

Final

Rohan Jay Arun

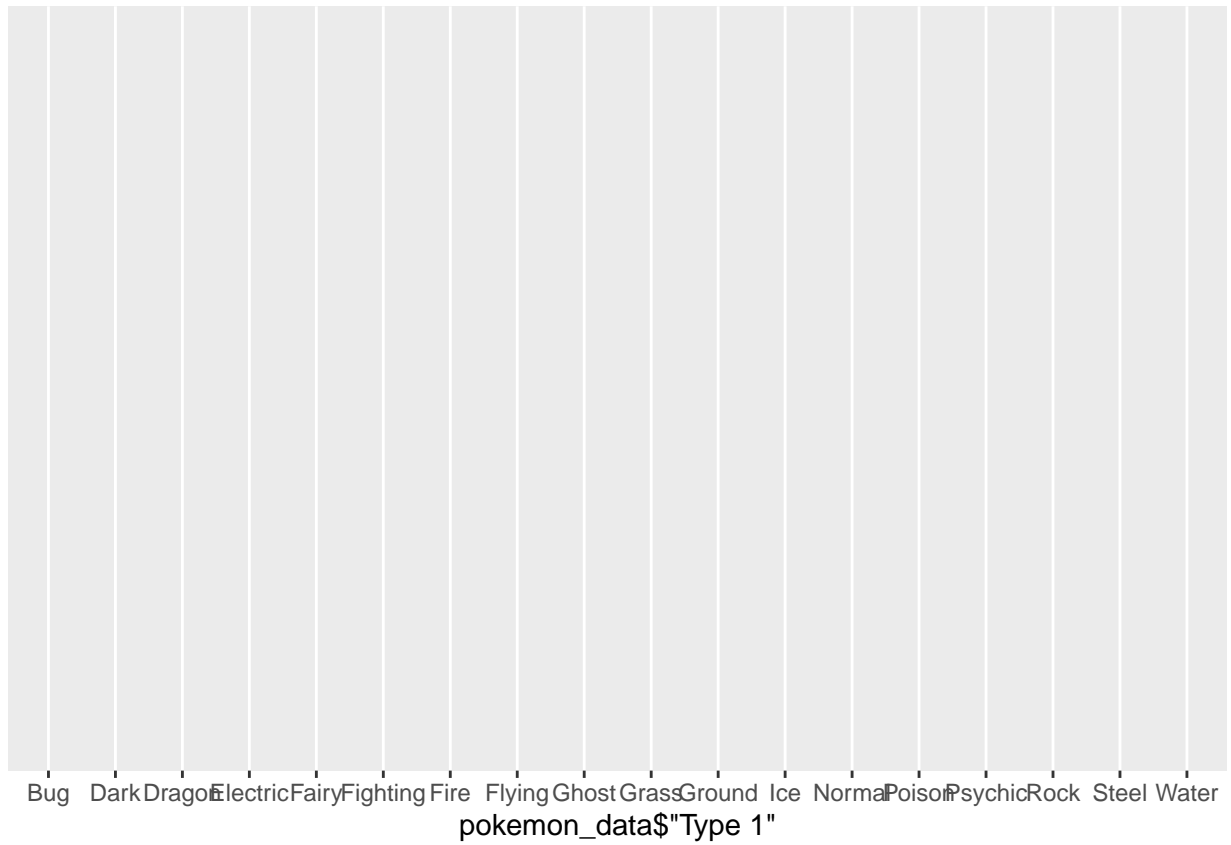
12/7/2021

```
library(ggplot2)
library(readr)
library(RColorBrewer)
pokemon_data <- read_csv('Pokemon.csv')

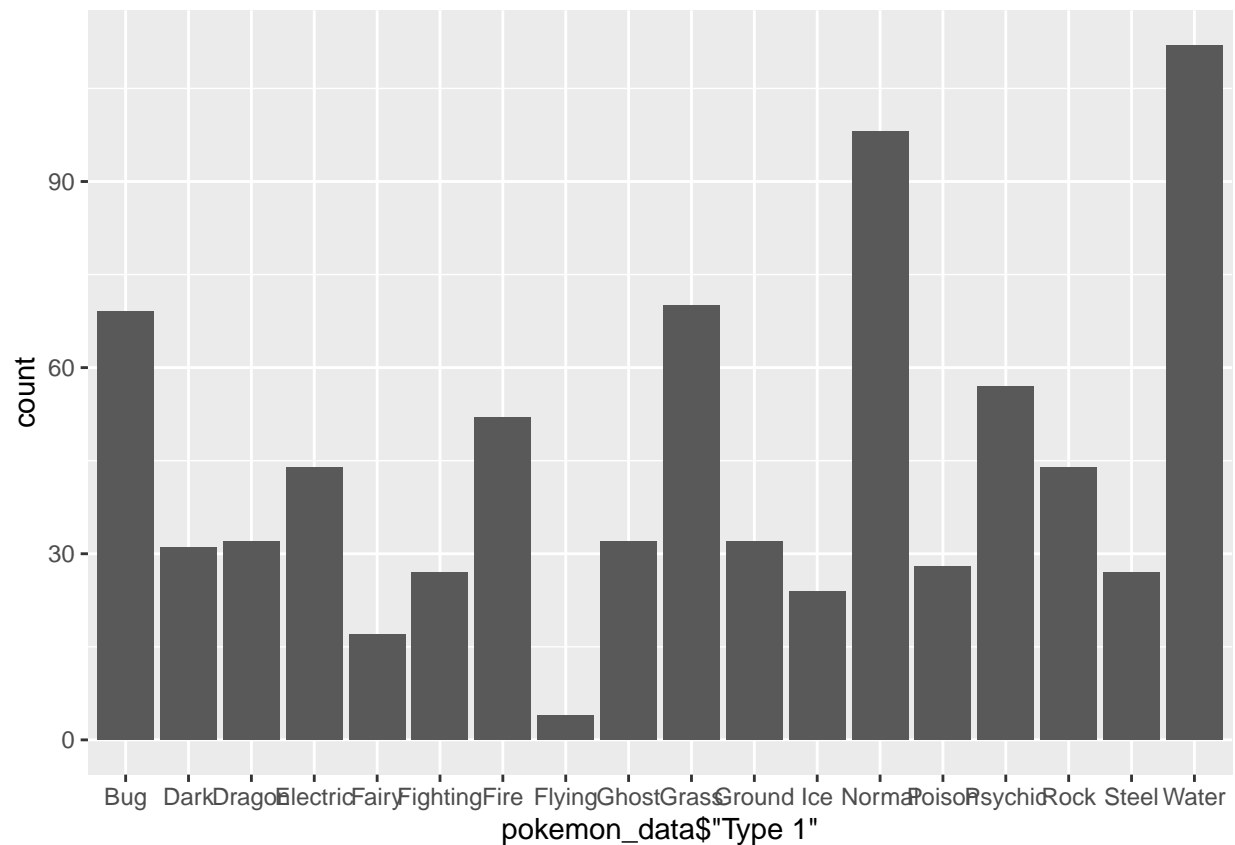
## Rows: 800 Columns: 13

## -- Column specification -----
## Delimiter: ","
## chr (3): Name, Type 1, Type 2
## dbl (9): #, Total, HP, Attack, Defense, Sp. Atk, Sp. Def, Speed, Generation
## lgl (1): Legendary

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
pokemonTypesBar <- ggplot(pokemon_data, aes(pokemon_data$"Type 1"))
#this repeat of the variable displays the graph we're saving to the variable
pokemonTypesBar
```

