Pokemon Dataset

Analysing 6 Generations of Pokemon

LA - 2 Exploratory Data Analysis

Submitted By:- Abhinandan S Vishwaroop,1NT21IS005, 5 A

Introduction

```
# The Complete Pokémon Dataset is an extensive and detailed collection of information about P
okémon, fictional creatures from the Pokémon franchise. This dataset goes beyond basic detail
s and encompasses a wide array of attributes, including Pokémon characteristics, abilities, b
ase stats, and evolution stages. It provides a comprehensive overview of all Pokémon across v
arious generations, offering a valuable resource for researchers, gamers, and enthusiasts. An
alysts can explore the dataset to uncover patterns, relationships, and trends within the Poké
mon world, making it a powerful tool for strategic gameplay analysis. The inclusion of additi
onal features like Mega Evolution and Legendary status adds depth to the dataset, making it a
comprehensive reference for understanding the intricate details of each Pokémon. In essence,
The Complete Pokémon Dataset serves as an exhaustive and organized repository, enhancing the
understanding and appreciation of the diverse Pokémon universe.
```

I am using a Pokemon Dataset which only has necessary attributes or columns required for the analysis part.

Explanation of each attribute

```
# Name: Each pokemon's name
# Type 1: Properties for pokemon and their moves (each type has its own strengths and weaknes
ses)
# Type 2: Some pokemon have two types
# Total: The sum of all stats below
# HP: Hit points (health)
# Attack: The base modifier for normal attacks
# Defense: The stat for protection against normal attacks
# SP Atk: The base modifier for special attacks
# SP Def: The stat for protection against special attacks
# Speed: Determines which pokemon attacks first in each round
# Generation: The generation of the pokemon
# Legendary: Whether or not pokemon is legendary
```

Task is to demonstrate 10 Visualization Functions

Let’s load the **Pokemon Dataset** into R

```
# Load the Dataset

pokemanz <- read.csv("C:\\Users\\abhiv\\OneDrive\\Desktop\\Pokemon.csv")


# Data Summary

summary(pokemanz)
```

```
##        X.            Name              Type1              Type2
##  Min.   :  1.0   Length:800         Length:800         Length:800
##  1st Qu.:184.8   Class :character   Class :character   Class :character
##  Median :364.5   Mode  :character   Mode  :character   Mode  :character
##  Mean   :362.8
##  3rd Qu.:539.2
##  Max.   :721.0
##      Total             HP             Attack       Defense
##  Min.   :180.0   Min.   :  1.00   Min.   :  5   Min.   :  5.00
##  1st Qu.:330.0   1st Qu.: 50.00   1st Qu.: 55   1st Qu.: 50.00
##  Median :450.0   Median : 65.00   Median : 75   Median : 70.00
##  Mean   :435.1   Mean   : 69.26   Mean   : 79   Mean   : 73.84
##  3rd Qu.:515.0   3rd Qu.: 80.00   3rd Qu.:100   3rd Qu.: 90.00
##  Max.   :780.0   Max.   :255.00   Max.   :190   Max.   :230.00
##      SpAtk            SpDef           Speed          Generation
##  Min.   : 10.00   Min.   : 20.0   Min.   :  5.00   Min.   :1.000
##  1st Qu.: 49.75   1st Qu.: 50.0   1st Qu.: 45.00   1st Qu.:2.000
##  Median : 65.00   Median : 70.0   Median : 65.00   Median :3.000
##  Mean   : 72.82   Mean   : 71.9   Mean   : 68.28   Mean   :3.324
##  3rd Qu.: 95.00   3rd Qu.: 90.0   3rd Qu.: 90.00   3rd Qu.:5.000
##  Max.   :194.00   Max.   :230.0   Max.   :180.00   Max.   :6.000
##  Legendary
##  Mode :logical
##  FALSE:735
##  TRUE :65
##
##
##
```

```
#To find missing values
any(is.na(pokemanz))
```

```
## [1] FALSE
```

It returned **False** means the dataset has no missing values.

```
# Retrieves the names of all the columns present in the dataset

names(pokemanz)
```

```
##  [1] "X."         "Name"       "Type1"      "Type2"      "Total"
##  [6] "HP"         "Attack"     "Defense"    "SpAtk"      "SpDef"
## [11] "Speed"      "Generation" "Legendary"
```

```
# Gives the structure
str(pokemanz)
```

```
## 'data.frame':    800 obs. of  13 variables:
##  $ X.         : int  1 2 3 3 4 5 6 6 6 7 ...
##  $ Name       : chr  "Bulbasaur" "Ivysaur" "Venusaur" "VenusaurMega Venusaur" ...
##  $ Type1      : chr  "Grass" "Grass" "Grass" "Grass" ...
##  $ Type2      : chr  "Poison" "Poison" "Poison" "Poison" ...
##  $ Total      : int  318 405 525 625 309 405 534 634 634 314 ...
##  $ HP         : int  45 60 80 80 39 58 78 78 78 44 ...
##  $ Attack     : int  49 62 82 100 52 64 84 130 104 48 ...
##  $ Defense    : int  49 63 83 123 43 58 78 111 78 65 ...
##  $ SpAtk      : int  65 80 100 122 60 80 109 130 159 50 ...
##  $ SpDef      : int  65 80 100 120 50 65 85 85 115 64 ...
##  $ Speed      : int  45 60 80 80 65 80 100 100 100 43 ...
##  $ Generation: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Legendary : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
# Outputs the initial rows of the dataset

head(pokemanz)
```

```
##   X.                  Name Type1  Type2 Total HP Attack Defense SpAtk SpDef
## 1  1             Bulbasaur Grass Poison   318 45     49      49    65    65
## 2  2               Ivysaur Grass Poison   405 60     62      63    80    80
## 3  3              Venusaur Grass Poison   525 80     82      83   100   100
## 4  3 VenusaurMega Venusaur Grass Poison   625 80    100     123   122   120
## 5  4            Charmander  Fire          309 39     52      43    60    50
## 6  5            Charmeleon  Fire          405 58     64      58    80    65
##   Speed Generation Legendary
## 1    45          1     FALSE
## 2    60          1     FALSE
## 3    80          1     FALSE
## 4    80          1     FALSE
## 5    65          1     FALSE
## 6    80          1     FALSE
```

Load all the necessary libraries required for the analysis.

```
# `ggplot2` is a powerful and flexible data visualization package in R. Developed by Hadley W
ickham, it follows the Grammar of Graphics principles, allowing users to create complex and c
ustomized plots with ease.With a wide range of geoms, themes, and options, `ggplot2` is widel
y used for producing publication-quality plots, making it a preferred choice for data scienti
sts, statisticians, and researchers.

library(ggplot2)

# `repr` in R is a package that provides a flexible and extensible framework for representing
data structures, enabling enhanced object representations in various contexts. It allows user
s to customize the way R objects are displayed, making it particularly useful for improving t
he interpretability and visual representation of complex data structures in interactive compu
ting environments like Jupyter notebooks.

library(repr)

# `dplyr` is a popular R package for data manipulation and transformation. It provides a set
of functions that streamline common data manipulation tasks. With intuitive verbs like `filte
r`, `mutate`, and `group_by`, `dplyr` simplifies code, making it readable and efficient. It's
widely used in data analysis for tasks like filtering rows, creating new variables, and aggre
gating data, enhancing the overall workflow.

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
```

```
# `tidyr` is an R package designed for tidy data manipulation. Developed by Hadley Wickham, i
t complements the `dplyr` package by providing functions like `gather` and `spread` for resha
ping datasets. `tidyr` focuses on organizing data into a tidy format where each variable has
its column, each observation has its row, and each value has its cell. It's instrumental in c
leaning and restructuring data for effective analysis and visualization.

library(tidyr)

# `Hmisc` is an R package designed for advanced statistical analysis and data exploration. It
extends the functionality of base R by providing additional tools for handling missing data,
creating informative summary statistics, and conducting advanced regression modeling. `Hmisc`
is particularly useful in biomedical and clinical research for its comprehensive set of funct
ions addressing various aspects of statistical analysis and data manipulation.

library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

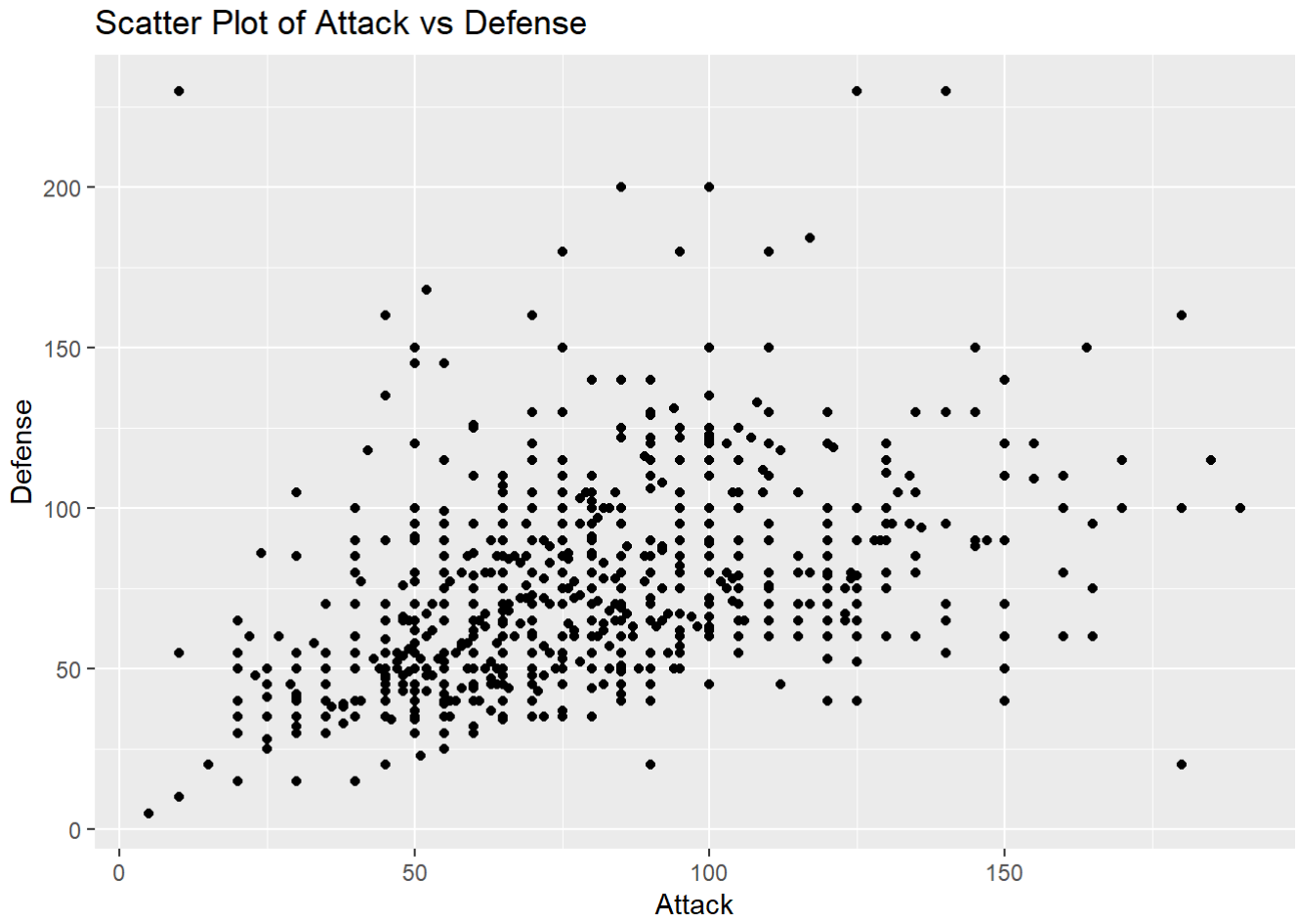
```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# `corrplot` is an R package for visualizing correlation matrices. It provides functions to c
reate visually appealing and informative correlation plots. `corrplot` is widely used for exp
loring relationships between variables by displaying color-coded correlation coefficients. It
supports various customization options, making it easy to interpret complex correlation struc
tures in datasets, aiding researchers and analysts in making data-driven decisions.

library(corrplot)
```

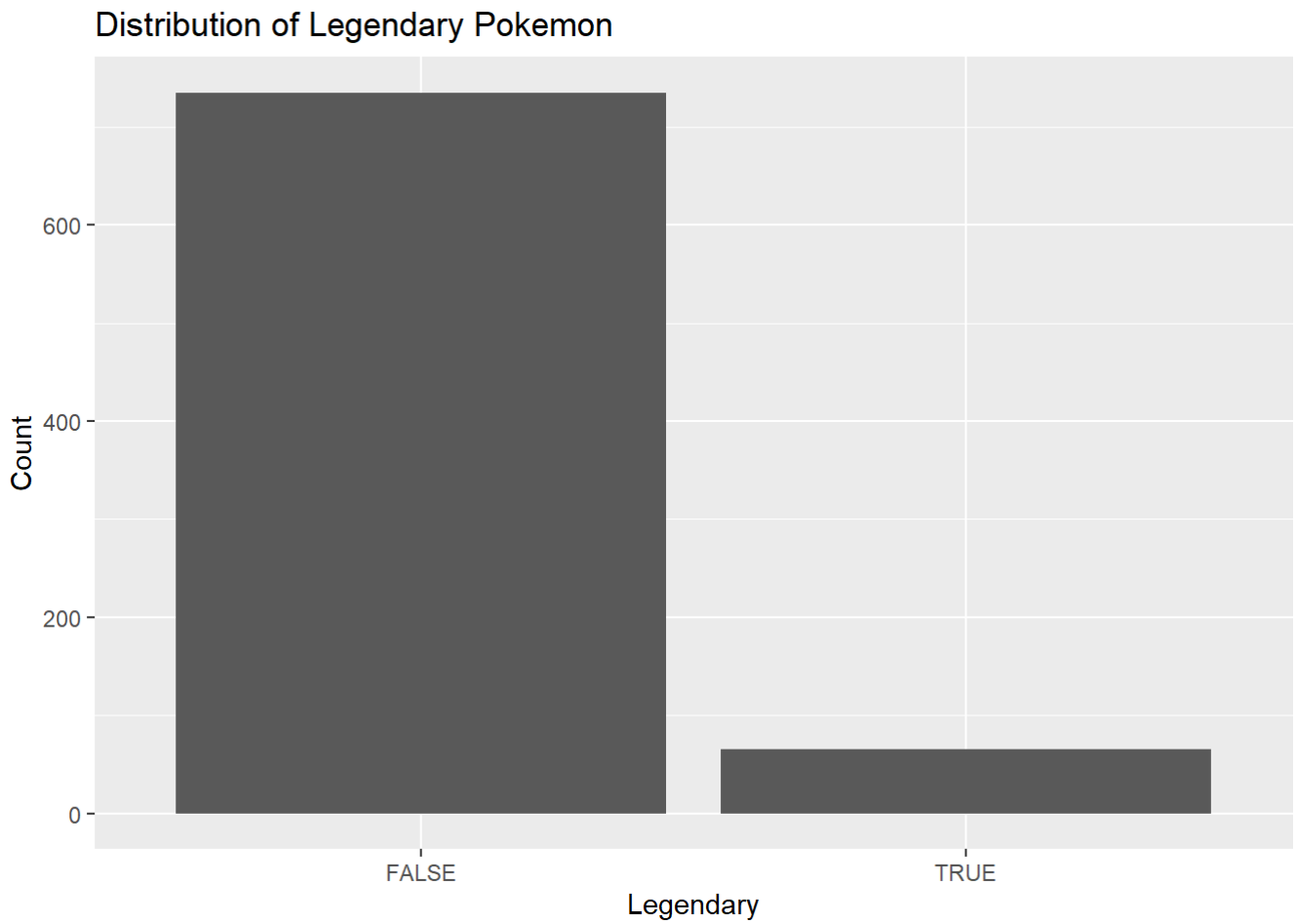
```
## corrplot 0.92 loaded
```

Basic Plots

