STATS 2107
Statistical Modelling and Inference II
Solutions
Workshop 4: Sampling distributions part 1

Matt Ryan

Semester 2 2022

Contents

The sampling distribution of the sample mean

What is a sampling distribution?

Suppose $Y_1, Y_2, \ldots, Y_n$ is a random sample, and T is a statistic on the Y_i. Then the distribution of T is called the *sampling distribution*.

The sample mean

For example, suppose each $Y_i \sim N(\mu, \sigma^2)$ and $T = \bar{Y}$. Then the sampling distribution is

$$\bar{Y} \sim N\left(\mu, \frac{\sigma^2}{n}\right).$$

What is meant by sampling distribution?

$$\begin{array}{cccccc}
Y_{11}, & Y_{12}, & \dots, & Y_{1n} & \rightarrow & T_1 \\
Y_{21}, & Y_{22}, & \dots, & Y_{2n} & \rightarrow & T_2 \\
Y_{31}, & Y_{32}, & \dots, & Y_{3n} & \rightarrow & T_3 \\
Y_{41}, & Y_{42}, & \dots, & Y_{4n} & \rightarrow & T_4 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}$$

Does the practice match the theory?

In theory, if our data is normal, the sample mean is normal. Let's test this.

1. Consider samples of size 3, $Y_1, Y_2, Y_3 \sim N(5, 2^2)$.
2. Every time we take a sample, calculate the mean

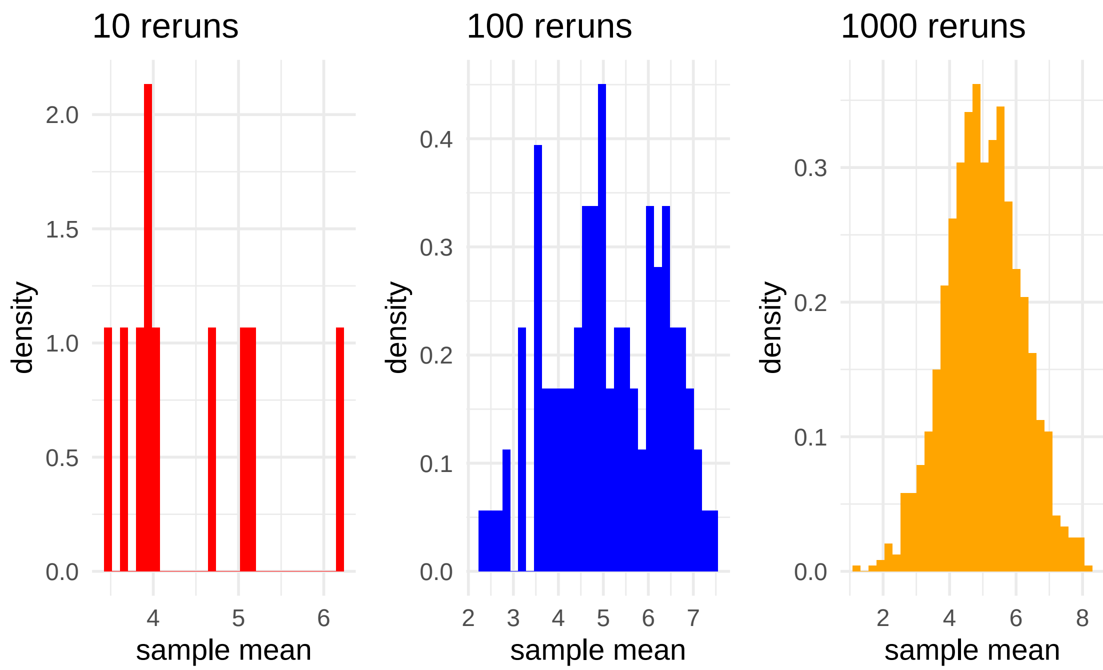
$$\bar{Y} = \frac{1}{3}\left(Y_1 + Y_2 + Y_3\right).$$