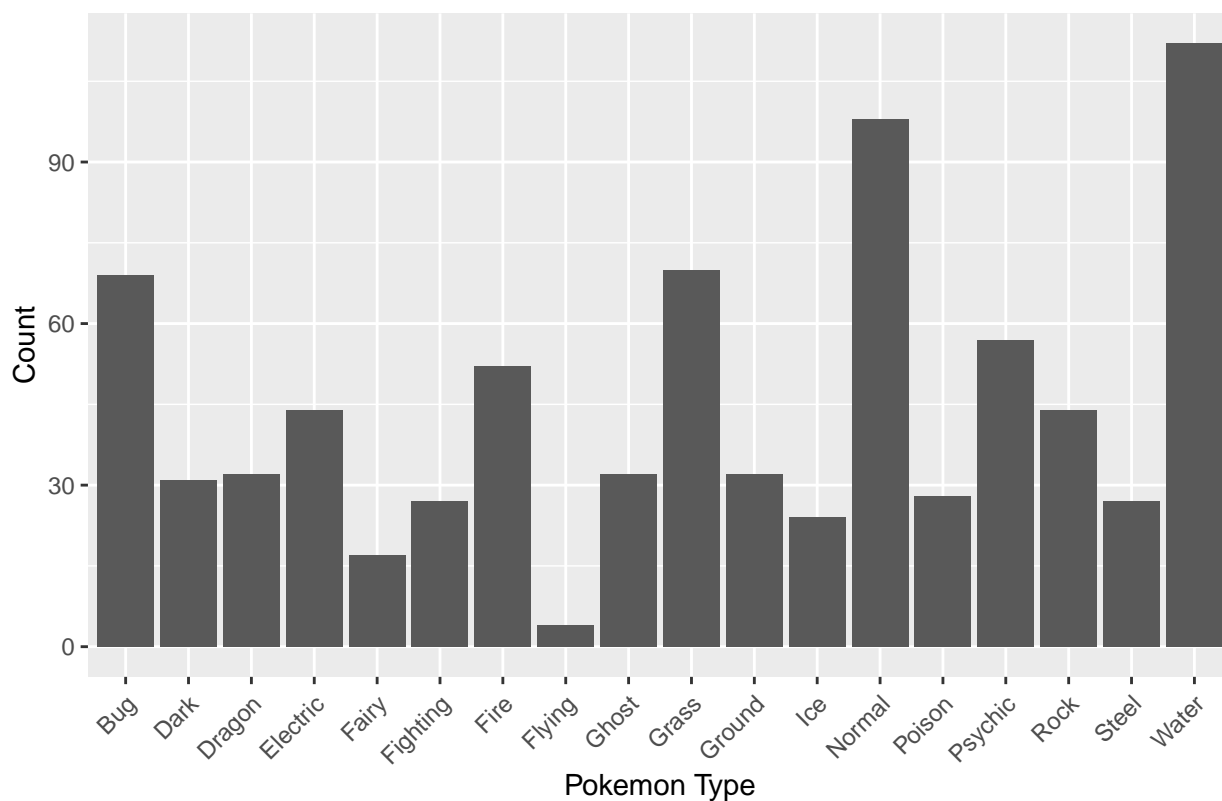


```
pokemonTypesBar <- pokemonTypesBar + geom_bar()
pokemonTypesBar
```



```
pokemonTypesBar <- ggplot(pokemon_data, aes(pokemon_data$"Type 1")) +
  geom_bar() +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=.95)) +
  #the angle is specified by "angle" and the position of labels
  #is specified by "hjust" (horizontal) and "vjust" (vertical)
  labs(title = "Pokemon Types Bar Chart",
        x = "Pokemon Type", y = "Count")
pokemonTypesBar
```