```
# Scatter Plot of Attack vs Defense

ggplot(data = pokemanz, aes(x = Attack, y = Defense)) +
  geom_point() +
  labs(title = "Scatter Plot of Attack vs Defense", x = "Attack", y = "Defense")
```



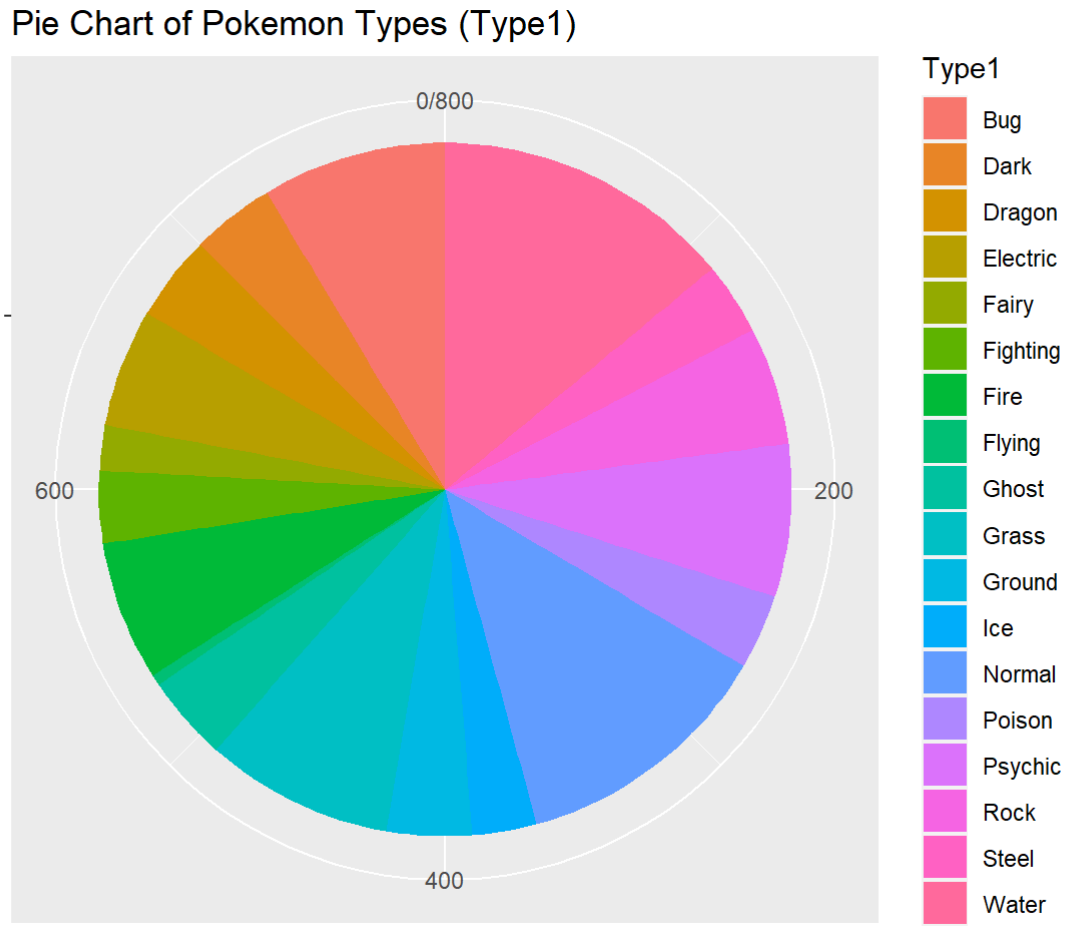

```
# Distribution of Legendary Pokemon using BarPlot

ggplot(data = pokemanz, aes(x = Legendary)) +
  geom_bar() +
  labs(title = "Distribution of Legendary Pokemon", x = "Legendary", y = "Count")
```



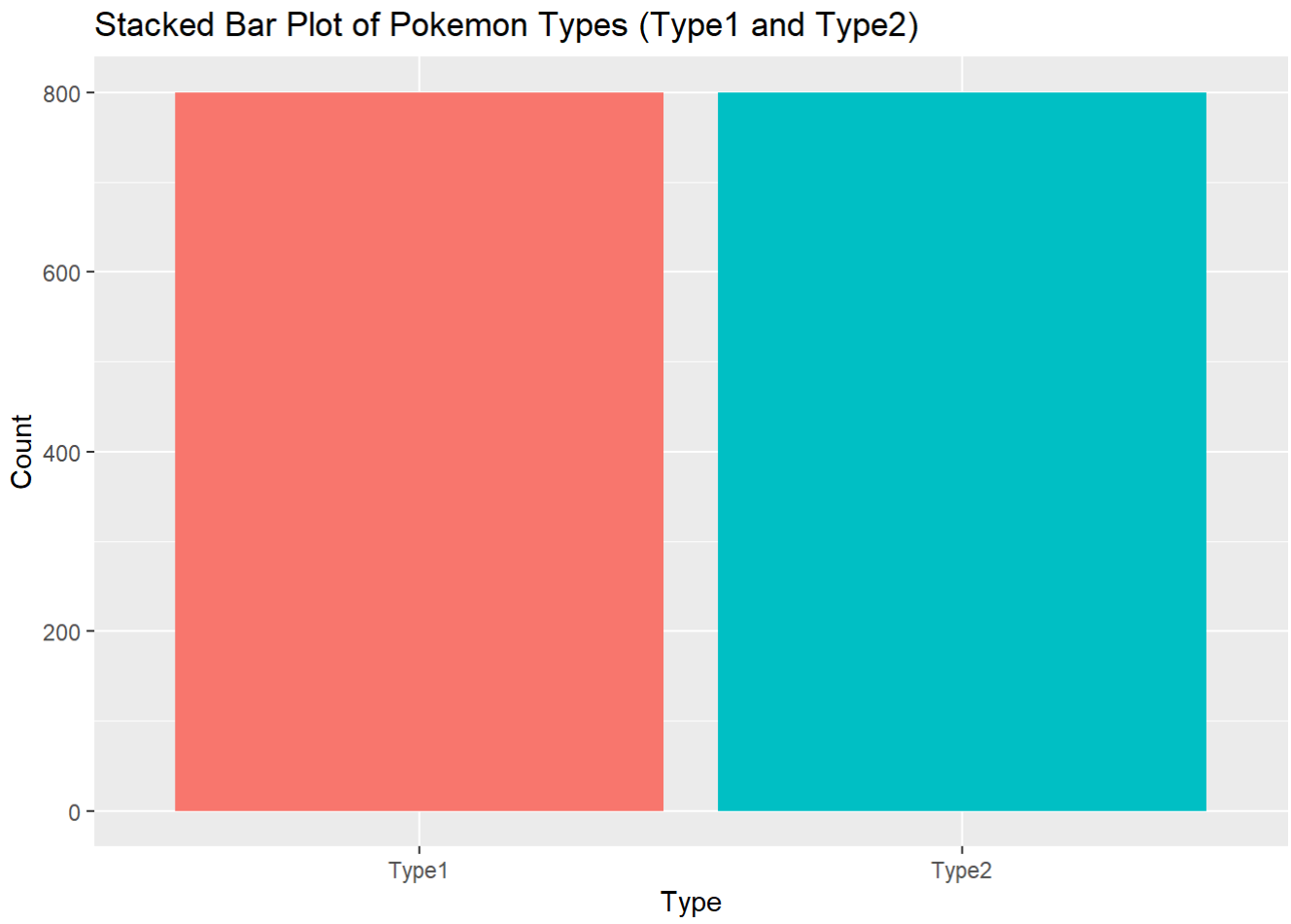
Analysis

We can observe that most of the pokemons are not LEGENDARY

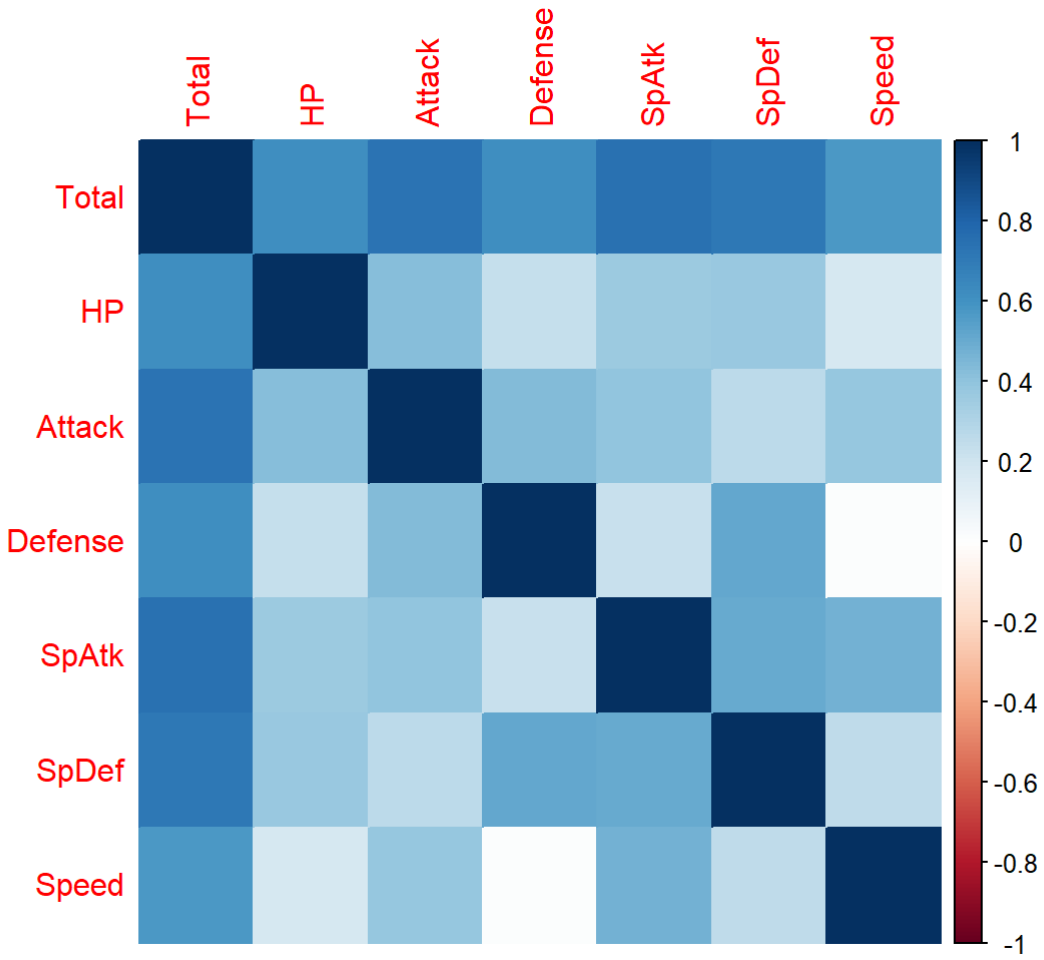
```
# Pie Chart of Pokemon Types (Type1)

type_counts <- pokemanz %>% count(Type1)
ggplot(data = type_counts, aes(x = "", y = n, fill = Type1)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Pie Chart of Pokemon Types (Type1)", x = NULL, y = NULL)
```




```
# Stacked Bar Plot of Pokemon Types (Type1 and Type2)

pokemanz_long <- pokemanz %>% gather(key = "Type", value = "Value", Type1, Type2)
ggplot(data = pokemanz_long, aes(x = Type, fill = Type)) +
  geom_bar() +
  labs(title = "Stacked Bar Plot of Pokemon Types (Type1 and Type2)", x = "Type", y = "Coun
t") +
  theme(legend.position="none")
```