3. Generate 10, 100, and 1000 samples to look at the distribution.
4. Is it normal?

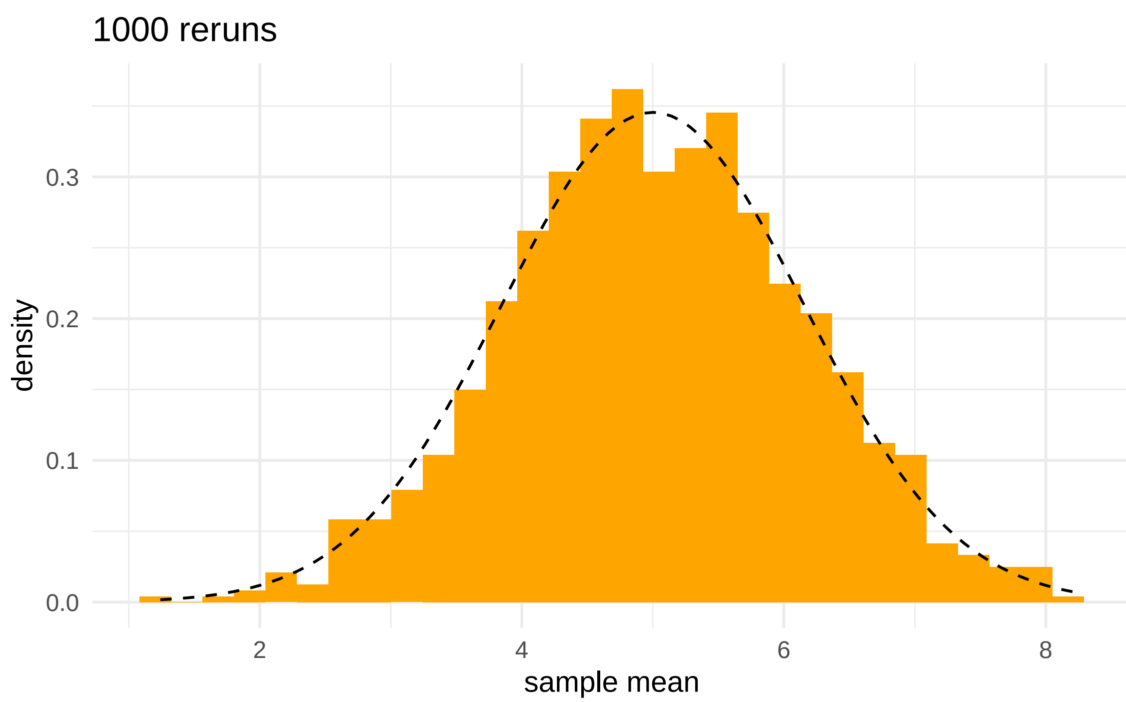
Some R code to do this

```
# Set up some parameters
N <- 10
mu <- 5
sig <- 2
n <- 3

# Get the samples and calculate the mean
norm_sample_3_10 <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
  map_dbl(mean) #Hey look, a new function!
```

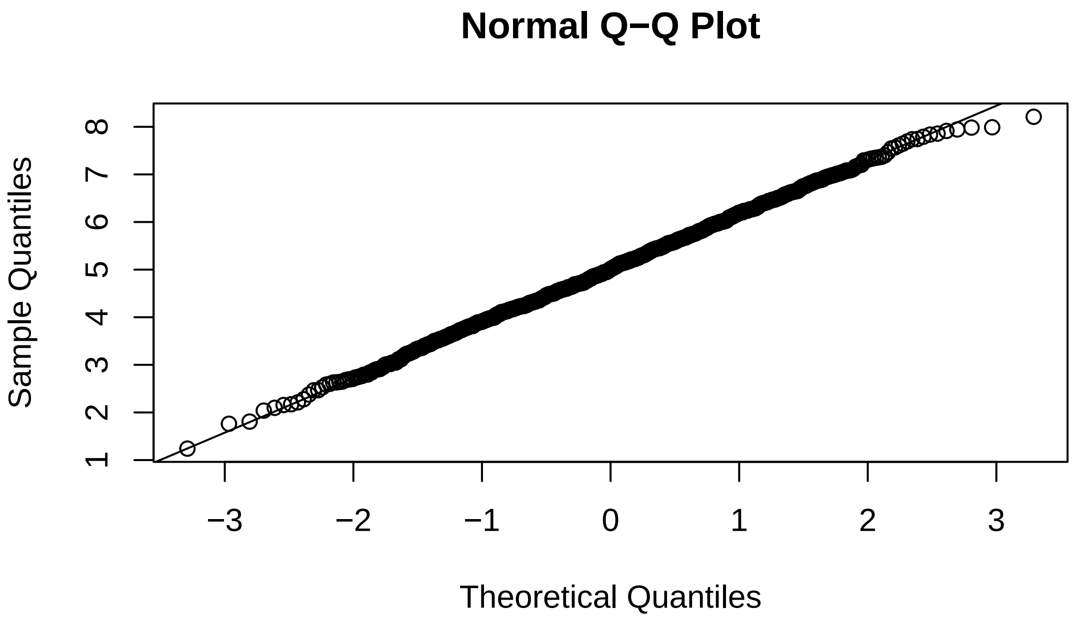
Histograms



Is this normal?



QQplot



Your turn

What to do

1. Adapt the given code to produce the histograms for $N = 10, 100, 1000$.

Solutions:
The adapted code is below, including code to generate the histograms.

```
# I use the patchwork library for the plots, so run
# library(patchwork) once it is installed.
# Set up some parameters
N <- 10
mu <- 5
sig <- 2
n <- 3

# Get the samples and calculate the mean
## N = 10
norm_sample_3_10 <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
  map_dbl(mean)

## N = 100
N <- 100
norm_sample_3_100 <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
  map_dbl(mean)

## N = 1000
N <- 1000
```