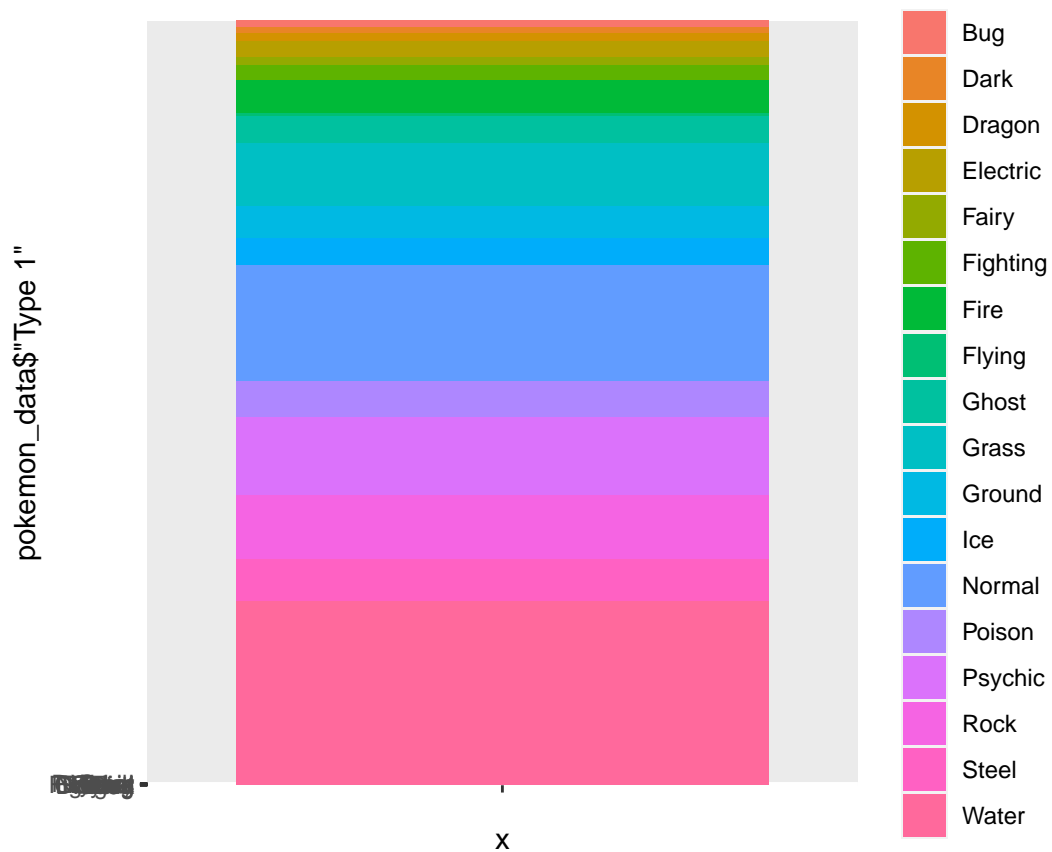
Pokemon Types Bar Chart



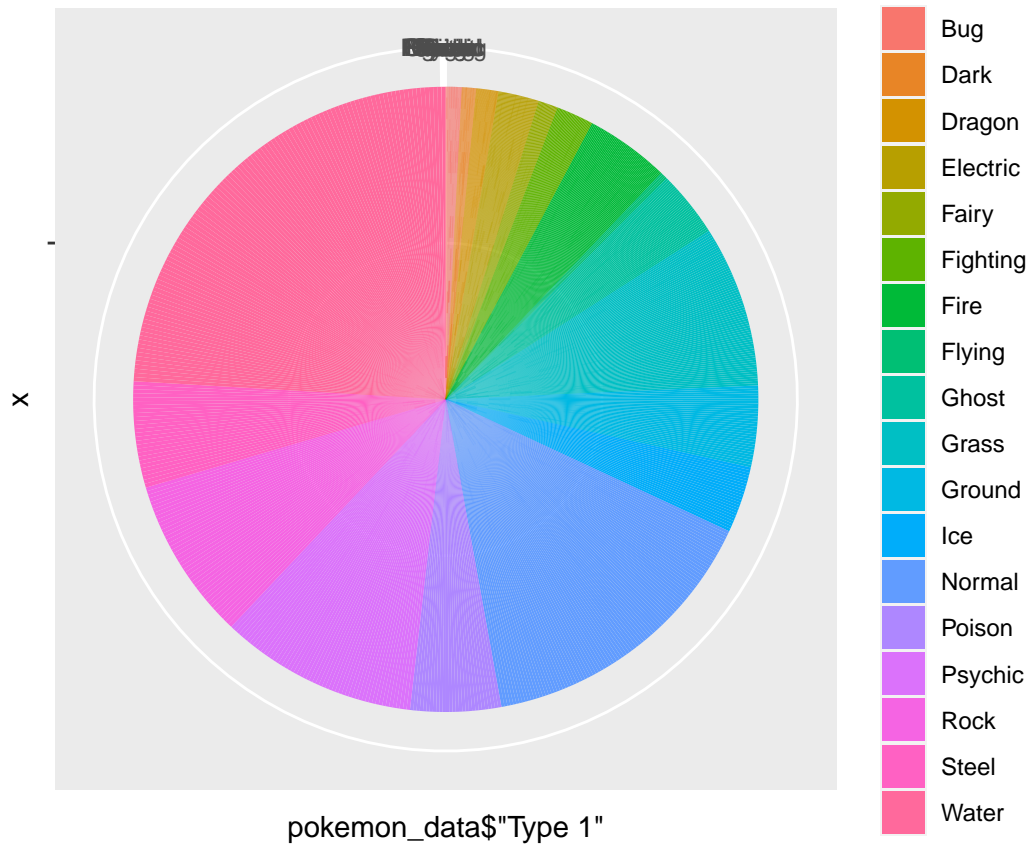
```

pokemonTypesVerticalBar <- ggplot(pokemon_data,
  aes(x = "",
      y = pokemon_data$"Type 1",
      fill = pokemon_data$"Type 1")) +
  geom_bar(stat="identity")
  #the 'stat = "identity"' part of this code tells
  #ggplot to use the y-values we put in
pokemonTypesVerticalBar

```

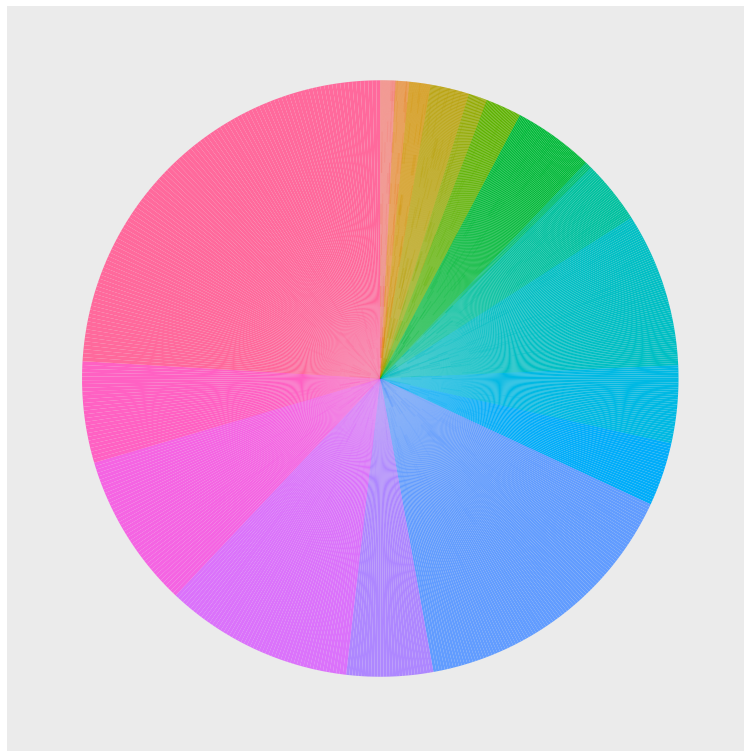


```
pokemonTypesPie <- pokemonTypesVerticalBar +
  coord_polar("y", start=0, direction = -1)
  #'start' determines where the graph begins, and 'direction'
  # dictates whether or not the plot should be clockwise or not:
  # -1 -> clockwise, 1 - counter-clockwise.
pokemonTypesPie
```



```
pokemonTypesPie <- pokemonTypesPie + labs(title = "Pokemon Types Pie Chart",
  x = "", y = "", fill = "Pokemon Types") +
  theme(axis.text.x=element_blank(),
    axis.ticks.y=element_blank(),
    panel.grid = element_blank())
pokemonTypesPie
```

Pokemon Types Pie Chart

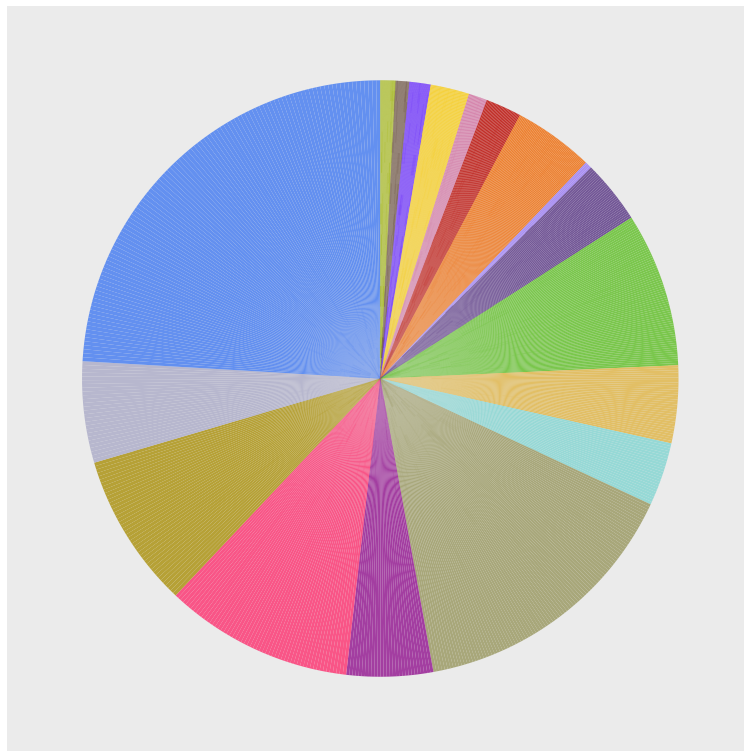


Pokemon Types



```
pokemonTypesPie <- pokemonTypesPie + scale_fill_manual(values=
  c("#A6B91A", "#705746", "#6F35FC", "#F7D02C",
    "#D685AD", "#C22E28", "#EE8130", "#A98FF3",
    "#735797", "#7AC74C", "#E2BF65", "#96D9D6",
    "#A8A77A", "#A33EA1", "#F95587", "#B6A136",
    "#B7B7CE", "#6390F0"))
pokemonTypesPie
```

