MA2003B Application of Multivariate Methods in Data Science

Module 1: Interdependence Relationships

Instructor: Raul Gomez
Email: rgomez@tec.mx
Office: A4-123A

August 14, 2025

Chapter 1

An Introduction to the Tidyverse

The Tidyverse is a collection of R packages designed for data science. It provides a consistent and user-friendly interface for data manipulation, visualization, and analysis. Although formally introduced as an ecosystem in 2016, many of these packages were developed by Hadley Wickham between 2007–2014.

1.1 Tidy Data

The core idea of behind the design of the Tidyverse is Wickham's definition of "tidy data" [3].

Definition 1.1.1. *Tidy Data* is a way of structuring datasets to make them easier to work with. In tidy data:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

This structure allows for easier data manipulation, analysis, and visualization, as it aligns with the way data is typically processed in R. In this section we will describe the basics of the Tidyverse but, for more information on its use, the interested reader can check the following references: [5, 6, 7].

1.1.1 Tibbles and pipes

In the Tidyverse, data is often represented using a special type of data frame called a *tibble*. Tibbles are more user-friendly than traditional data frames, as they provide better printing and subsetting behavior. They also allow for more intuitive handling of data. In particular, tibbles are designed to work seamlessly with the pipe operator (`%>%`), which allows for chaining together multiple operations (usually known as "verbs" in the context of the tidyverse) in a clear and concise manner. To illustrate these ideas, in the next section we will perform data cleaning on the "WHO" dataset, to make sure that the associated tibble satisfies the tidy data principles. This is usually the first step in Exploratory Data Analysis (EDA) and is crucial for ensuring that the data is in a suitable format for analysis and visualization.

1.1.2 An Example: Tidying the WHO dataset

Before starting, we need to load the required libraries and load the WHO dataset:

```
library("tidyverse")
library("here")
library("cowplot")
library("patchwork")
library("krulRutils")
library("ISLR2")
library("magrittr")

options(scipen = 999) # Disable scientific notation
data(who)
```

We can now start our "tidying" process. The main problem with this dataset is that it contains variables (like "new_sp_m014") that are not very clear and that contain more than one piece of information. So we need to separate these variables and introduce better names for them and their associated values.

```
# We start by creating the variable "who_tidy"
# that contains the cleaned WHO dataset
who_tidy <- who %>%
  # We use pivot_longer to eliminate the variables starting with "new"
  # and use them as values instead
  pivot_longer(
    cols = starts_with("new"),
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  ) %>%
  mutate(
    key = if_else(
      startsWith(key, "newrel"),
      sub("newrel", "new_rel", key),
      key
    ),
    cases = as.integer(cases)
  ) %>%
  separate(key, into = c("new", "type", "sexage"), sep = "_") %>%
  separate(sexage, into = c("sex", "age"), sep = 1) %>%
  select(-new, -iso2, - iso3)  %T>%
  # Finally, we save the tidy data in both RDS and CSV formats
  saveRDS(here("data", "who_tidy.rds")) %>%
  write_csv(here("data", "who_tidy.csv")) %>%
  glimpse()
```

```
Rows: 76,046
```

```
Columns: 6
$ country <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", "A~
$ year    <dbl> 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, 19~
$ type    <chr> "sp", "sp", "sp", "sp", "sp", "sp", "sp", "sp", "sp", "sp", "s~
$ sex     <chr> "m", "m", "m", "m", "m", "m", "m", "f", "f", "f", "f", "f", "f~
$ age     <chr> "014", "1524", "2534", "3544", "4554", "5564", "65", "014", "1~
$ cases   <int> 0, 10, 6, 3, 5, 2, 0, 5, 38, 36, 14, 8, 0, 1, 30, 129, 128, 90~
```

We now want to convert the columns "type", "sex", and "age" into factors. We start by constructing the following lookup tables:

```
who_type_lookup_tbl <- tibble(
  code = c("ep", "rel", "sn", "sp"),
  label = c(
    "Extrapulmonary TB",
    "Relapse case",
    "Smear-Negative pulmonary TB",
    "Smear-Positive pulmonary TB"
  )
)

who_sex_lookup_tbl <- tibble(
  code = c("f", "m"),
  label = c("Female", "Male")
)

who_age_lookup_tbl <- tibble(
  code = c(
    "014",
    "1524",
    "2534",
    "3544",
    "4554",
    "5564",
    "65"
  ),
  label = c(
    "0-14",
    "15-24",
    "25-34",
    "35-44",
    "45-54",
    "55-64",
    "65+"
  )
)
```