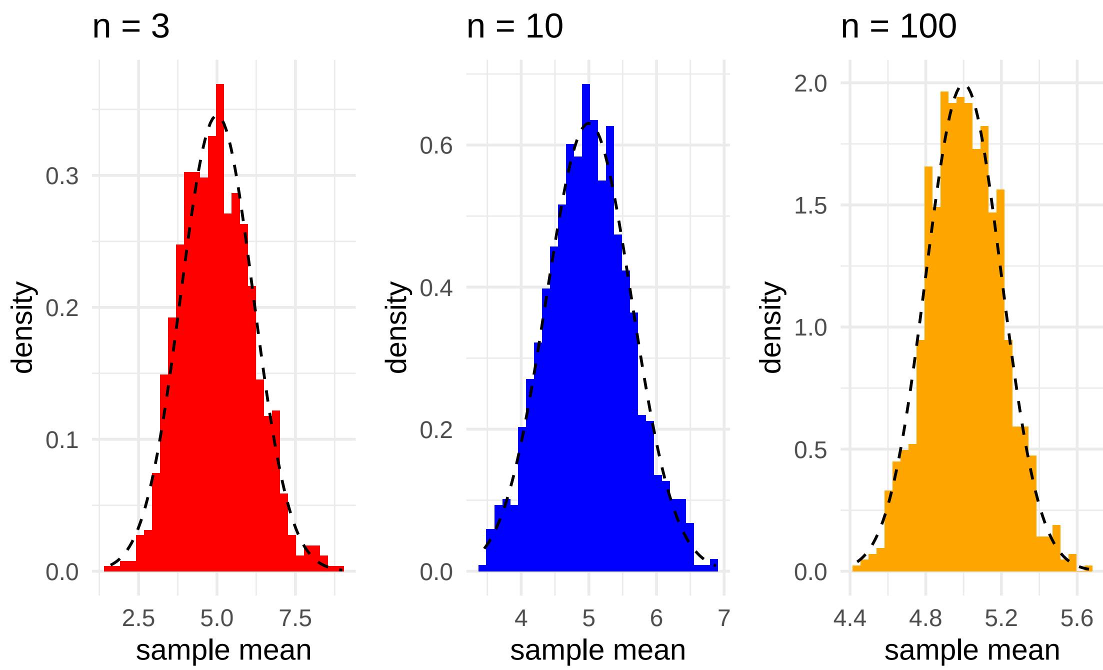
```r
norm_sample_3_1000 <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
  map_dbl(mean)

## Generate the plots, make them look pretty.
p1 <- ggplot(data = tibble(x = norm_sample_3_10),
             aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 fill = "red") +
  theme_minimal() +
  labs(x = "sample mean", y = "density",
       title = "10 reruns")
p2 <- ggplot(data = tibble(x = norm_sample_3_100),
             aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 fill = "blue") +
  theme_minimal() +
  labs(x = "sample mean", y = "density",
       title = "100 reruns")
p3 <- ggplot(data = tibble(x = norm_sample_3_1000),
             aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 fill = "orange") +
  theme_minimal() +
  labs(x = "sample mean", y = "density",
       title = "1000 reruns")
## Use patchworks to display them side by side.
p1 + p2 + p3
```

2. Explore the distribution as you increase n.

Solutions:
Lets look at 3 examples. Fix $N = 1000$, and look at $n = 3, 10, 100$. Looking at the plots, the densities are matching well.

```r
# I use the patchwork library for the plots, so run
# library(patchwork) once it is installed.
# Set up some parameters
N <- 1000
mu <- 5
sig <- 2
n <- 3

# Get the samples and calculate the mean
## n = 3
norm_sample_3_1000 <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
  map_dbl(mean)

## n = 10
n <- 10
norm_sample_10_1000 <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
```