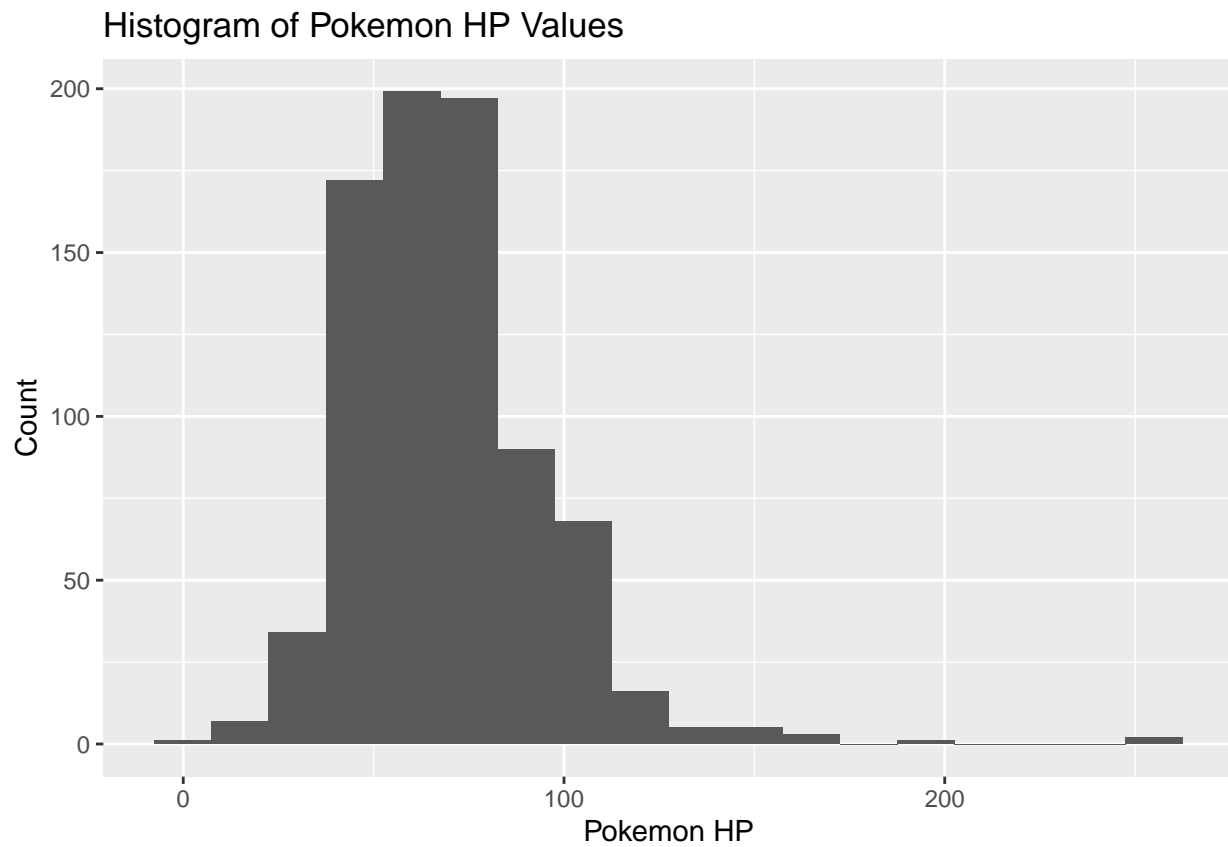
Pokemon Types Pie Chart



Pokemon Types

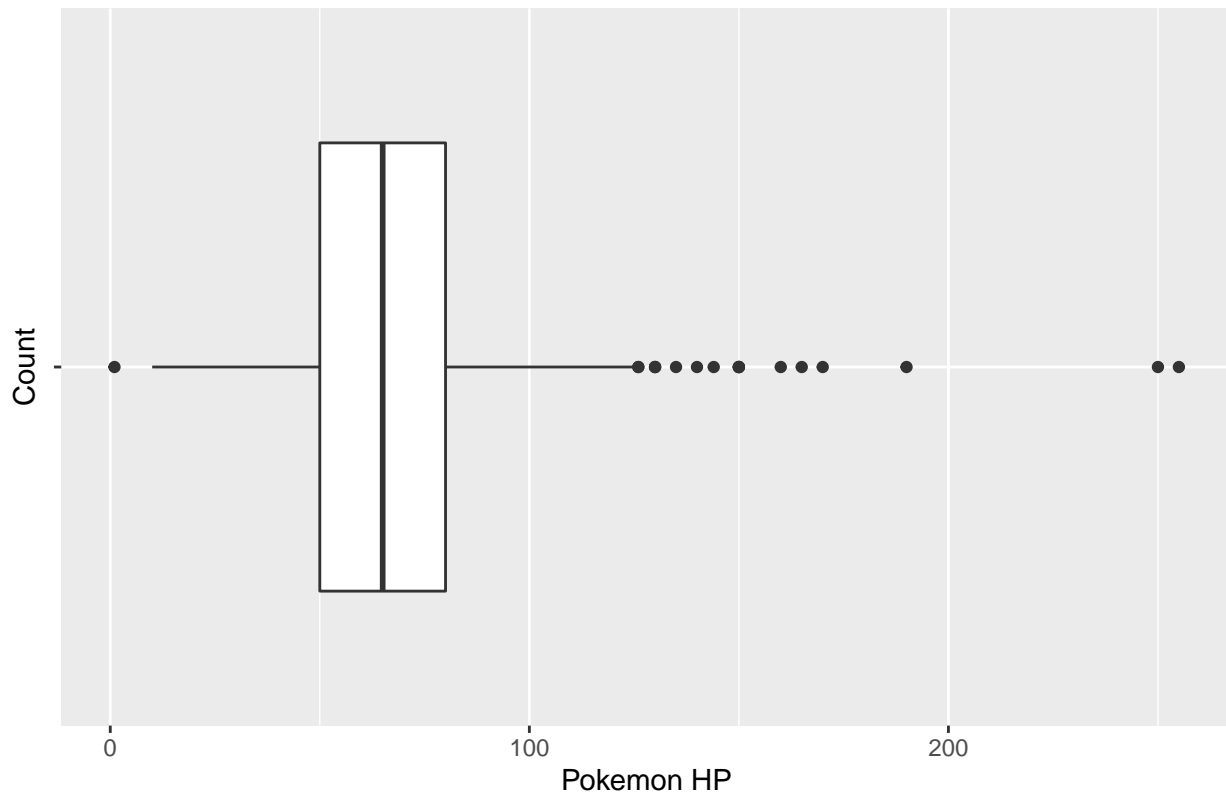


```
pokemonHPHist <- ggplot(pokemon_data, aes(pokemon_data$"HP")) +
  geom_histogram(binwidth = 15) +
  labs(title = "Histogram of Pokemon HP Values",
        x = "Pokemon HP", y = "Count")
pokemonHPHist
```

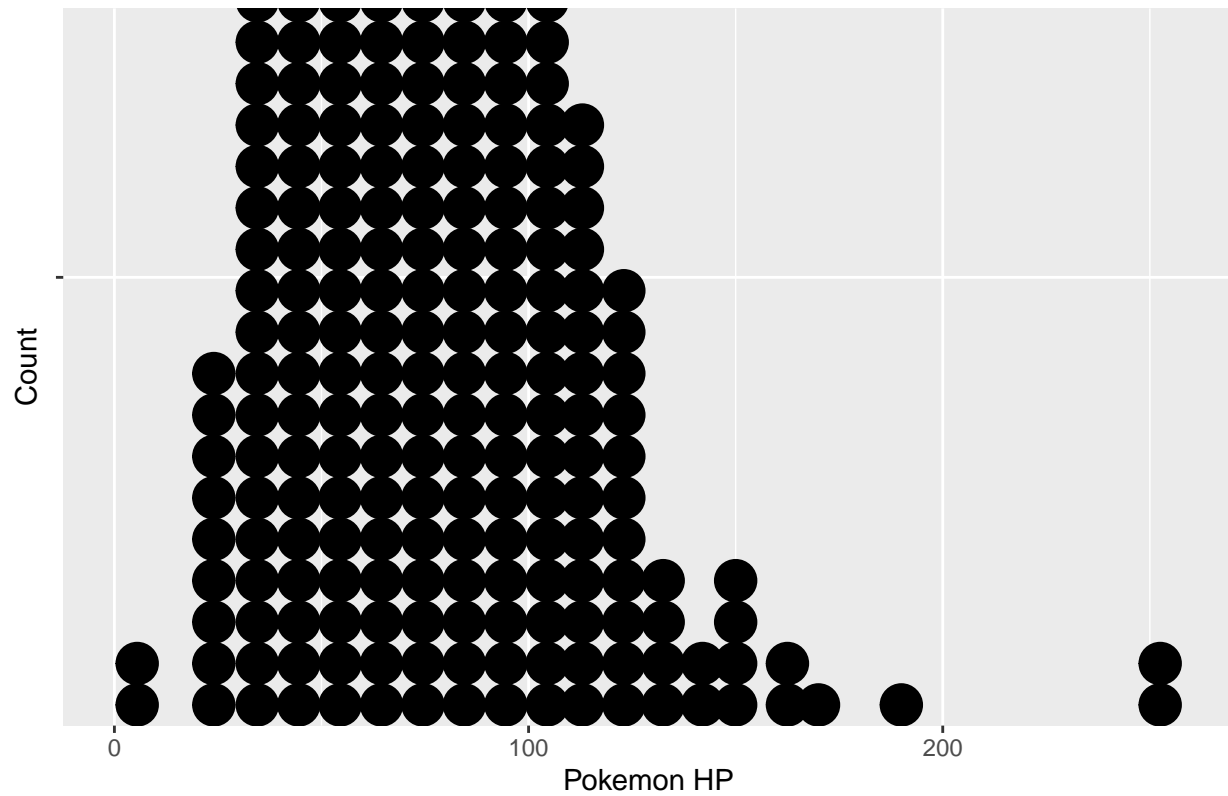
```
pokemonHPBoxPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"HP", y="")) +  
  geom_boxplot() +  
  labs(title = "Distribution of Pokemon HP Values",  
        x = "Pokemon HP", y = "Count")  
pokemonHPBoxPlot
```