We can now append factor columns for the corresponding variables in the WHO dataset:

```
who_factor <- who_tidy %>%
  convert_codes_to_factor(
    code_col = type,
    lookup_tbl = who_type_lookup_tbl,
    lookup_code_col = code,
    lookup_label_col = label,
  ) %>%
  convert_codes_to_factor(
    code_col = sex,
    lookup_tbl = who_sex_lookup_tbl,
    lookup_code_col = code,
    lookup_label_col = label,
  ) %>%
  convert_codes_to_factor(
    code_col = age,
    lookup_tbl = who_age_lookup_tbl,
    lookup_code_col = code,
    lookup_label_col = label,
  ) %>%
  glimpse()
```

```
Rows: 76,046
Columns: 9
$ country     <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan"~
$ year        <dbl> 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997~
$ type        <chr> "sp", "sp", "sp", "sp", "sp", "sp", "sp", "sp", "sp", "sp"~
$ sex         <chr> "m", "m", "m", "m", "m", "m", "m", "f", "f", "f", "f", "f"~
$ age         <chr> "014", "1524", "2534", "3544", "4554", "5564", "65", "014"~
$ cases       <int> 0, 10, 6, 3, 5, 2, 0, 5, 38, 36, 14, 8, 0, 1, 30, 129, 128~
$ type_factor <fct> Smear-Positive pulmonary TB, Smear-Positive pulmonary TB, ~
$ sex_factor  <fct> Male, Male, Male, Male, Male, Male, Male, Female, Female, ~
$ age_factor  <fct> 0-14, 15-24, 25-34, 35-44, 45-54, 55-64, 65+, 0-14, 15-24,~
```

Finally, we want to use this information to analyze the number of cases of tuberculosis by type and sex. We start by generating the corresponding frequency tibble.

```
who_type_sex_tbl <- who_factor %>%
  count(type_factor, sex_factor, wt = cases, name = "cases") %>%
  print(n = Inf)
```

```
# A tibble: 8 x 3
  type_factor                sex_factor    cases
  <fct>                      <fct>         <int>
1 Extrapulmonary TB          Female       941880
2 Extrapulmonary TB          Male        1044299
3 Relapse case               Female      1201596
```

```
4 Relapse case                 Male        2018976
5 Smear-Negative pulmonary TB Female      2439139
6 Smear-Negative pulmonary TB Male        3840388
7 Smear-Positive pulmonary TB Female     11324409
8 Smear-Positive pulmonary TB Male       20586831
```

1.2 The "Grammar of Graphics" and ggplot2

The concept of the "Grammar of Graphics" was introduced by Leland Wilkinson in his 1999 book *The Grammar of Graphics* [8]. This provides a framework for understanding how to create visualizations in a systematic way. It breaks down the process of creating a plot into components such as data, aesthetics, geometries, statistics, coordinates, and themes. The `ggplot2` package [4] implements this grammar in R, allowing users to build complex visualizations by layering these components. For more information on the Grammar of Graphics, its history and applications, the interested reader can consult the following references: [1, 2].

1.2.1 Layers in ggplot2

In `ggplot2`, plots are constructed by adding multiple *layers* that define the components of the visualization. Each layer corresponds to a specific aspect of the plot, and together they form the complete graphic. These layers include:

1. **Data:** The dataset to be visualized (usually a data frame or tibble). This is the source of the information for the plot.
2. **Aesthetics:** Mappings that relate variables in the data to visual properties of the plot, such as:
   - Position on the x- and y-axes.
   - Color, fill.
   - Size, shape.
   - Transparency.

   These mappings define *what* data is shown and *how* it is represented visually.
3. **Geometries:** Geometric objects that display the data, for example:
   - `geom_point()` for scatterplots.
   - `geom_line()` for line graphs.
   - `geom_col()` for bar charts.
   - `geom_histogram()` for histograms.

   The geometry controls *how* the data is drawn.
4. **Statistical transformations:** Optional calculations applied to the data before plotting, such as:

- `stat_bin()` for binning data in histograms.
- `stat_smooth()` for fitting smooth curves.

These are preprocessing steps for the data visualization.

5. **Scales:** Define how data values are translated into visual properties, for example mapping numeric values to colors or shapes. Scales also control axis ticks, legends, and guides.
6. **Coordinates:** The coordinate system used for the plot, such as Cartesian (`coord_cartesian()`), polar (`coord_polar()`), or flipped coordinates (`coord_flip()`).
7. **Facets:** Methods to split the data into subsets and display multiple plots arranged in a grid:
   - `facet_wrap()`
   - `facet_grid()`
8. **Labels:** `labs()` adds titles, axis labels, and captions.
9. **Themes:** `theme()` controls the overall appearance of the plot (fonts, background, gridlines).

Each layer can be combined and customized to build complex and elegant plots. This *layered grammar* allows you to think of your graphic as a composition of independent components rather than a single, monolithic object.

1.2.2 Plotting with ggplot2

We will illustrate these concepts by plotting the number of cases of tuberculosis by type and sex, using the frequency tibble we created earlier.

```
who_type_sex_plot <- who_type_sex_tbl %>%
  ggplot(aes(x = type_factor, y = cases, fill = sex_factor)) +
  geom_col(position = "dodge") +
  c_scale_fill("C rose", "C blue") +
  labs(
    title = "Tuberculosis Cases by Type and Sex",
    x = "Tuberculosis Type",
    y = "Cases",
    fill = "Sex"
  ) +
  theme_krul()
```

We can now save and plot the graph:

```
ggsave(
  filename = here("images", "who_plot.png"),
  plot = who_type_sex_plot,
  width = 12,
  height = 6,
  dpi = 300
```

```
)
who_type_sex_plot
```