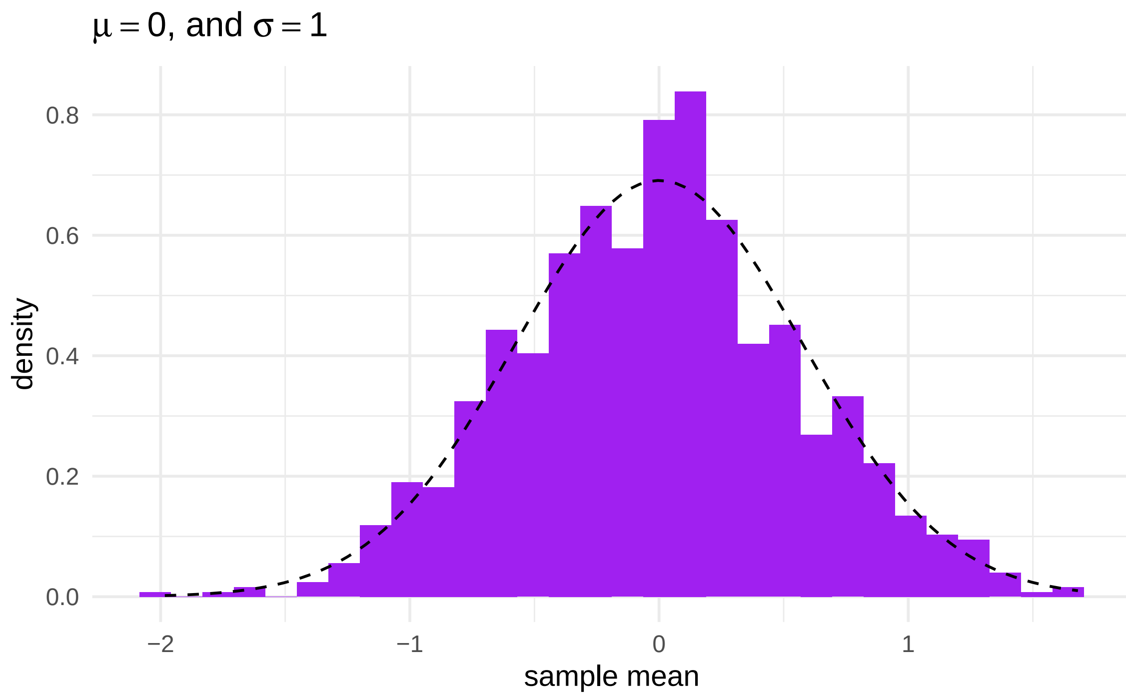
```r
  map_dbl(mean)

## n = 100
n <- 100
norm_sample_100_1000 <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
  map_dbl(mean)

## Generate the plots, make them look pretty.
p1 <- ggplot(data = tibble(x = norm_sample_3_1000),
             aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 fill = "red") +
  theme_minimal() +
  labs(x = "sample mean", y = "density",
       title = "n = 3 ") +
  stat_function(fun = dnorm, args = list(mean = mu, sd = sig/sqrt(3)),
                lty = 2) # This function plots the density on top.
p2 <- ggplot(data = tibble(x = norm_sample_10_1000),
             aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 fill = "blue") +
  theme_minimal() +
  labs(x = "sample mean", y = "density",
       title = "n = 10") +
  stat_function(fun = dnorm, args = list(mean = mu, sd = sig/sqrt(10)),
                lty = 2) # This function plots the density on top.
p3 <- ggplot(data = tibble(x = norm_sample_100_1000),
             aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 fill = "orange") +
  theme_minimal() +
  labs(x = "sample mean", y = "density",
       title = "n = 100") +
  stat_function(fun = dnorm, args = list(mean = mu, sd = sig/sqrt(100)),
                lty = 2) # This function plots the density on top.
## Use patchworks to display them side by side.
p1 + p2 + p3
```



3. Explore the distribution as you change μ and σ^2.

Solutions:
Let's present one exploration for $\mu = 0$ and $\sigma = 1$. We present 1 plot.

```
# Set up some parameters
N <- 1000
mu <- 0
sig <- 1
n <- 3

norm_samp <- N %>%
  rerun(rnorm(n, mu, sig)) %>%
  map_dbl(mean)

ggplot(data = tibble(x = norm_samp),
             aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 fill = "purple") +
  theme_minimal() +
  labs(x = "sample mean", y = "density",
       title = latex2exp::TeX("$\\mu = 0$, and $\\sigma = 1$")) +
  stat_function(fun = dnorm, args = list(mean = mu, sd = sig/sqrt(n)),
                lty = 2)
```



Non-normal data

The problem

Our distributional result relies on the fact that $Y_i \sim N(\mu, \sigma^2)$, although we know

$$\mathrm{E}[\bar{Y}] = \mu$$

and

$$\mathrm{Var}(\bar{Y}) = \frac{\sigma^2}{n}\,.$$

CLT to the rescue?

Let $Y_1, Y_2, \ldots, Y_n$ be independent independent and identically distributed random variables with $\mathrm{E}[Y_i] = \mu$ and $\mathrm{Var}(Y_i) = \sigma^2 < \infty$. Define

$$U_n = \frac{\bar{Y} - \mu}{\sigma/\sqrt{n}}\,.$$

Then the distribution of U_n converges to the standard normal distribution function as $n \to \infty$.

The problem

The CLT only kicks in for large n, the worse the distribution, the larger the n needed.

χ^2_5

Let's explore the sampling distribution of the sample mean for $Y_1, Y_2, \ldots, Y_n \sim \chi^2_5$. We will

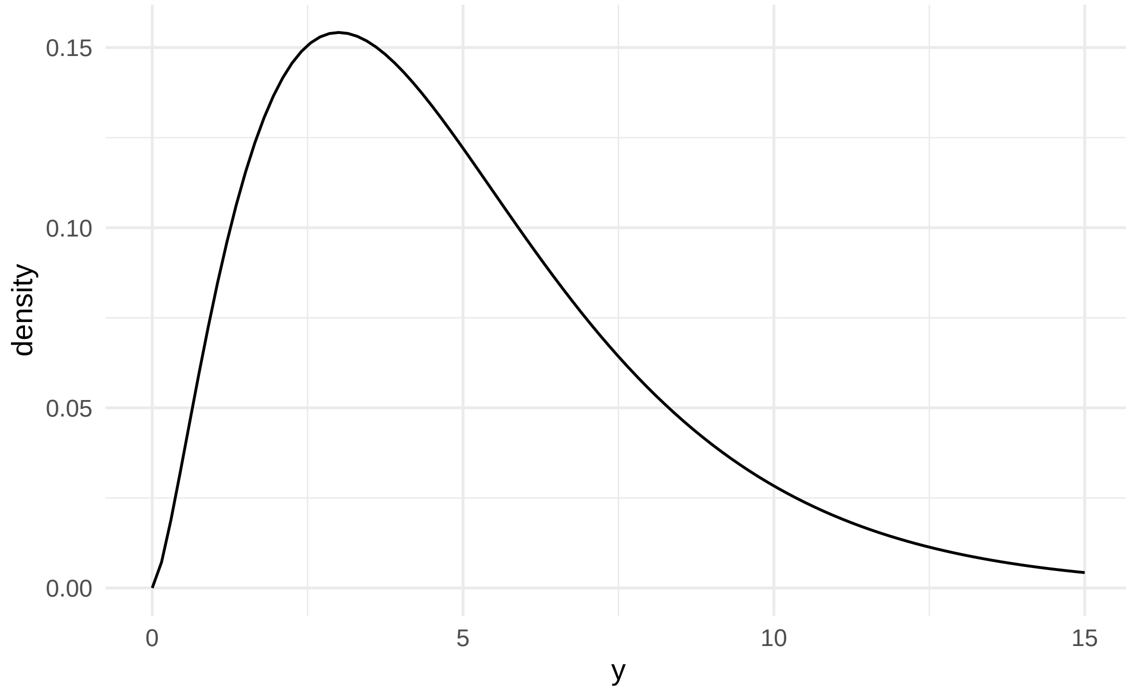
1. Consider samples of size 3, $Y_1, Y_2, Y_3 \sim \chi^2_5$.

