,
  geom_qq_line(distribution = qnorm, dparams = list(mean =
  stat_qq_band(distribution = "norm", dparams = list(mean
  theme_minimal() +
  labs(title = "Example 2: QQ Plot",  x = "Normal quantile
qqplot2
```



All the values were within the bands, so the data follows a normal distribution!

Now we've shown that 1) data that does not look normal in the histogram may still be normally distributed, if you look at its QQ plot, and 2) data where it is unclear if it follows a normal distribution in the histogram can show normality in the QQ plot.

Why are QQ plots better than histograms at showing if the data follows a certain distribution?

QQ plots compare sample quantiles and theoretical quantiles directly and thus provides effective identification in deviations from the assumed distribution and possible skewness. These deviations are more noticeable because they are represented as departures from a straight line in the QQ plot. Since histograms simply plot distributions, it can be challenging to assess the skewness and the shape of the tails, particularly for small sample sizes.

For these reasons, a QQ plot is advantageous in showing whether a sample fits a particular distribution and provides more information about skewness and tail performances than a histogram.

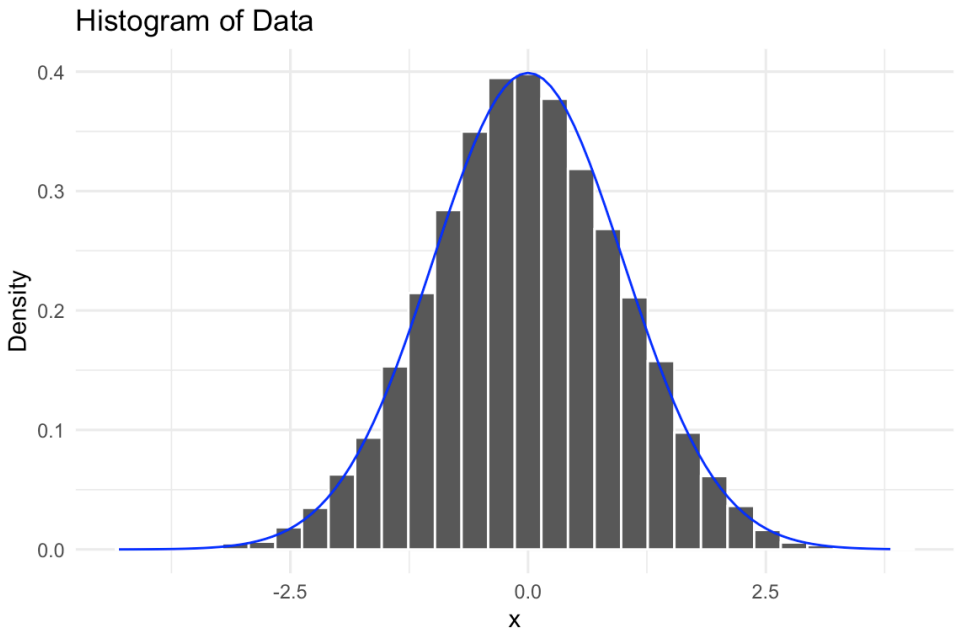
QQ plots Intervals and Distribution Tests

Another tool that statisticians can use to determine if a sample follows a certain distribution is a distribution test. Today, we'll be using the Kolmogorov–Smirnov test to test for normality, the same test used to determine QQ plot confidence intervals.

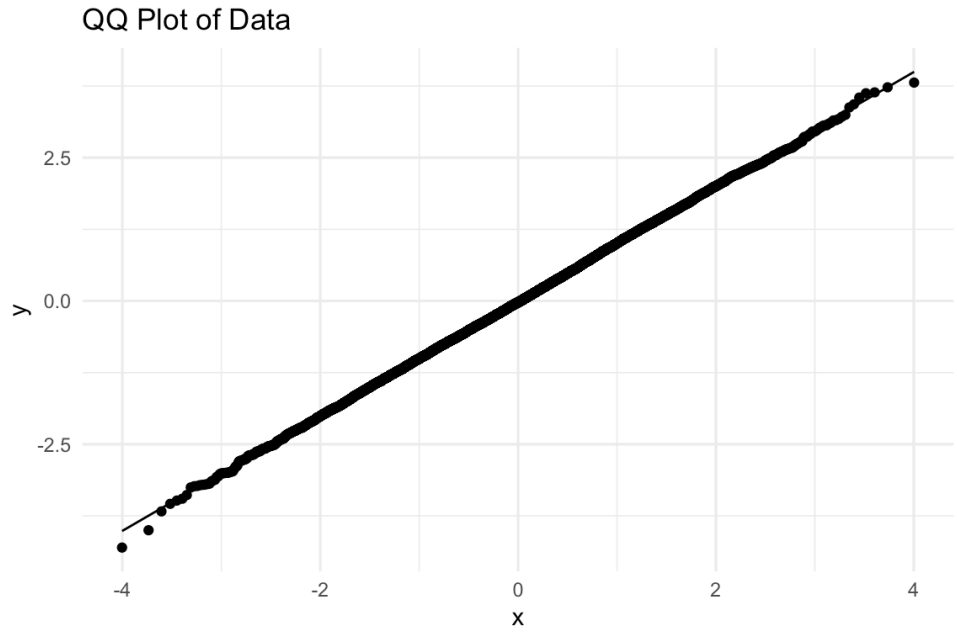
I've generated some normal data in "distTest_data.csv". Below is code to upload this CSV, plot its histogram, and its QQ plot.