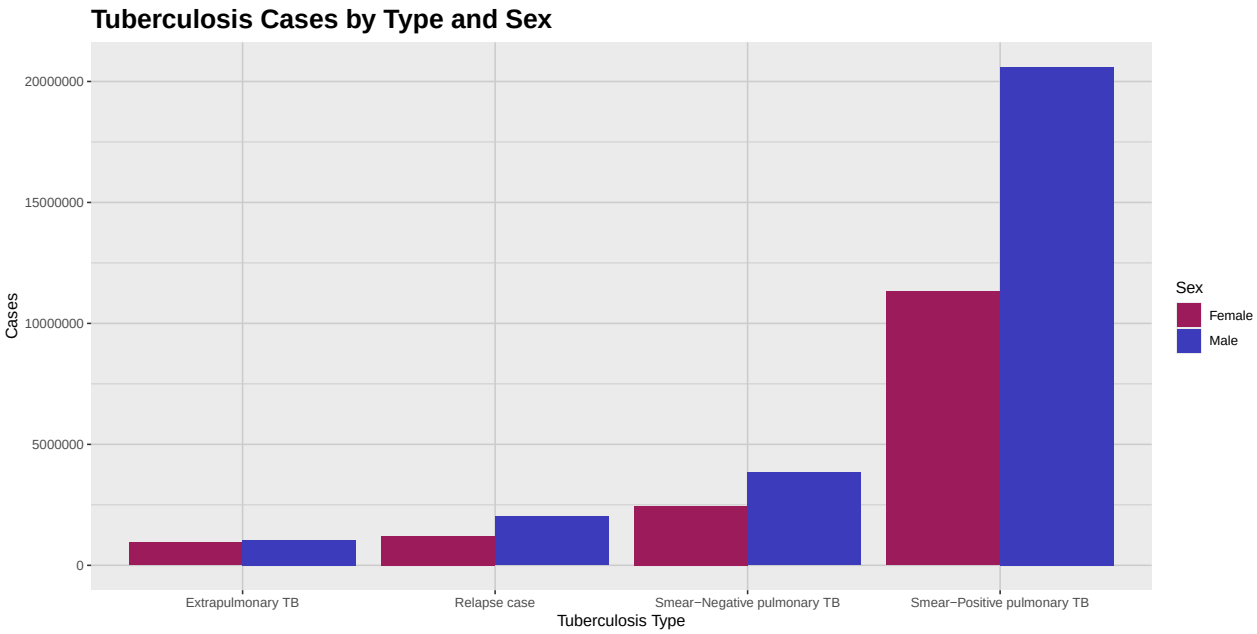

Figure 1.1: This plot shows a comparative analysis of the number of cases of tuberculosis by type and sex. As we can see from the plot, the number of cases is consistently higher for the male population across all types of tuberculosis.

1.3 Activity: Tidying the Billboard dataset

The problem with the Billboard dataset, according to the principles of "tidy data", is that the weeks in which a song appeared in the Billboard chart are represented as columns, which makes it difficult to analyze the data. Using the techniques we have learned so far, the reader is required to tidy the Billboard dataset, so that it can be used to analyze the evolution of the ranking of the song "Who Let The Dogs Out" by "Baha Men" in the Billboard Hot 100 chart. The final visualization should look like the one shown below.

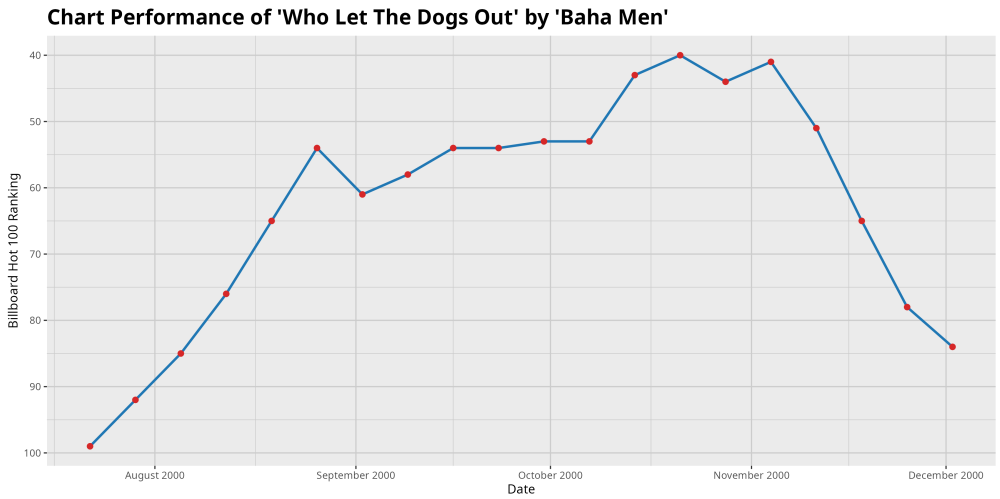

Figure 1.2: Evolution of the ranking of the song 'Who Let The Dogs Out' by 'Baha Men' in the Billboard Hot 100 chart. The song peaked at rank 40 in the week of October 21st, 2000, and remained in the top 100 for several weeks.

Bibliography

[1] Cleveland, W. S. (1993) *The Elements of Graphing Data*. Hobart Press.

[2] Hyndman, R. J., and Athanasopoulos, G. (2021) *Forecasting: Principles and Practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

[3] Wickham, H. (2014) Tidy data. *Journal of Statistical Software*, *59*(10), 1–23. https://doi.org/10.18637/jss.v059.i10

[4] Wickham, H. (2016) *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag.

[5] Wickham, H., and Grolemund, G. (2017) *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media.