```
# Correlation Heatmap of Numeric Variables

cor_matrix <- cor(pokemanz[, c("Total", "HP", "Attack", "Defense", "SpAtk", "SpDef", "Spee
d")])
corrplot(cor_matrix, method = "color")
```



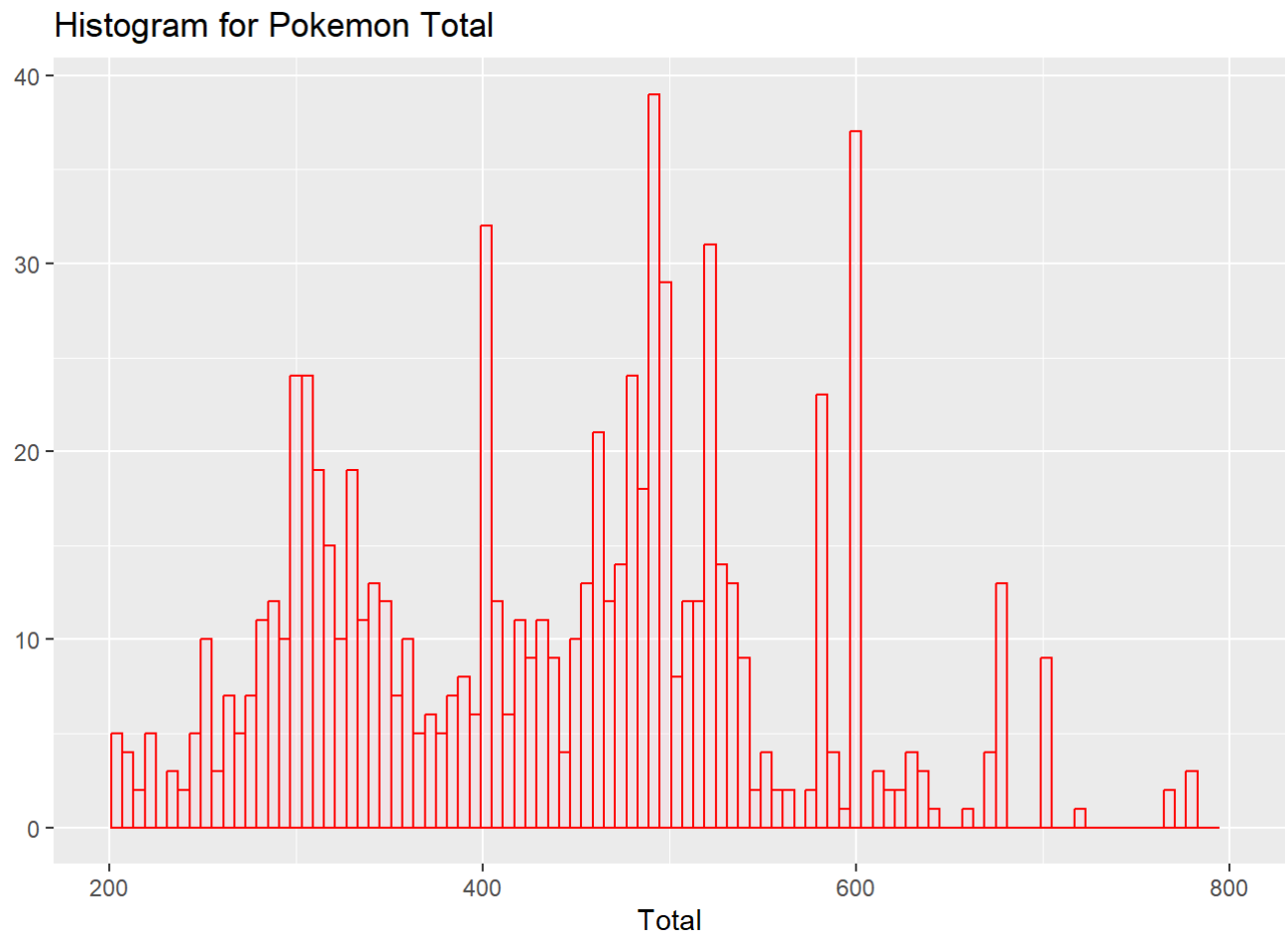
Some other Visualizations:-

```
# The below chunk of code produces a histogram for Total attribute

qplot(pokemanz$Total, geom ="histogram", binwidth = 6, main = "Histogram for Pokemon Total",
      xlab = "Total", fill = I("#ffc8dd"), col = I("red"), alpha = I(.2),
      xlim = c(200, 800))
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

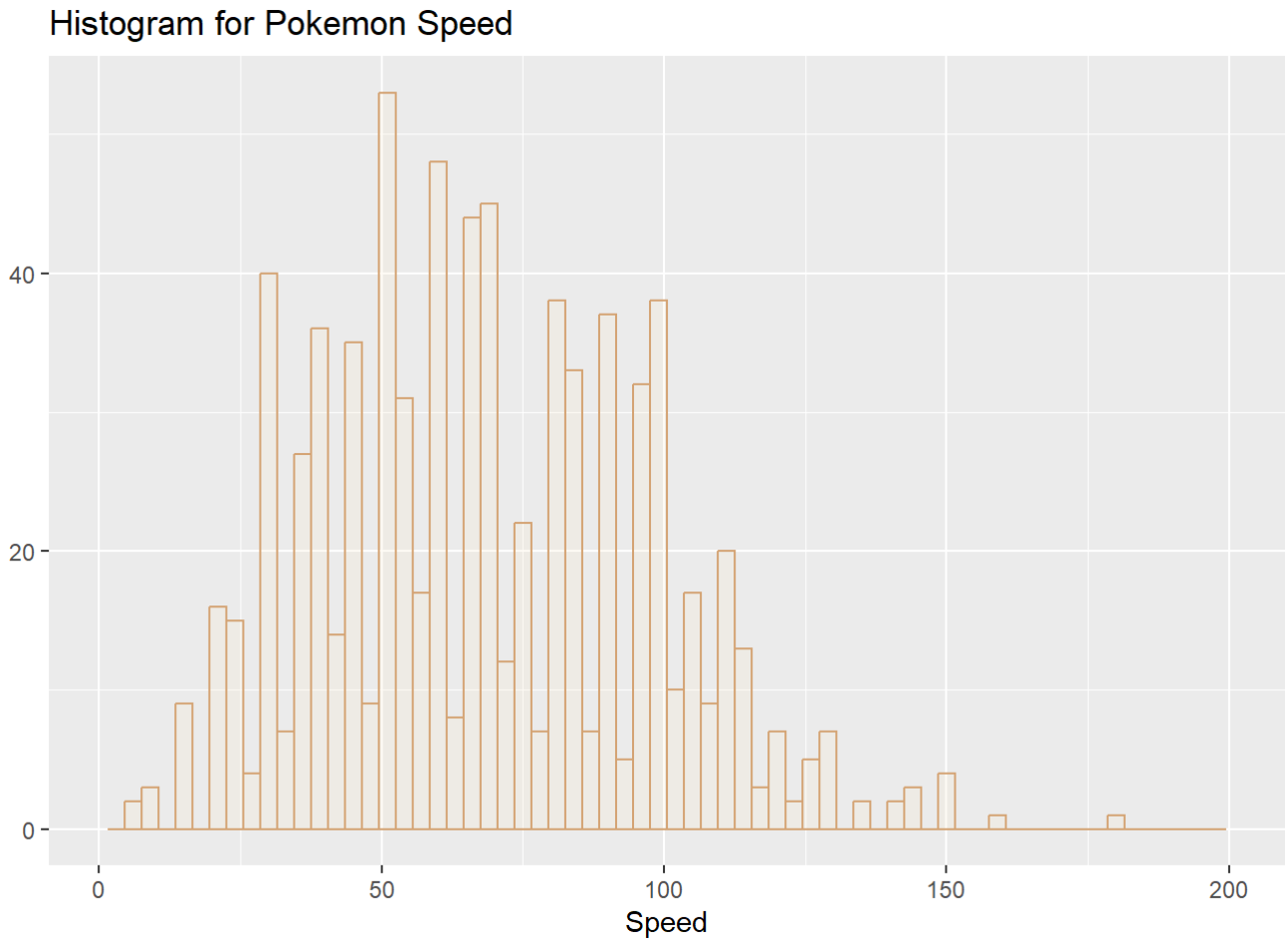
```
## Warning: Removed 7 rows containing non-finite values (`stat_bin()`).
```

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



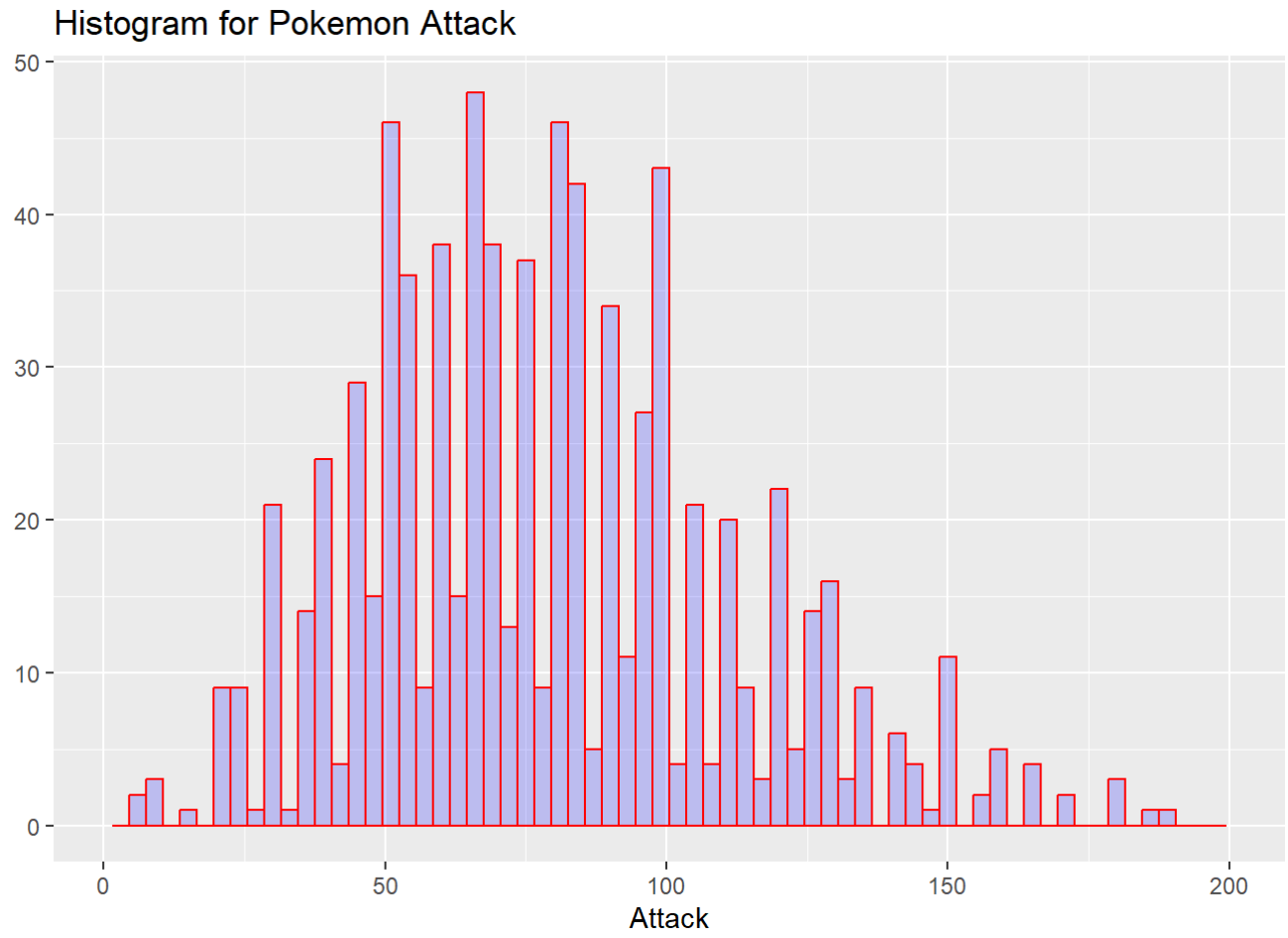

```
# # The below chunk of code produces a histogram for Speed attribute

qplot(pokemanz$Speed, geom ="histogram", binwidth = 3, main = "Histogram for Pokemon Speed",
      xlab = "Speed", fill = I("#faedcd"), col = I("#d4a373"), alpha = I(.2),
      xlim = c(1, 200))
```

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



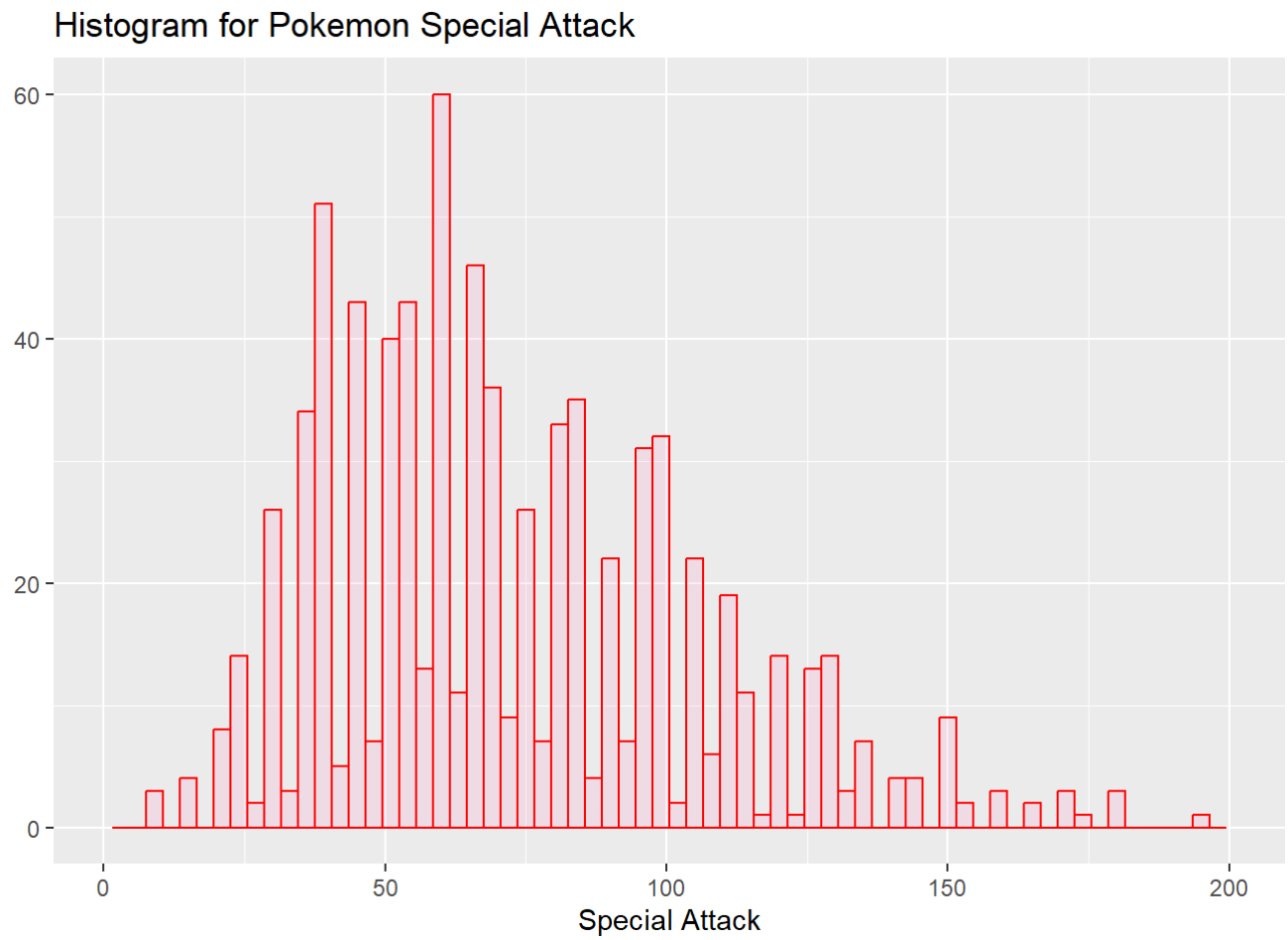

```
# # The below chunk of code produces a histogram for Attack attribute

qplot(pokemanz$Attack, geom ="histogram", binwidth = 3, main = "Histogram for Pokemon Attac
k",
      xlab = "Attack", fill = I("blue"), col = I("red"), alpha = I(.2),
      xlim = c(1, 200))
```