```
distTest_data <- read.csv("distTest_data.csv")

hist1 <- ggplot(distTest_data, aes(x = x)) +
  geom_histogram(aes(y = after_stat(density)), col = "whi
  stat_function(
    fun = dnorm,
    args = list(mean = 0, sd = 1),
    col = 'blue'
  ) +
  theme_minimal() +
```

```
  ylab("Density") + xlab("x") + ggtitle("Histogram of Data
hist1
```




```
qq_dist <- ggplot(distTest_data, aes(sample = x)) +
  geom_qq() + # Add qq points
  geom_qq_line() + # Add qq line
  theme_minimal() + ggtitle("QQ Plot of Data")
qq_dist
```



Looks normal (literally)!

Let's see what the Kolmogorov–Smirnov Distribution test says.

```
ks.test(distTest_data$x, pnorm)
```

```
    Asymptotic one-sample Kolmogorov-Smirnov test

data:  distTest_data$x
D = 0.010917, p-value = 0.04411
alternative hypothesis: two-sided
```

That's a low p-value (< 0.05). Why is that?

Let's look at how we generated the data.

```
#Generate data
#| eval: FALSE
set.seed(1)
distTest_data1 <- tibble(
  x = rnorm(16000)
)

add_data <- tibble(
  x = c(-4)
)

distTest_data1 <- bind_rows(distTest_data1, add_data)

write.csv(distTest_data1, "distTest_data.csv", row.names =
```

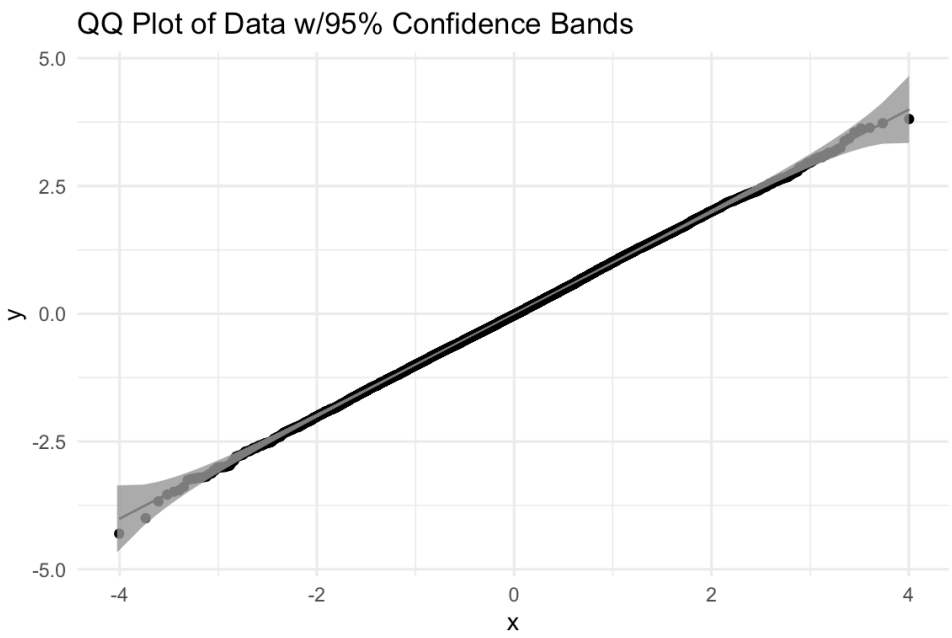
Do you see something different with this code? Look at the sample size of the data, and then at "add_data" and "bind_rows".

I generated 16,000 normal data points and simply added a -4. Because the sample size is so large (making the standard error extremely small), by adding just one outlier, the null of the data being normal is rejected, even though this data is clearly normal.

This particular situation is one where the QQ plot with confidence intervals has an advantage over the distribution test, since just a quick visual check would let us know if the data is normal. Let's see it below.