Distribution of Pokemon HP Values



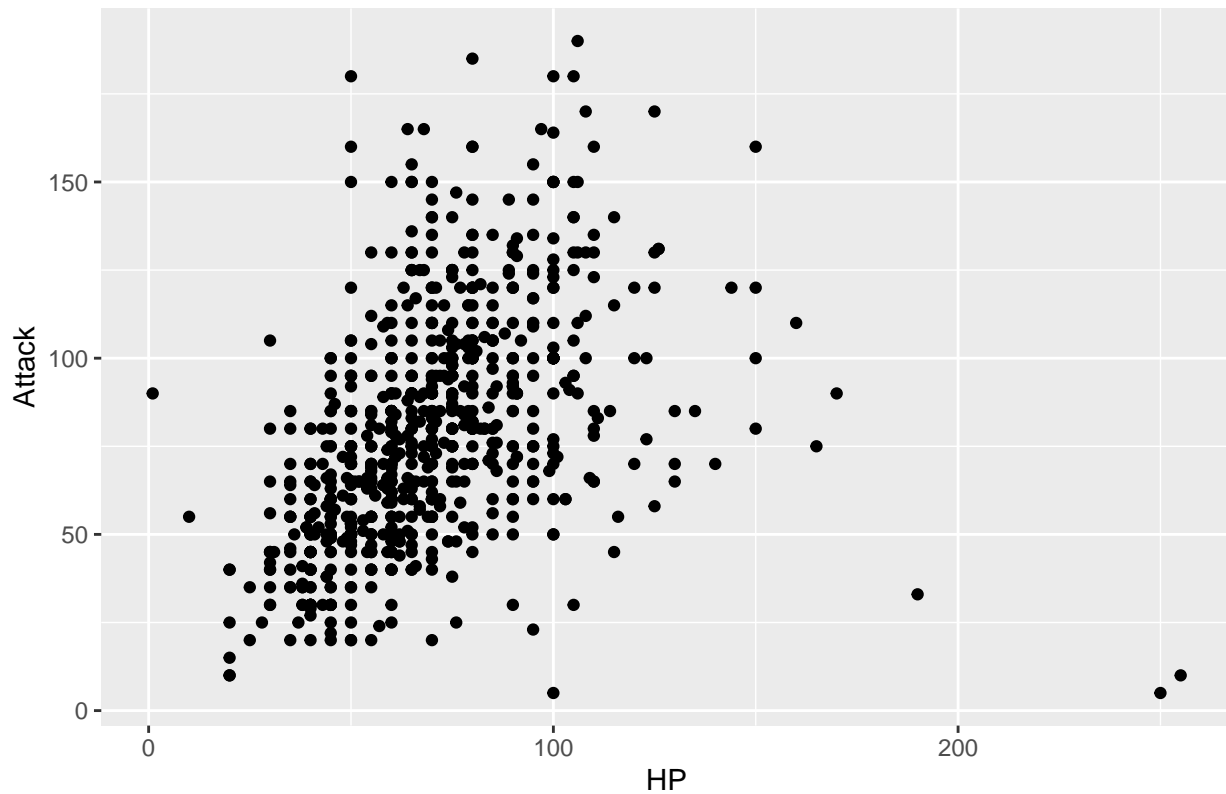
```
pokemonDotPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"HP", y="")) +  
  geom_dotplot(binwidth = 10) +  
  #In this case, binwidth denotes the number of points represented  
  #by one dot. The size of the dots is affected by the binwidth.  
  labs(title = "Distribution of Pokemon HP Values",  
        x="Pokemon HP", y = "Count")  
pokemonDotPlot
```

Distribution of Pokemon HP Values



```
pokemonScatterPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"HP",
                                                y=pokemon_data$"Attack")) +
  geom_point() +
  labs(title = "Pokemon HP Values vs. Pokemon Attack",
        x="HP", y = "Attack")
pokemonScatterPlot
```

Pokemon HP Values vs. Pokemon Attack

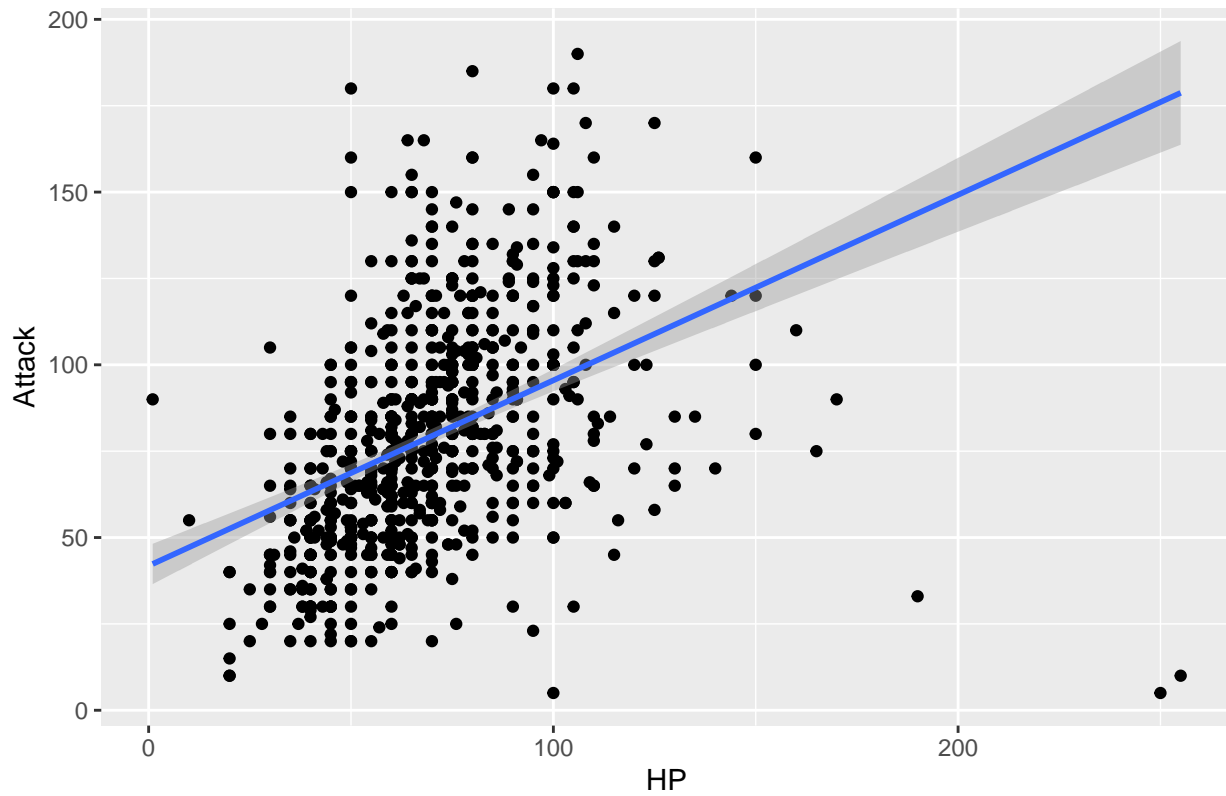


```
pokemonScatterPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"HP",  
                                                y=pokemon_data$"Attack")) +  
  geom_point() +  
  geom_smooth(method='lm') +  
  labs(title = "Pokemon HP Values vs. Pokemon Attack",  
        x="HP", y = "Attack")
```

```
pokemonScatterPlot
```