[6] Wickham, H. (2019) Functional programming with purrr. *UseR! Conference*.

[7] Wickham, H., et al. (2019) Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

[8] Wilkinson, L. (1999) *The Grammar of Graphics*. Springer-Verlag.

Chapter 2

Skewness, Kurtosis and QQ Plots

As it is well known, any normal distribution is completely determined by its mean and standard deviation. However, many real-world datasets do not follow a normal distribution. In this chapter, we will explore some other distributions and show how the concepts of skewness and kurtosis can help us understand its shape. We will also introduce QQ plots, which are useful graphical tools for comparing the distribution of a dataset against a theoretical distribution, which is usually taken to be normal. But before starting, let's load the necessary libraries and set some options.

```
library("tidyverse")
library("here")
library("patchwork")
library("krulRutils")
library("ISLR2")
library("magrittr")
library("e1071") # for skewness and kurtosis functions

options(scipen = 999)
```

2.1 Skewness

Definition 2.1.1. The *skewness* of a random variable X is defined as the third standardized moment, given by

$$\gamma_1 = \frac{\mu_3}{\sigma^3} = \frac{\mathbb{E}[(X-\mu)^3]}{\sigma^3},$$

where $\mu = \mathbb{E}[X]$ is the mean and $\sigma = \sqrt{\mathbb{E}[(X-\mu)^2]}$ is the standard deviation of X.

Skewness measures the asymmetry of a probability distribution. A distribution is said to be *positively skewed* (or right-skewed) if it has a longer tail on the right side, and *negatively skewed* (or left-skewed) if it has a longer tail on the left side. A perfectly symmetric distribution has a skewness of zero. To illustrate this idea, we will introduce the gamma distributions.

Definition 2.1.2. We say that a continuous random variable X follows a *Gamma distribution* with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$, if its associated probability density function

(PDF) is given by

$$f_X(x;\alpha,\beta) = \begin{cases} \dfrac{\beta^\alpha}{\Gamma(\alpha)}\, x^{\alpha-1}e^{-\beta x}, & x > 0, \\ 0, & x \leq 0, \end{cases} \tag{2.1}$$

where $\Gamma(\alpha)$ denotes the Gamma function, defined as

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1}e^{-t}\,dt.$$

The parameters α and β control the shape and rate of the distribution, respectively.

Among other applications, the Gamma distribution is frequently used to model the number of total insurance claims over a given time period.

Remark 2.1.3. The skewness of a Gamma distribution is given by

$$\gamma_1 = \frac{2}{\sqrt{\alpha}}.$$

This means that the skewness is always positive, and it approaches zero as α increases. In other words, the distribution becomes more symmetric as α becomes larger.

We will now show how to plot the PDF of the Gamma distribution for $\alpha = 2$ and $\beta = 1$, highlighting the mean and median.

```
# Parameters
alpha <- 2
beta <- 1

# Create sequence of x values
x_data <- seq(0, 10, length.out = 200)

# Calculate density values
gamma_distribution_tbl <- tibble(
  x_data = x_data,
  density = dgamma(x_data, shape = alpha, rate = beta)
)

# Calculate mean and median
mean_val <- alpha / beta
median_val <- qgamma(0.5, shape = alpha, rate = beta)

# Plot
gamma_distribution_tbl %>%
ggplot(aes(x = x_data, y = density)) +
  geom_line(
```

```
  color = c_palette["C blue"],
  linewidth = 1
) +
geom_vline(
  xintercept = mean_val,
  color = c_palette["C red"],
  linetype = "dashed",
  linewidth = 1
) +
geom_vline(
  xintercept = median_val,
  color = c_palette["C green"],
  linetype = "dotted",
  linewidth = 1
) +
annotate(
  "text",
  x = mean_val,
  y = 0.35,
  label = "Mean",
  color = c_palette["C red"],
  hjust = -0.1
) +
annotate(
  "text",
  x = median_val,
  y = 0.35,
  label = "Median",
  color = c_palette["C green"],
  hjust = 1.1
) +
labs(
  title = paste(
    "Gamma Distribution PDF (shape =", alpha, ", rate =", beta, ")"
  ),
  x = "x",
  y = "Density"
) +
theme_krul()
```