```
#QQ Plot of Normal Data
qq_dist1 <- ggplot(distTest_data, aes(sample = x)) +
  geom_qq() + # Add qq points
  geom_qq_line() + # Add qq line
  stat_qq_band() + #Add confidence bands
  theme_minimal() + ggtitle("QQ Plot of Data w/95% Confide
```

```
qq_dist1
```



Real Life Data Examples

Now let's see what you've learned.

We have two real life data exercises. We'll have you guess the underlying distribution of the data from a few choices, and then we'll show you the QQ plots with confidence intervals.

The first will be using a built in R dataset that recorded the height of 812 men in inches.

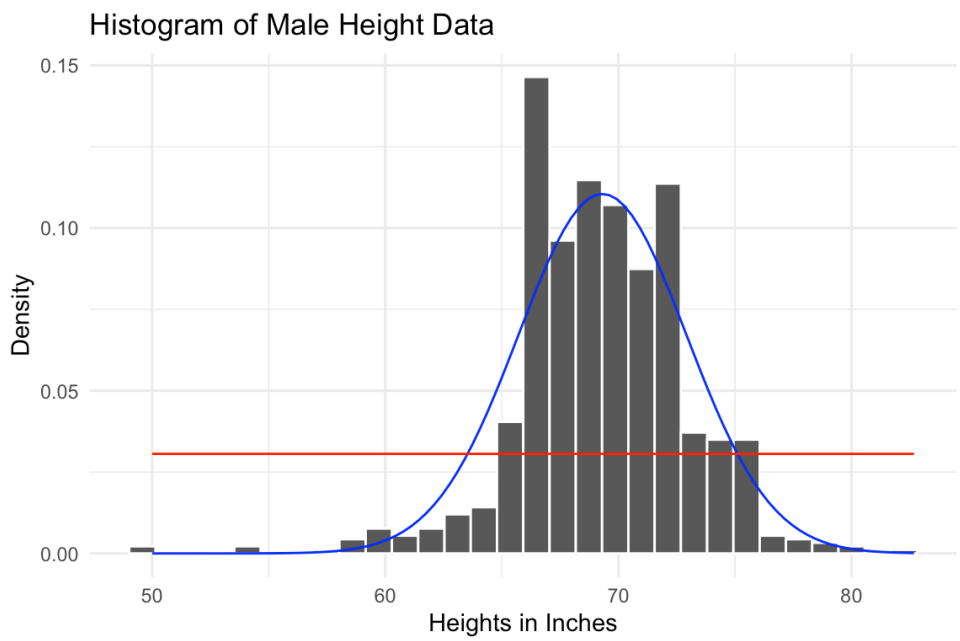
Men's Height Data Exercise

Do you think the the height of men is normally or uniformly distributed?

Let's look at a histogram of the data.

```
heights <- read.csv("heights.csv")

male_heights <- subset(heights, sex == "Male")  #Taking or

mu = mean(male_heights$height)
sigma = sd(male_heights$height)
```

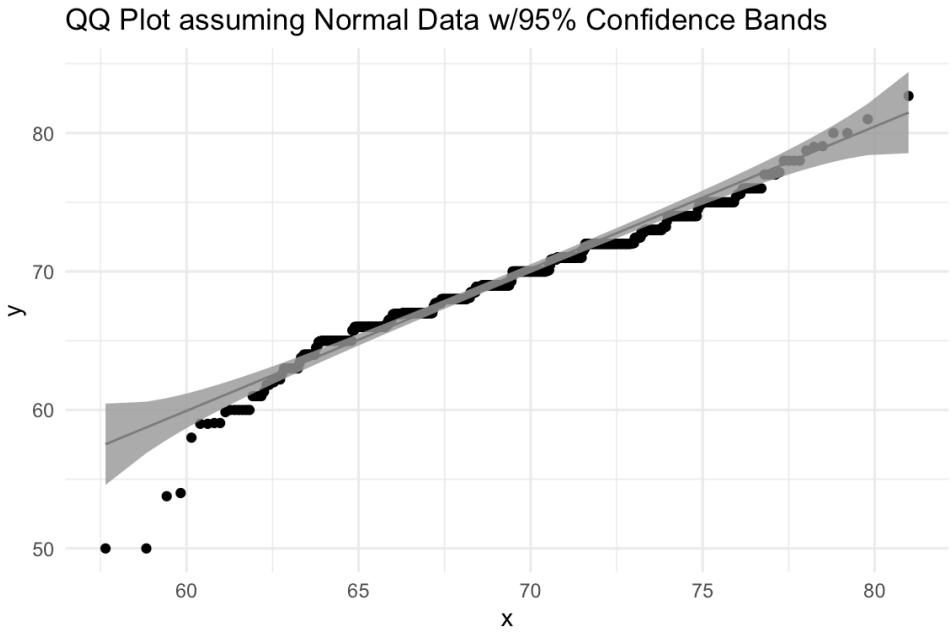
```
min = min(male_heights$height)
max = max(male_heights$height)

