Plotting Distribution Fits

Tynan

2023-10-21

Plots

Made this on 2023-05-02, but it needs more work.

Made several different iterations of the distribution plots. This is still a WIP. Need to add normal distribution, was less interested in that at first.

Each iteration added either a new distribution or refined the code a bit to make it more compact.

Currently, the functions both generate a data set based on a common distribution, plot it, and then assess the fit of the data to the distribution.

Need to pull in some real-world data to demonstrate the use case of comparing the fit to different distributions.

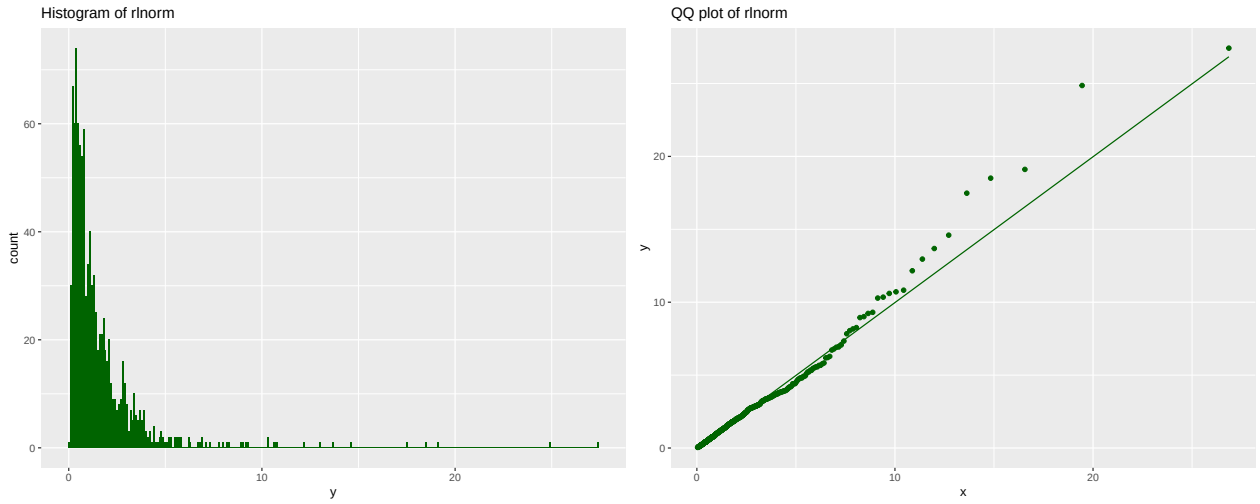
Also, should add some stat fit tests to complement the plot fit test.

Versions of Distribution Plots

- This version can fit log-normal, Weibull, and Beta distributions.
- The control flow determines plot settings. But it's a bit repetitive, could be more compact.
- NOTE: different params are needed for each distribution.

```
qqplot_and_histogram_v1 <- function(distribution_type, sample_size, params) {

  library(cowplot)
  library(glue)

  df <-  if (distribution_type == 'rlnorm') {
               as.data.frame( rlnorm(sample_size))

             } else if(distribution_type == 'rweibull') {
               as.data.frame( rweibull(sample_size, shape = params[1], scale = params[2]))

             } else if(distribution_type == 'rbeta') {
               as.data.frame( rbeta(sample_size, shape1 = params[1], shape2 = params[2]))

             } else if(distribution_type == 'rpois') {
               as.data.frame( rpois(sample_size, lambda = params[1]))

             }

  df <- df %>%
        mutate(distribution = as.character(distribution_type),
               x            = row_number()) %>%
        rename(y = 1) %>%
```

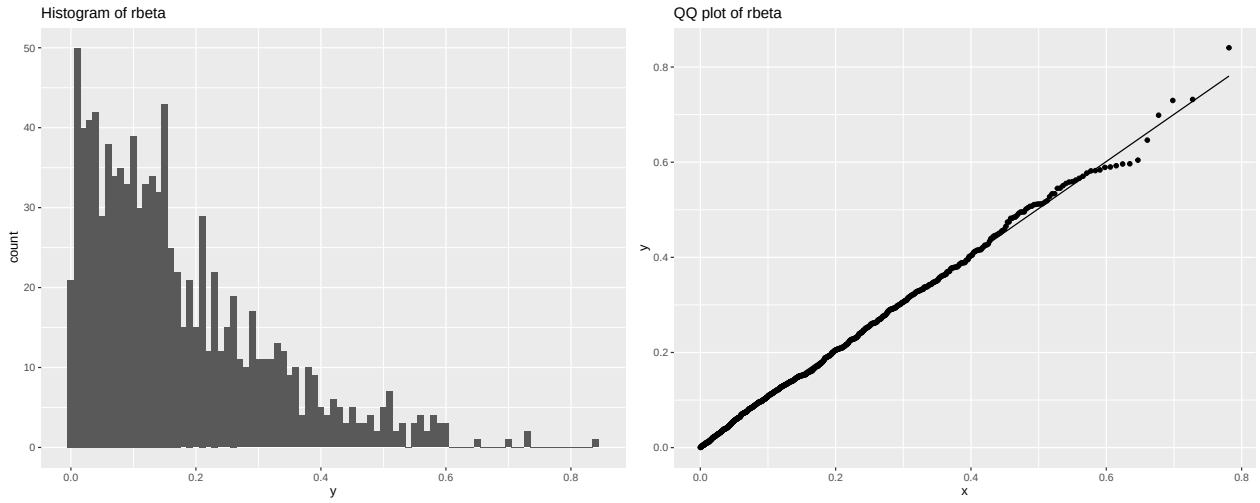
```
        dplyr::select(distribution, x, y)

  p1 <- df %>%
        ggplot(.) +
          aes(x=y) +
          if (distribution_type == 'rpois') {
          geom_bar()
          } else {
          geom_histogram(binwidth = if_else(distribution_type %in% c('rlnorm', 'rweibull'),  0.1, 0.01)
                         fill = 'darkgreen')
          }

    p1 <- p1 + labs(title = glue('Histogram of {distribution_type}'))

          if (distribution_type == 'rlnorm') {
           p2 <- df %>%
                 ggplot() +
                 aes(sample = y) +
                 stat_qq(distribution      = stats::qlnorm, color = 'darkgreen') +
                 stat_qq_line(distribution = stats::qlnorm, color = 'darkgreen')  +
                 labs(title = glue('QQ plot of {distribution_type}'))

          } else if(distribution_type %in% c('rweibull')) {
           p2 <- df %>%
                 ggplot() +
                 aes(sample = y) +
                 stat_qq(distribution      = stats::qweibull, dparams = list(shape = params[1], scale =
                 stat_qq_line(distribution = stats::qweibull, dparams = list(shape = params[1], scale =
                 labs(title = glue('QQ plot of {distribution_type}'))

          } else if(distribution_type %in% c('rbeta')) {
             p2 <- df %>%
                 ggplot() +
                 aes(sample = y) +
                 stat_qq(distribution      = stats::qbeta, dparams = list(shape1 = params[1], shape2 = p
                 stat_qq_line(distribution = stats::qbeta, dparams = list(shape1 = params[1], shape2 = p
                 labs(title = glue('QQ plot of {distribution_type}'))

          } else if(distribution_type %in% c('rpois')) {
             p2 <- df %>%
                 ggplot() +
                 aes(sample = y) +
                 stat_qq(distribution      = stats::qpois, dparams = list(lambda = params[1]), color =
                 stat_qq_line(distribution = stats::qpois, dparams = list(lambda = params[1]), color =
                 labs(title = glue('QQ plot of {distribution_type}'))

          }

  cowplot::plot_grid(p1, p2)

}

qqplot_and_histogram_v1('rlnorm', 1000, c(5, 5))
```



- Does not fit Poisson distribution, but combines the Weibull and Beta conditions into the same section.

```
qqplot_and_histogram_v2 <- function(distribution_type, sample_size, params) {

  library(cowplot)
  library(glue)

  df <-  if (distribution_type == 'rlnorm') {
              as.data.frame( rlnorm(sample_size))

            } else if(distribution_type == 'rweibull') {
              as.data.frame( rweibull(sample_size, shape = params[1], scale = params[2]))

            } else if(distribution_type == 'rbeta') {
              as.data.frame( rbeta(sample_size, shape1 = params[1], shape2 = params[2]))

            }

  df <- df %>%
        mutate(distribution = as.character(distribution_type),
               x            = row_number()) %>%
        rename(y = 1) %>%
        dplyr::select(distribution, x, y)

  p1 <- df %>%
        ggplot(.) +
          aes(x=y) +
          geom_histogram(binwidth = if_else(distribution_type %in% c('rlnorm', 'rweibull'),  0.1, 0.01))
          labs(title = glue('Histogram of {distribution_type}'))

         if (distribution_type == 'rlnorm') {
          p2 <- df %>%
                ggplot() +
                aes(sample = y) +
                stat_qq(distribution      = stats::qlnorm) +
                stat_qq_line(distribution = stats::qlnorm)  +
                labs(title = glue('QQ plot of {distribution_type}'))
```

```
        } else if(distribution_type %in% c('rweibull', 'rbeta')) {
         p2 <- df %>%
                ggplot() +
                aes(sample = y) +
                stat_qq(distribution = if(distribution_type == 'rweibull') {stats::qweibull} else {sta
                        dparams      = if(distribution_type == 'rweibull') {
                                                list(shape  = params[1], scale = params[2])}
                                           else { list(shape1 = params[1], shape2 = params[2]) }
                                         ) +
                stat_qq_line(distribution = if(distribution_type == 'rweibull') {stats::qweibull} else
                             dparams      = if(distribution_type == 'rweibull') {
                                                    list(shape  = params[1], scale = params[2])}
                                               else { list(shape1 = params[1], shape2 = params[2]) }
                                                  ) +
                labs(title = glue('QQ plot of {distribution_type}'))

        }

  cowplot::plot_grid(p1, p2)


}

qqplot_and_histogram_v2('rbeta', 1000, c(1, 5))
```