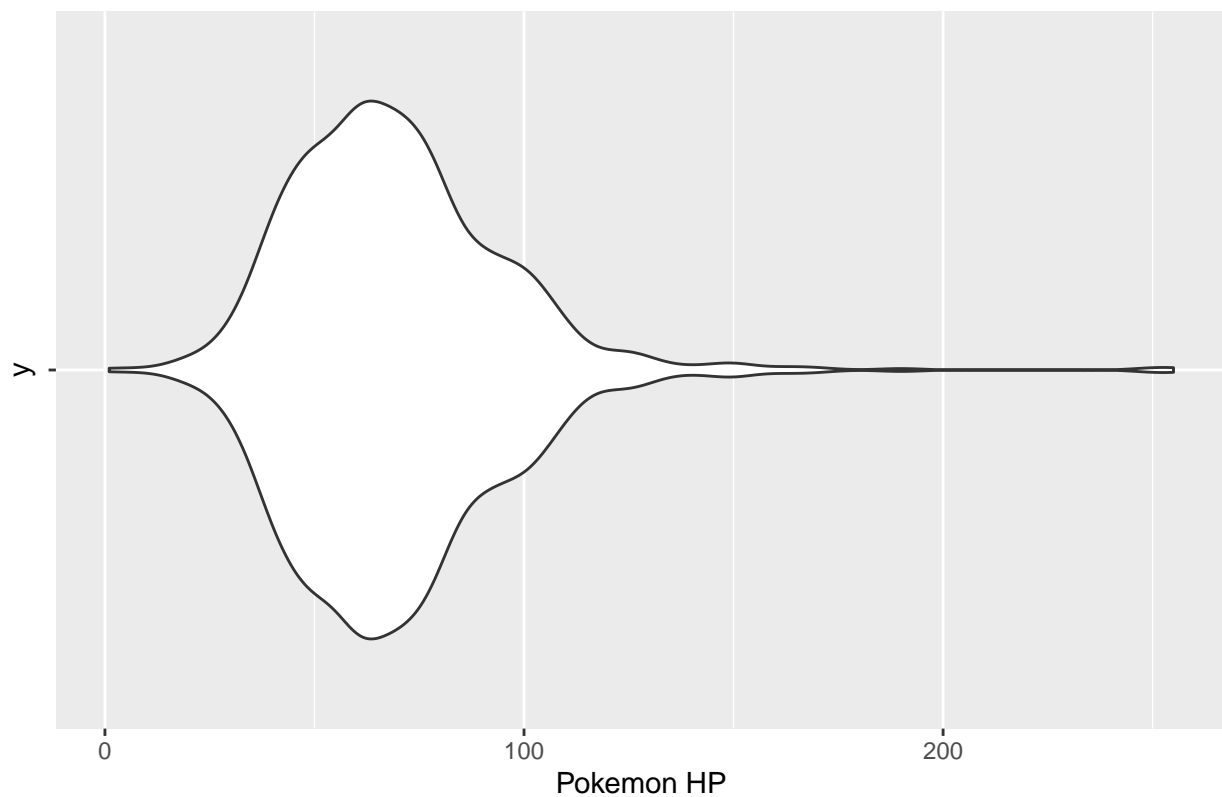
```
## `geom_smooth()` using formula 'y ~ x'
```

Pokemon HP Values vs. Pokemon Attack



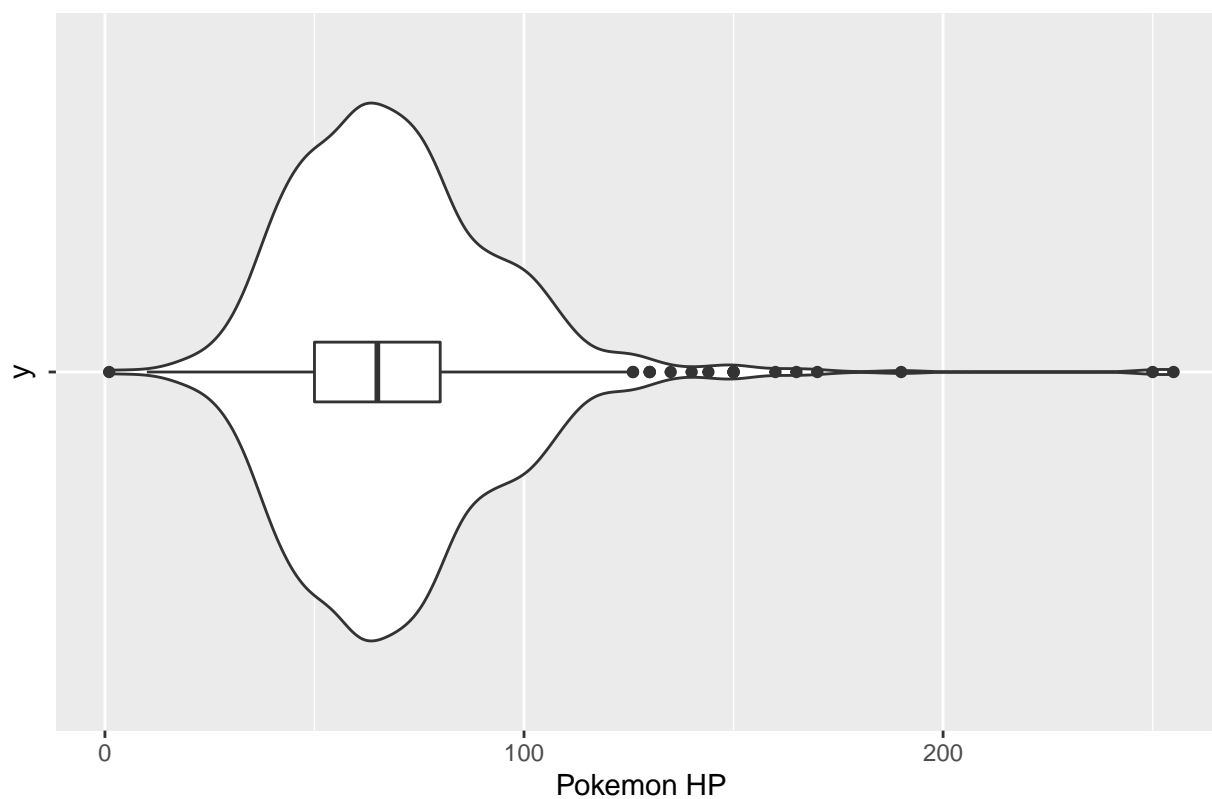
```
pokemonHPViolinPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"HP", y="")) +  
  geom_violin() +  
  labs(title = "Distribution of Pokemon HP Values",  
        x = "Pokemon HP")  
pokemonHPViolinPlot
```