2. Every time we take a sample, calculate the mean

$$\bar{Y} = \frac{1}{3}\left(Y_1 + Y_2 + Y_3\right) .$$

3. Generate 10, 100, and 1000 samples to look at the distribution.
4. Is it normal? Expect to see $N(5, 10/3)$.

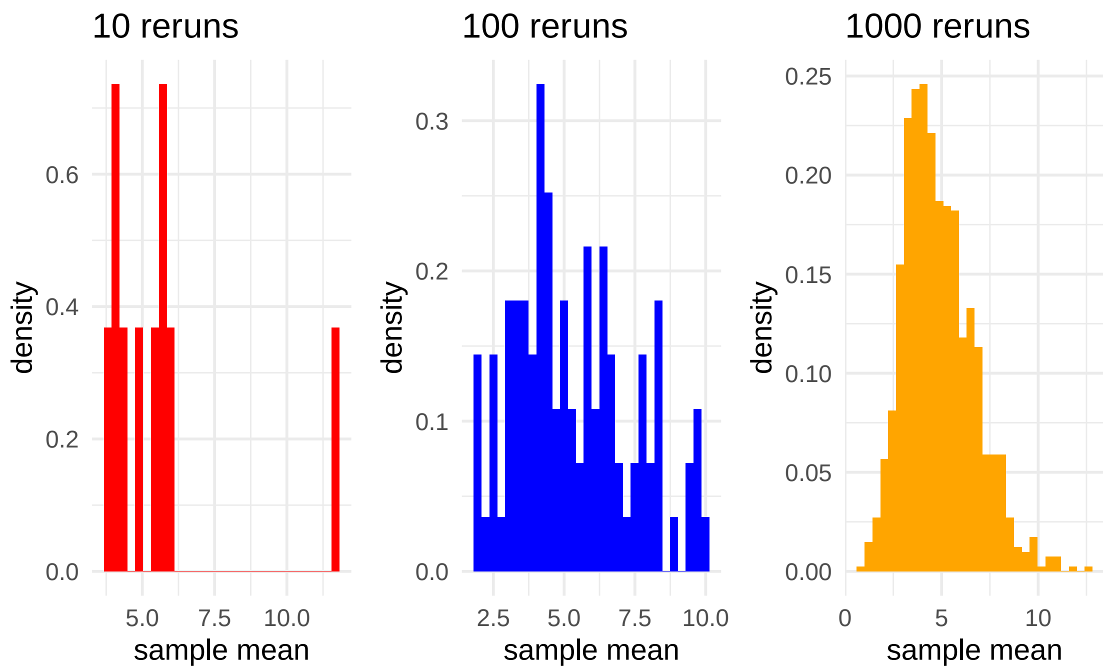
Is the χ^2_5 normal?