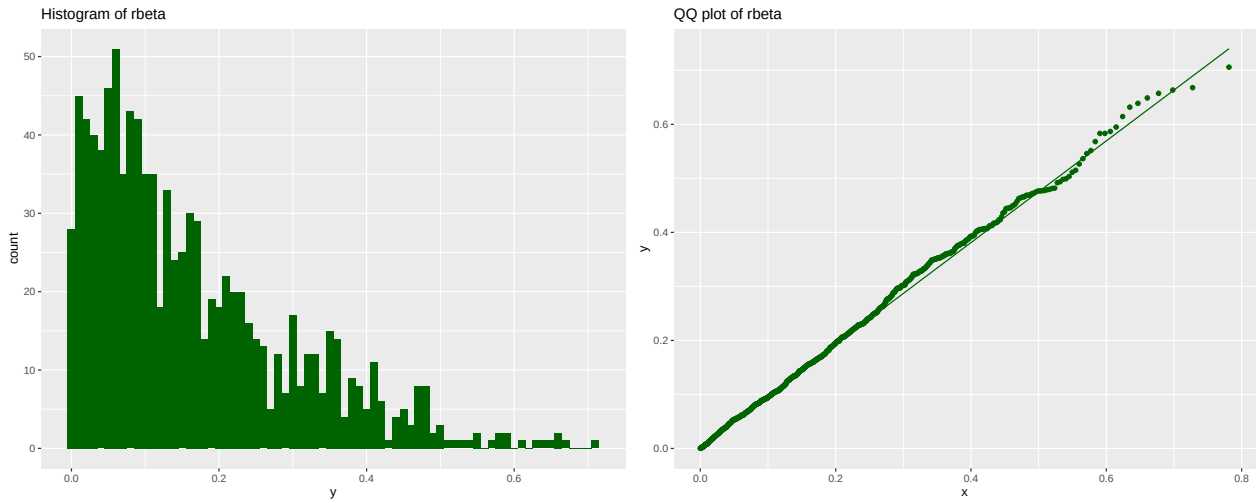


- This version is similar to qqplot_and_histogram_v1 above, need to come back and see what changed.

```
qqplot_and_histogram_v3 <- function(distribution_type, sample_size, params) {

  library(cowplot)
  library(glue)

  df <-  if (distribution_type == 'rlnorm') {
              as.data.frame( rlnorm(sample_size))

            } else if(distribution_type == 'rweibull') {
              as.data.frame( rweibull(sample_size, shape = params[1], scale = params[2]))
```

```
          } else if(distribution_type == 'rbeta') {
            as.data.frame( rbeta(sample_size, shape1 = params[1], shape2 = params[2]))

          } else if(distribution_type == 'rpois') {
            as.data.frame( rpois(sample_size, lambda = params[1]))

          }

df <- df %>%
      mutate(distribution = as.character(distribution_type),
             x            = row_number()) %>%
      rename(y = 1) %>%
      dplyr::select(distribution, x, y)

p1 <- df %>%
      ggplot(.) +
        aes(x=y) +
        if (distribution_type == 'rpois') {
        geom_bar()
        } else {
        geom_histogram(binwidth = if_else(distribution_type %in% c('rlnorm', 'rweibull'),  0.1, 0.01)
                       fill = 'darkgreen')
        }

  p1 <- p1 + labs(title = glue('Histogram of {distribution_type}'))

       if (distribution_type == 'rlnorm') {
        stat_qq_setting      <- stat_qq(distribution      = stats::qlnorm, color = 'darkgreen')
        stat_qq_line_setting <- stat_qq_line(distribution = stats::qlnorm, color = 'darkgreen')

       } else if(distribution_type %in% c('rweibull')) {
        stat_qq_setting      <- stat_qq(distribution      = stats::qweibull, dparams = list(shape = pa
        stat_qq_line_setting <- stat_qq_line(distribution = stats::qweibull, dparams = list(shape = pa

       } else if(distribution_type %in% c('rbeta')) {
        stat_qq_setting      <- stat_qq(distribution      = stats::qbeta, dparams = list(shape1 = para
        stat_qq_line_setting <- stat_qq_line(distribution = stats::qbeta, dparams = list(shape1 = para

       } else if(distribution_type %in% c('rpois')) {
        stat_qq_setting      <- stat_qq(distribution      = stats::qpois, dparams = list(lambda = para
        stat_qq_line_setting <- stat_qq_line(distribution = stats::qpois, dparams = list(lambda = para

       }

p2 <- df %>%
      ggplot() +
      aes(sample = y) +
      stat_qq_setting +
      stat_qq_line_setting +
      labs(title = glue('QQ plot of {distribution_type}'))

cowplot::plot_grid(p1, p2)
```

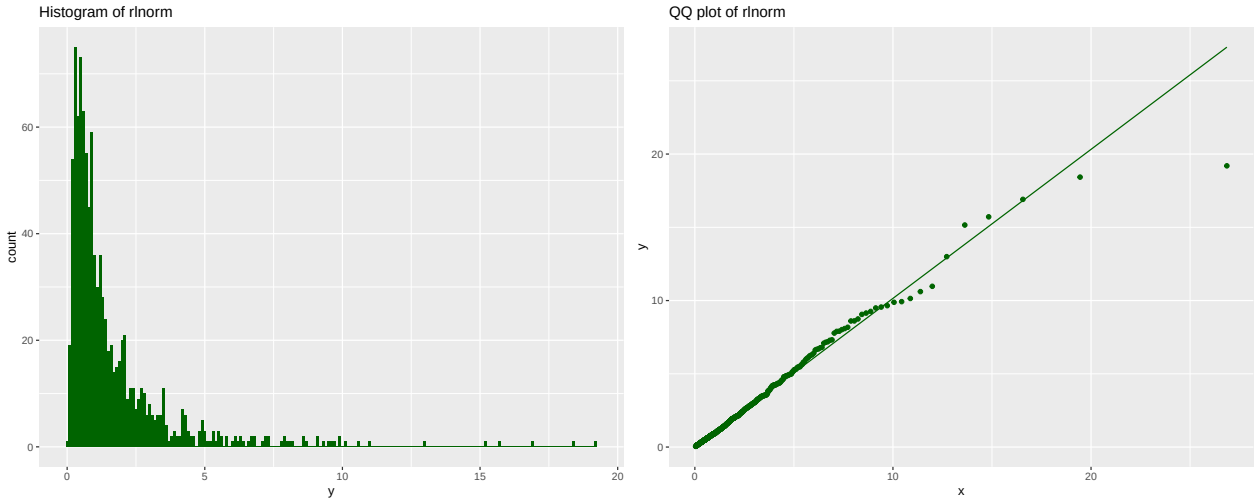
```
}

qqplot_and_histogram_v3('rbeta', 1000, c(1, 5))
```



- Final version
- The control flow for how the example data.frame is made has been updated.
- A list called 'dist_args' is created to pass the params to a much more compact plotting section.

```
# final version
qqplot_and_histogram <- function(distribution_type, sample_size, params) {

  library(cowplot)
  library(glue)

   if (distribution_type == 'rlnorm') {
      df        <- as.data.frame( rlnorm(sample_size))
      dist_args =  list(dist_func = stats::qlnorm,
                        dparams   = NULL)

    } else if(distribution_type == 'rweibull') {
      df        <- as.data.frame( rweibull(sample_size, shape = params[1], scale = params[2]))
      dist_args =  list(dist_func = stats::qweibull,
                        dparams   = list(shape  = params[1], scale = params[2]))

    } else if(distribution_type == 'rbeta') {
      df        <- as.data.frame( rbeta(sample_size, shape1 = params[1], shape2 = params[2]))
      dist_args =  list(dist_func = stats::qbeta,
                        dparams   = list(shape1 = params[1], shape2 = params[2]))

    } else if(distribution_type == 'rpois') {
      df        <- as.data.frame( rpois(sample_size, lambda = params[1]))
      dist_args =  list(dist_func = stats::qpois,
                        dparams   = list(lambda = params[1]))

    }

  df <- df %>%
```

```
        mutate(distribution = as.character(distribution_type),
               x            = row_number()) %>%
        rename(y = 1) %>%
        dplyr::select(distribution, x, y)

  p1 <- df %>%
        ggplot(.) +
          aes(x=y) +
          if (distribution_type == 'rpois') {
          geom_bar(fill = 'darkgreen')
          } else {
          geom_histogram(binwidth = if_else(distribution_type %in% c('rlnorm', 'rweibull'),  0.1, 0.01)
                         fill = 'darkgreen')
          }

  p1 <- p1 + labs(title = glue('Histogram of {distribution_type}'))

  p2 <- df %>%
        ggplot() +
        aes(sample = y) +
        stat_qq(distribution      = dist_args$dist_func, dparams = dist_args$dparams, color = 'darkgreen
        stat_qq_line(distribution = dist_args$dist_func, dparams = dist_args$dparams, color = 'darkgreen
        labs(title = glue('QQ plot of {distribution_type}'))

  cowplot::plot_grid(p1, p2)

}

qqplot_and_histogram('rlnorm', 1000)
```

