Homework 2

2023-10-09

PROBLEM 2

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

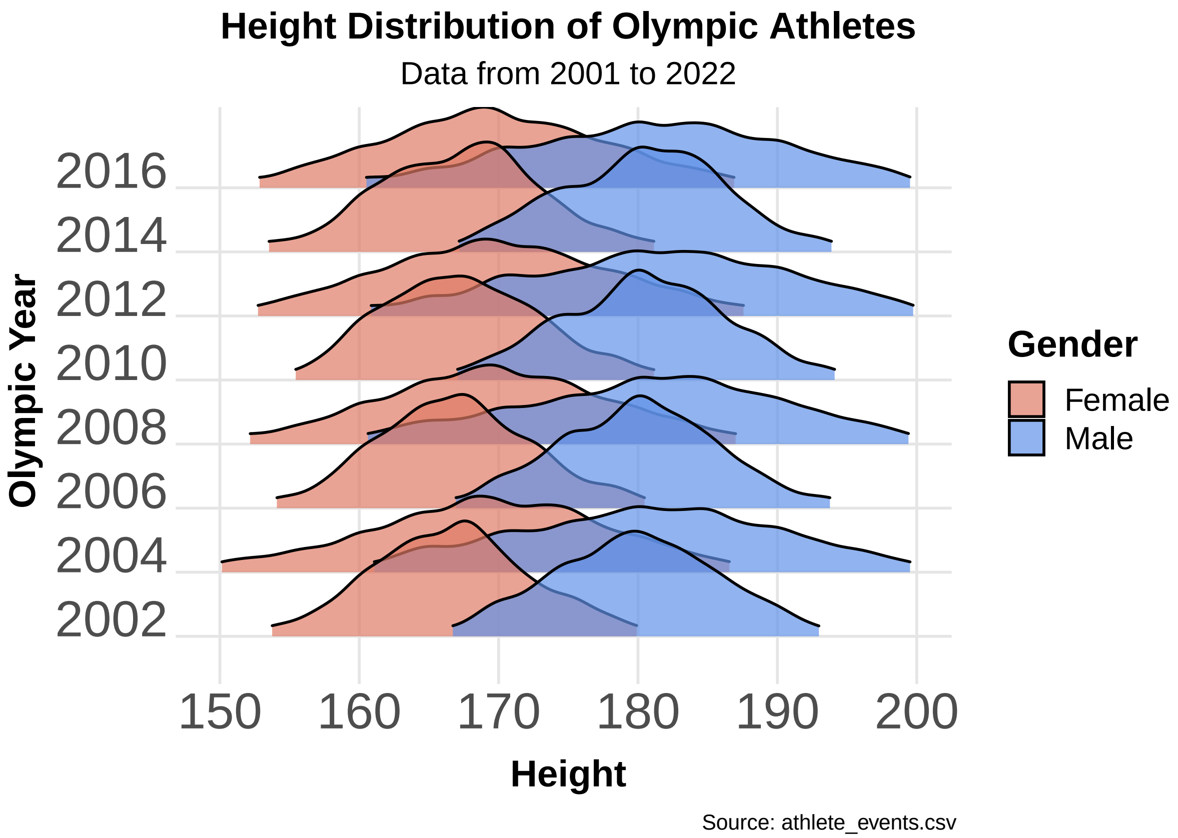
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## v readr     2.1.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(tidyr)
library(ggridges)

df_athlete <- read.csv('athlete_events.csv')

# Filtering columns Sex, Age, Height, Weight, Year
df_athlete <- df_athlete %>%
  dplyr::select(Sex, Age, Height, Weight, Year)

# Removing empty lines (NA values)
df_athlete <- df_athlete %>%
  drop_na()

# Filtering by year (year > 2000 and year < 2023)
df_athlete <- df_athlete %>%
  filter(Year > 2000 & Year < 2023)
```

```
# Creating a ridgeline plot for Heights
df_athlete %>%
  ggplot(aes(x = Height, y = Year, fill = Sex)) +
  geom_density_ridges(aes(y = as.factor(Year)), alpha = 0.7, rel_min_height = 0.09) + # Making the dens
  scale_fill_manual(values = c("#e07b65", "#6192e8"),
    name = bquote(bold("Gender")),
    labels = c("Female", "Male")) +
  scale_x_continuous(
    breaks = seq(150, 200, by = 10),   # Setting x-axis tick marks
    limits = c(150, 200),              # Setting x-axis limits
  ) +
  labs(
    title = "Height Distribution of Olympic Athletes",
    subtitle = "Data from 2001 to 2022",
    caption = "Source: athlete_events.csv",
    x = "Height",
    y = "Olympic Year"
  ) + theme_ridges() +
    theme(
    axis.title.y = element_text(hjust = 0.5, face = "bold"),  # Center, bold y-axis title
    plot.title = element_text(hjust = 0.5),  # Center plot title
    plot.subtitle = element_text(hjust = 0.5),  # Center plot subtitle
    axis.title.x = element_text(hjust = 0.5, face = "bold"),  # Center, bold x-axis title
    axis.text.x = element_text(size = 19, color = "#4d4d4d"),  # Adjusting font size and color for x-ax
    axis.text.y = element_text(size = 19, color = "#4d4d4d"),  # Adjusting font size and color for y-ax
    plot.caption = element_text(size = 7.8, hjust = 1.01) # Adjusting font size and location
  )
```

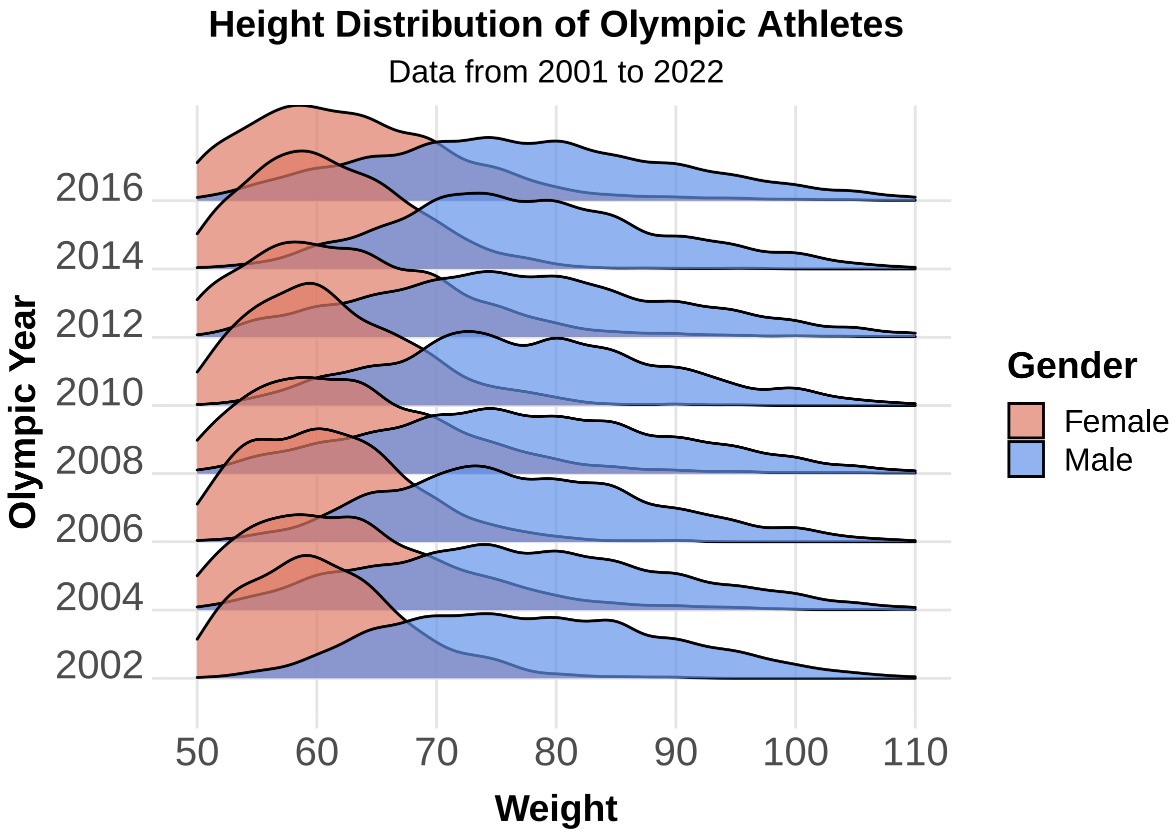
```
## Picking joint bandwidth of 1.27
```