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




```
# The below chunk of code produces a histogram for Special Attack attribute

qplot(pokemanz$SpAtk, geom ="histogram", binwidth = 3, main = "Histogram for Pokemon Special
Attack",
      xlab = "Special Attack", fill = I("#ff99c8"), col = I("red"), alpha = I(.2),
      xlim = c(1, 200))
```

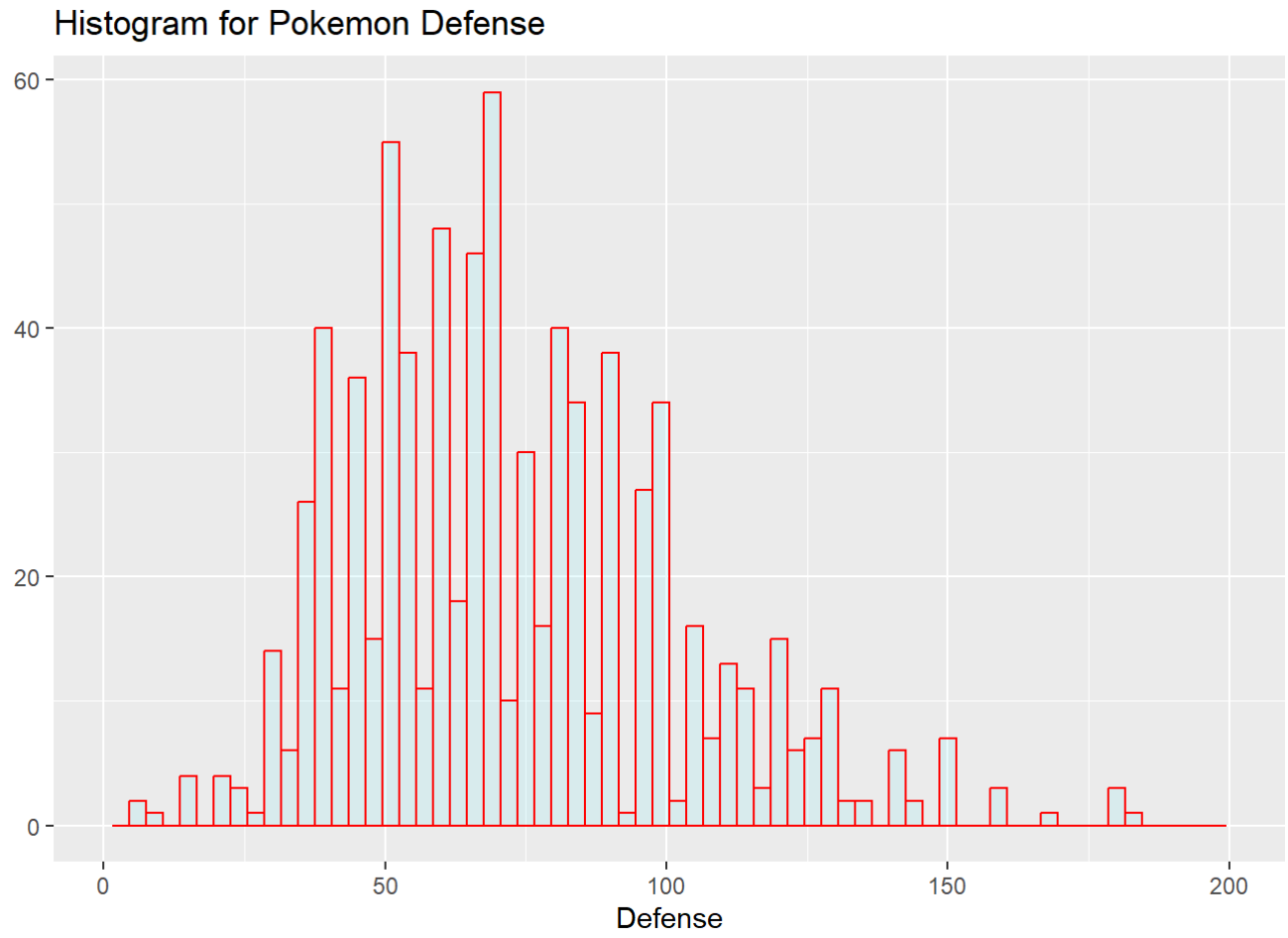
```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```




```
# The below chunk of code produces a histogram for Defense attribute

qplot(pokemanz$Defense, geom ="histogram", binwidth = 3, main = "Histogram for Pokemon Defens
e",
      xlab = "Defense", fill = I("#8eecf5"), col = I("red"), alpha = I(.2),
      xlim = c(1, 200))
```

```
## Warning: Removed 3 rows containing non-finite values (`stat_bin()`).
```

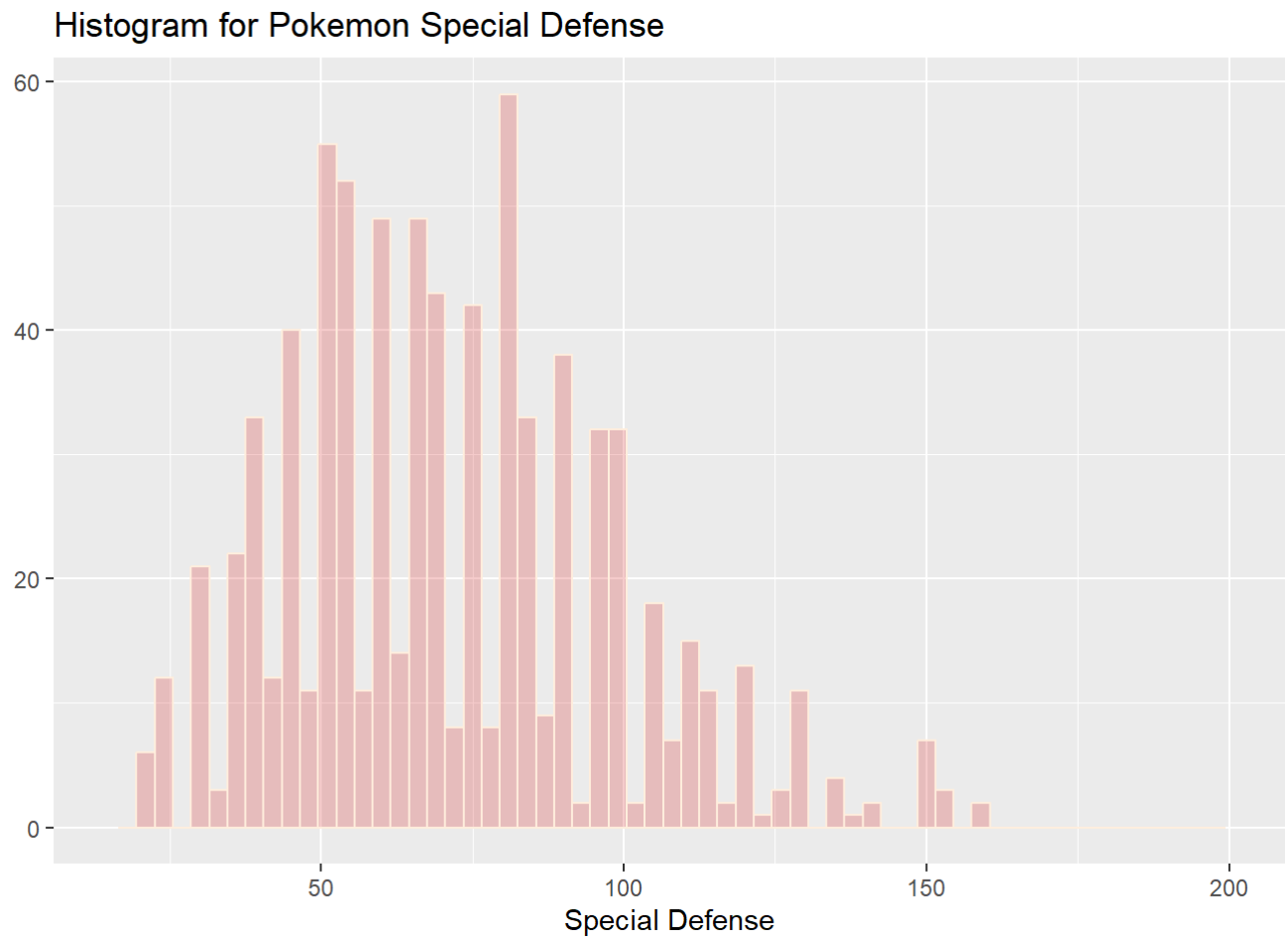
```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```




```
# The below chunk of code produces a histogram for Special Defense attribute

qplot(pokemanz$SpDef, geom ="histogram", binwidth = 3, main = "Histogram for Pokemon Special
Defense",
      xlab = "Special Defense", fill = I("#d00000"), col = I("#ffeedd"), alpha = I(.2),
      xlim = c(15, 200))
```

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

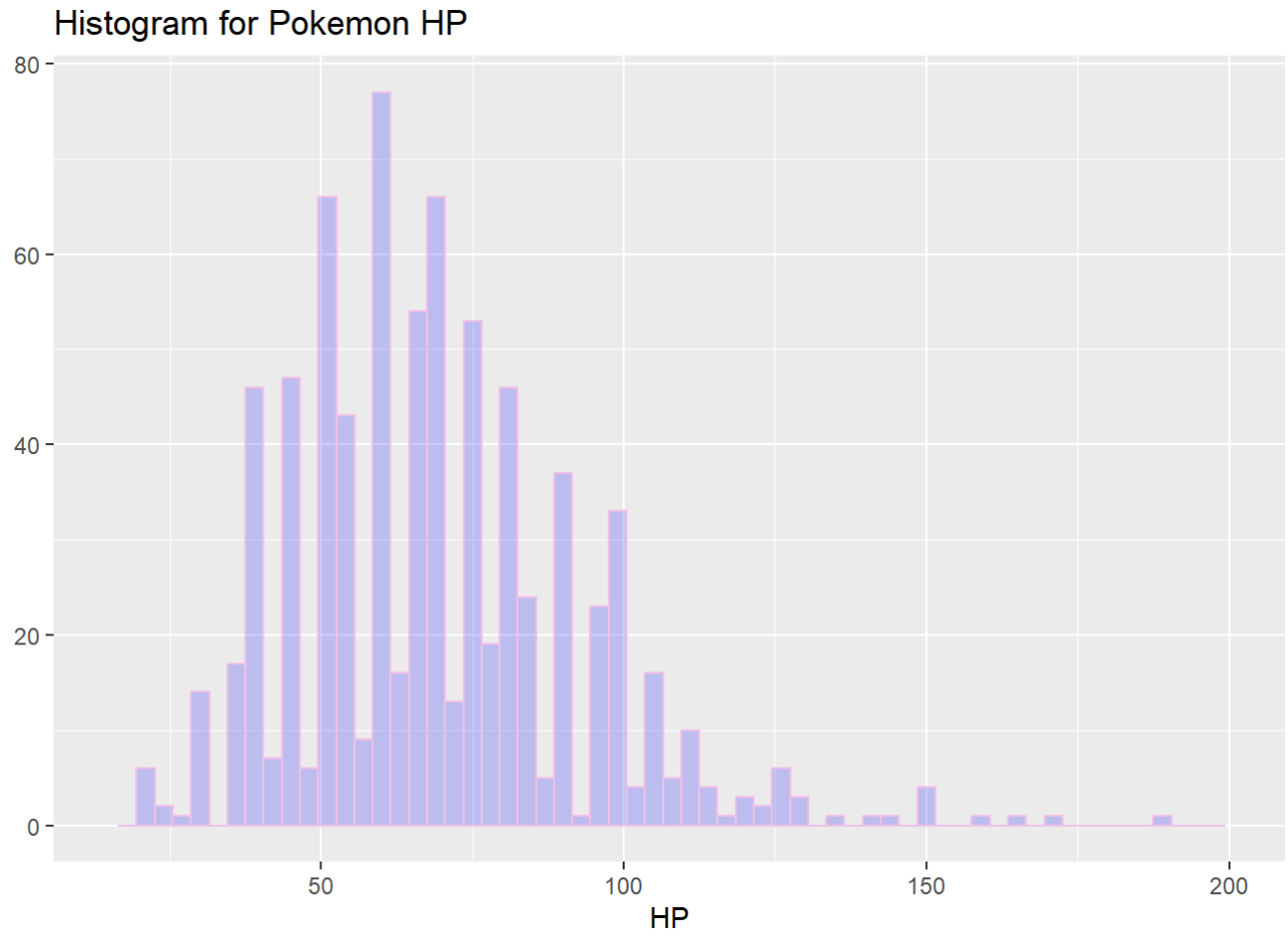
```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```