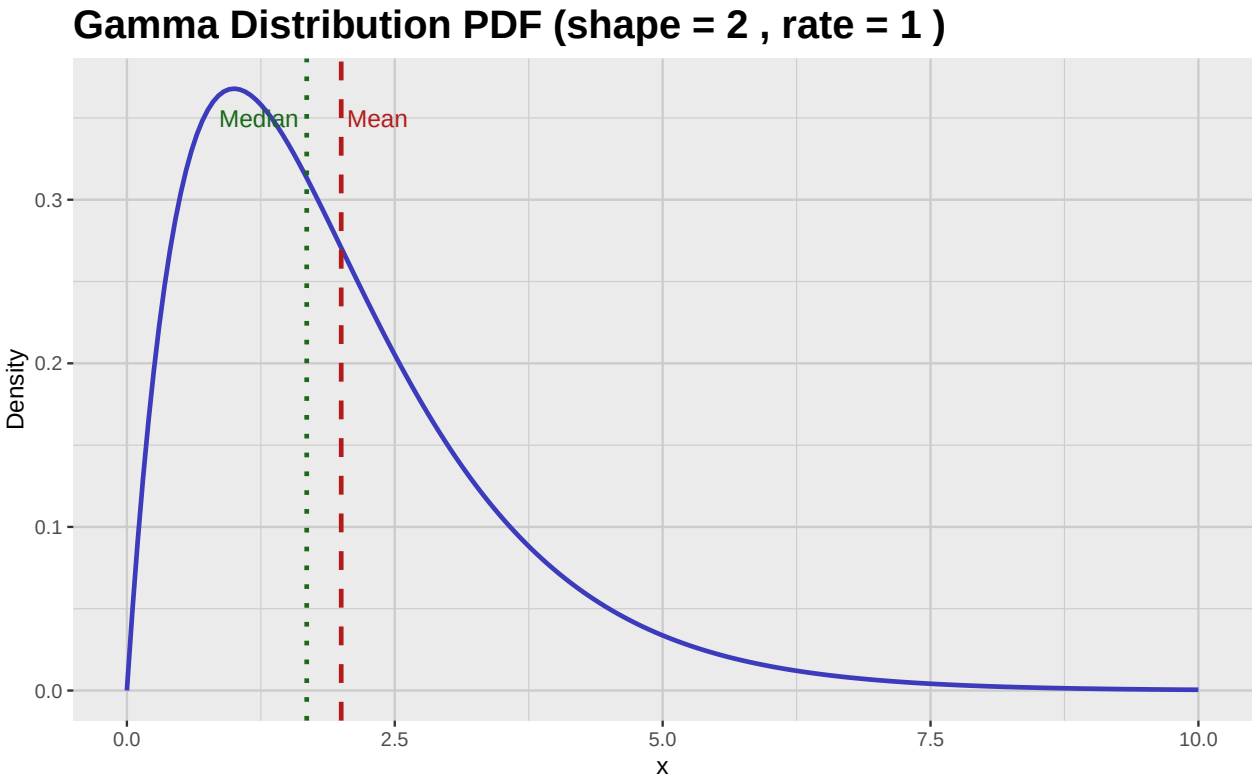

Figure 2.1: PDF of the gamma distribution with mean and median highlighted.

2.2 Kurtosis

Definition 2.2.1. The *kurtosis* of a random variable X is defined as the fourth standardized moment, given by

$$\beta_2 = \frac{\mu_4}{\sigma^4} = \frac{\mathbb{E}[(X-\mu)^4]}{\sigma^4},$$

where $\mu = \mathbb{E}[X]$ is the mean and $\sigma = \sqrt{\mathbb{E}[(X-\mu)^2]}$ is the standard deviation of X.

Kurtosis measures the "tailedness" of a probability distribution. A distribution with high kurtosis (also known as a *leptokurtic* distribution) has heavier tails and a sharper peak than a normal distribution, while a distribution with low kurtosis (also known as a *platykurtic* distribution) has lighter tails and a flatter peak. The kurtosis of a normal distribution is 3, which is why frequently we are interested in computing the *excess kurtosis* which is given by:

$$\gamma_2 = \beta_2 - 3.$$

To illustrate this idea, we will introduce the Laplace distribution.

Definition 2.2.2. We say that a continuous random variable X follows a *Laplace distribution* with location parameter $\mu \in \mathbb{R}$ and scale parameter $b > 0$, if its associated probability density function (PDF) is given by

$$f_X(x;\mu,b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right), \quad x \in \mathbb{R}. \tag{2.2}$$

The parameters μ and b control the location and scale of the distribution, respectively.

Among other applications, the Laplace distribution is often used in image processing and computer vision, particularly in modeling noise in images.

Remark 2.2.3. The excess kurtosis of a Laplace distribution is given by

$$\gamma_2 = 3.$$

This means that the Laplace distribution has heavier tails than a normal distribution.

We will now show how to plot the PDF of the Laplace distribution with mean zero and variance one.

```
# Parameters
mu <- 0      # mean (location)
b <- 1/sqrt(2)       # scale

# Sequence of x values covering enough range to see tails
x_data <- seq(-5, 5, length.out = 300)

# Laplace PDF function
dlaplace <- function(x, mu, b) {
  return(1/(2*b) * exp(-abs(x - mu)/b))
}

# Create data frame with PDF values
laplace_distribution_tbl<- tibble(
  x_data = x_data,
  density = dlaplace(x_data, mu, b)
)

# Plot PDF
laplace_distribution_tbl %>%
  ggplot(aes(x = x_data, y = density)) +
  geom_line(color = c_palette["C blue"], linewidth = 1) +
  labs(
    title = "Laplace Distribution PDF (mean = 0, var = 1)",
    x = "x",
    y = "Density"
  ) +
  theme_krul()
```