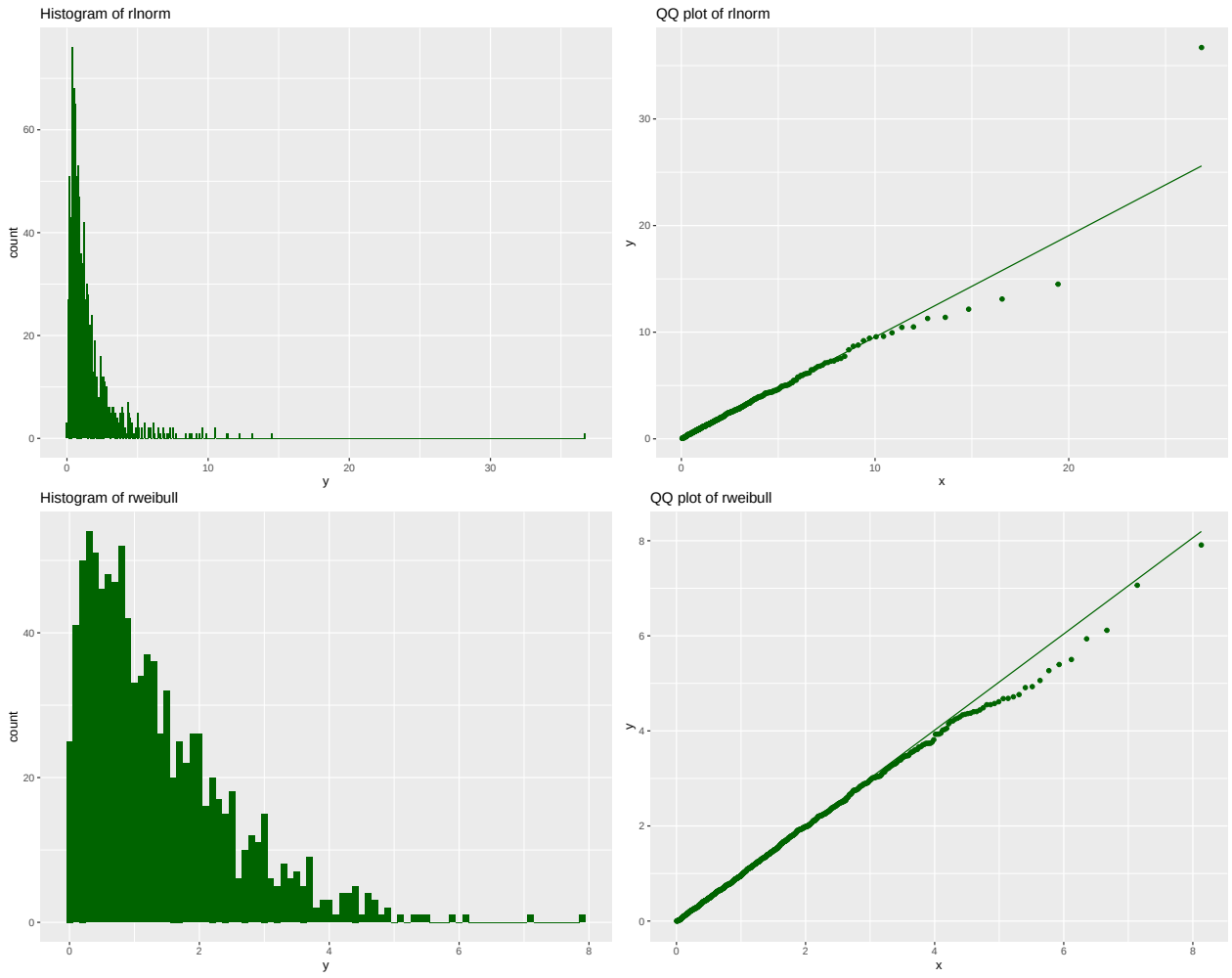


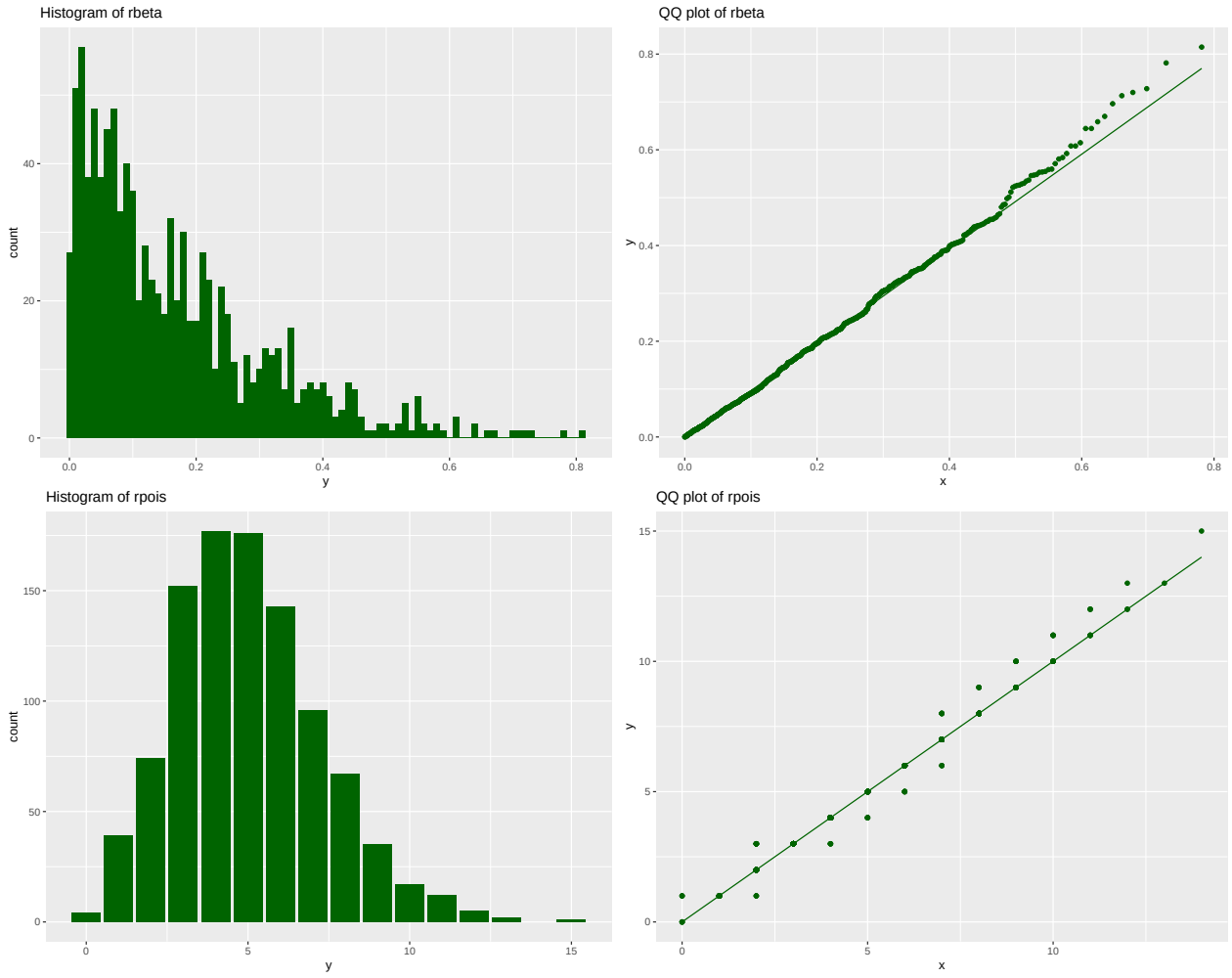
Final Version

- The final version of qqplot_and_histogram() is used when looping over different distributions.

```
params_list <- list(c('rlnorm'),
                    c('rweibull', 1.2, 1.5),
                    c('rbeta',  1, 5),
                    c('rpois', 5))
```

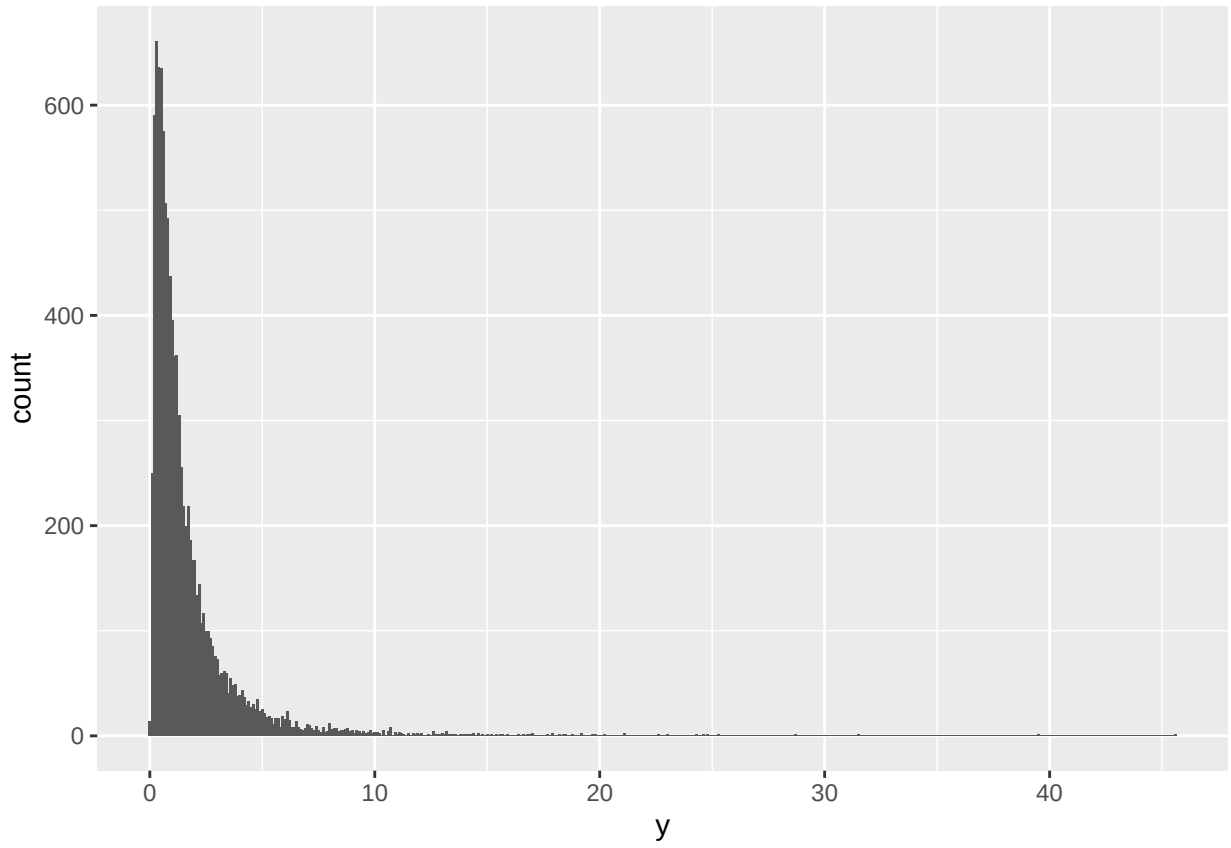
```
for (i in seq_along( params_list)) {

  dist   <- params_list[[i]][1]
  params <- as.numeric(params_list[[i]][2:3])

  print(qqplot_and_histogram(dist, 1000, params))

}
```