```
## Warning: Removed 1391 rows containing non-finite values
## (`stat_density_ridges()`).
```



```
# Creating a ridgeline plot for Weights
df_athlete %>%
  ggplot(aes(x = Weight, y = Year, fill = Sex)) +
  geom_density_ridges(aes(y = as.factor(Year)), alpha = 0.7) + # Making the density plots transparent
  scale_fill_manual(values = c("#e07b65", "#6192e8"),
    name = bquote(bold("Gender")),
    labels = c("Female", "Male")) +
  scale_x_continuous(
    breaks = seq(50, 120, by = 10),   # Setting x-axis tick marks
    limits = c(50, 110)             # Setting x-axis limits
  ) +
  labs(
    title = "Height Distribution of Olympic Athletes",
    subtitle = "Data from 2001 to 2022",
    x = "Weight",
    y = "Olympic Year"
  ) + theme_ridges() +
    theme(
    axis.title.y = element_text(hjust = 0.5, face = "bold"),  # Center, bold y-axis title
    plot.title = element_text(hjust = 0.5),  # Center plot title
    plot.subtitle = element_text(hjust = 0.5),  # Center plot subtitle
    axis.title.x = element_text(hjust = 0.5, face = "bold"),  # Center, bold x-axis title
    axis.text.x = element_text(size = 15, color = "#4d4d4d"),  # Adjusting font size and color for x-ax
    axis.text.y = element_text(size = 15, color = "#4d4d4d")  # Adjusting font size and color for y-axi
  )
```

```
## Picking joint bandwidth of 1.69
## Warning: Removed 3873 rows containing non-finite values
## (`stat_density_ridges()`).
```



What is the difference ? We see that for the Weight we have right skewed distribution.

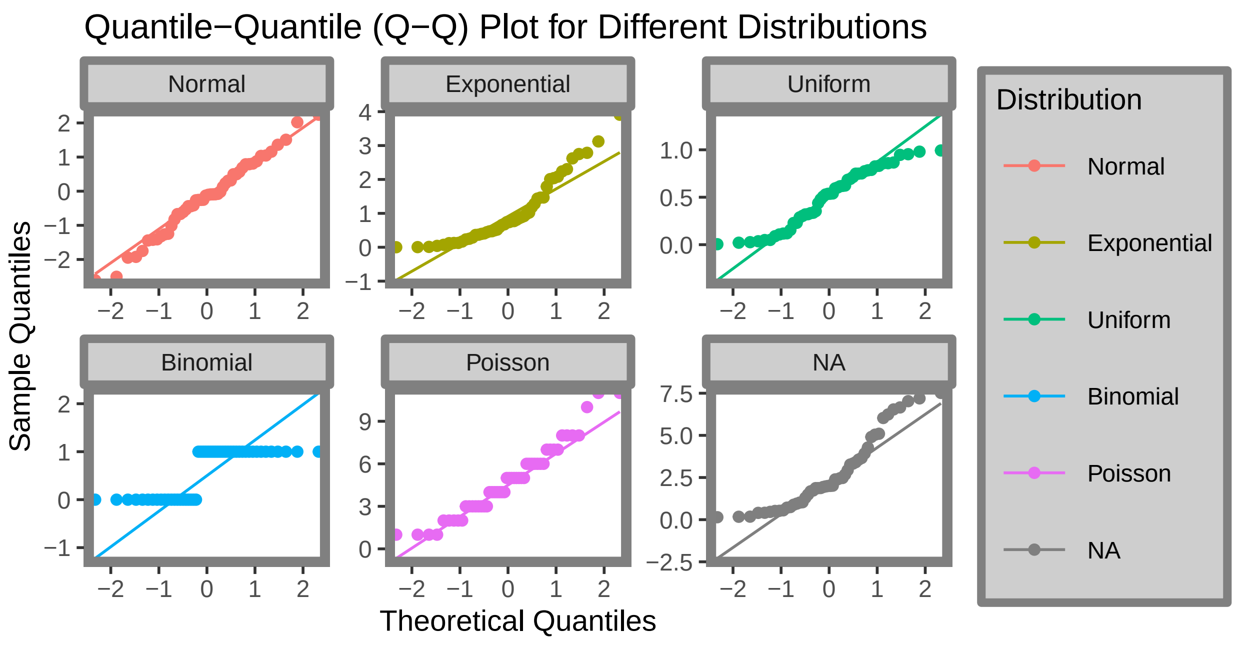
PROBLEM 3

```r
# Setting a seed for reproducibility
set.seed(2013)
# Generating random data from different distributions
n <- 50 # Sample size
normal_data <- rnorm(n)
exponential_data <- rexp(n, rate = 1)
uniform_data <- runif(n)
chi_squared_data <- rchisq(n, df = 3)
binomial_data <- rbinom(n, size = 1, prob = 0.5)
poisson_data <- rpois(n, lambda = 5)
```

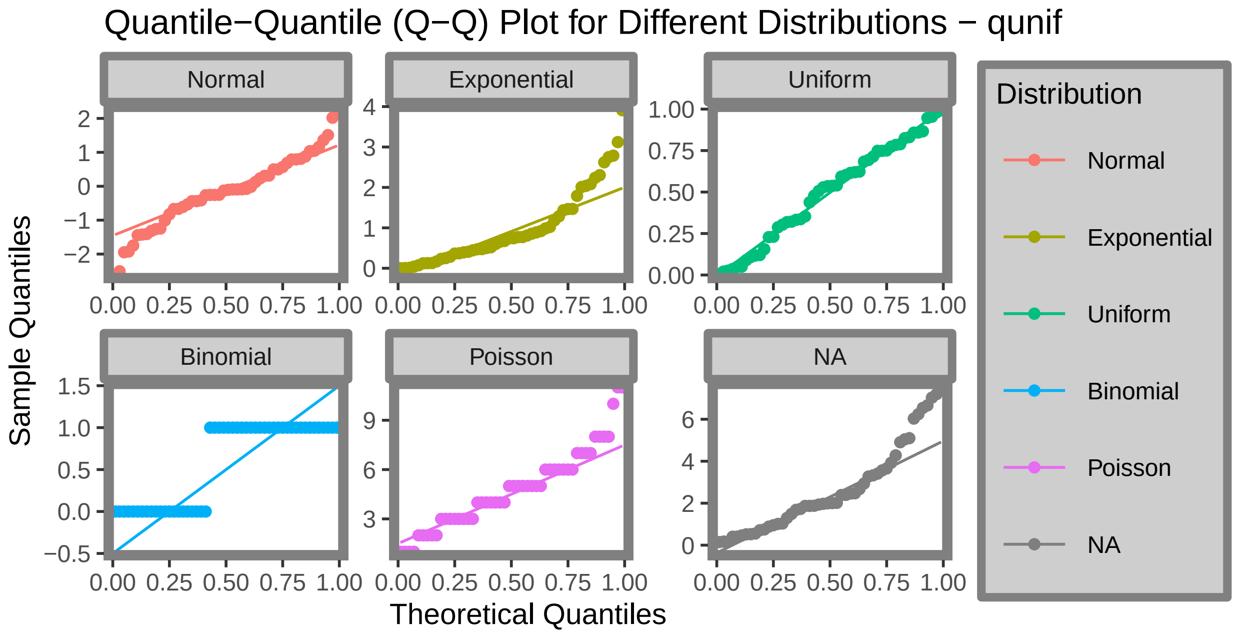
```r
# Combining into a single data frame with labels
df <- data.frame(
  Data = c(normal_data, exponential_data, uniform_data, chi_squared_data, binomial_data, poisson_data),
  Distribution = rep(c("Normal", "Exponential", "Uniform", "Chi-Squared", "Binomial", "Poisson"), each =
)