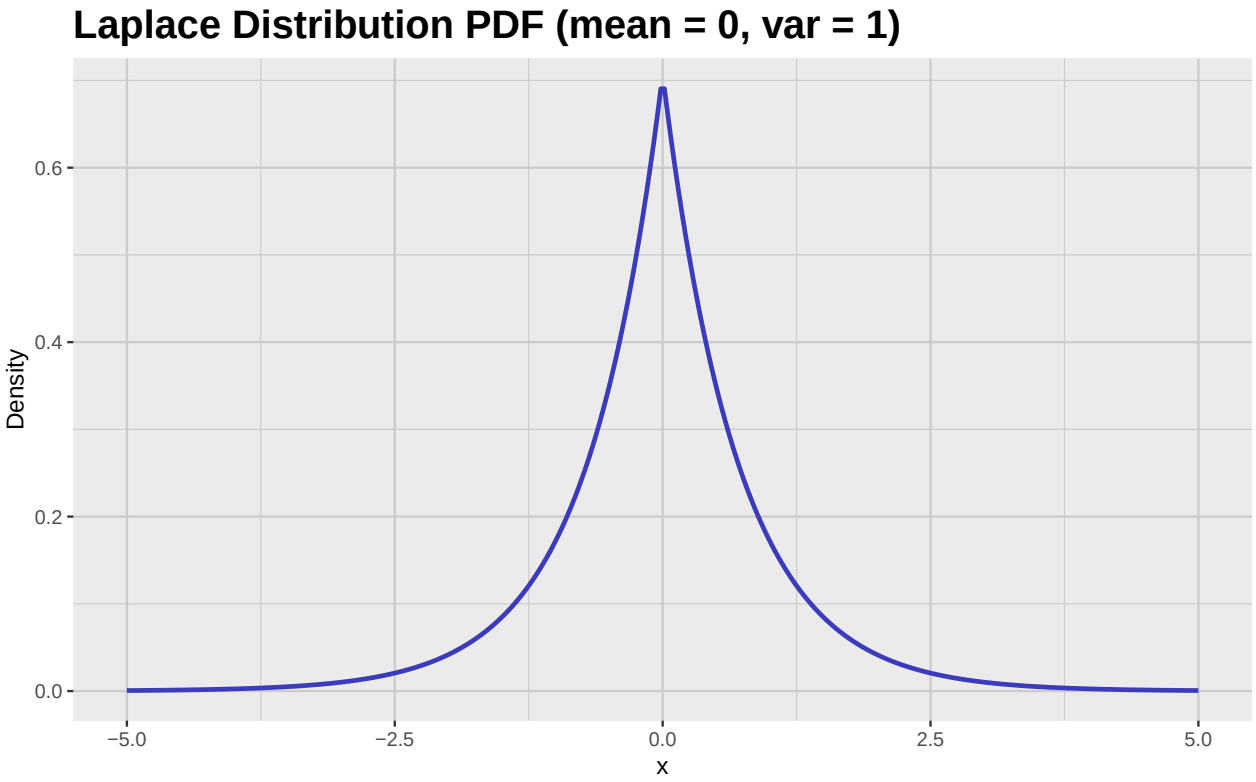

Figure 2.2: PDF of the Laplace distribution with mean zero and variance one.

2.3 QQ Plots

QQ plots, or quantile-quantile plots, are graphical tools used to compare the distribution of a dataset against a theoretical distribution, such as the normal distribution. They are particularly useful for assessing the normality of the data.

To construct a QQ plot, you should follow these steps:

1. **Sort the sample data:** Arrange the data in increasing order:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}.$$

2. **Calculate plotting positions (empirical quantiles):** For each ordered value $x_{(i)}$, compute the corresponding probability

$$p_i = \frac{2i-1}{2n}, \quad i = 1, 2, \ldots, n.$$

3. **Find theoretical quantiles of the normal distribution:** Compute the theoretical quantiles from the inverse cumulative distribution function (CDF) of the normal distribution:

$$q_i = \Phi^{-1}(p_i),$$

where Φ^{-1} is the quantile function of the standard normal distribution.

4. **Plot the points:** Plot the pairs $(q_i, x_{(i)})$ with q_i on the x-axis and $x_{(i)}$ on the y-axis.
5. **Add a reference line:** Typically, we add a line through the first and third quartiles to help assess normality.

The closer the points lie to this line, the more the sample resembles the specified normal distribution.

To illustrate these ideas, we will now generate samples from the Gamma and Laplace distributions, and then we will construct their associated QQ plots and histograms. First we construct the samples:

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

# Sample size
n <- 300
# Generate random samples from the Gamma distribution
gamma_sample <- rgamma(n, shape = alpha, rate = beta)

gamma_sample_tbl <- tibble(
  sample = gamma_sample
)

# Generate uniform random numbers in (0,1)
u <- runif(n)

# Apply inverse CDF of Laplace
laplace_sample <- ifelse(u < 0.5,
                         mu + b * log(2*u),
                         mu - b * log(2*(1 - u)))

laplace_sample_tbl <- tibble(
  sample = laplace_sample
)
```

We will now construct the corresponding QQ plots and histograms. We will start with the Gamma distribution samples:

```
gamma_qq_plot <- gamma_sample_tbl %>%
  ggplot(aes(sample = sample)) +
  geom_qq(
    color = c_palette["C red"],
    size = 0.3
  ) +
  geom_qq_line(
    color = c_palette["C blue"],
    linewidth = 0.5
  ) +
  labs(
    title = "QQ Plot",
```