```
# The below chunk of code produces a histogram for Pokemon HP attribute

qplot(pokemanz$HP, geom ="histogram", binwidth = 3, main = "Histogram for Pokemon HP",
      xlab = "HP", fill = I("blue"), col = I("#f1c0e8"), alpha = I(.2),
      xlim = c(15, 200))
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).
```

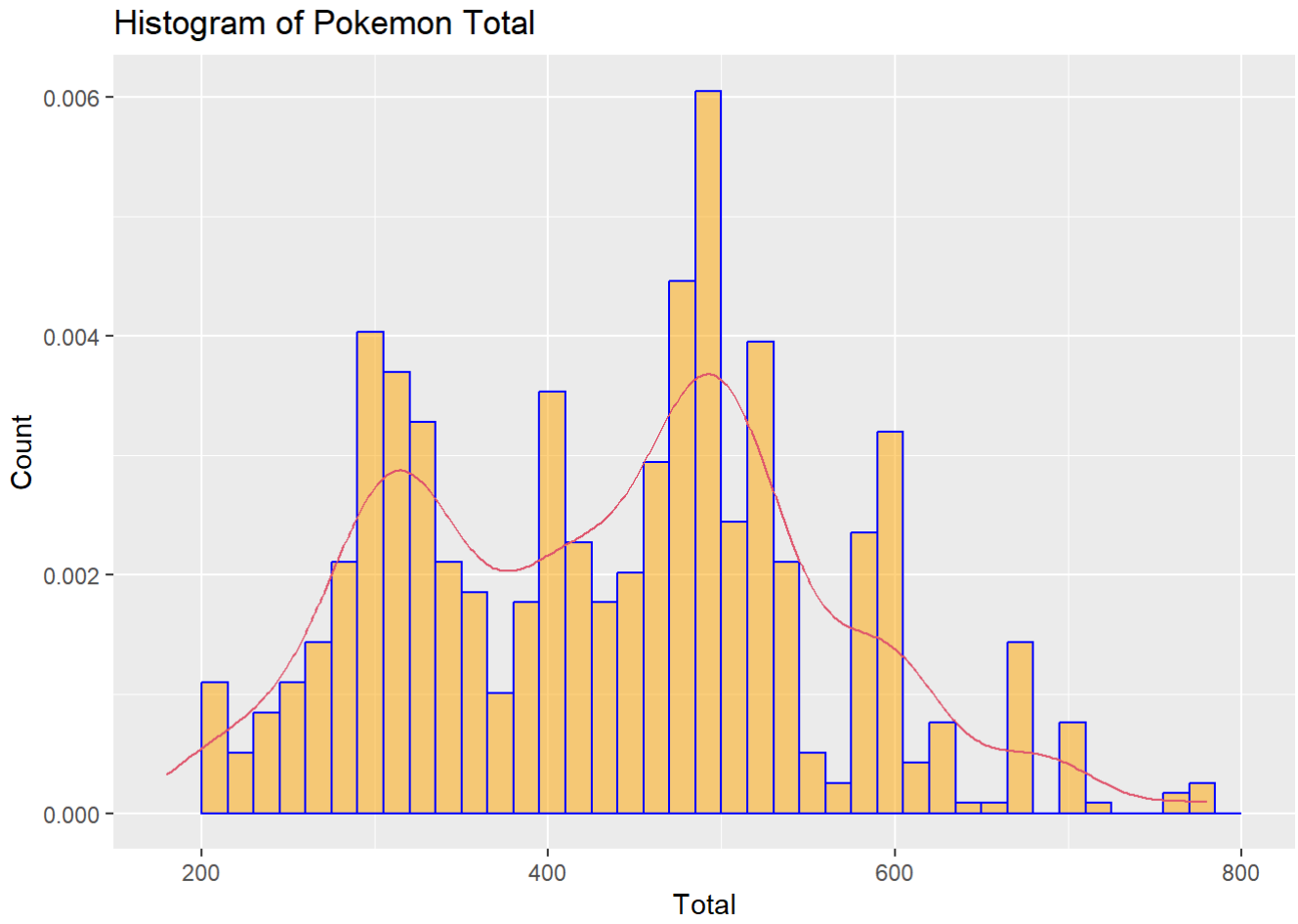
```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



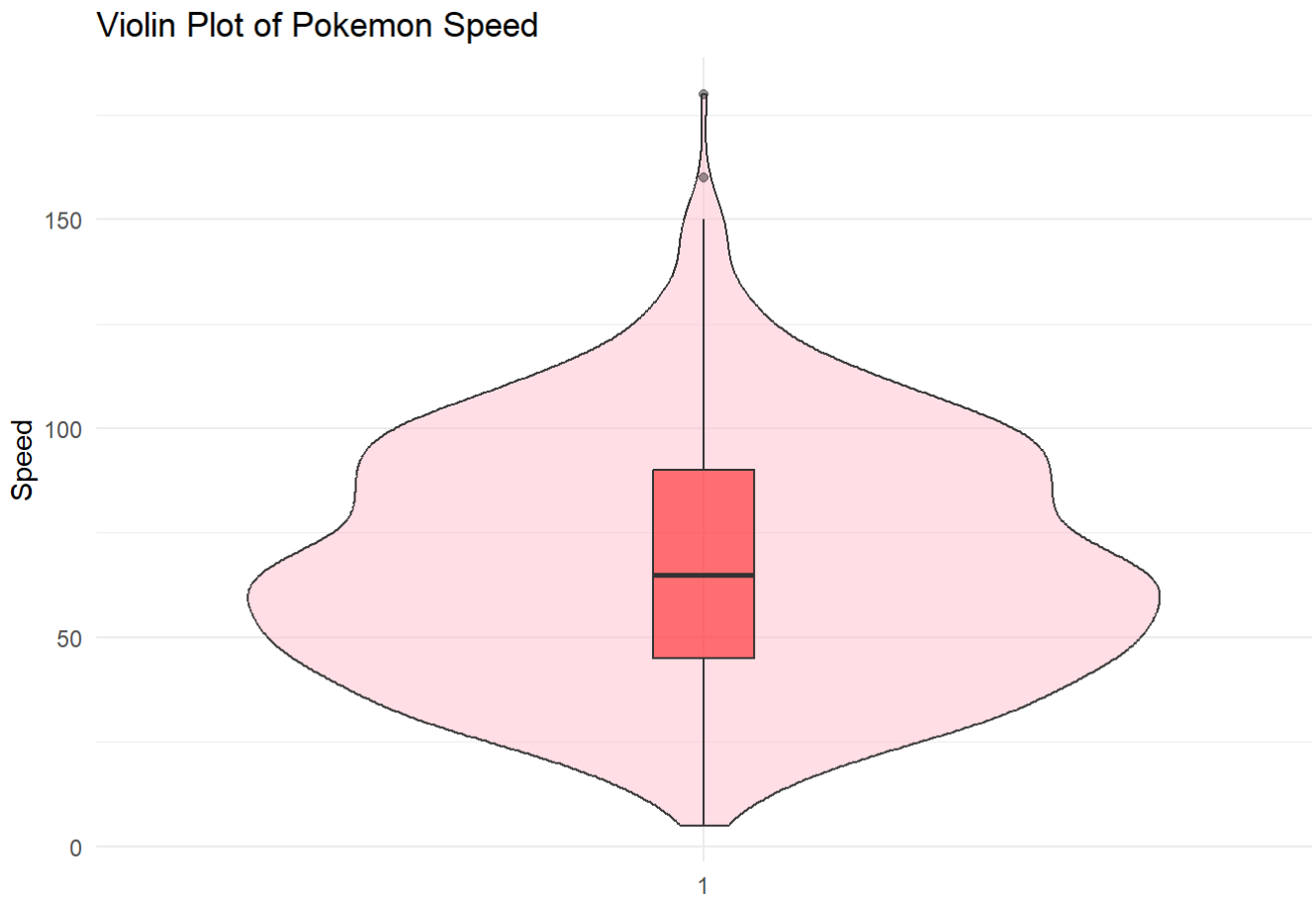

```
# The given code creates a histogram and density plot for the "Total" attribute in the "pokem
anz" dataset using the ggplot2 package in R.

ggplot(data=pokemanz, aes(pokemanz$Total)) +
  geom_histogram(aes(y=..density..),
                 breaks = seq(200, 800, by = 15),
                 col = "blue",
                 fill = "orange",
                 alpha = .5) +
  geom_density(col = 2) +
  labs(title = "Histogram of Pokemon Total") +
  labs(x = "Total", y = "Count")
```

```
## 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.
```

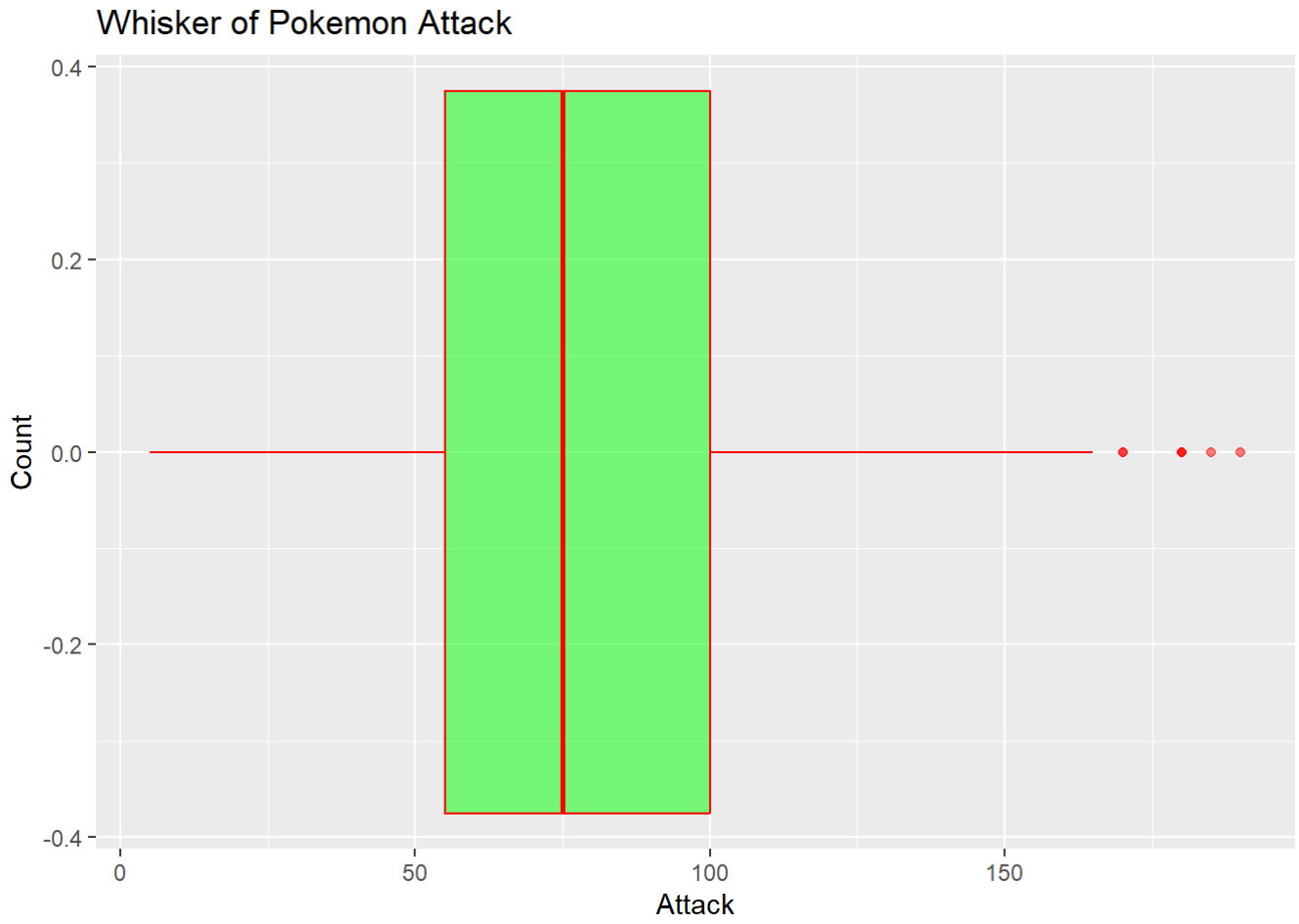
```
## Warning: Use of `pokemanz$Total` is discouraged.
## ℹ Use `Total` instead.
## Use of `pokemanz$Total` is discouraged.
## ℹ Use `Total` instead.
```



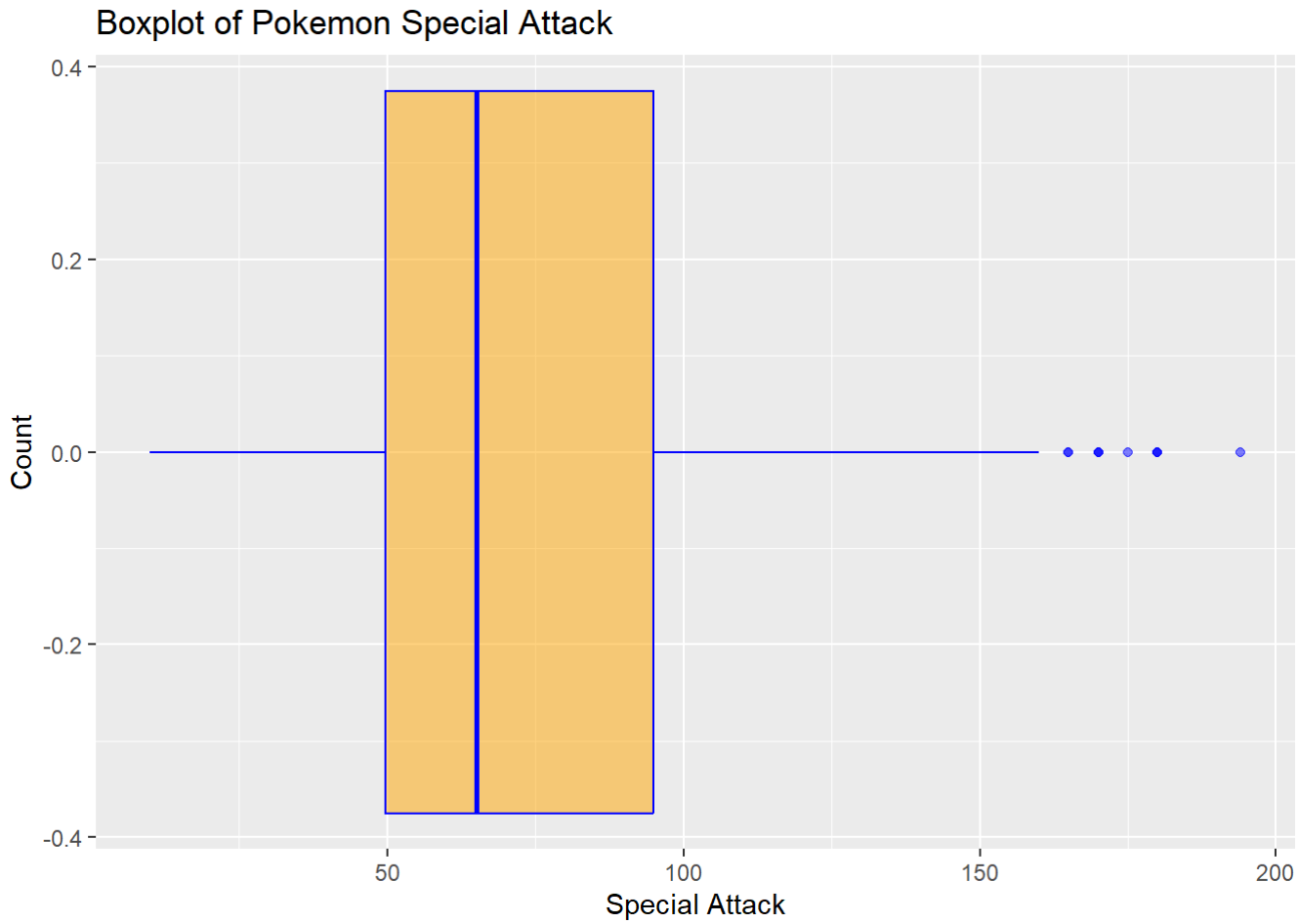

```
# The given code produces a combined plot with both a violin plot and a box plot to visualize
the distribution of the 'Speed' variable in the Pokémon dataset.

ggplot(data = pokemanz, aes(x = factor(1), y = Speed)) +
  geom_violin(fill = "pink", alpha = 0.5) +
  geom_boxplot(width = 0.1, fill = "red", alpha = 0.5) +
  labs(title = "Violin Plot of Pokemon Speed") +
  labs(x = "", y = "Speed") +
  theme_minimal()
```




```
# The provided R code creates a boxplot for the "Attack" attribute in the "pokemanz" dataset
using the ggplot2 package.

ggplot(data = pokemanz, aes(x = Attack)) +
  geom_boxplot(col = "red", fill = "green", alpha = 0.5) +
  labs(title = "Whisker of Pokemon Attack") +
  labs(x = "Attack", y = "Count")
```