hist2 <- ggplot(male_heights, aes(x=height)) +
    geom_histogram(aes(y = after_stat(density)), col = "wh
    theme_minimal() +
    stat_function(
      fun = dnorm,
      args = list(mean = mu, sd = sigma),
      col = 'blue'
    ) +
    stat_function(
      fun = dunif,
      args = list(min = min, max = max),
      col = 'red'
    ) +
    ylab("Density") + xlab("Heights in Inches") + ggtitle(
hist2
```



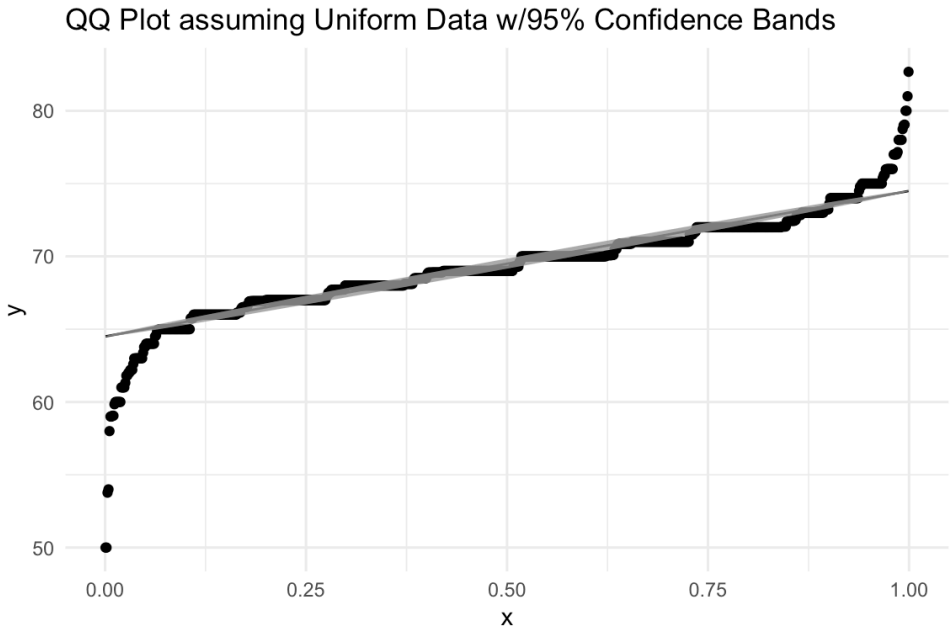
The red line is the uniform distribution, and the blue is the normal.

The code below creates the QQ plots of the height data assuming a normal distribution and a uniform distribution.

```
#Normal QQ plot
qqn <- ggplot(male_heights, aes(sample=height)) +
  geom_qq(distribution = qnorm, dparams = list(mean = mu,
  geom_qq_line(dparams = list(mean = mu, sd = sigma)) + #
  stat_qq_band(distribution = "norm", dparams = list(mean
  theme_minimal() + ggtitle("QQ Plot assuming Normal Data
qqn
```




```
#Uniform QQ plot
qqun <- ggplot(male_heights, aes(sample=height)) +
  geom_qq(distribution = qunif) + # Add qq points, specify
  geom_qq_line(distribution = qunif) + # Add qq line, spec
  stat_qq_band(distribution = "unif") +
  theme_minimal() + ggtitle("QQ Plot assuming Uniform Data
qqun
```



Note

Question 2: Now, after seeing the QQ plots and their bands, what

> distribution do you think the height of men follows?

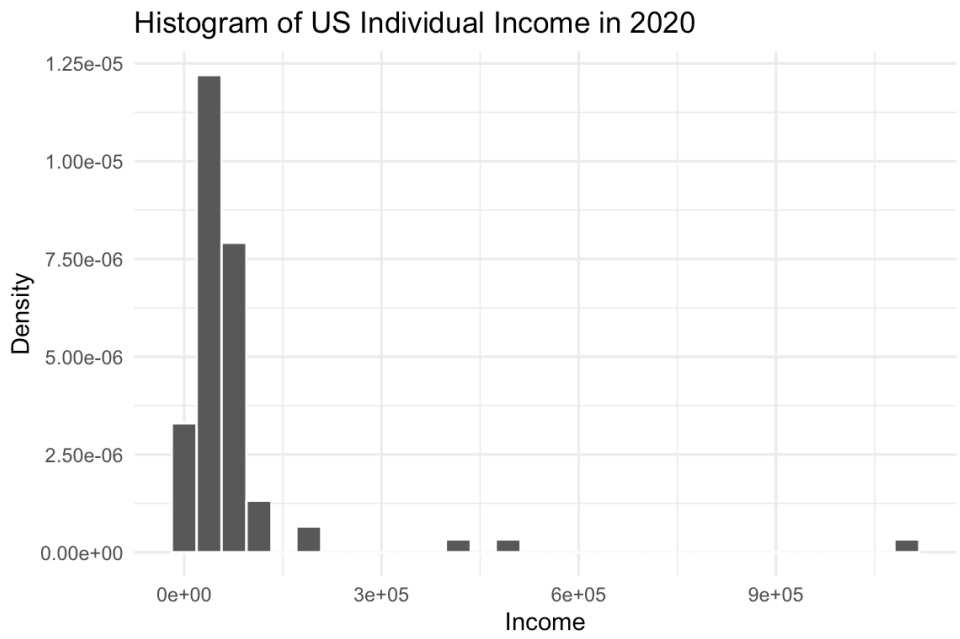
Income Data Exercise

In the second example, we will use wage and salary data from 65535 people in the US in 2020. We randomly sampled 80 observations from the original dataset. This data was collected by US Bureau of Labor Statistics.

What distribution do you think the income for individuals in the US in 2020 would follow? Normal, log-normal, or uniform?

Let's start with the histogram of the original income data.