```
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_krul()

gamma_histogram <- gamma_sample_tbl %>%
  ggplot(aes(x = sample)) +
  geom_histogram(
    bins = 30,
    fill = c_palette["C blue"],
    color = "black",
    alpha = 0.7
  ) +
  labs(
    title = "Histogram",
    x = "Sample Values",
    y = "Density"
  ) +
  theme_krul()

gamma_plot <- gamma_qq_plot + gamma_histogram +
  plot_layout(ncol = 2) +

  plot_annotation(
    title = "Gamma Distribution: QQ Plot and Histogram",
    theme = theme(
      plot.title = element_text(
        size = 24,
        face = "bold"
      )
    )
  )

gamma_plot
```

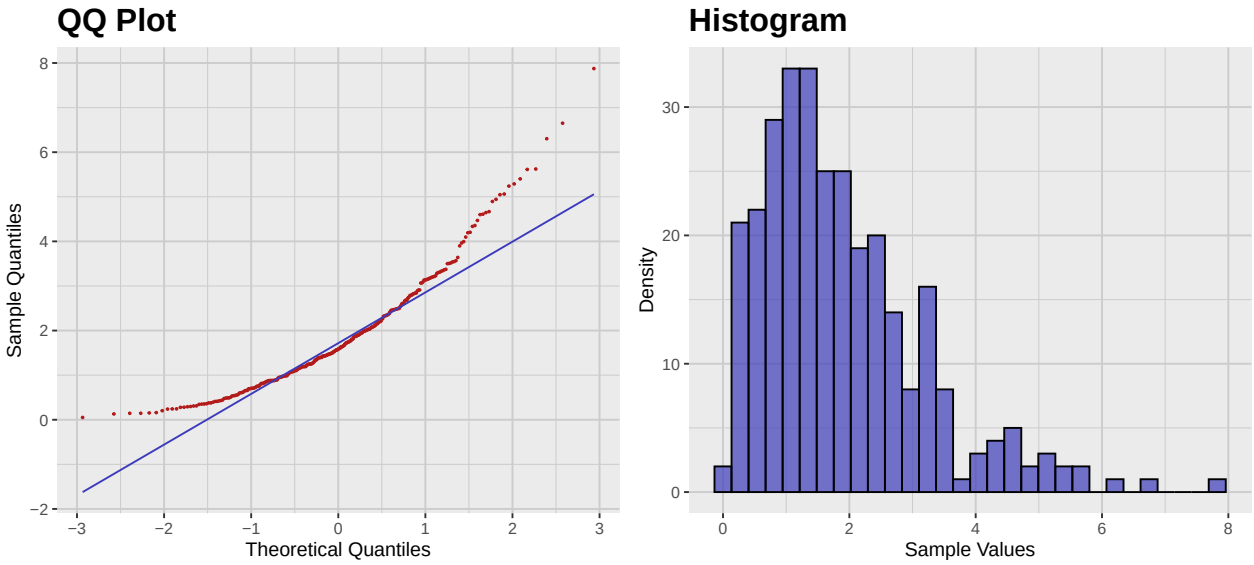

Figure 2.3: QQ plot and histogram of the Gamma distribution samples.

We will now construct the QQ plot and histogram for the Laplace distribution samples:

```
laplace_qq_plot <- laplace_sample_tbl %>%
  ggplot(aes(sample = sample)) +
  geom_qq(
    color = c_palette["C red"],
    size = 0.3
  ) +
  geom_qq_line(
    color = c_palette["C blue"],
    linewidth = 0.5
  ) +
  labs(
    title = "QQ Plot",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_krul()

laplace_histogram <- laplace_sample_tbl %>%
  ggplot(aes(x = sample)) +
  geom_histogram(
    bins = 30,
    fill = c_palette["C blue"],
    color = "black",
    alpha = 0.7
```

```
  ) +
  labs(
    title = "Histogram",
    x = "Sample Values",
    y = "Density"
  ) +
  theme_krul()
laplace_plot <- laplace_qq_plot + laplace_histogram +
  plot_layout(ncol = 2) +