# Specifying the order of facets
df$Distribution <- factor(df$Distribution, levels = c("Normal", "Exponential", "Uniform", "Chi.Squared"

# Creating (Q-Q) plot
```

```r
ggplot(df, aes(sample = Data, color = Distribution)) +
  geom_qq() + stat_qq_line() +
  facet_wrap(~ Distribution, scales = "free") +
  labs(
    title = "Quantile-Quantile (Q-Q) Plot for Different Distributions",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  guides(color = guide_legend(title = "Distribution")) +
  theme(
    panel.background = element_blank(),  # Removing background
    panel.grid.major = element_blank(),  # Removing major grid lines
    panel.grid.minor = element_blank(),   # Removing minor grid lines
    panel.border = element_rect(fill = NA, color = "#808080", size = 3.5), # Coloring panel borders
    strip.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling, colori
    strip.clip = "off", # Making strip borders rounded
    legend.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling the le
    legend.key = element_rect(fill = "#cecece"), # Filling key backgrounds
    legend.key.size = unit(2,"lines") # Making legend background longer
  ) +  theme(aspect.ratio = 0.74) # Getting a bit rectangular shape for the panels
```

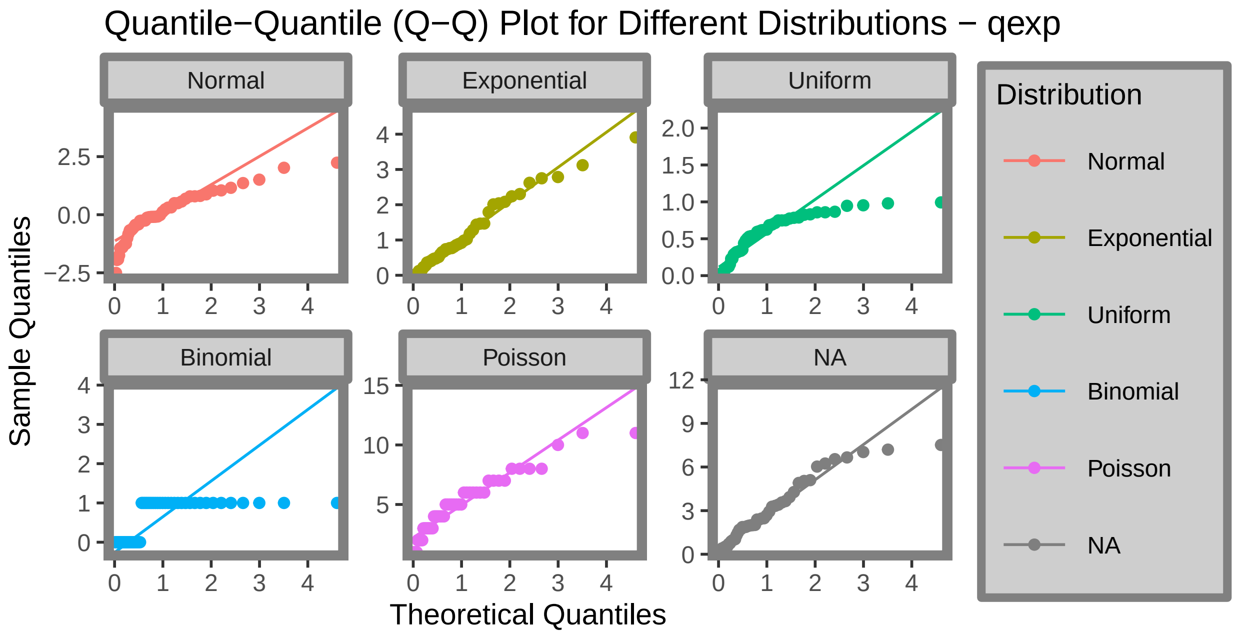
```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



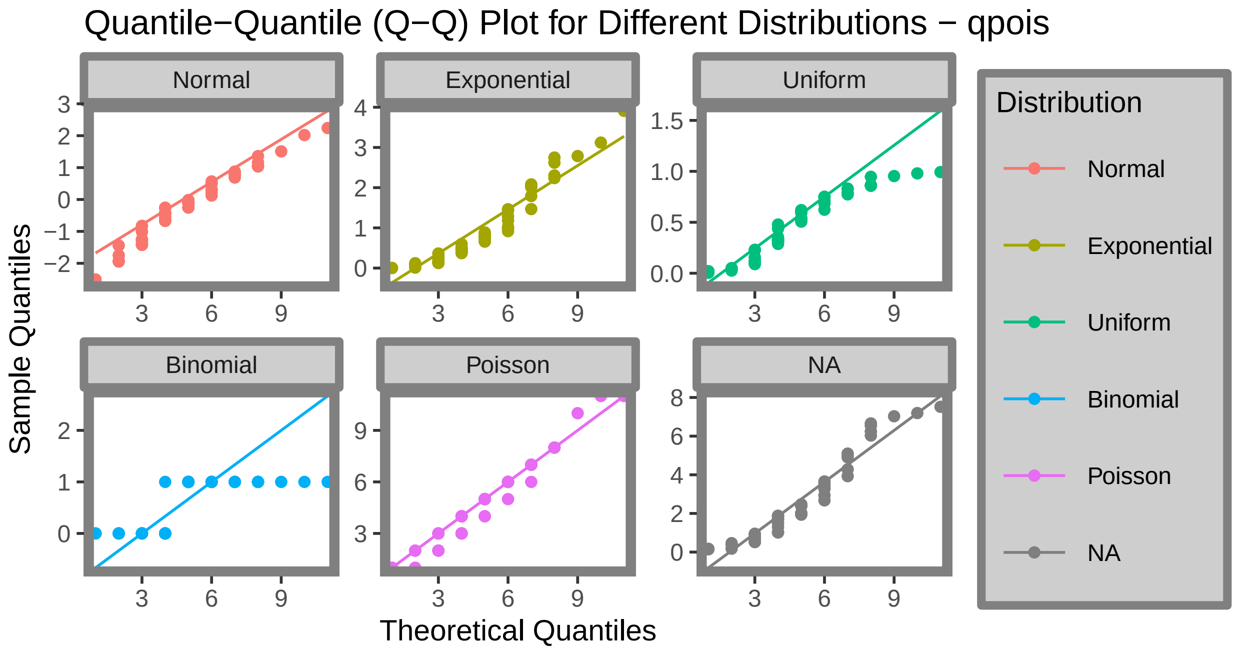
```r
# Combining into a single data frame with labels
df <- data.frame(
  Data = c(normal_data, exponential_data, uniform_data, chi_squared_data, binomial_data, poisson_data),
  Distribution = rep(c("Normal", "Exponential", "Uniform", "Chi-Squared", "Binomial", "Poisson"), each
)

# Specifying the order of facets
df$Distribution <- factor(df$Distribution, levels = c("Normal", "Exponential", "Uniform", "Chi.Squared"

# Creating (Q-Q) plot
ggplot(df, aes(sample = Data, color = Distribution)) +
  geom_qq(distribution = qunif) + stat_qq_line(distribution = qunif) +
  facet_wrap(~ Distribution, scales = "free") +
  labs(
    title = "Quantile-Quantile (Q-Q) Plot for Different Distributions - qunif",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  guides(color = guide_legend(title = "Distribution")) +
  theme(
    panel.background = element_blank(),  # Removing background
    panel.grid.major = element_blank(),  # Removing major grid lines
    panel.grid.minor = element_blank(),   # Removing minor grid lines
    panel.border = element_rect(fill = NA, color = "#808080", size = 3.5), # Coloring panel borders
    strip.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling, colori
    strip.clip = "off", # Making strip borders rounded
    legend.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling the le
    legend.key = element_rect(fill = "#cecece"), # Filling key backgrounds
    legend.key.size = unit(2,"lines") # Making legend background longer
  ) +  theme(aspect.ratio = 0.74) # Getting a bit rectangular shape for the panels
```