```
CPSworkers <- read_csv("sampled_CPS.csv", show_col_types =

inc_hist <- ggplot(CPSworkers,aes(x=incwage))+
  geom_histogram(aes(y=after_stat(density)),bins = 30, co

inc_hist
```

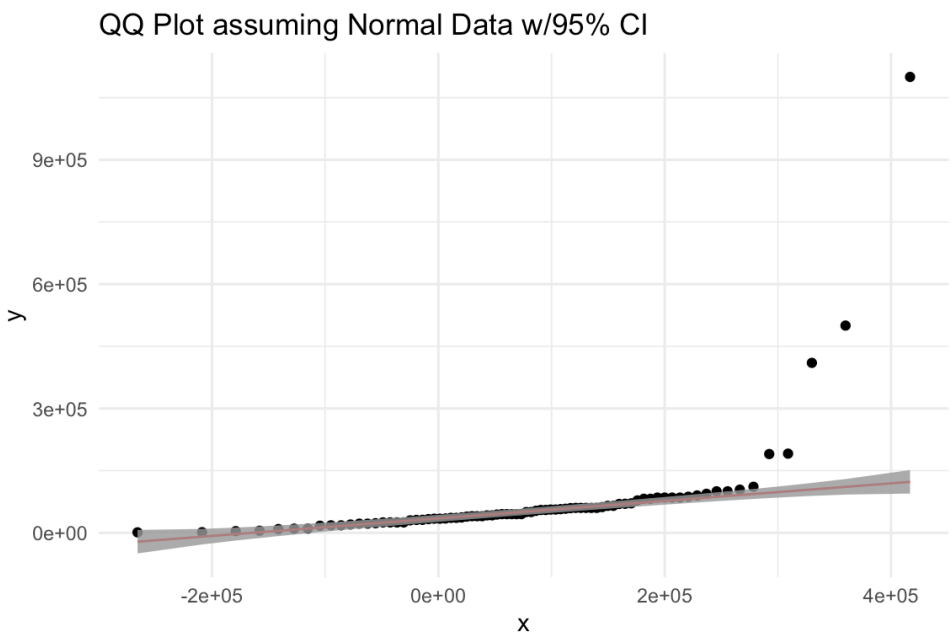


It is extremely right-skewed. We have a lot of billionaires.

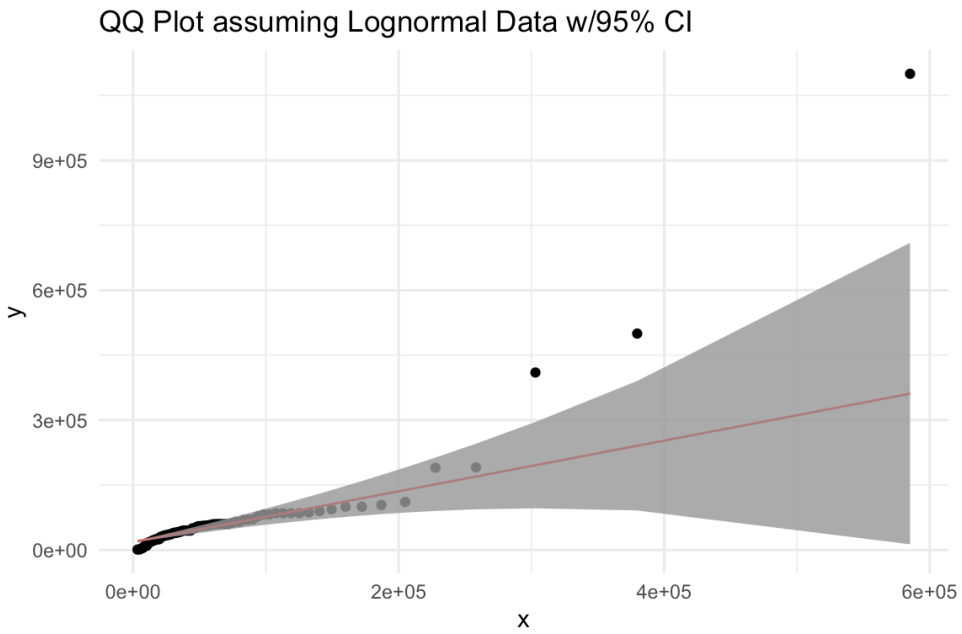
Let's also look at the data's QQ plots with confidence bands to further explore what distribution it could follow.

The code below creates the QQ plots, assuming first a normal distribution, then a log-normal distribution, and a uniform distribution.

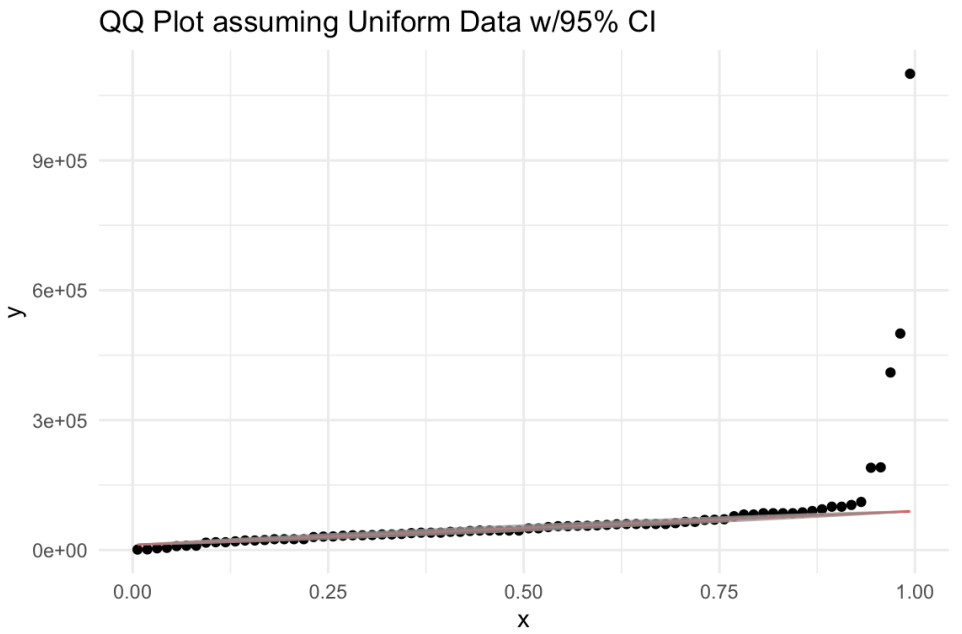
```
mean2 <- mean(CPSworkers$incwage)
sd2 <- sd(CPSworkers$incwage)