  plot_annotation(
    title = "Laplace Distribution: QQ Plot and Histogram",
    theme = theme(
      plot.title = element_text(
        size = 24,
        face = "bold"
      )
    )
  )
laplace_plot
```

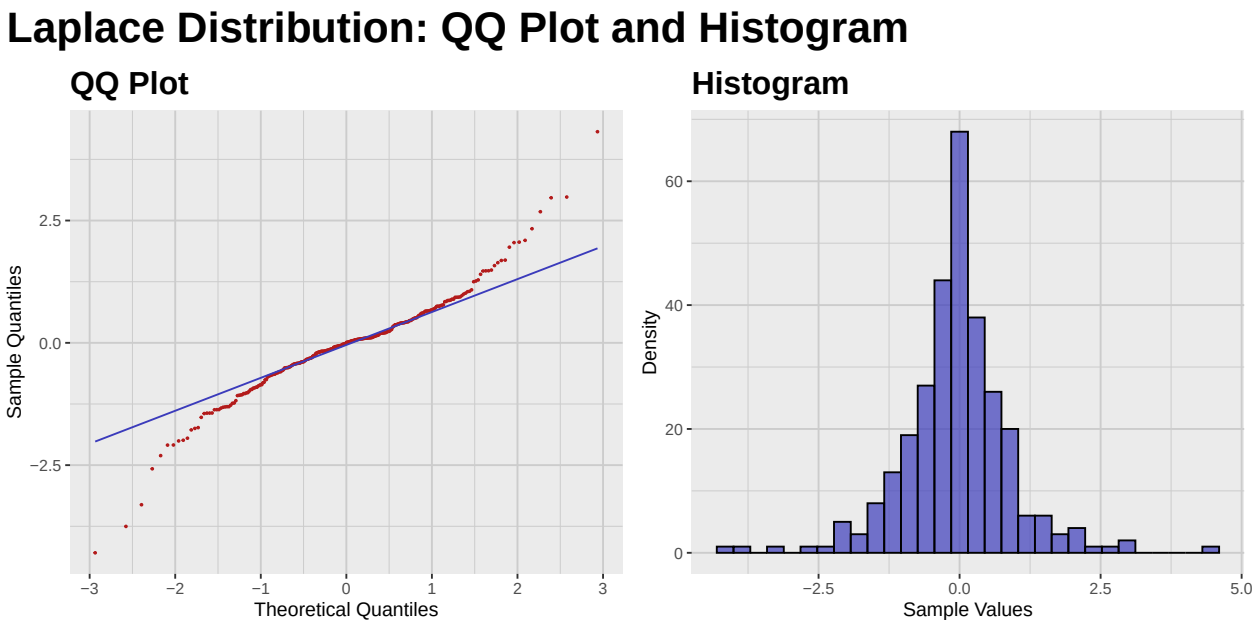

Figure 2.4: QQ plot and histogram of the Laplace distribution samples.

2.4 Sample Skewness and Kurtosis

So far we have defined skewness and kurtosis for random variables. However, in practice, we often work with samples rather than entire populations. Therefore, we need to define sample skewness

and sample kurtosis.

Definition 2.4.1. Let $x_1, x_2, \ldots, x_n$ be a sample of size n. The *sample mean* $\bar{x}$ and *sample variance* s^2 are given by

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

The *sample skewness* g_1 given by

$$g_1 = \frac{n}{(n-1)(n-2)}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^3.$$

The *sample excess kurtosis* g_2 is given by

$$g_2 = \frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}.$$

As an example, we will compute the sample skewness and sample excess kurtosis of the Gamma and Laplace samples we generated earlier. For the Gamma sample we have:

```
# Compute sample skewness and excess kurtosis for Gamma distribution
gamma_sample_tbl %>%
  summarise(
    sample_skewness = skewness(sample, type = 2),
    sample_excess_kurtosis = kurtosis(sample, type = 2) - 3
  ) %>%
glimpse()
```

```
Rows: 1
Columns: 2
$ sample_skewness        <dbl> 1.293772
$ sample_excess_kurtosis <dbl> -0.8356471
```

For the Laplace sample we have:

```
# Compute sample skewness and excess kurtosis for Laplace distribution
laplace_sample_tbl %>%
  summarise(
    sample_skewness = skewness(sample, type = 2),
    sample_excess_kurtosis = kurtosis(sample, type = 2)
  ) %>%
glimpse()
```

```
Rows: 1
Columns: 2
$ sample_skewness        <dbl> -0.07876743
$ sample_excess_kurtosis <dbl> 3.726374
```

2.5 Activity: Assessing Normality

For each of the following distributions, construct a sample of size 300 and compute their associated sample Skewness and Kurtosis. After that, construct and compare their associated QQ plots and histograms.

1. Exponential distribution with rate parameter $\lambda = 1$.
2. Uniform distribution on the interval $[0, 1]$.
3. Beta distribution with shape parameters $\alpha = 2$ and $\beta = 5$.
4. Cauchy distribution with location parameter $x_0 = 0$ and scale parameter $\gamma = 1$.
5. Chi-squared distribution with degrees of freedom $k = 3$.

Chapter 3

Normality Testing

Although QQ plots are a good way to visually assess the normality of a dataset, they can be subjective and imprecise. Therefore, it becomes necessary to develop more objective and precise statistical tests to determine whether the data we are analyzing follows a normal distribution. In this chapter, we will explore some commonly used statistical tests for normality and discuss some transformations that can be performed on our data to make it more normal-like.

3.1 Scaling and Standardization

Definition 3.1.1. An *affine transformation* of a random variable X is a transformation of the form

$$Y = aX + b$$

where a and b are constants.

Remark 3.1.2. In the context of statistics, affine transformations are often refered to as *scaling* (even when they also include a translational component).

Now assume that X is a random variable with mean μ and standard deviation σ. Then the random variable

$$Y = aX + b$$

will have mean $a\mu + b$ and standard deviation $|a|\sigma$. In particular, if we choose $a = 1/\sigma$ and $b = -\mu/\sigma$, then the random variable

$$Z = \frac{X - \mu}{\sigma}$$

will have mean 0 and standard deviation 1. This particular affine transformation is called a *standardization*, and the resulting random variable Z is called the *standardized* version of X. Z is also sometimes called the *z-score* of X.

Remark 3.1.3. Notice that if X is not normally distributed, then Z will *not* be the standard normal distribution. Standardization does not make a non-normal distribution normal.

Observation 3.1.4. Notice that, by definition,

$$\gamma_1(Z) = \mathbb{E}[Z^3] = \frac{\mathbb{E}[(X - \mu)^3]}{\sigma^3} = \gamma_1(X)$$

and

$$\gamma_2(Z) = \mathbb{E}[Z^4] - 3 = \frac{\mathbb{E}[(X-\mu)^4]}{\sigma^4} - 3 = \gamma_2(X).$$