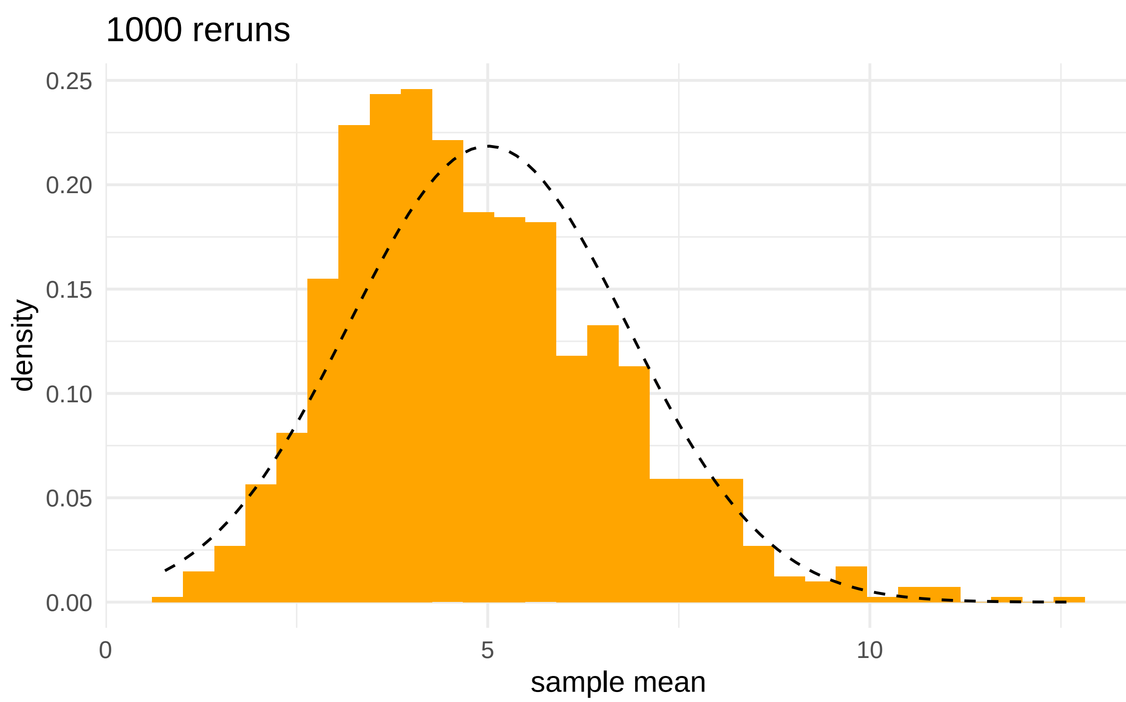
Some R code to do this

```
# Set up some parameters
N <- 10
df <- 5
n <- 3

# Get the samples and calculate the mean
chi_sample_3_10 <- N %>%
  rerun(rchisq(n, df)) %>%
  map_dbl(mean)
```

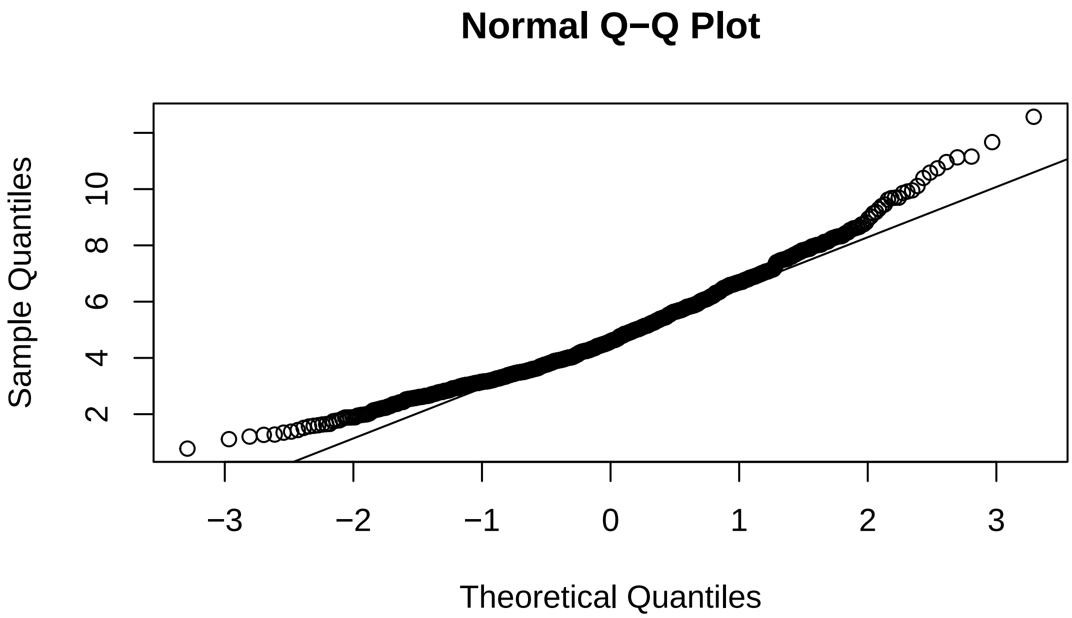
Histograms



Is this normal?