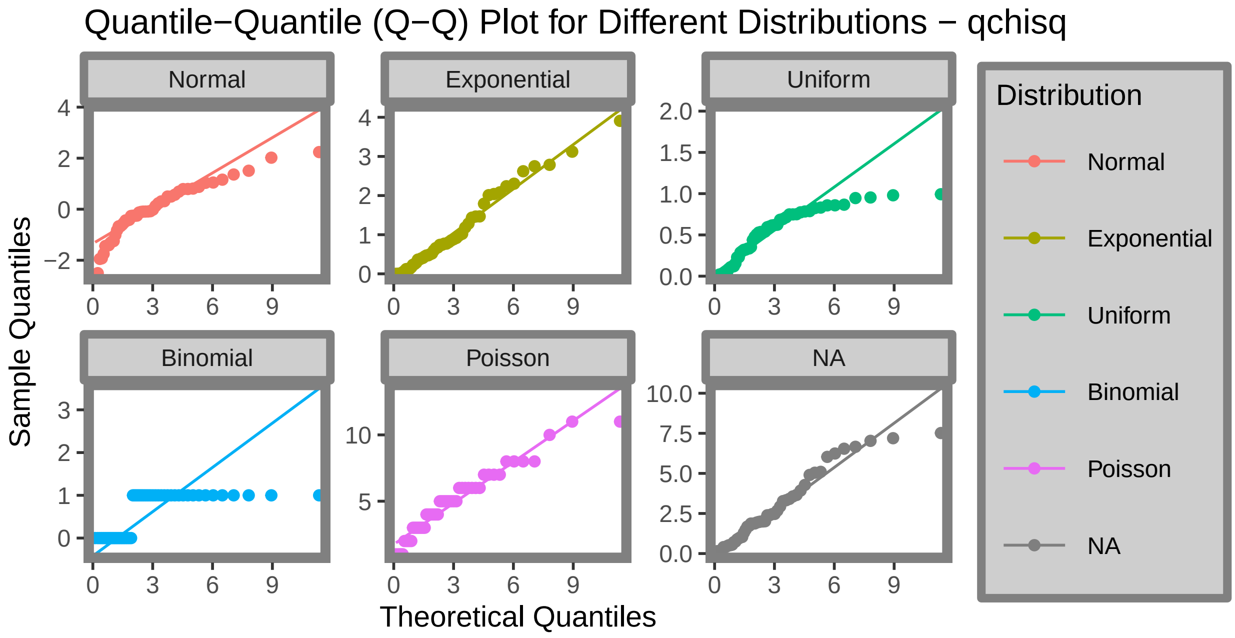
```r
# Combining into a single data frame with labels
df <- data.frame(
  Data = c(normal_data, exponential_data, uniform_data, chi_squared_data, binomial_data, poisson_data),
  Distribution = rep(c("Normal", "Exponential", "Uniform", "Chi-Squared", "Binomial", "Poisson"), each =
)

# Specifying the order of facets
df$Distribution <- factor(df$Distribution, levels = c("Normal", "Exponential", "Uniform", "Chi.Squared"

# Creating (Q-Q) plot
ggplot(df, aes(sample = Data, color = Distribution)) +
  geom_qq(distribution = qexp) + stat_qq_line(distribution = qexp) +
  facet_wrap(~ Distribution, scales = "free") +
  labs(
    title = "Quantile-Quantile (Q-Q) Plot for Different Distributions - qexp",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  guides(color = guide_legend(title = "Distribution")) +
  theme(
    panel.background = element_blank(),  # Removing background
    panel.grid.major = element_blank(),  # Removing major grid lines
    panel.grid.minor = element_blank(),   # Removing minor grid lines
    panel.border = element_rect(fill = NA, color = "#808080", size = 3.5), # Coloring panel borders
    strip.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling, colori
    strip.clip = "off", # Making strip borders rounded
    legend.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling the le
    legend.key = element_rect(fill = "#cecece"), # Filling key backgrounds
    legend.key.size = unit(2,"lines") # Making legend background longer
  ) +  theme(aspect.ratio = 0.74) # Getting a bit rectangular shape for the panels
```



```
# Combining into a single data frame with labels
df <- data.frame(
  Data = c(normal_data, exponential_data, uniform_data, chi_squared_data, binomial_data, poisson_data),
  Distribution = rep(c("Normal", "Exponential", "Uniform", "Chi-Squared", "Binomial", "Poisson"), each =
)

# Specifying the order of facets
df$Distribution <- factor(df$Distribution, levels = c("Normal", "Exponential", "Uniform", "Chi.Squared"

# Creating (Q-Q) plot
ggplot(df, aes(sample = Data, color = Distribution)) +
  geom_qq(distribution = qpois, dparams = list(lambda = 5)) +
  stat_qq_line(distribution = qpois, dparams = list(lambda = 5)) +
  facet_wrap(~ Distribution, scales = "free") +
  labs(
    title = "Quantile-Quantile (Q-Q) Plot for Different Distributions - qpois",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  guides(color = guide_legend(title = "Distribution")) +
  theme(
    panel.background = element_blank(),  # Removing background
    panel.grid.major = element_blank(),  # Removing major grid lines
    panel.grid.minor = element_blank(),   # Removing minor grid lines
    panel.border = element_rect(fill = NA, color = "#808080", size = 3.5), # Coloring panel borders
    strip.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling, colori
    strip.clip = "off", # Making strip borders rounded
    legend.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling the le
    legend.key = element_rect(fill = "#cecece"), # Filling key backgrounds
    legend.key.size = unit(2,"lines") # Making legend background longer
  ) +  theme(aspect.ratio = 0.74) # Getting a bit rectangular shape for the panels
```