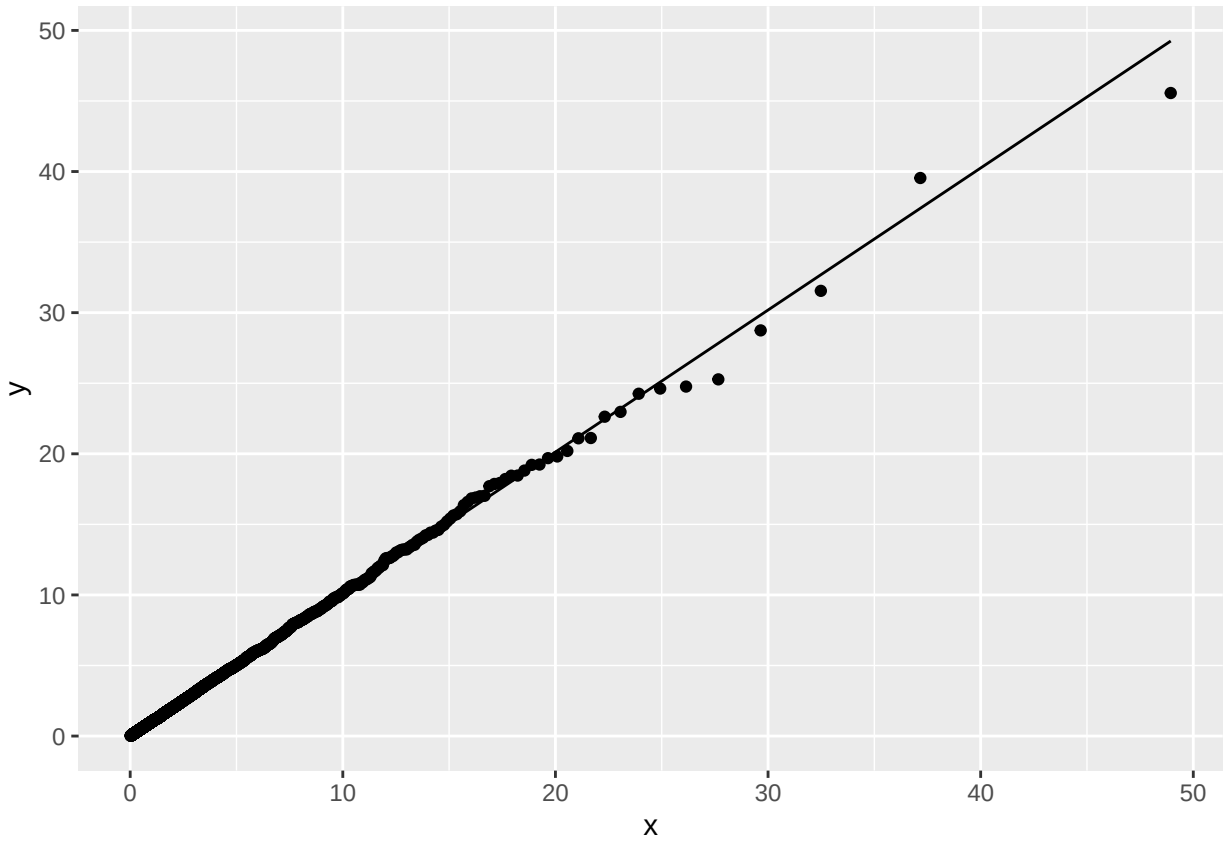


Preliminary Individul Distribution Plots

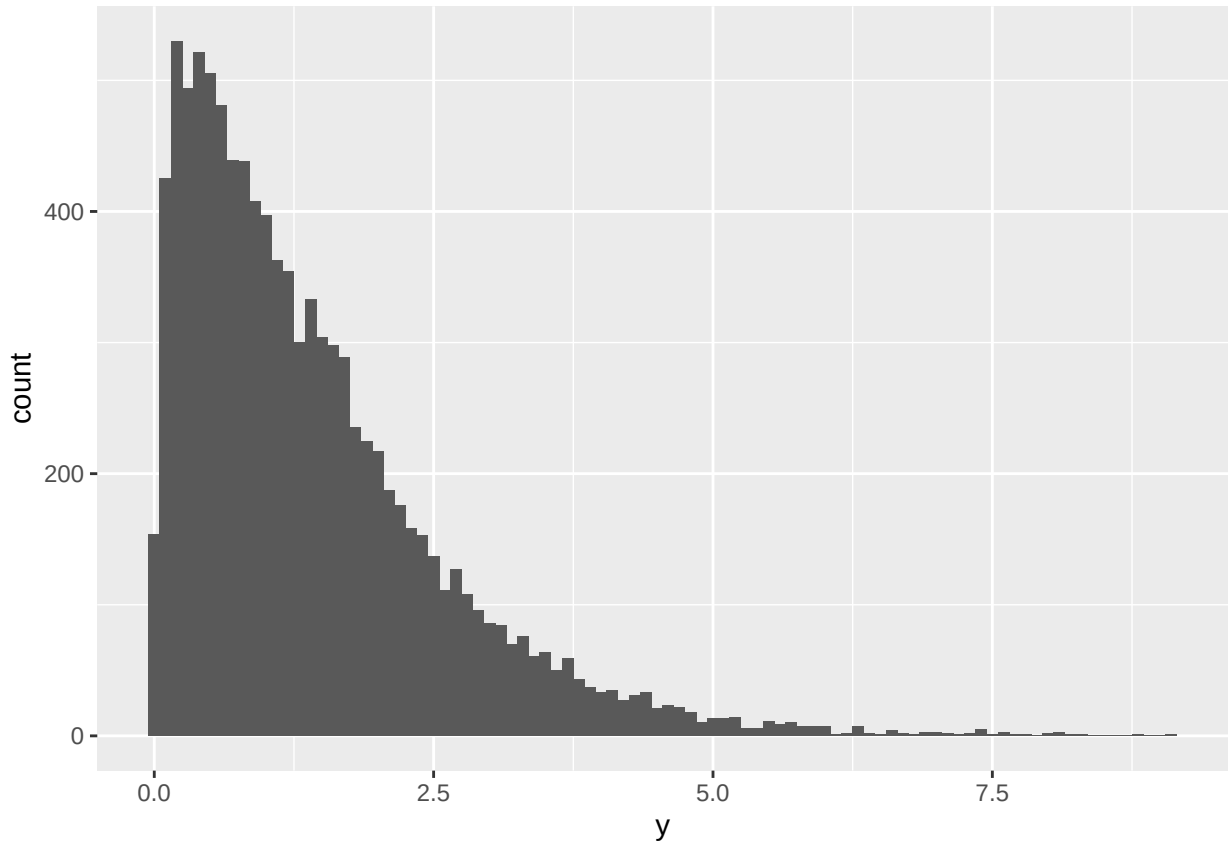
```
y_dlnorm <- as.data.frame(rlnorm(10000)) %>%
              mutate(x = row_number()) %>%
              rename(y = 1)

y_dlnorm %>%
ggplot(.) +
  aes(x=y) +
  geom_histogram(binwidth=0.1)
```



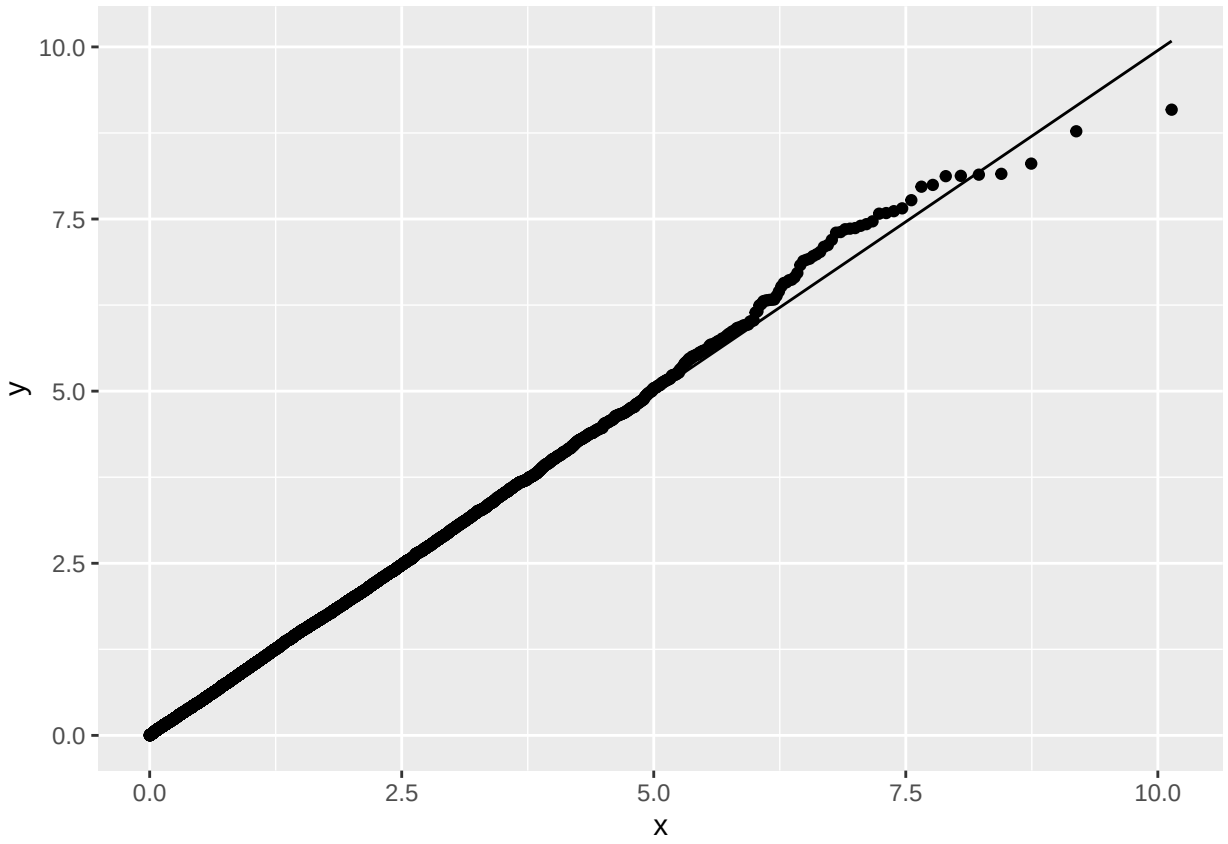

```
y_dlnorm %>%
ggplot(., aes(sample = y)) +
   stat_qq(distribution = stats::qlnorm) +
   stat_qq_line(distribution = stats::qlnorm)
```