Distribution of Pokemon HP Values



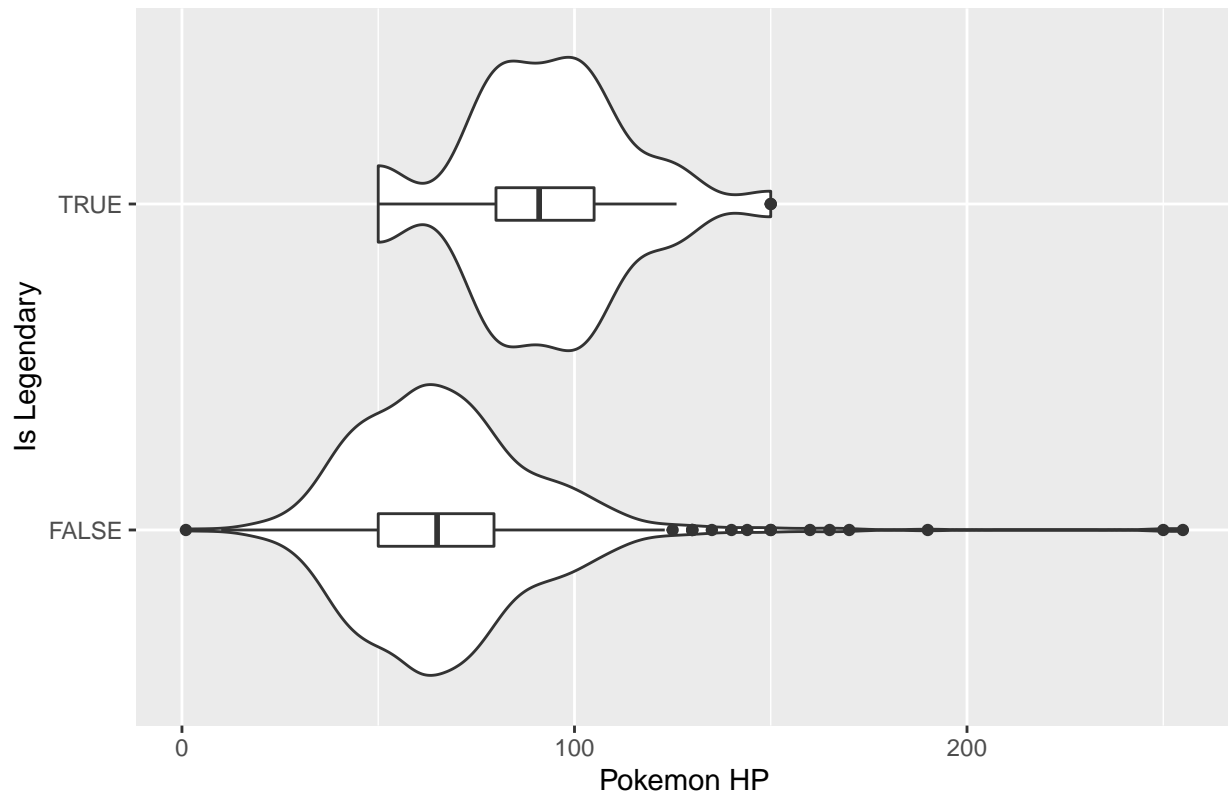
```
pokemonHPViolinPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"HP", y="")) +  
  geom_violin() +  
  geom_boxplot(width=0.1) +  
  labs(title = "Distribution of Pokemon HP Values",  
        x = "Pokemon HP")  
pokemonHPViolinPlot
```

Distribution of Pokemon HP Values



```
pokemonHPViolinPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"HP",  
                                                  pokemon_data$"Legendary")) +  
  geom_violin() +  
  geom_boxplot(width=0.1) +  
  labs(title = "Distribution of Pokemon HP Values",  
        x = "Pokemon HP", y="Is Legendary")  
pokemonHPViolinPlot
```