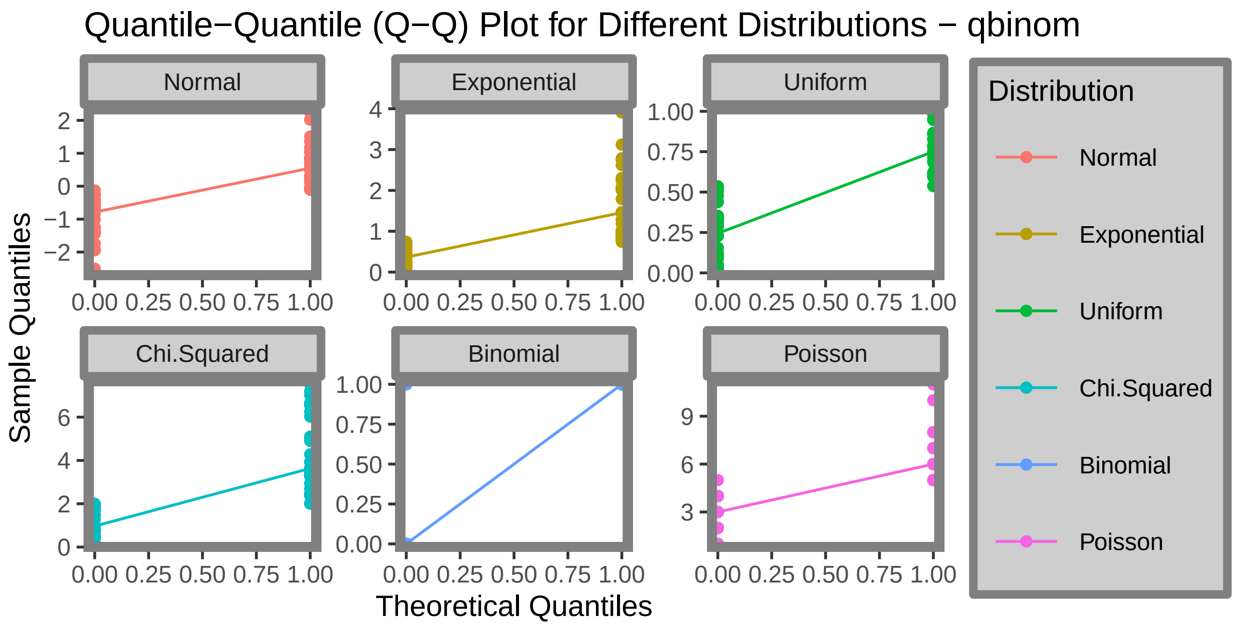


```
# Combining into a single data frame with labels
df <- data.frame(
  Data = c(normal_data, exponential_data, uniform_data, chi_squared_data, binomial_data, poisson_data),
  Distribution = rep(c("Normal", "Exponential", "Uniform", "Chi-Squared", "Binomial", "Poisson"), each
)

# Specifying the order of facets
df$Distribution <- factor(df$Distribution, levels = c("Normal", "Exponential", "Uniform", "Chi.Squared"

# Creating (Q-Q) plot
ggplot(df, aes(sample = Data, color = Distribution)) +
  geom_qq(distribution = qchisq, dparams = list(df = 3)) +
  stat_qq_line(distribution = qchisq, dparams = list(df = 3)) +
  facet_wrap(~ Distribution, scales = "free") +
  labs(
    title = "Quantile-Quantile (Q-Q) Plot for Different Distributions - qchisq",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  guides(color = guide_legend(title = "Distribution")) +
  theme(
    panel.background = element_blank(),  # Removing background
    panel.grid.major = element_blank(),  # Removing major grid lines
    panel.grid.minor = element_blank(),   # Removing minor grid lines
    panel.border = element_rect(fill = NA, color = "#808080", size = 3.5), # Coloring panel borders
    strip.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling, colori
    strip.clip = "off", # Making strip borders rounded
    legend.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling the le
    legend.key = element_rect(fill = "#cecece"), # Filling key backgrounds
    legend.key.size = unit(2,"lines") # Making legend background longer
  ) +  theme(aspect.ratio = 0.74) # Getting a bit rectangular shape for the panels
```



```
# Combining into a single data frame with labels
df <- data.frame(
  Data = c(normal_data, exponential_data, uniform_data, chi_squared_data, binomial_data, poisson_data),
  Distribution = rep(c("Normal", "Exponential", "Uniform", "Chi.Squared", "Binomial", "Poisson"), each =
)

# Specifying the order of facets
df$Distribution <- factor(df$Distribution, levels = c("Normal", "Exponential", "Uniform", "Chi.Squared"
```

```r
# Creating (Q-Q) plot
ggplot(df, aes(sample = Data, color = Distribution)) +
  geom_qq(distribution = qbinom, dparams = list(size = 1, prob = 0.5)) +
  stat_qq_line(distribution = qbinom, dparams = list(size = 1, prob = 0.5)) +
  facet_wrap(~ Distribution, scales = "free") +
  labs(
    title = "Quantile-Quantile (Q-Q) Plot for Different Distributions - qbinom",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  guides(color = guide_legend(title = "Distribution")) +
  theme(
    panel.background = element_blank(),  # Removing background
    panel.grid.major = element_blank(),  # Removing major grid lines
    panel.grid.minor = element_blank(),   # Removing minor grid lines
    panel.border = element_rect(fill = NA, color = "#808080", size = 3.5), # Coloring panel borders
    strip.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling, colori
    strip.clip = "off", # Making strip borders rounded
    legend.background = element_rect(fill = "#cecece", color = "#808080", size = 1.5), # Filling the le
    legend.key = element_rect(fill = "#cecece"), # Filling key backgrounds
    legend.key.size = unit(2,"lines") # Making legend background longer
  ) +  theme(aspect.ratio = 0.74) # Getting a bit rectangular shape for the panels
```



Give summary and explanation. What is the difference between those distributions, and which one of them is different, and why ?

For qnorm, qunif and qexp we have build in parameters that do not need to be specified, on the other hand if we plot qpois, qchisq, qbinom without specifying we will not see our distributions nor reference lines. That is the main difference. The Q-Q plots help determine how well the data fits the given distribution. We see that the generated random data points in most cases fall on that reference line when they are plotted against their distributions, because we define quantiles of theoretical and sample distributions to be the same.

Additionally, Against qunif, for example chi.squared, exponential, poisson are right skewed because the points' upward trend shows that the sample quantiles are much greater than the theoretical quantiles.

Can be said, that all besides binomial are left skewed against qexp. The sample quantiles are going to be much lower than the theoretical quantiles.

Against qpois, we see that, uniform and normal are left skewed. Exponential and chi.squared are right skewed.

Against qchisq, we see that exponential is almost the same with parameters rate = 1 and we have chi.square parameters df = 3. Also, normal and uniform is are left skewed.

PROBLEM 4 1. Generate random samples

```
library(ggridges)
library(moments)
library(evd)
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
##
## Attaching package: 'VGAM'
```

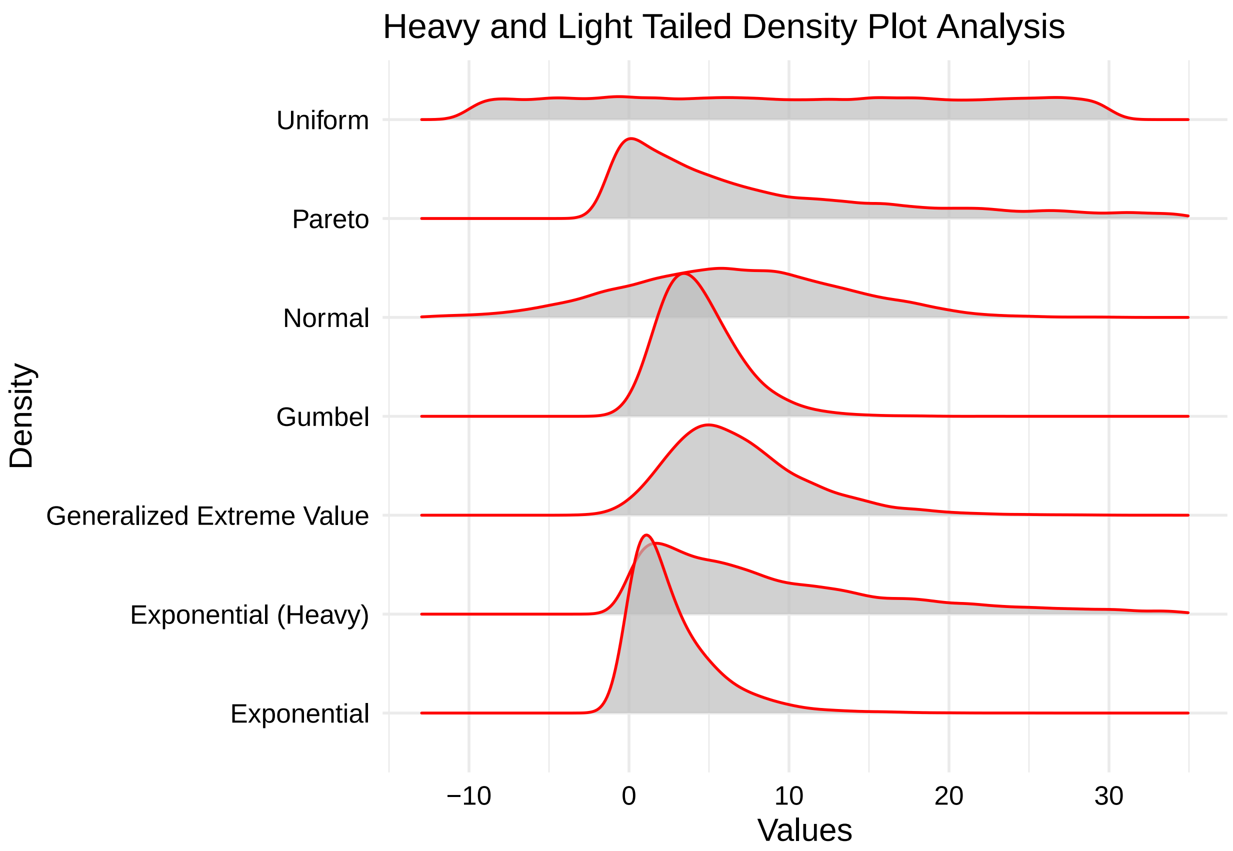
```
## The following objects are masked from 'package:evd':
##
##     dfrechet, dgev, dgpd, dgumbel, pfrechet, pgev, pgpd, pgumbel,
##     qfrechet, qgev, qgpd, qgumbel, rfrechet, rgev, rgpd, rgumbel,
##     venice
```

```
# Generating data for different distributions
set.seed(2013)

n <- 10000
data_uniform <- runif(n, min = -10, max = 30)
data_pareto <- (rpareto(n, scale = 1, shape = 1) - 1.13 ) * 9.5
data_normal <- rnorm(n, mean = 6.5, sd = 6.8)
data_gumbel <- rgumbel(n, location = 3.3, scale = 1.95)
data_gev <- rgev(n, location = 5, scale = 3.4, shape = 0)
data_exp_heavy <- rexp(n, rate = 0.1)
data_exp <- rexp(n, rate = 0.36)

# Combing data into a single data frame with labels
df <- data.frame(
  Distribution = factor(
    rep(
      c("Uniform", "Pareto", "Normal", "Gumbel", "Generalized Extreme Value", "Exponential (Heavy)", "E
      each = n
    ),
    levels = c("Exponential", "Exponential (Heavy)", "Generalized Extreme Value", "Gumbel", "Normal","Pa
  ),
  Data = c(data_uniform, data_pareto, data_normal, data_gumbel, data_gev, data_exp_heavy, data_exp)
```

```r
)

# Creating the ridgeline plot
ggplot(df, aes(x = Data, y = Distribution, fill = Distribution)) +
  geom_density_ridges(alpha = 0.7, fill = "grey", color = "red") +
  labs(
    title = "Heavy and Light Tailed Density Plot Analysis",
    x = "Values",
    y = "Density"
  ) +
  scale_x_continuous(
    breaks = seq(-10, 30, by = 10),   # Setting x-axis tick marks
    limits = c(-13, 35)              # Setting x-axis limits
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(size = 10, color = "black"), #For X axis
    axis.text.y = element_text(size = 10, color = "black"), # For Y axis
    axis.title.x = element_text(hjust = 0.5, size = 12, color = "black"), # For X title
    axis.title.y = element_text(hjust = 0.5, size = 12, color = "black"), # For Y title
    legend.position = "none"
  )
```

```
## Picking joint bandwidth of 0.88
```

```
## Warning: Removed 2417 rows containing non-finite values
## (`stat_density_ridges()`).
```