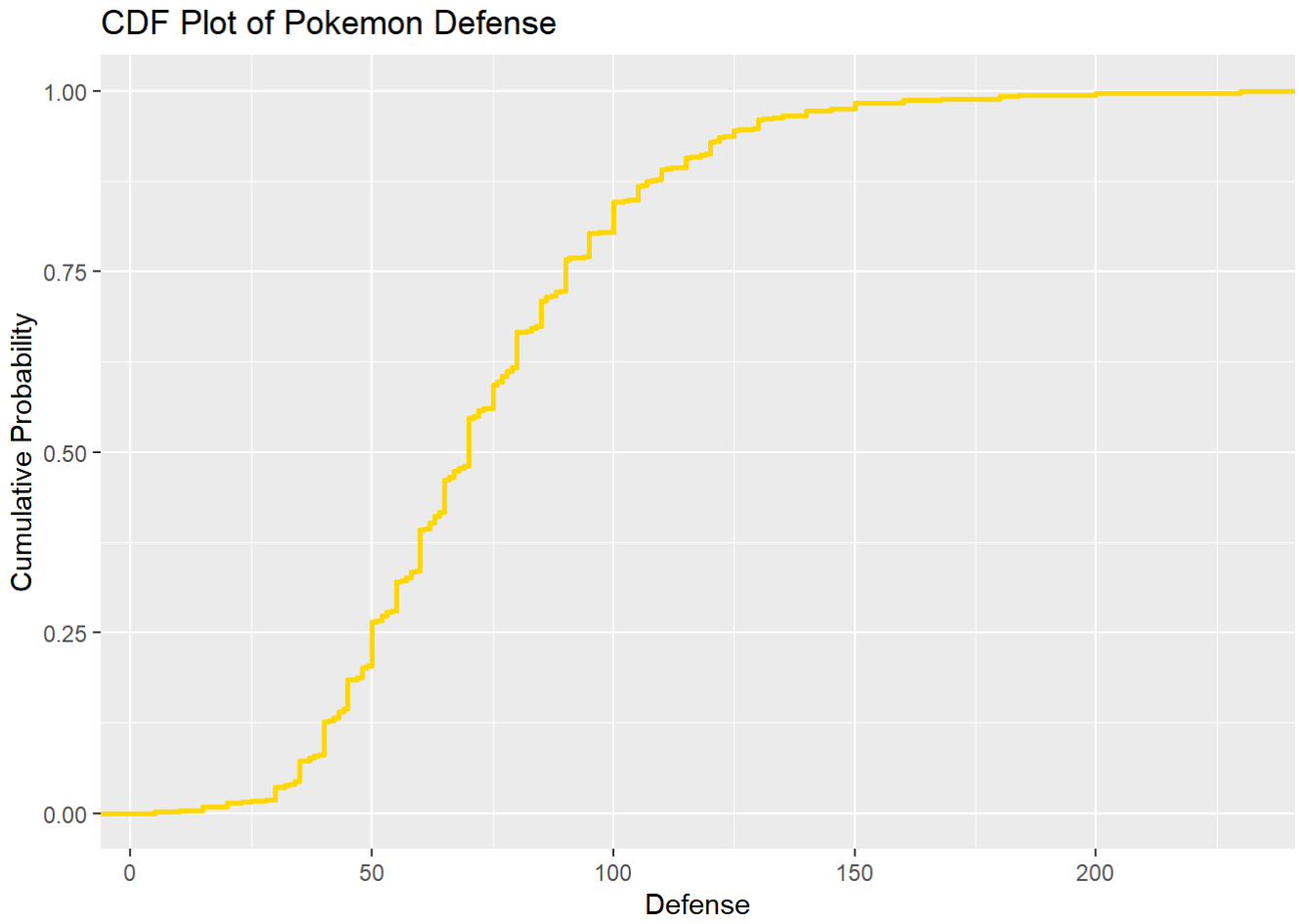



```
#The provided R code generates a boxplot to visualize the distribution of the "Special Attac
k" attribute in the Pokémon dataset.

ggplot(data = pokemanz, aes(x = SpAtk)) +
  geom_boxplot(col = "blue", fill = "orange", alpha = 0.5) +
  labs(title = "Boxplot of Pokemon Special Attack") +
  labs(x = "Special Attack", y = "Count")
```



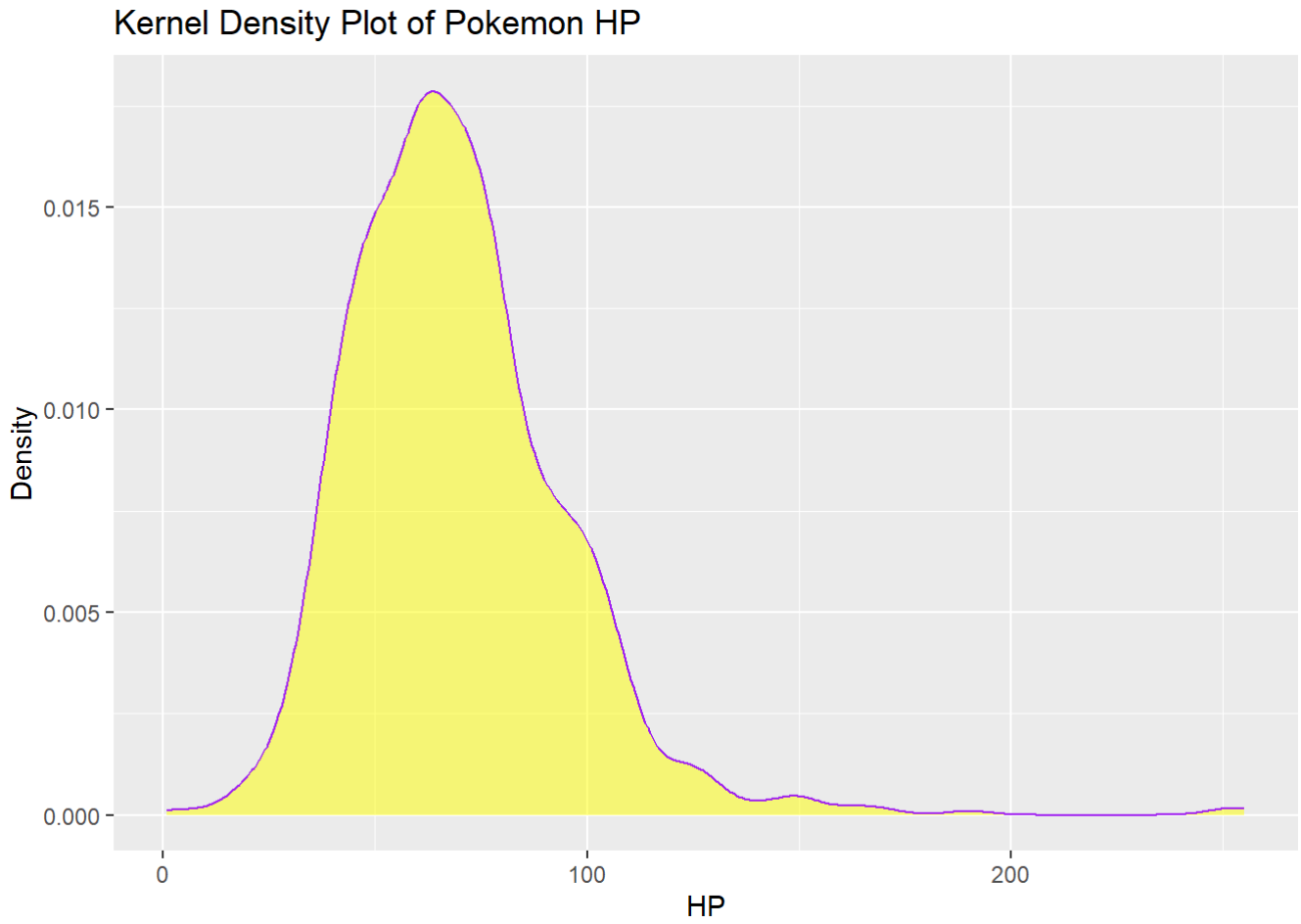

```
#  Cumulative Distribution Function (CDF) plot represents the cumulative probability distribu
tion of a continuous random variable. In the context of data visualization, a CDF plot shows
the probability that a random variable takes on a value less than or equal to a given value.

# In the CDF plot:

# The x-axis represents the values of the variable.
# The y-axis represents the cumulative probability that the variable is less than or equal to
the corresponding x-axis value.

ggplot(data = pokemanz, aes(x = Defense)) +
  stat_ecdf(geom = "step", col = "gold", size = 1) +
  labs(title = "CDF Plot of Pokemon Defense") +
  labs(x = "Defense", y = "Cumulative Probability")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# A kernel density plot, often referred to as a kernel density estimate (KDE), is a non-param
etric way to estimate the probability density function of a continuous random variable.

ggplot(data = pokemanz, aes(x = pokemanz$HP)) +
  geom_density(col = "purple", fill = "yellow", alpha = 0.5) +
  labs(title = "Kernel Density Plot of Pokemon HP") +
  labs(x = 'HP', y = "Density")
```

```
## Warning: Use of `pokemanz$HP` is discouraged.
## ℹ Use `HP` instead.
```



I am using ***density plots and facet wraps*** to analyze each attribute.

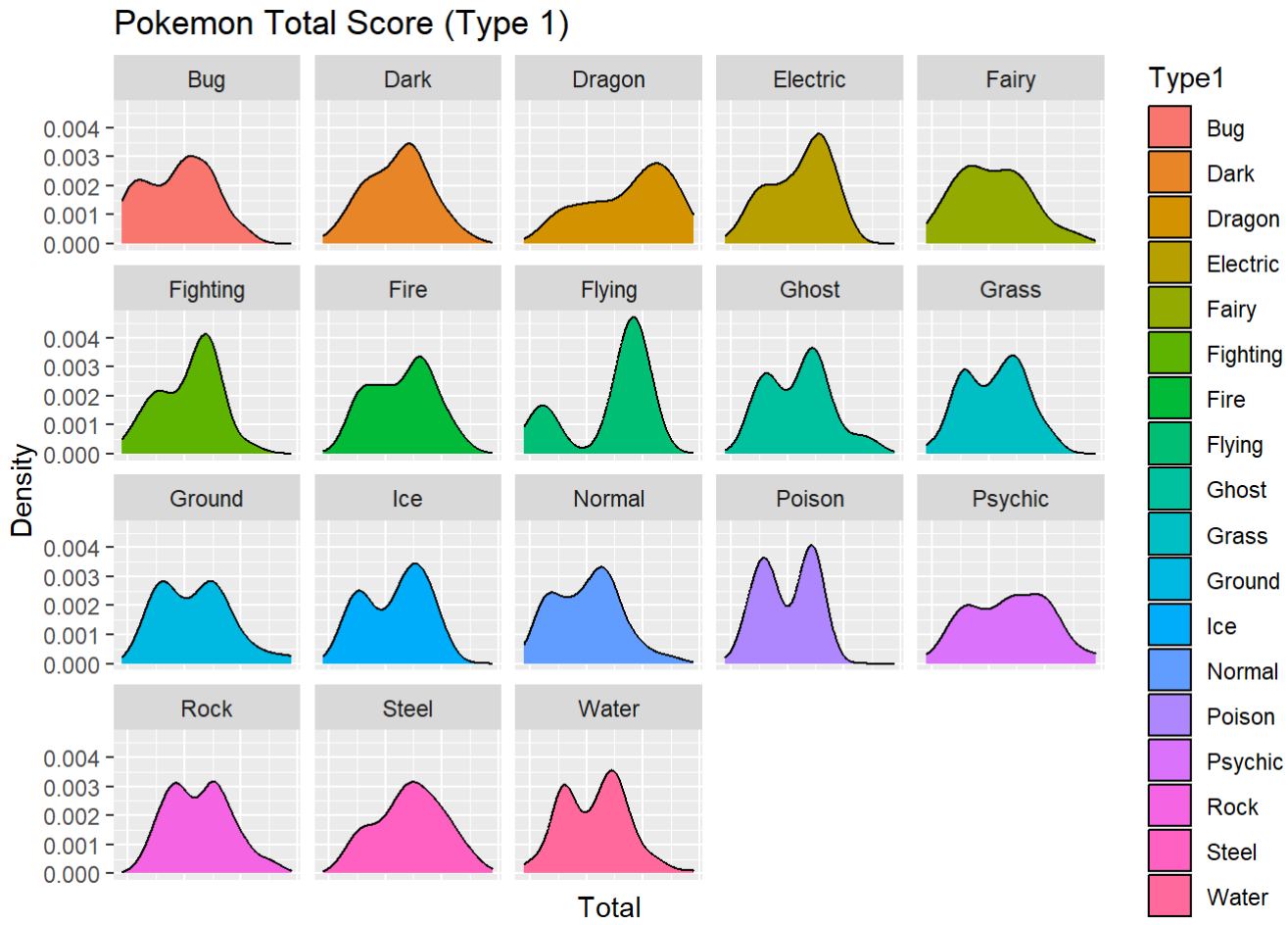
facet_wrap() is a function in ggplot2 (an R package for creating static, animated, and interactive visualizations) that allows you to create a grid of small multiples (facets) based on a categorical variable. It's a convenient way to split your data into subsets and create separate plots for each subset, displayed in a grid.