# Normal QQ plot
qqn2 <- ggplot(CPSworkers, aes(sample=incwage)) +
  geom_qq(distribution = qnorm, dparams = list(mean = mean
  geom_qq_line(dparams = list(mean = mean2, sd = sd2), co

qqn2
```




```
# Log-adjust the income variable
lnwage <- log(CPSworkers$incwage)
mean3 <- mean(lnwage)
sd3 <- sd(lnwage)

# Log-normal QQ plot (mean and sd same as normal distr.)
qqln <- ggplot(CPSworkers, aes(sample=incwage)) +
  geom_qq(distribution = qlnorm, dparams = list(meanlog =
  geom_qq_line(distribution = qlnorm, dparams = list(mean
  stat_qq_band(distribution = "lnorm", dparams = list(mean
  theme_minimal() + ggtitle("QQ Plot assuming Lognormal Da
qqln
```



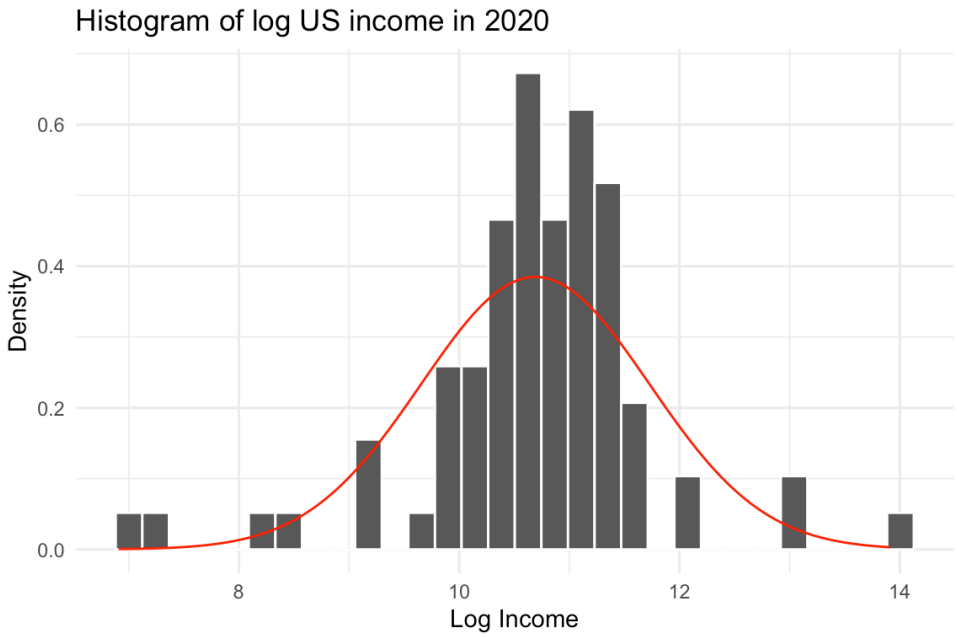