2. Analyze each distribution statistically and explain your findings. You may want to look at the mean, variance, kurtosis, skewness, or any other statistical feature that can help differentiate between a light-tailed and a heavy-tailed distribution. Also, you can explain the difference using parameter comparison for the same distribution.

Uniform Distribution: Mean: (min + max) / 2, which is 10. Variance: (max - min)^2 / 12, which is 70. Kurtosis: The kurtosis for this uniform distribution is approximately 1.8. It is important to note that the kurtosis of a uniform distribution is always constant and does not depend on the specific bounds of the distribution. Indicating a lack of heavy tails. Skewness: The skewness is 0, as the distribution is symmetric. So, the Uniform distribution is light-tailed and has low kurtosis with no skewness.

Pareto Distribution: This transformation - (rpareto(n, scale = 1, shape = 1) - 1.13) * 9.5 aims to shift and scale the random values generated from the Pareto distribution. Kurtosis: Pareto distributions have high kurtosis, indicating heavy tails. It contains 80% or more of the probabilities.

```
kurtosis(data_pareto)
```

```
## [1] 1832.491
```

Skewness: The skewness is positive for the given parameters, indicating a right-skewed distribution.

```
skewness(data_pareto)
```

```
## [1] 38.49526
```

So, the Pareto distribution is heavy-tailed with high kurtosis and positive skewness.

Normal Distribution: Mean: 6.5 Variance: 6.8^2 Kurtosis:

```
library(moments)
kurtosis(data_normal)
```

```
## [1] 2.975004
```

This value tells that we have a bit heavy tails and a fat central portion. It is close to 3 we could say that it is heavy. Skewness:

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:VGAM':
##
##     fisherz, logistic, logit

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
skew(data_normal)
```

```
## [1] -0.0009505221
```

This suggests a very mild leftward skew, but the distribution is almost symmetrical. A skewness value close to 0 suggests that the data distribution is nearly symmetric.

Gumbel Distribution: Mean: 3.3 Variance: 1.95^2 Kurtosis:

```
kurtosis(data_gumbel)
```

```
## [1] 5.374056
```

Kurtosis values greater than 3 (excess kurtosis) indicate a distribution with heavier tails and more peaked in the center. Skewness:

```
skewness(data_gumbel)
```

```
## [1] 1.152792
```

This means that the tail of the distribution extends to the right. And our bulk is located a bit left.

Generalized Extreme Value: Mean: 5 Variance: 3.4^2 Kurtosis:

```
kurtosis(data_gev)
```

```
## [1] 5.245115
```

Means has heavier tails and also a higher propensity for extreme values. Skewness:

```
skewness(data_gev)
```

```
## [1] 1.129111
```

Indicates a tendency for larger values in the dataset, leading to a right-skewed distribution.

Exponential Heavy:

```
mean(data_exp_heavy)
```

```
## [1] 9.879738
```

```
var(data_exp_heavy)
```

```
## [1] 97.08354
```

```
kurtosis(data_exp_heavy)
```

```
## [1] 9.45747
```

```r
skewness(data_exp_heavy)
```

```
## [1] 2.047256
```

The mean is approximately 9.88, which represents the central tendency of the data. The variance is relatively high, indicating a wide spread of data points. The kurtosis is approximately 9.46, which suggests that the dataset has a positive excess kurtosis, meaning it has heavier tails. The skewness is approximately 2.05, indicating a positive skew, which means the distribution is skewed to the right.

Exponential:

```r
mean(data_exp)
```

```
## [1] 2.752747
```

```r
var(data_exp)
```

```
## [1] 7.657532
```

```r
kurtosis(data_exp)
```

```
## [1] 10.12934
```

```r
skewness(data_exp)
```

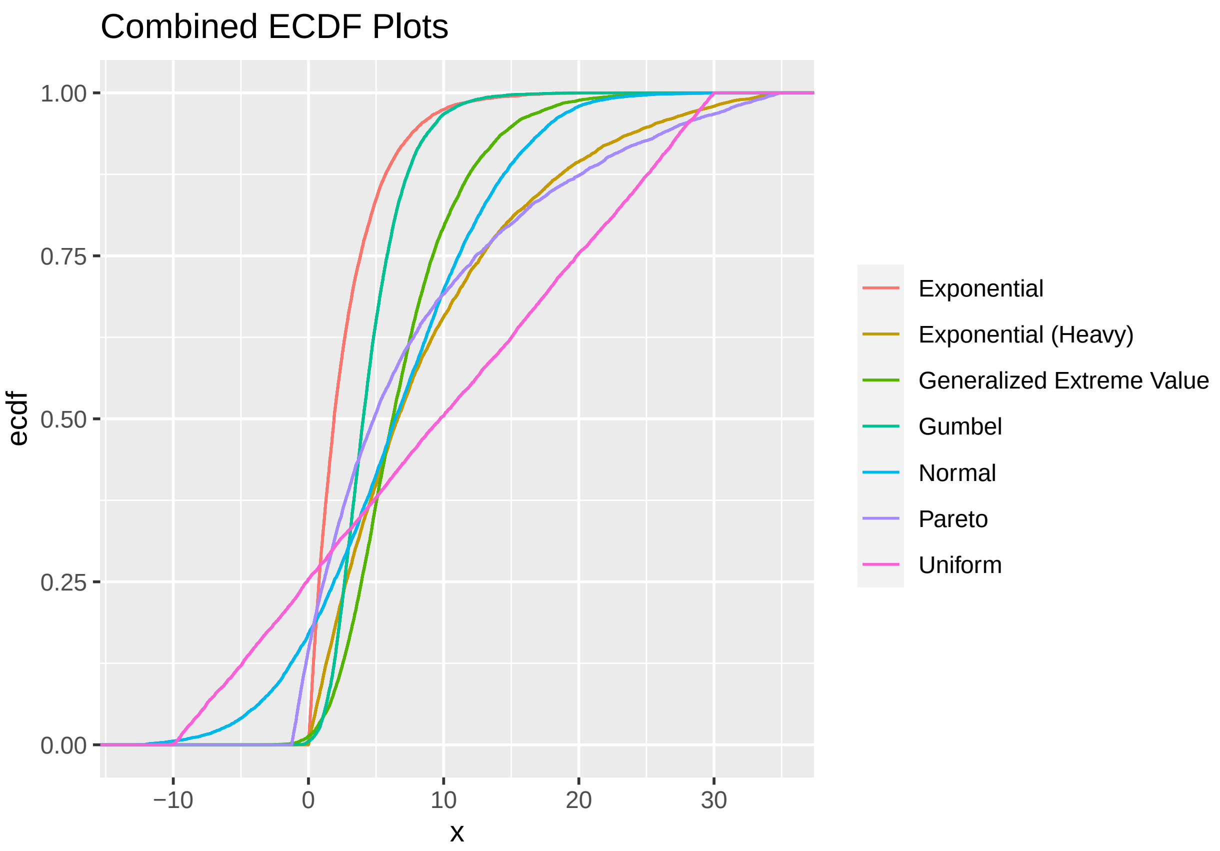
```
## [1] 2.083361
```

Exhibits heavy-tailed behavior with high kurtosis and a strong positive skew to the right.

3. Visualize the generated distributions

```r
# Create a data frame with all the data
combined_data <- data.frame(
  Distribution = factor(rep(c("Uniform", "Pareto", "Normal", "Gumbel", "Exponential (Heavy)", "Exponenti
                            each = length(data_uniform))),
  x = c(data_uniform, data_pareto, data_normal, data_gumbel, data_exp_heavy, data_exp, data_gev)
)

# Create a single ECDF plot
ecdf_plot <- ggplot(combined_data, aes(x = x, color = Distribution)) +
  geom_step(stat = "ecdf") +
  labs(title = "Combined ECDF Plots") +
  scale_color_discrete(guide = guide_legend(title = NULL)) +
  scale_x_continuous(
    breaks = seq(-10, 30, by = 10),   # Setting x-axis tick marks
    limits = c(-13, 35)             # Setting x-axis limits
  )


# Display the combined ECDF plot
print(ecdf_plot)
```

```
## Warning: Removed 2417 rows containing non-finite values (`stat_ecdf()`).
```



Visually explain the difference between light-tailed and heavy-tailed distributions.

As we say that uniform is surely not heavy-tailed. Heavy-tailed distributions have a greater probability of extreme events or outliers compared to distributions with lighter tails. We can say that how much the slope is stricter than more heavy tailed it is.

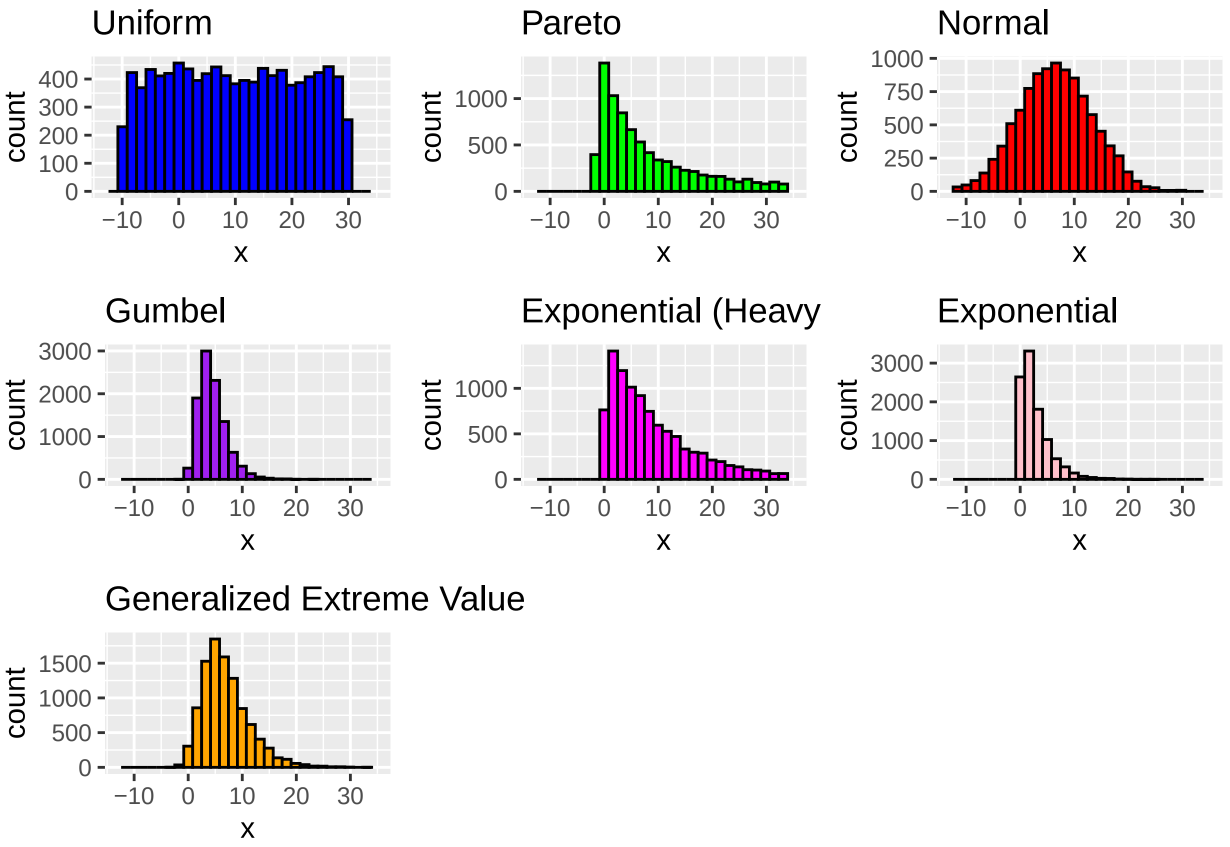
4. Histogram Analysis

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# Creating histograms for each distribution
hist_plots <- list(
  ggplot(data.frame(x = data_uniform), aes(x = x)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  labs(title = "Uniform") +
    scale_x_continuous(
    breaks = seq(-10, 30, by = 10),
    limits = c(-13, 35)
  ),

  ggplot(data.frame(x = data_pareto), aes(x = x)) +
  geom_histogram(bins = 30, fill = "green", color = "black") +
  labs(title = "Pareto") +
    scale_x_continuous(
    breaks = seq(-10, 30, by = 10),
    limits = c(-13, 35)
  ),

  ggplot(data.frame(x = data_normal), aes(x = x)) +
  geom_histogram(bins = 30, fill = "red", color = "black") +
  labs(title = "Normal")+
    scale_x_continuous(
    breaks = seq(-10, 30, by = 10),
    limits = c(-13, 35)
  ),

  ggplot(data.frame(x = data_gumbel), aes(x = x)) +
  geom_histogram(bins = 30, fill = "purple", color = "black") +
  labs(title = "Gumbel")+
    scale_x_continuous(
    breaks = seq(-10, 30, by = 10),
    limits = c(-13, 35)
  ),

  ggplot(data.frame(x = data_exp_heavy), aes(x = x)) +
  geom_histogram(bins = 30, fill = "magenta", color = "black") +
  labs(title = "Exponential (Heavy)")+
    scale_x_continuous(
    breaks = seq(-10, 30, by = 10),
    limits = c(-13, 35)
  ),

  ggplot(data.frame(x = data_exp), aes(x = x)) +
  geom_histogram(bins = 30, fill = "pink", color = "black") +
  labs(title = "Exponential")+
    scale_x_continuous(
    breaks = seq(-10, 30, by = 10),
    limits = c(-13, 35)
  ),

  ggplot(data.frame(x = data_gev), aes(x = x)) +
  geom_histogram(bins = 30, fill = "orange", color = "black") +
  labs(title = "Generalized Extreme Value") +
    scale_x_continuous(
    breaks = seq(-10, 30, by = 10),
    limits = c(-13, 35)
  )
)

# Arranging the histograms in a grid
grid.arrange(grobs = hist_plots, ncol = 3)
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).

## Warning: Removed 2104 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 2 rows containing missing values (`geom_bar()`).
## Warning: Removed 26 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 2 rows containing missing values (`geom_bar()`).
## Removed 2 rows containing missing values (`geom_bar()`).

## Warning: Removed 284 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 2 rows containing missing values (`geom_bar()`).

## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 2 rows containing missing values (`geom_bar()`).

## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



Observe and explain the histogram plots for these distributions.

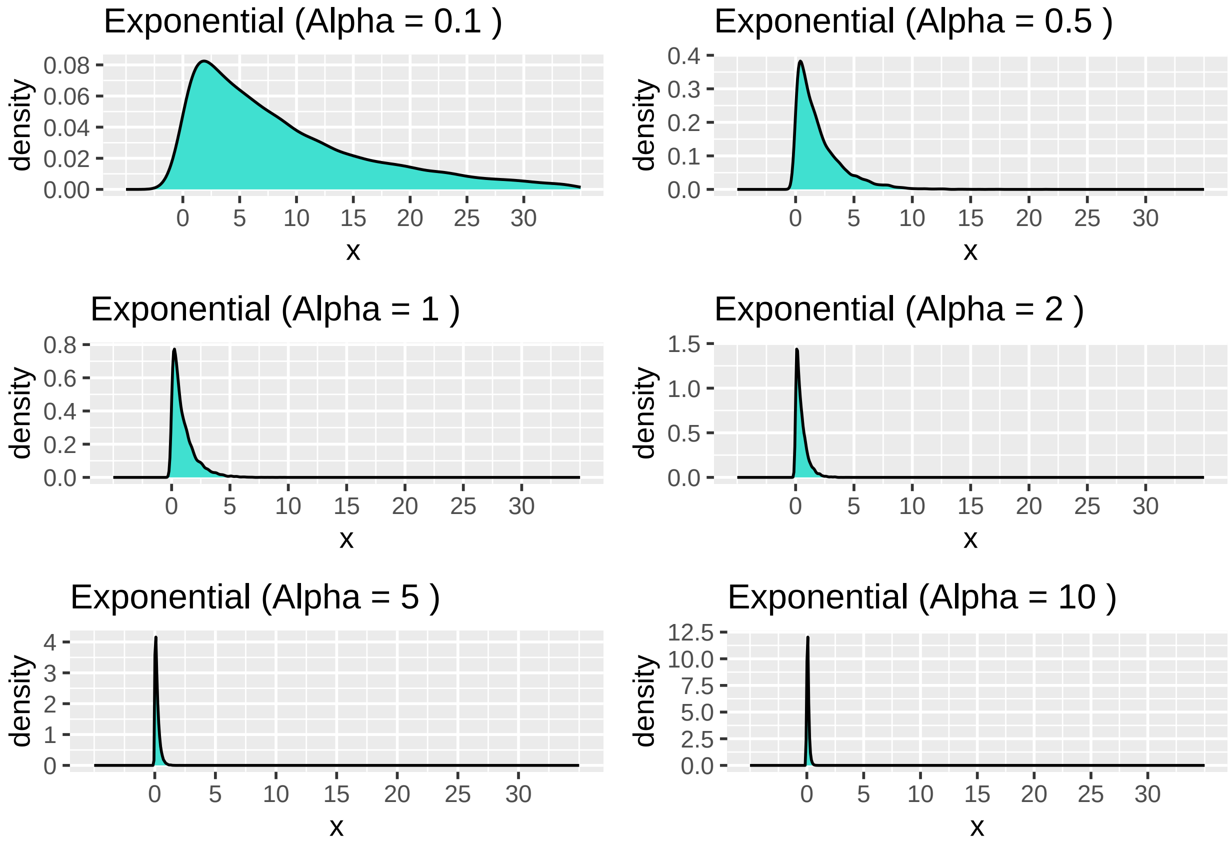
Heavy-tailed distributions have long tails that extend further from the central peak. Some heavy-tailed distributions exhibit an exponential-like shape in the tails, meaning the frequency of observations decreases at a slower rate, similar to an exponential decay. Pareto, Gumbel, Exponential (Heavy), Exponential and Generalized Extreme Value, all are right skewed with a heavy tail. For Normal, we see a little left skewed histogram, and Uniform we surely see that it is not skewed nor has any heavy tail.

5. Exponential Distribution Comparison