facet_wrap(~variable)

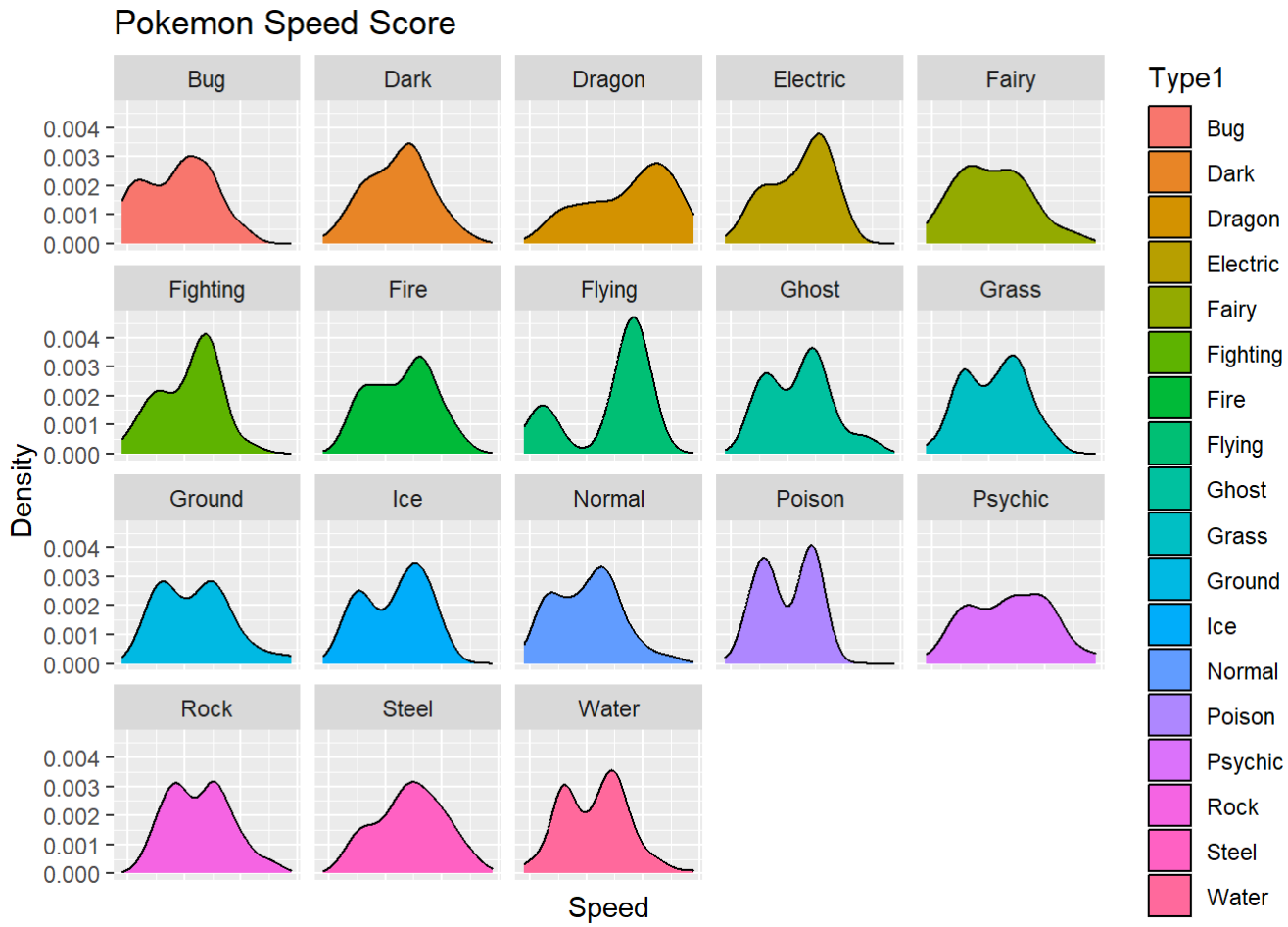
The tilde (~) symbol is used to specify the variable you want to use for faceting.

variable is the categorical variable based on which you want to create separate plots.

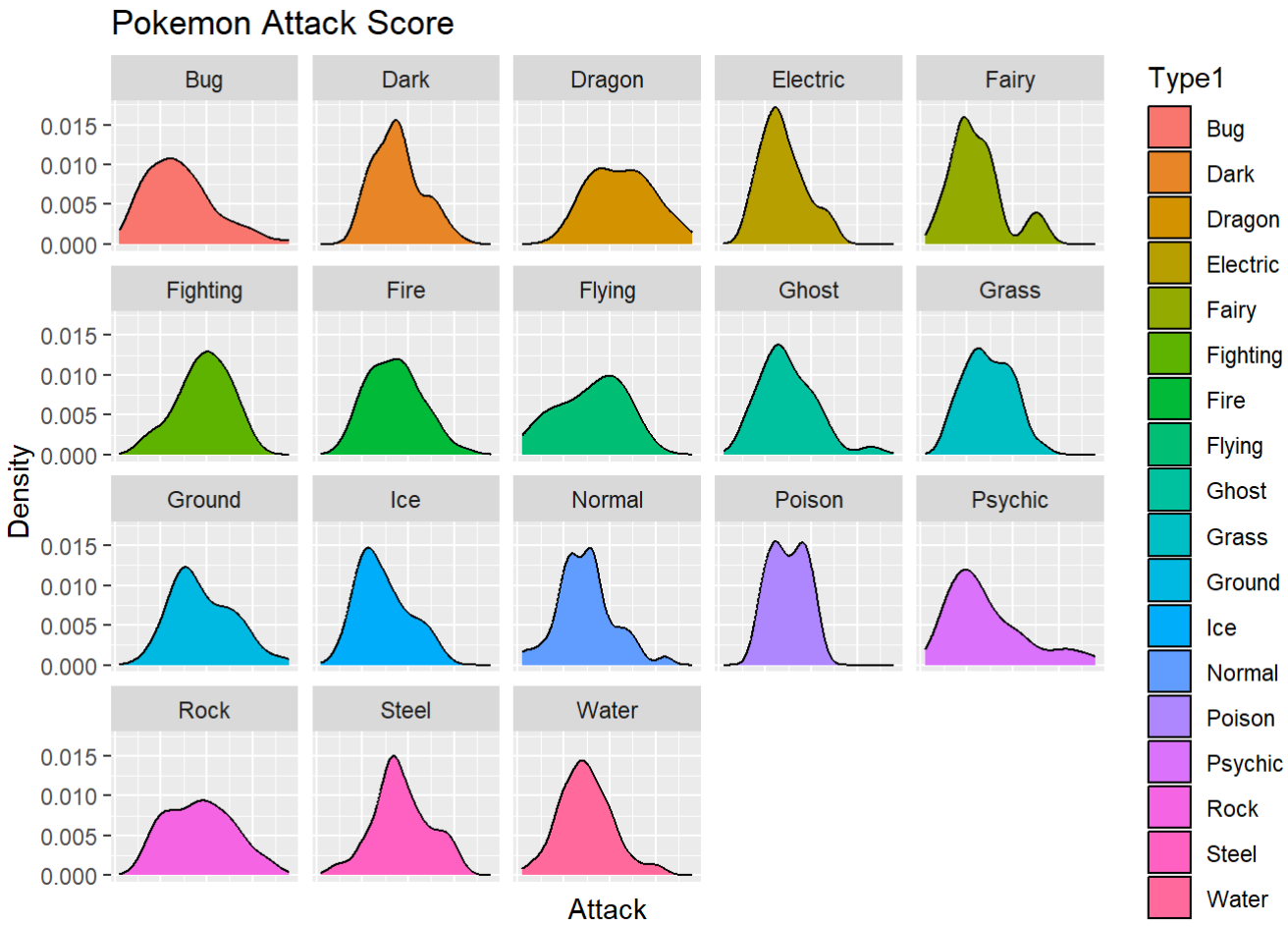
```
# This R code generates a faceted density plot using the ggplot2 package.
# This code creates a set of density plots using ggplot2 in R for the Total scores of Pokemo
n, grouped by the Type1 variable.

pokemanz_plot01 <- ggplot(pokemanz, aes(x=Total, fill=Type1)) + geom_density(alpha = 1)
pokemanz_plot01 <- pokemanz_plot01 +
  facet_wrap(~Type1) +
  labs(x = "Total", y = "Density", title = "Pokemon Total Score (Type 1)") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

pokemanz_plot01
```



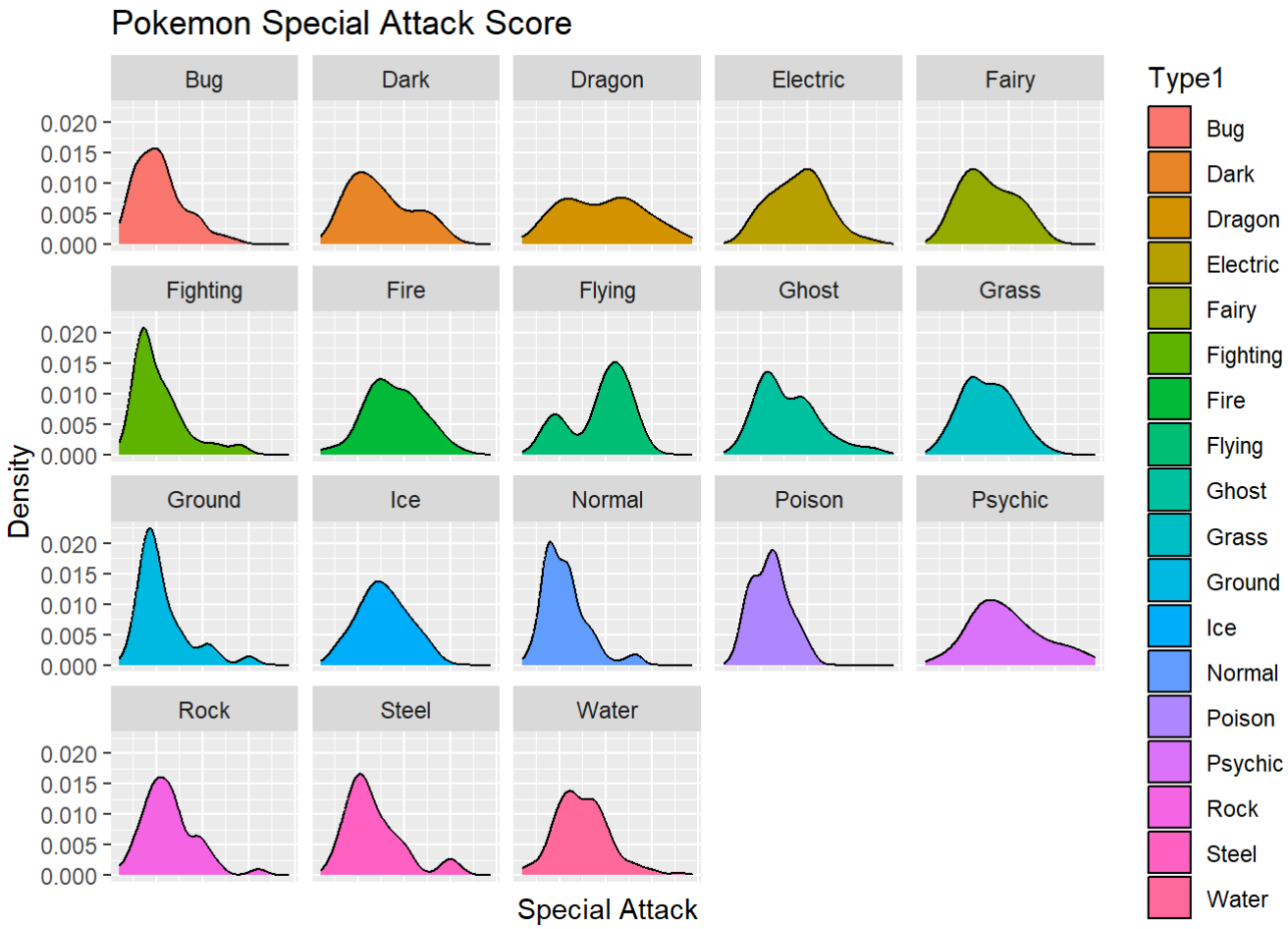

```
pokemanz_plot02 <- ggplot(pokemanz, aes(x=Total, fill=Type1)) + geom_density(alpha = 1)
pokemanz_plot02<- pokemanz_plot02 +
  facet_wrap(~Type1) +
  labs(x = "Speed", y = "Density", title ="Pokemon Speed Score") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

pokemanz_plot02
```