```
# Uniform QQ plot
qqun2 <- ggplot(CPSworkers, aes(sample=incwage)) +
  geom_qq(distribution = qunif)+
  geom_qq_line(distribution = qunif, col ="red")+  geom_q

qqun2
```



> **Note**
>
> Question 3: After seeing QQ plots with their bands, what distribution do you think the income data follows?

Now we'll log-adjust the income variable. Let's see what the histogram looks like afterwards:

```
lnwage_hist <- ggplot(CPSworkers,aes(x=lnwage))+
  geom_histogram(aes(y=after_stat(density)),bins = 30, co
    fun = dnorm,
    args = list(mean = mean3, sd = sd3),
    col="red"
  ) + theme_minimal() + ylab("Density") + xlab("Log Income

lnwage_hist
```



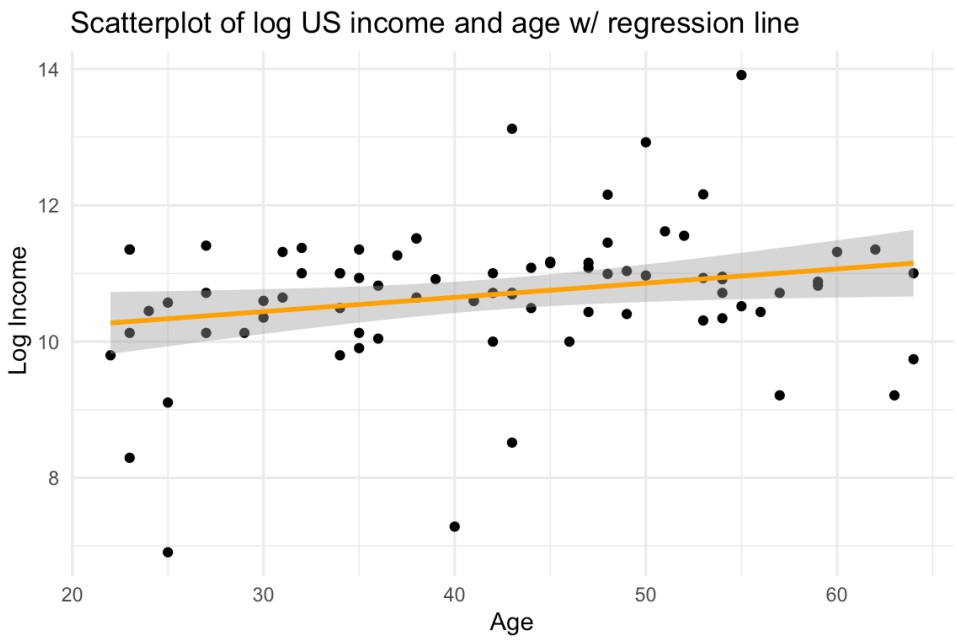
Looks normal now!

We know that age is a huge factor contributing to one's income level, so we can run a simple linear regression to explore more about this relationship.

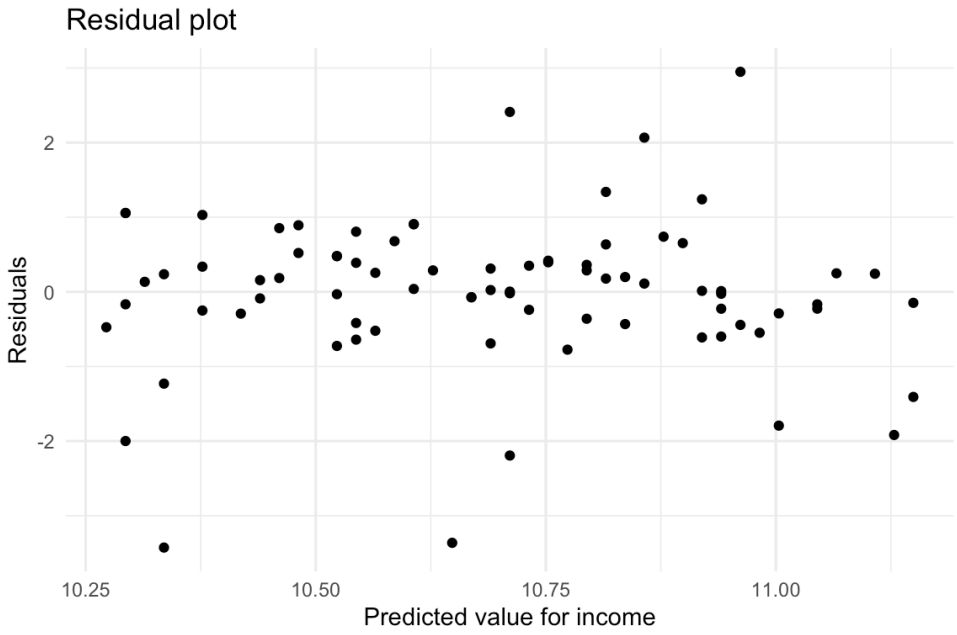
We'll fit the regression $\ln(\text{wage})_i = \beta_0 + \beta_1 \text{Age}_i + \epsilon_i$. We will also make a residual plot and a QQ plot of the residuals to check if the residuals are uncorrelated and normally distributed.