QQplot



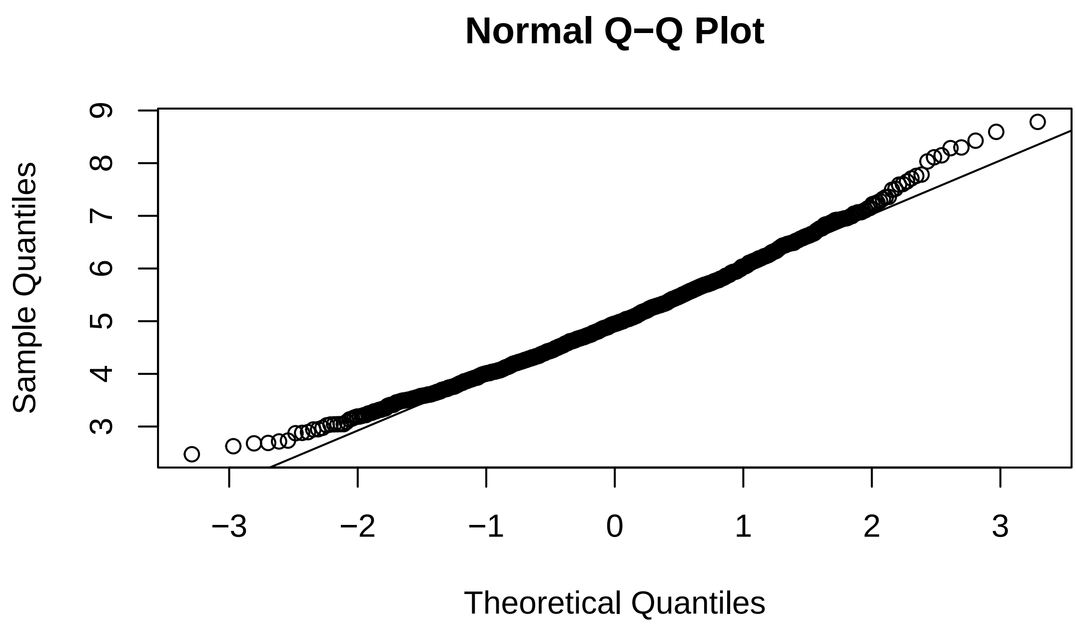
Your turn

What to do

1. Explore the distribution of the sample mean as you increase the sample size n from the χ^2_5. When does it start to become normal?

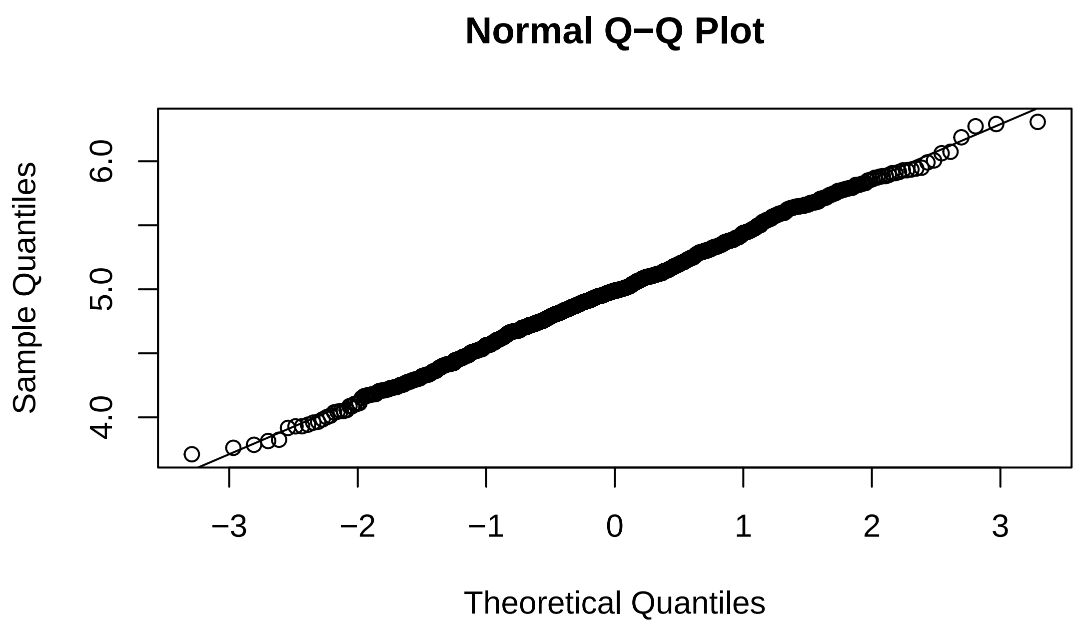
Solutions:
Let's explore different values for n. First up, let's look at $n = 10$.

```
# Set up some parameters
N <- 1000
df <- 5
n <- 10

# Get the samples and calculate the mean
chi_sample_10_1000 <- N %>%
  rerun(rchisq(n, df)) %>%
  map_dbl(mean)

qqnorm(chi_sample_10_1000)
qqline(chi_sample_10_1000)
```



Solutions:
Still a bit fat at the tails. Let's jump to $n = 50$.

```
# Set up some parameters
N <- 1000
df <- 5
n <- 50

# Get the samples and calculate the mean
chi_sample_50_1000 <- N %>%
  rerun(rchisq(n, df)) %>%
  map_dbl(mean)

qqnorm(chi_sample_50_1000)
qqline(chi_sample_50_1000)
```