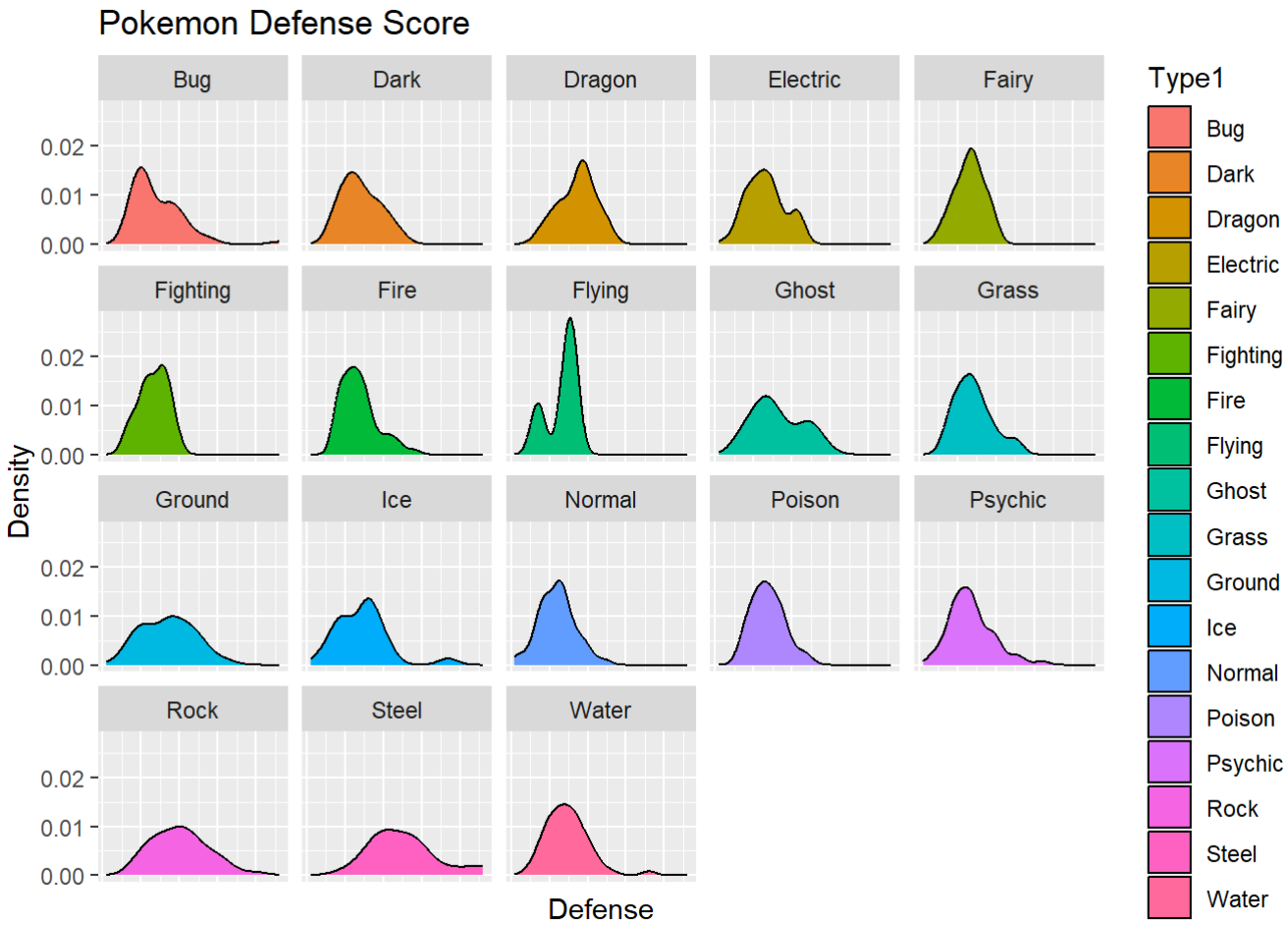

```
pokemanz_plot03 <- ggplot(pokemanz, aes(x=Attack, fill=Type1)) + geom_density(alpha = 1)
pokemanz_plot03 <- pokemanz_plot03 +
  facet_wrap(~Type1) +
  labs(x = "Attack", y = "Density", title ="Pokemon Attack Score") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

pokemanz_plot03
```



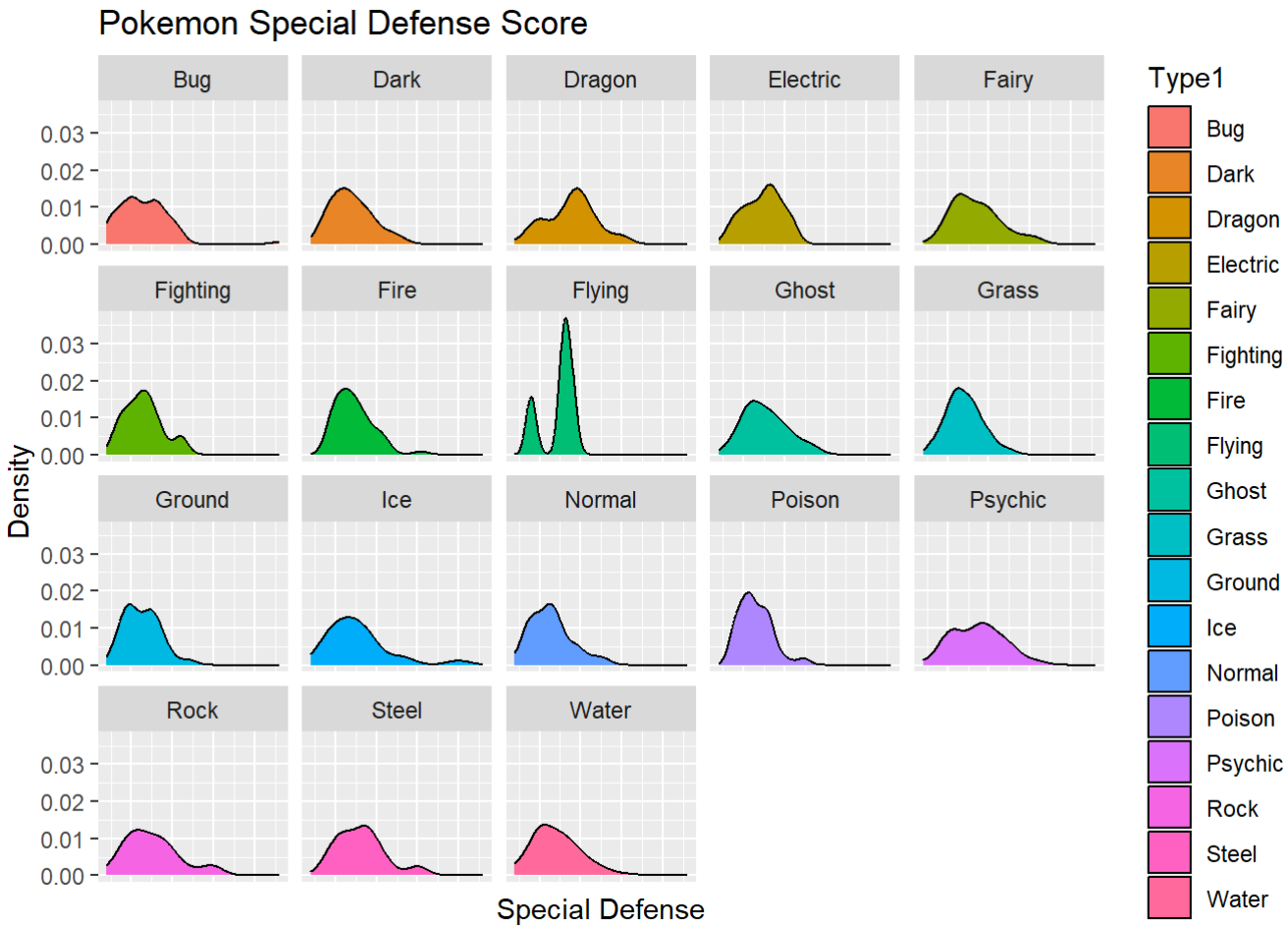

```
pokemanz_plot04 <- ggplot(pokemanz, aes(x=SpAtk, fill=Type1)) + geom_density(alpha = 1)
pokemanz_plot04 <- pokemanz_plot04 +
  facet_wrap(~Type1) +
  labs(x = "Special Attack", y = "Density", title ="Pokemon Special Attack Score") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

pokemanz_plot04
```



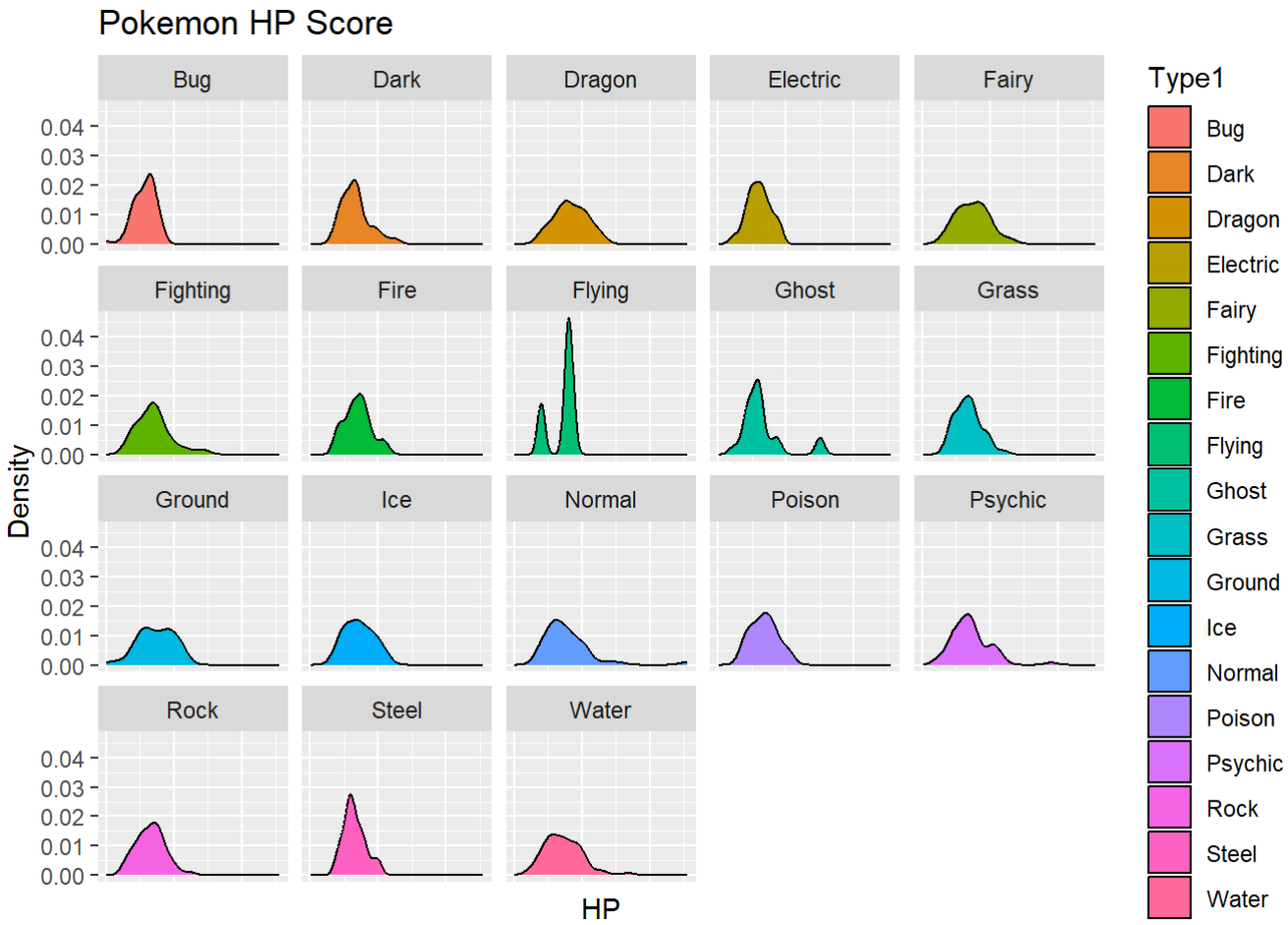

```
pokemanz_plot05 <- ggplot(pokemanz, aes(x=Defense, fill=Type1)) + geom_density(alpha = 1)
pokemanz_plot05<- pokemanz_plot05 +
  facet_wrap(~Type1) +
  labs(x = "Defense", y = "Density", title ="Pokemon Defense Score") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

pokemanz_plot05
```




```
pokemanz_plot06 <- ggplot(pokemanz, aes(x=SpDef, fill=Type1)) + geom_density(alpha = 1)
pokemanz_plot06<- pokemanz_plot06 +
  facet_wrap(~Type1) +
  labs(x = "Special Defense", y = "Density", title ="Pokemon Special Defense Score") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

pokemanz_plot06
```




```
pokemanz_plot07 <- ggplot(pokemanz, aes(x=HP, fill=Type1)) + geom_density(alpha = 1)
pokemanz_plot07<- pokemanz_plot07 +
  facet_wrap(~Type1) +
  labs(x = "HP", y = "Density", title ="Pokemon HP Score") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

pokemanz_plot07
```



Different Types of Visualizations used for Analysis

1. Bar Plot
2. Scatter Plot
3. Correlation Heatmap
4. Density Plot
5. Pie Chart
6. Stacked Bar Plot
7. Histogram
8. Facet Wrap
9. Violin Plot
10. Box Plot
11. Whisker Plot
12. Cumulative Distribution Function (CDF) Plot
13. Kernel Density Plot**

Conclusion

1. Total Score Distribution: The density plots for the Total scores of Pokemon, grouped by Type1, provide insights into the distribution of Pokemon based on their overall strength. Different types exhibit varying total score distributions.

2. Speed Distribution: The density plots for the Speed scores of Pokemon, grouped by Type1, show the speed distribution within each Pokemon type.

3. Attack and Special Attack Distributions: The density plots for Attack and Special Attack scores reveal the distribution of offensive capabilities among different Pokemon types.

4. Defense and Special Defense Distributions: The density plots for Defense and Special Defense scores showcase the defensive capabilities of Pokemon across different types.

5. HP Distribution: The density plots for HP scores illustrate the distribution of hit points, representing the health of Pokemon, for various types.

6. Legendary Pokemon Distribution: The bar plot shows the distribution of Legendary Pokemon, indicating that most Pokemon in the dataset are not legendary.

7. Pokemon Type Distribution: The pie chart and stacked bar plot provide insights into the distribution of Pokemon types (Type1 and Type2), indicating the prevalence of different types in the dataset.

8. Correlation Heatmap: The correlation heatmap visualizes the relationships between numeric attributes, helping identify patterns and potential correlations among different stats.

9. Cumulative Distribution Function (CDF) Plot: The CDF plot for Pokemon Defense illustrates the cumulative probability distribution of defense scores, offering insights into the overall defensive capabilities of Pokemon.

10. Kernel Density Plot for HP: The kernel density plot for HP shows the probability density function of hit points, giving an overview of the distribution of health scores among Pokemon.

Github Link :- https://github.com/abhivishwaroop11/Pokemon-Dataset-Analysis-Using-R (https://github.com/abhivishwaroop11/Pokemon-Dataset-Analysis-Using-R)