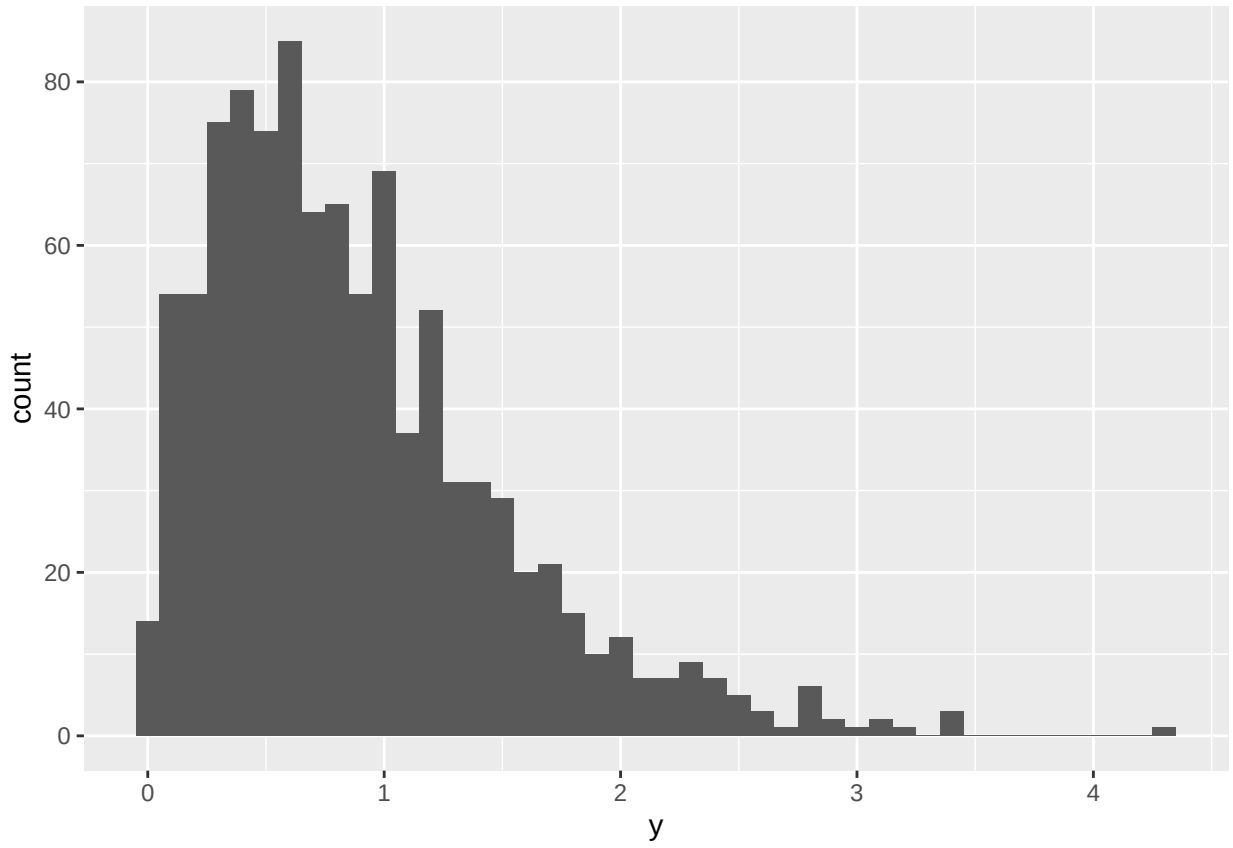

```
weibull <- as.data.frame(rweibull(10000, shape = 1.2, scale = 1.5)) %>%
              mutate(x = row_number()) %>%
              rename(y = 1)

weibull %>%
ggplot(.) +
  aes(x=y) +
  geom_histogram(binwidth=0.1)
```



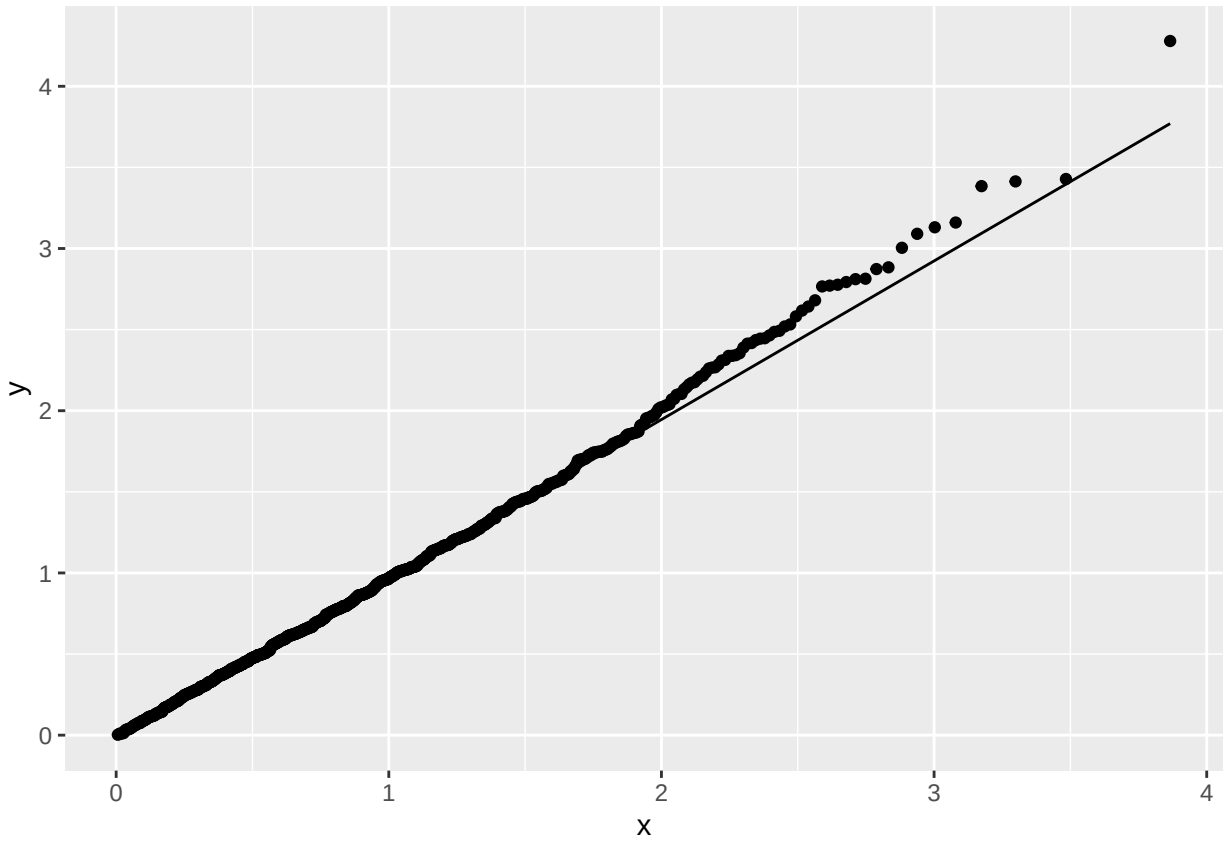

```
weibull %>%
ggplot() +
   aes(sample = y) +
   stat_qq(distribution      = stats::qweibull, dparams = list(shape = 1.2, scale = 1.5)) +
   stat_qq_line(distribution = stats::qweibull, dparams = list(shape = 1.2, scale = 1.5))
```