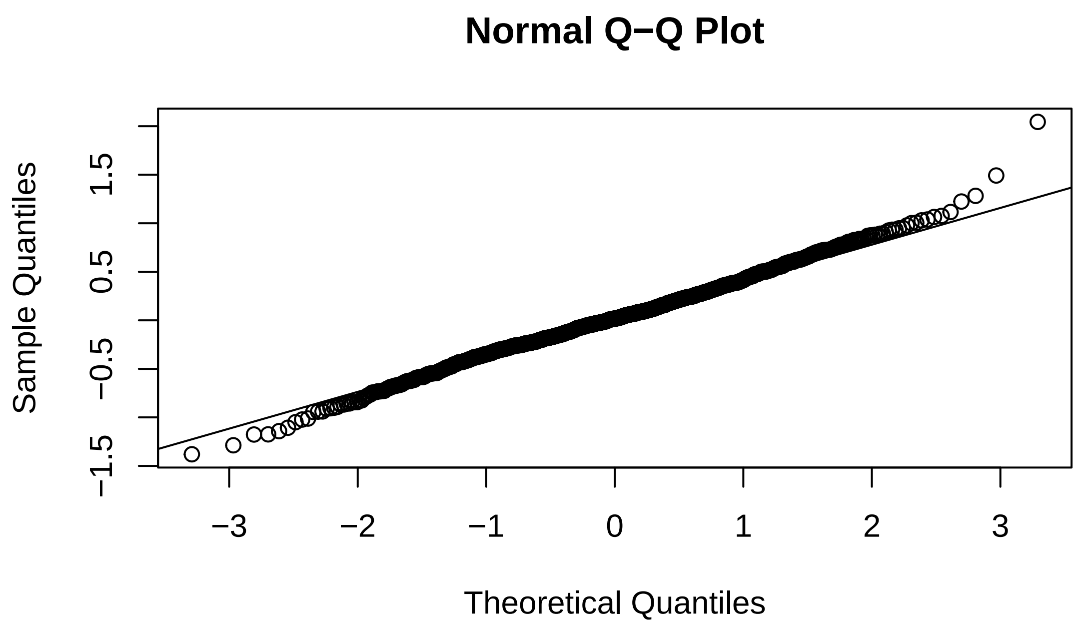


Solutions:
Much better, but still a little wavy. You will find that as you increase n larger, it will still be a little dodgy at the tails, but the bulk is approximately normal.

2. Look at the t_5 distribution. Explore the sampling distribution of the sample mean. When does it start to become normal?

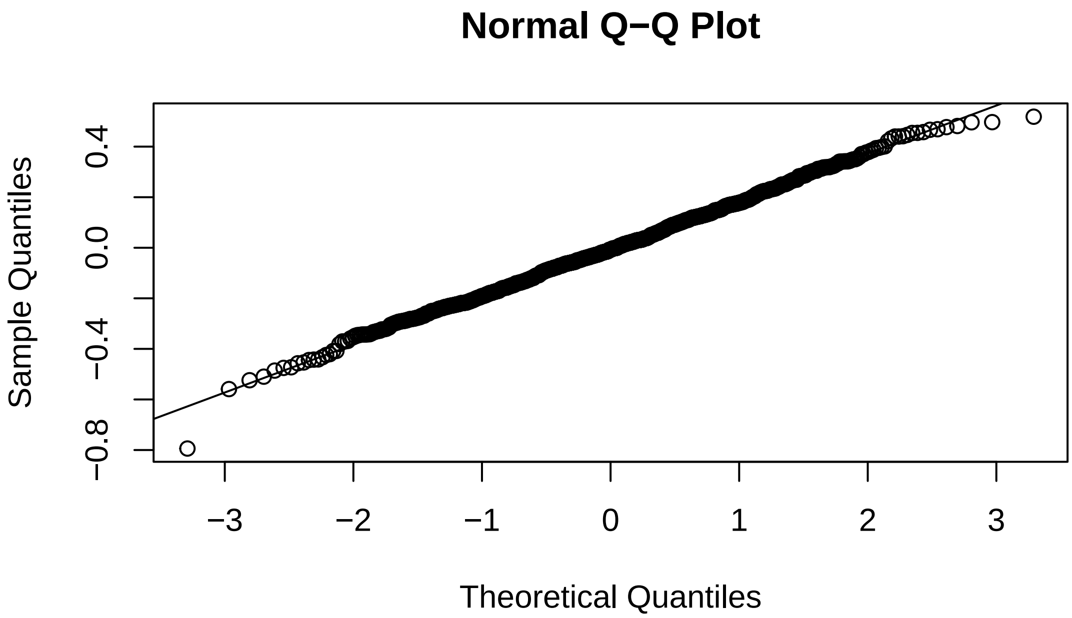
Solutions:
Let's start at $n = 10$.

```r
# Set up some parameters
N <- 1000
df <- 5
n <- 10

# Get the samples and calculate the mean
t_sample_10_1000 <- N %>%
  rerun(rt(n, df)) %>%
  map_dbl(mean)

qqnorm(t_sample_10_1000)
qqline(t_sample_10_1000)
```



Solutions:
Not too bad, but off on the tails. $n = 50$.

```
# Set up some parameters
N <- 1000
df <- 5
n <- 50

# Get the samples and calculate the mean
t_sample_50_1000 <- N %>%
  rerun(rt(n, df)) %>%
  map_dbl(mean)

qqnorm(t_sample_50_1000)
qqline(t_sample_50_1000)
```



Solutions:
Much better.

3. If you had a dataset with no knowledge of its distribution, how might you explore the sampling distribution of the sample mean?

Solutions:
I would use bootstrapping. If your original sample, say $x_1, x_2, \ldots, x_n$ is representative of the population, bootstrapping is the principal of treating this sample *as* the population. You then resample from your data *with replacement* and treat this as a new sample. You then use this to get an understanding of the distribution of your statistics from your original sample.