```r
# Number of data points
n <- 10000

# Generating exponential distributions with different alpha values
alphas <- c(0.1, 0.5, 1, 2, 5, 10)
```

```
exp_distributions <- lapply(alphas, function(alpha) rexp(n, rate = alpha))

# Creating density plots for comparison
plots <- lapply(1:length(exp_distributions), function(i) {
  ggplot(data.frame(x = exp_distributions[[i]]), aes(x = x)) +
    geom_density(fill = "turquoise", color = "black") +
    labs(title = paste("Exponential (Alpha =", alphas[i], ")"))
} +
    scale_x_continuous(
    breaks = seq(0, 30, by = 5),
    limits = c(-5, 35)
  ))

# Arranging the density plots in a grid
grid.arrange(grobs = plots, ncol = 2)
```

```
## Warning: Removed 301 rows containing non-finite values (`stat_density()`).
```



```
# Analyzing the findings
print("Alpha Values and Tailedness:")
```

```
## [1] "Alpha Values and Tailedness:"
```

```
for (i in 1:length(alphas)) {
  if (alphas[i] < 1) {
    cat(paste("Alpha =", alphas[i], "results in a heavy-tailed distribution.\n"))
  } else if (alphas[i] > 1) {
    cat(paste("Alpha =", alphas[i], "results in a light-tailed distribution.\n"))
  } else {
    cat(paste("Alpha =", alphas[i], "results in an exponential distribution with a standard tail.\n"))
  }
}
```

```
## Alpha = 0.1 results in a heavy-tailed distribution.
## Alpha = 0.5 results in a heavy-tailed distribution.
## Alpha = 1 results in an exponential distribution with a standard tail.
## Alpha = 2 results in a light-tailed distribution.
## Alpha = 5 results in a light-tailed distribution.
## Alpha = 10 results in a light-tailed distribution.
```

Compare the distributions and explain your findings. Which one are light tailed and heavy tailed? Why that happens?

The alpha parameter controls the rate at which the exponential distribution decreases. A smaller alpha results in a slower decrease, leading to heavier tails, while a larger alpha results in a faster decrease, leading to lighter tails. Light-tailed distributions have tails that decrease quickly, and extreme values are unlikely. Heavy-tailed distributions have tails that decrease more slowly, allowing for a higher probability of extreme values. When we increase alpha, we see that the distribution is becoming more and more light tailed. A lower alpha leads to a heavier-tailed distribution.

6. Summarize your findings Write a brief conclusion on insights obtained from the histograms, PDFs, and statistical analysis, highlighting the key differences between heavy-tailed and light-tailed distributions.

It is clear that heavy-tailed distributions exhibit characteristics such as long tails that extend further from the central peak. Some of these distributions show an exponential-like shape in the tails, indicating a slower decrease in frequency as we move away from the central peak. In summary, distributions with heavier tails have a greater probability of extreme events or compared to those with lighter tails. The steepness of the slope in the tails is a key factor in determining the heaviness of the tail.