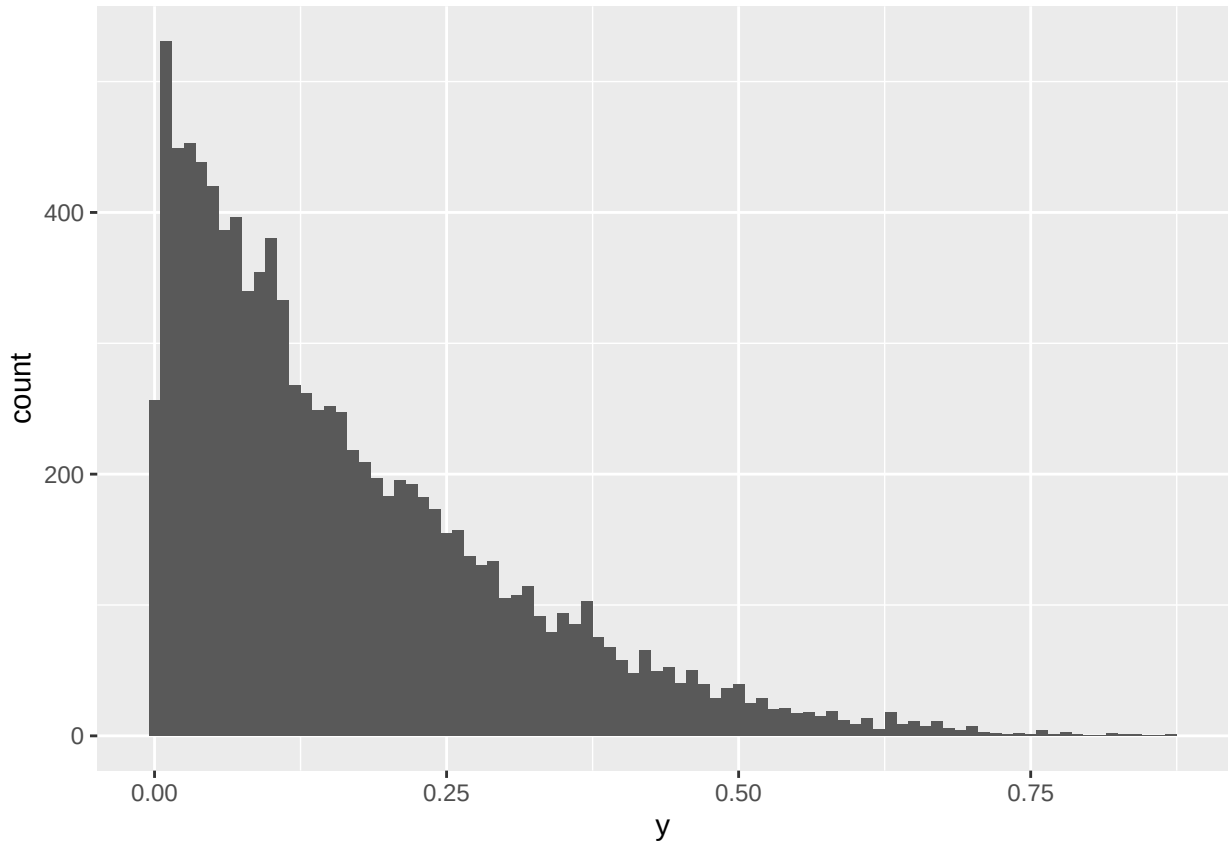

```
weibull_2 <- as.data.frame(rweibull(1000, shape = 1.5, scale = 1)) %>%
                mutate(x = row_number()) %>%
                rename(y = 1)

weibull_2 %>%
    ggplot() +
    aes(x = y) +
    geom_histogram(binwidth = 0.1)
```



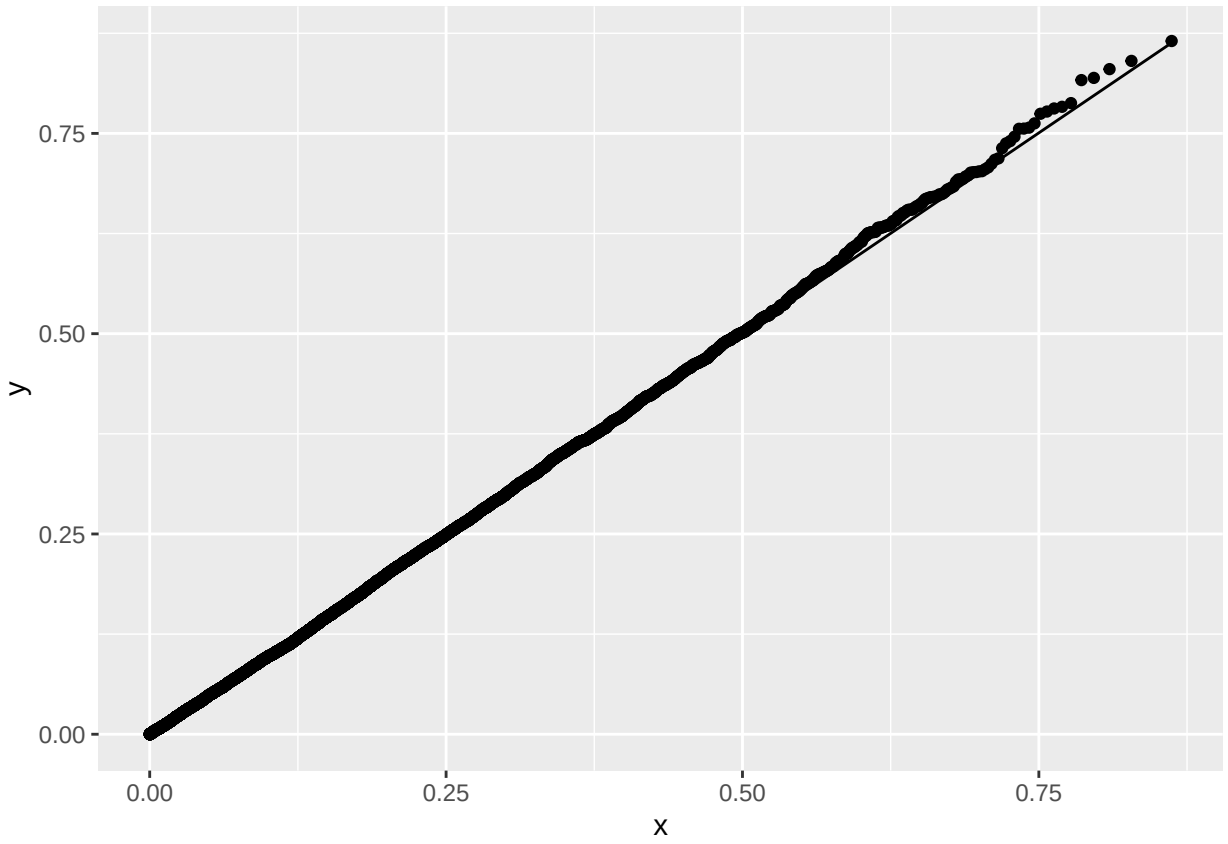

```
weibull_2 %>%
  ggplot() +
  aes(sample = y) +
  stat_qq(distribution      = stats::qweibull, dparams = list(shape = 1.5, scale = 1)) +
  stat_qq_line(distribution = stats::qweibull, dparams = list(shape = 1.5, scale = 1))
```



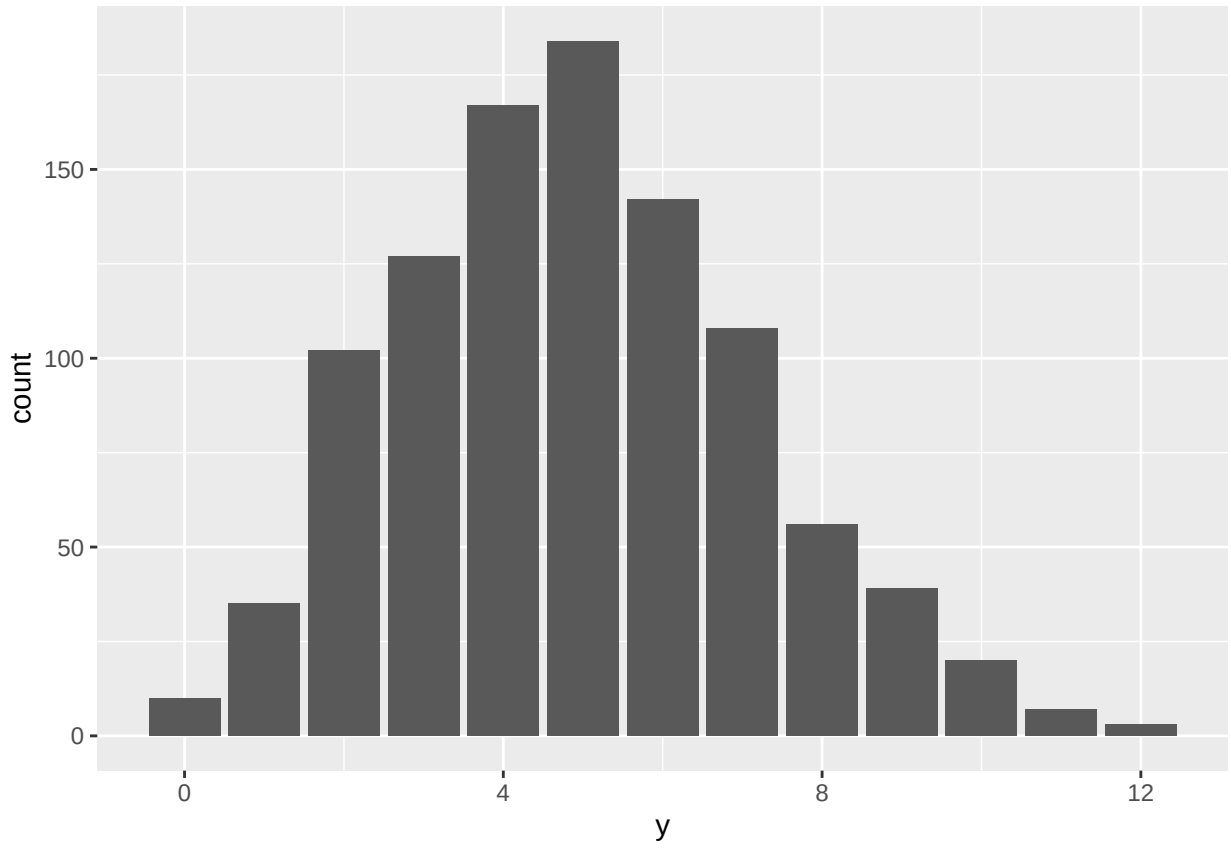

```
beta <- as.data.frame(rbeta(10000, shape1 = 1, shape2 = 5)) %>%
              mutate(x = row_number()) %>%
              rename(y = 1)

beta %>%
    ggplot() +
    aes(x = y) +
    geom_histogram(binwidth = 0.01)
```