```
# Adjust the data to log-normal
lnwage <- log(CPSworkers$incwage)

# Fit the regression model
model <- lm(lnwage ~ age, data = CPSworkers)

#scatter plot with the fitted model
scatter_line <- ggplot(CPSworkers, aes(x = age, y = lnwage
```

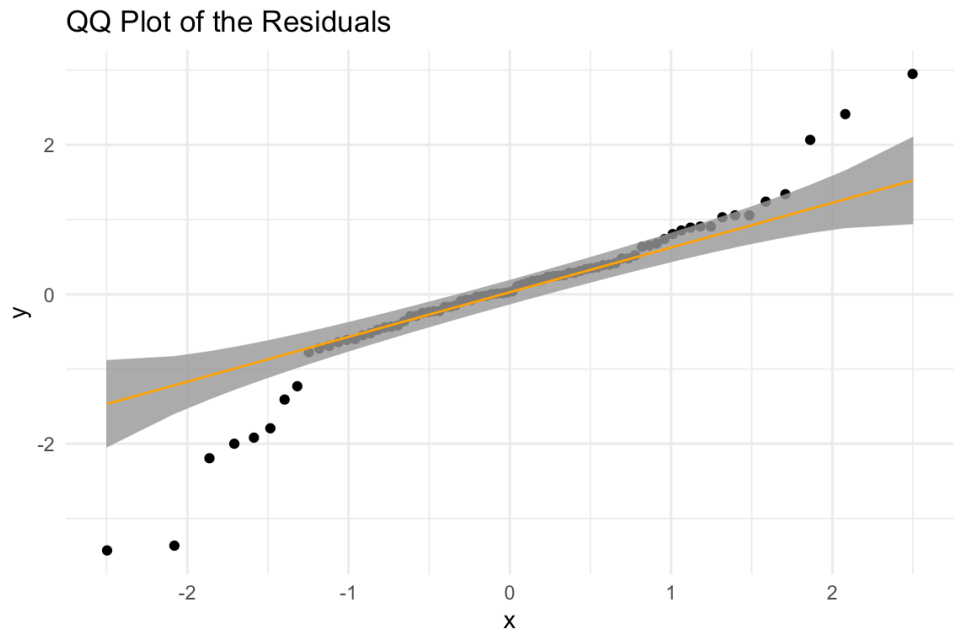
```
  geom_point() +
  geom_smooth(color = "orange",
              method = "lm",
              formula = y ~ x) +
  theme_minimal()+
  labs(title = "Scatterplot of log US income and age w/ re
       x = "Age",
       y = " Log Income")

print(scatter_line)
```




```
#residual plot
residual <- ggplot(model, aes(x = fitted(model), y = resi
  geom_point()+
  theme_minimal()+
  labs(title = "Residual plot",
       x = "Predicted value for income",
       y = "Residuals")

print(residual)
```




```
#QQ plot
results = broom::augment(model)
qqpr <- ggplot(results, aes(sample = .resid)) +
  geom_qq() +
  stat_qq_band() +
  geom_qq_line(col = "orange") + theme_minimal() + ggtitle
qqpr
```



Note

Question 4: After seeing these plots, do you believe a linear model is correct for this relationship?

Sources

https://www.stat.auckland.ac.nz/~ihaka/courses/787/lectures-quantiles2-handouts.pdf
https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test
https://en.wikipedia.org/wiki/Empirical_distribution_function
https://statisticsbyjim.com/graphs/qq-plot/