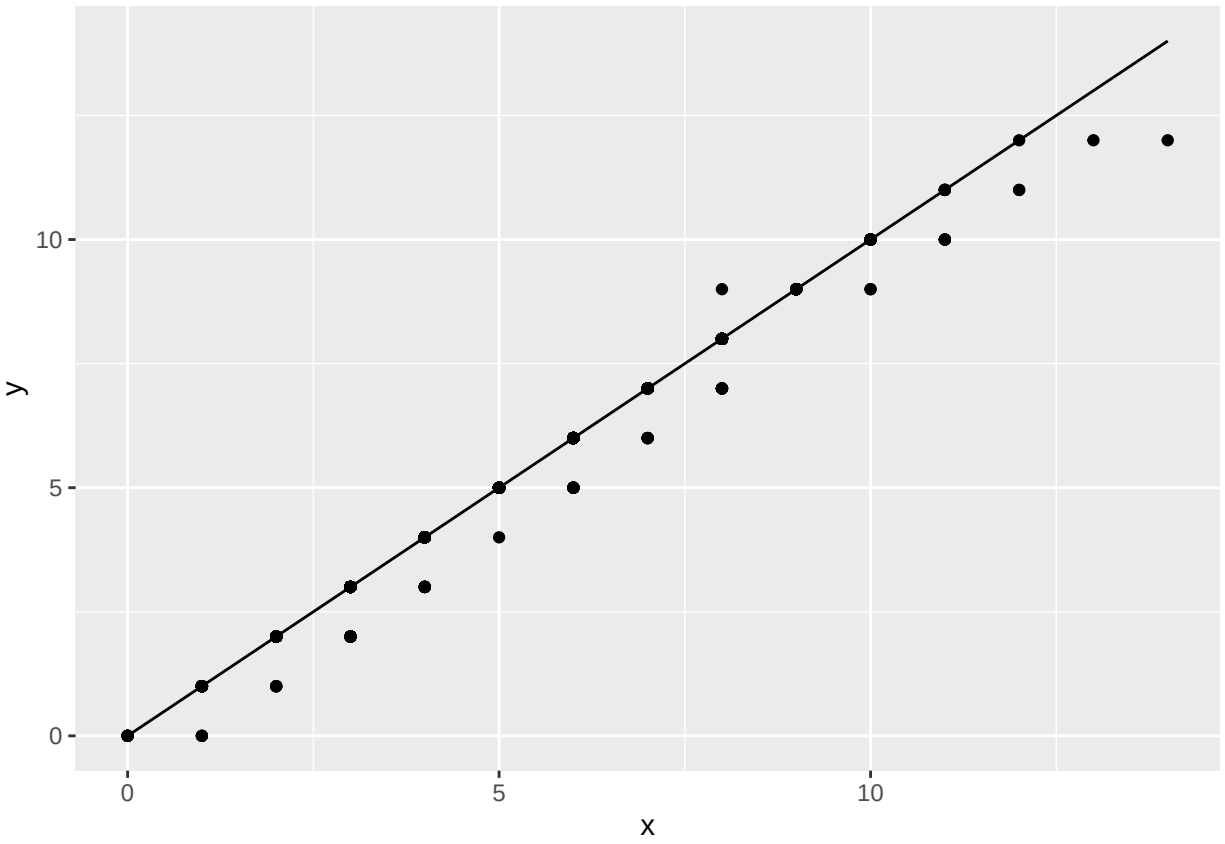

```
beta %>%
ggplot() +
  aes(sample = y) +
  stat_qq(distribution      = stats::qbeta, dparams = list(shape1 = 1, shape2 = 5)) +
  stat_qq_line(distribution = stats::qbeta, dparams = list(shape1 = 1, shape2 = 5))
```




```
poisson <- as.data.frame(rpois(1000, lambda = 5))  %>%
              mutate(x = row_number()) %>%
              rename(y = 1)

poisson %>%
    ggplot() +
    aes(x = y) +
    geom_bar()
```




```
poisson %>%
  ggplot() +
    aes(sample = y) +
    stat_qq(distribution      = stats::qpois, dparams = list(lambda = 5)) +
    stat_qq_line(distribution = stats::qpois, dparams = list(lambda = 5))
```



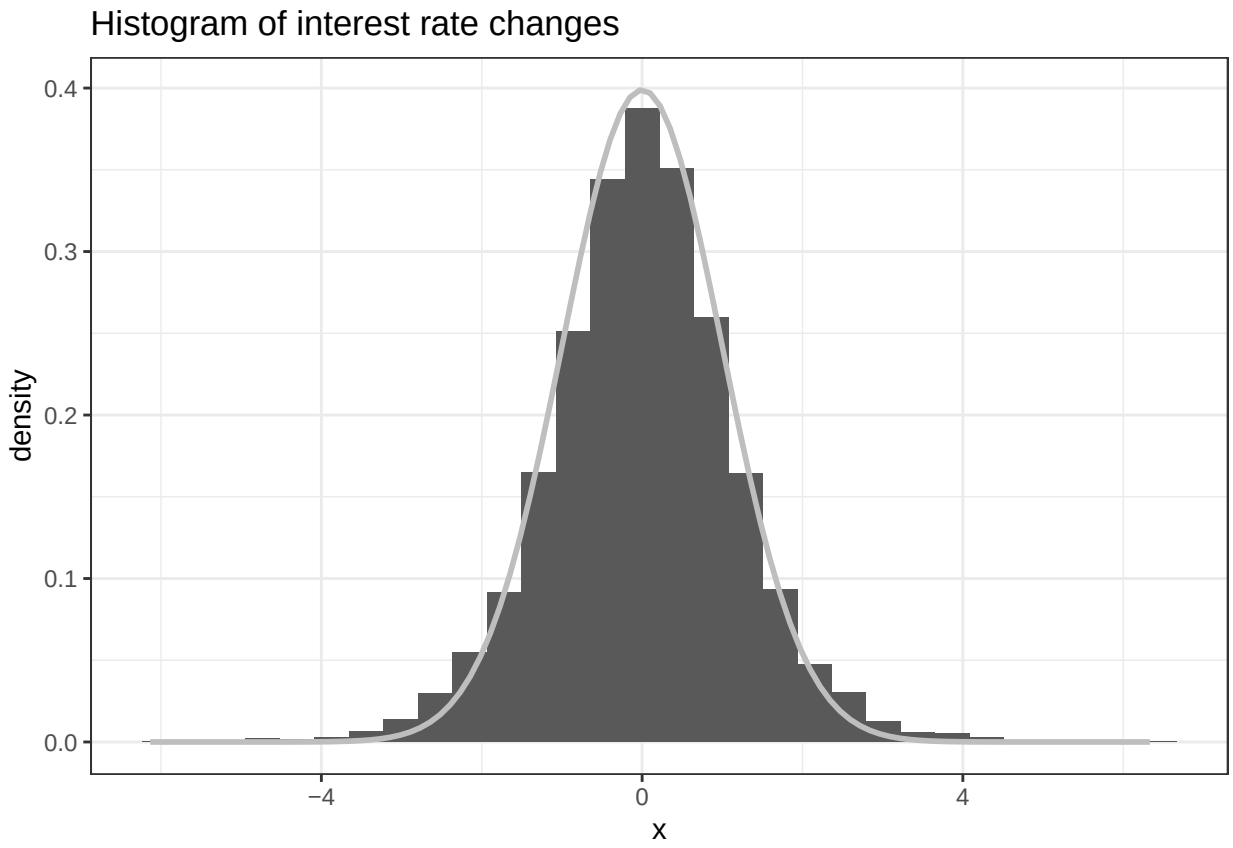
Misc Plots

- These are other distributions plots, could incorporate them into the main function, but need to get the qq line plot set-up.
- This is for Student t Distribution.

```
data <- data.frame(x=rt(10000, df=7))

data %>%
ggplot(aes(x=x)) +
  geom_histogram(aes(y = ..density..)) +
  stat_function(fun       = dnorm,
                linewidth = 1,
                color     = 'gray',
                args      = list()
                ) +
  labs(title="Histogram of interest rate changes") +
  theme_bw()
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

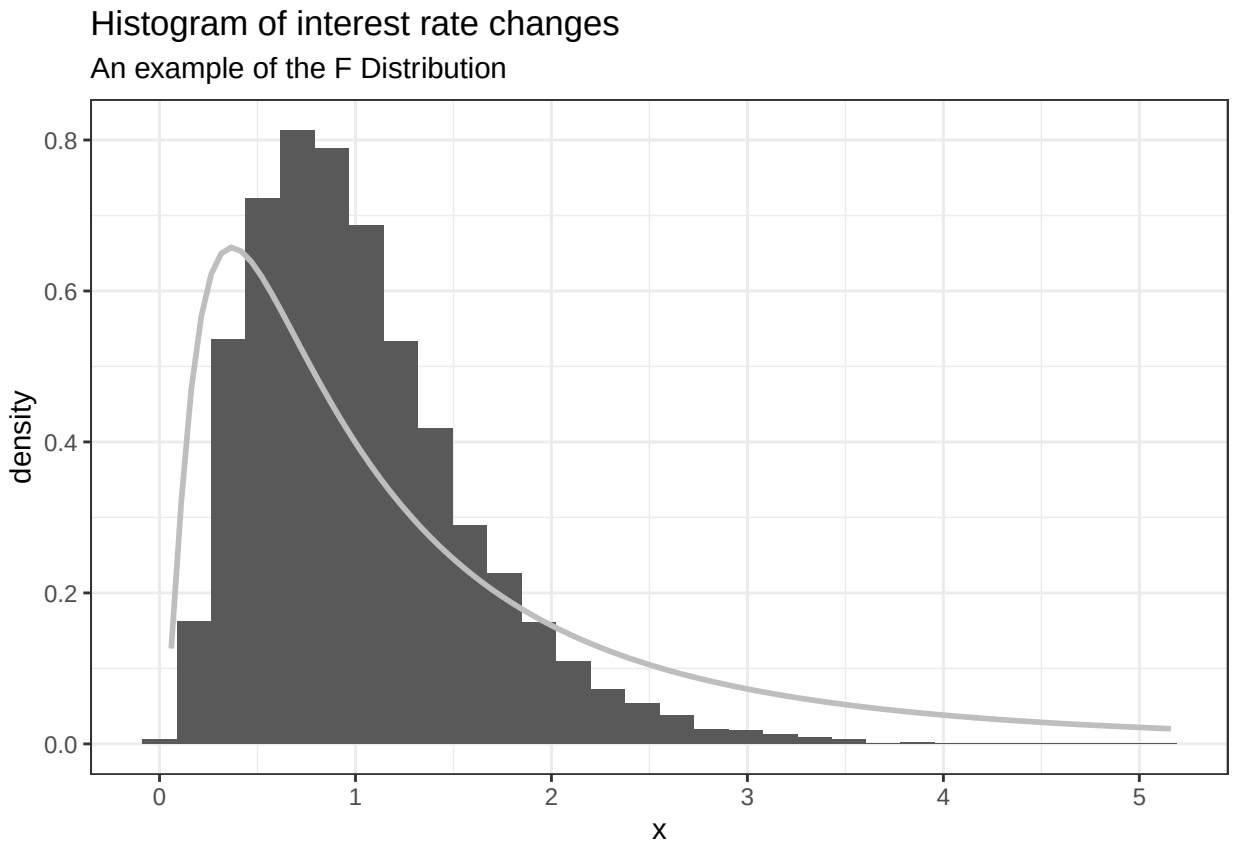
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- For The F Distribution

```
data <- data.frame(x=rf(10000, df1=7, df2=120))

data %>%
ggplot(aes(x=x)) +
  geom_histogram(aes(y = ..density..)) +
  stat_function(fun       = dlnorm,
                linewidth = 1,
                color     = 'gray',
                args      = list()) +
  labs(title="Histogram of interest rate changes",
       subtitle = "An example of the F Distribution") +
  theme_bw()
```

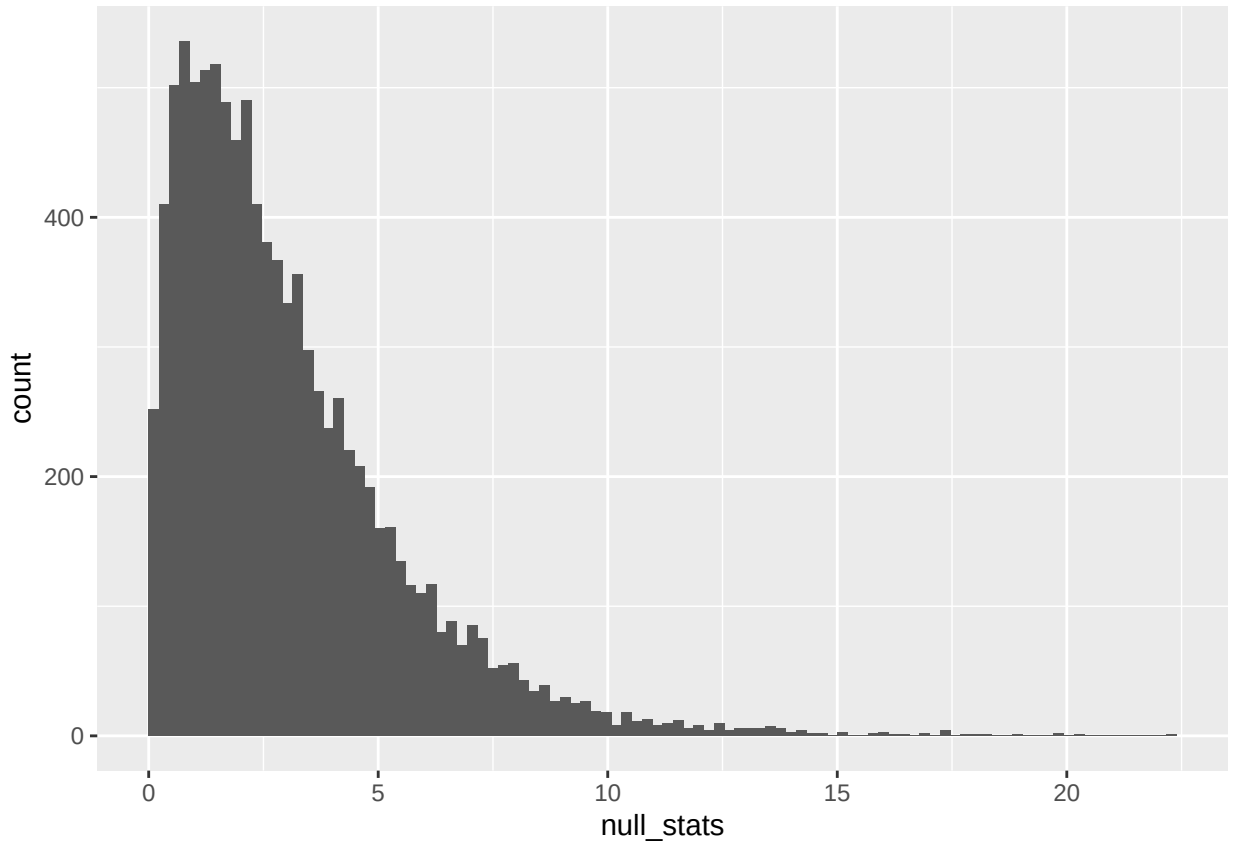
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



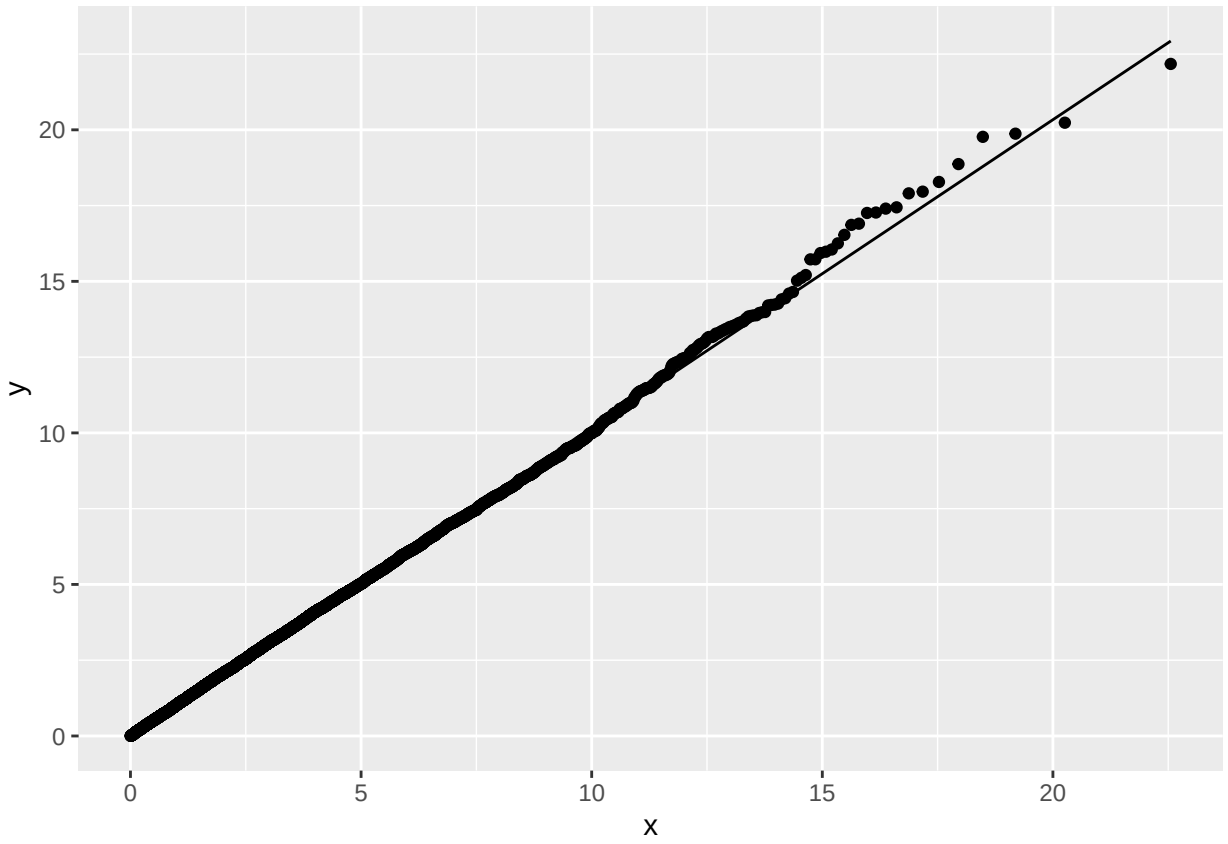
Chi Square Distribution Plot

```
oestat = function(o, e){
  sum( (e-o)^2/e )
}

set.seed(1)
B = 10000
# here we pick an arbitrary length / not the same as for Celegans
n = 2847

expected = rep(n/4 ,4)
oenull = replicate(B, oestat(e=expected, o=rmultinom(1,size = n, prob = rep(1/4,4))))
```

```
ggplot(data.frame(null_stats = oenull)) +
  geom_histogram(aes(x = null_stats), bins = 100, boundary=0)
```




```
ggplot(data.frame(stat = oenull), aes(sample = stat)) +
  stat_qq(distribution = stats::qchisq, dparams = list(df = 3)) +
  stat_qq_line(distribution = stats::qchisq, dparams = list(df = 3))
```


20
15
10
5
0
y
0
5
10
15
20
x