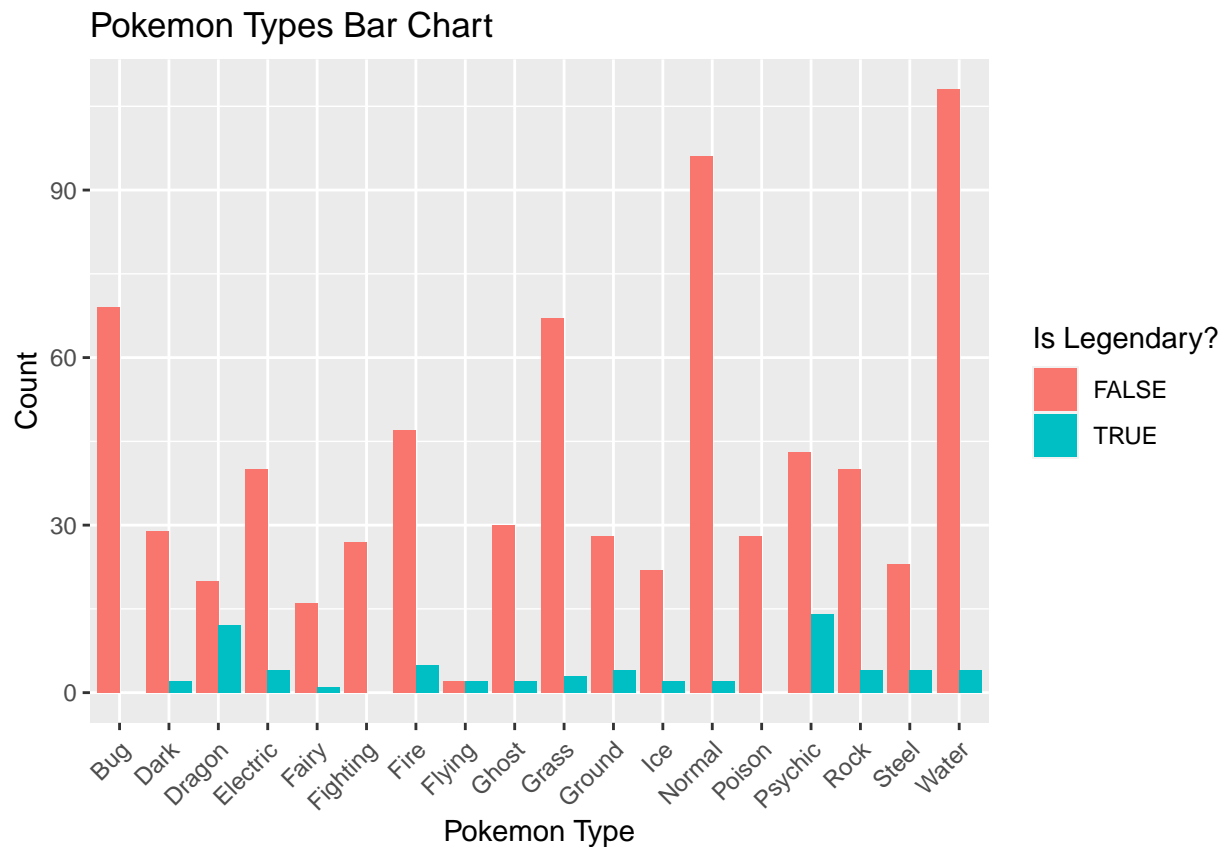
Distribution of Pokemon HP Values



```
pokemonTypesBar <- ggplot(pokemon_data, aes(fill=pokemon_data$"Legendary",
                                             x=pokemon_data$"Type 1")) +
  geom_bar(position = position_dodge(preserve = 'single')) +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=.95)) +

  #the angle is specified by "angle," and the position of the
  #labels is specified by "hjust" (horizontal) & "vjust" (vertical)

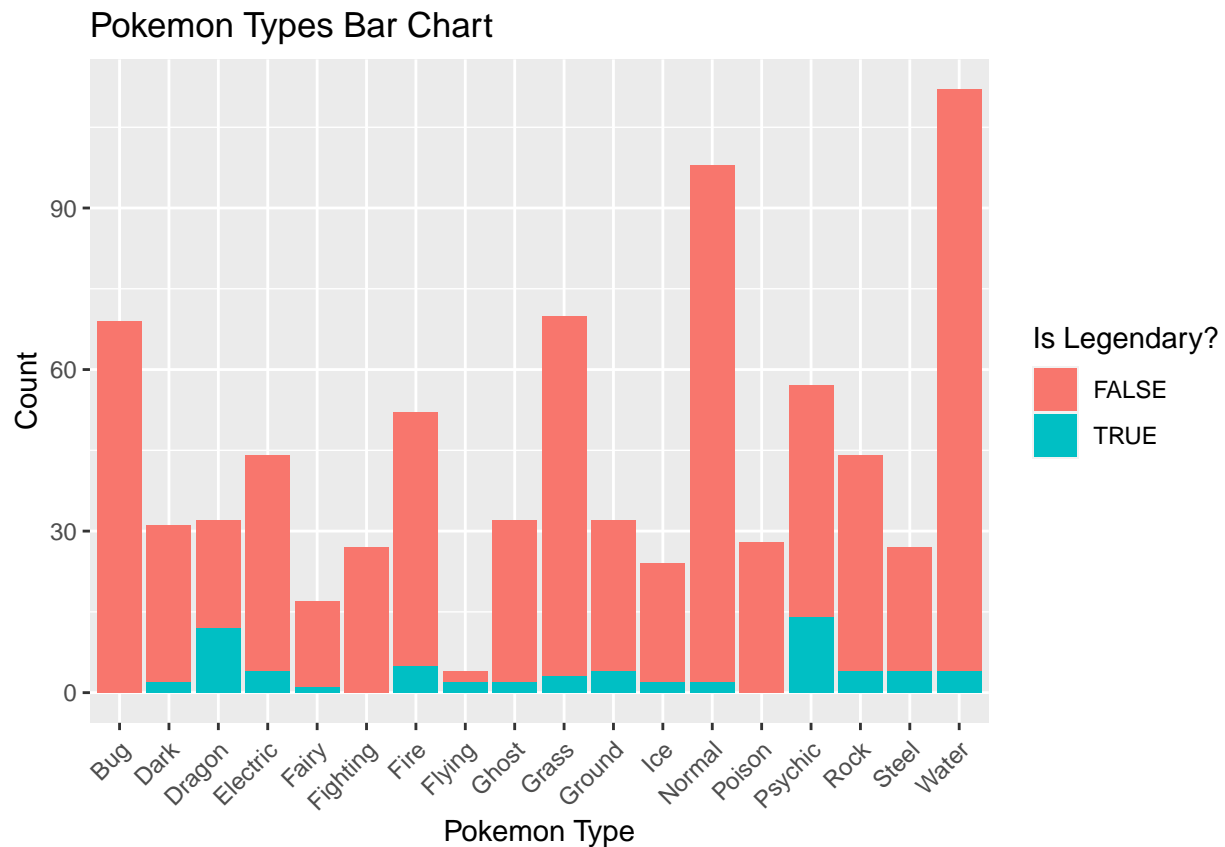
  labs(title = "Pokemon Types Bar Chart", x = "Pokemon Type",
        y = "Count", fill = "Is Legendary?")
pokemonTypesBar
```

```
pokemonTypesBar <- ggplot(pokemon_data, aes(fill=pokemon_data$"Legendary",
                                             x=pokemon_data$"Type 1")) +
  geom_bar(position = 'stack') +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=.95)) +

  #the angle is specified by "angle," and the position of the
  #labels is specified by "hjust" (horizontal) & "vjust" (vertical)

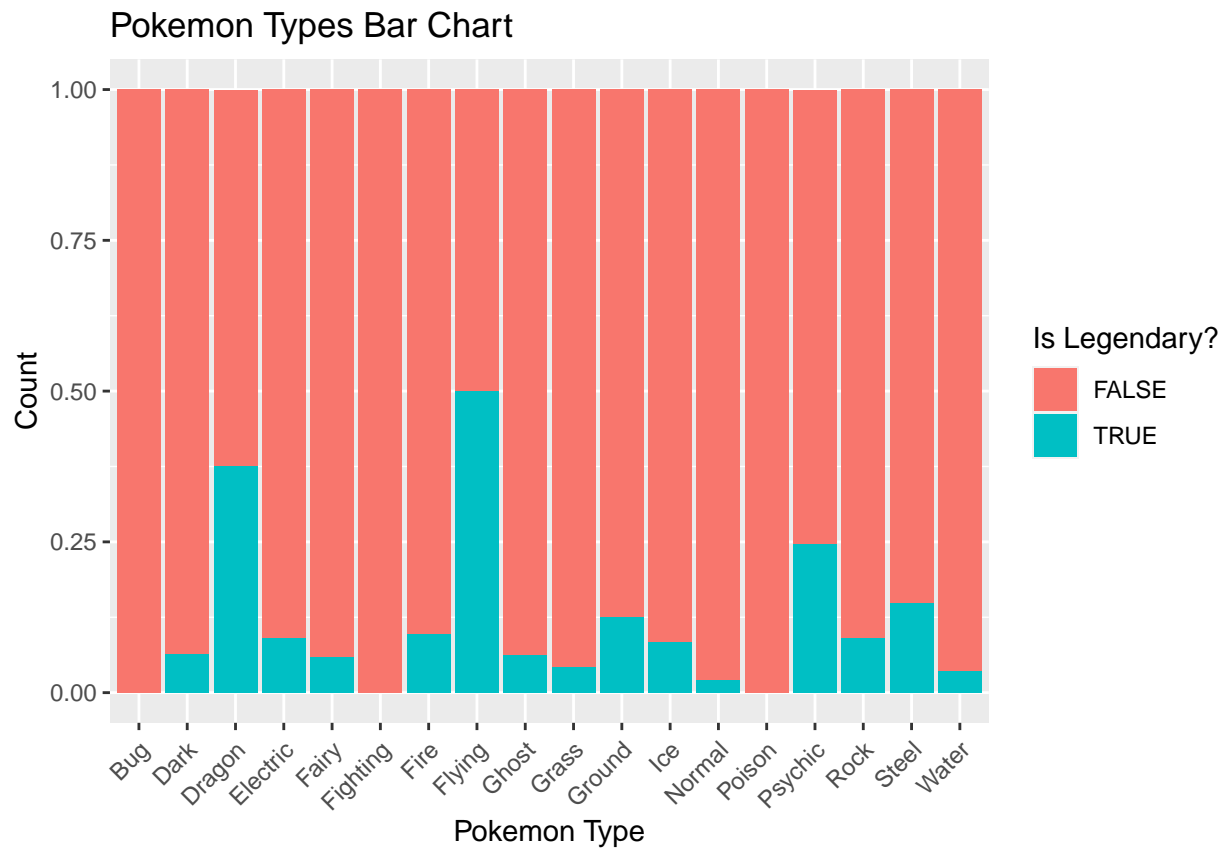
  labs(title = "Pokemon Types Bar Chart", x = "Pokemon Type",
        y = "Count", fill = "Is Legendary?")
pokemonTypesBar
```



```
pokemonTypesBar <- ggplot(pokemon_data, aes(fill=pokemon_data$"Legendary",
                                             x=pokemon_data$"Type 1")) +
  geom_bar(position = 'fill') +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=.95)) +

  #the angle is specified by "angle," and the position of the
  #labels is specified by "hjust" (horizontal) & "vjust" (vertical)

  labs(title = "Pokemon Types Bar Chart", x = "Pokemon Type",
        y = "Count", fill = "Is Legendary?")
pokemonTypesBar
```

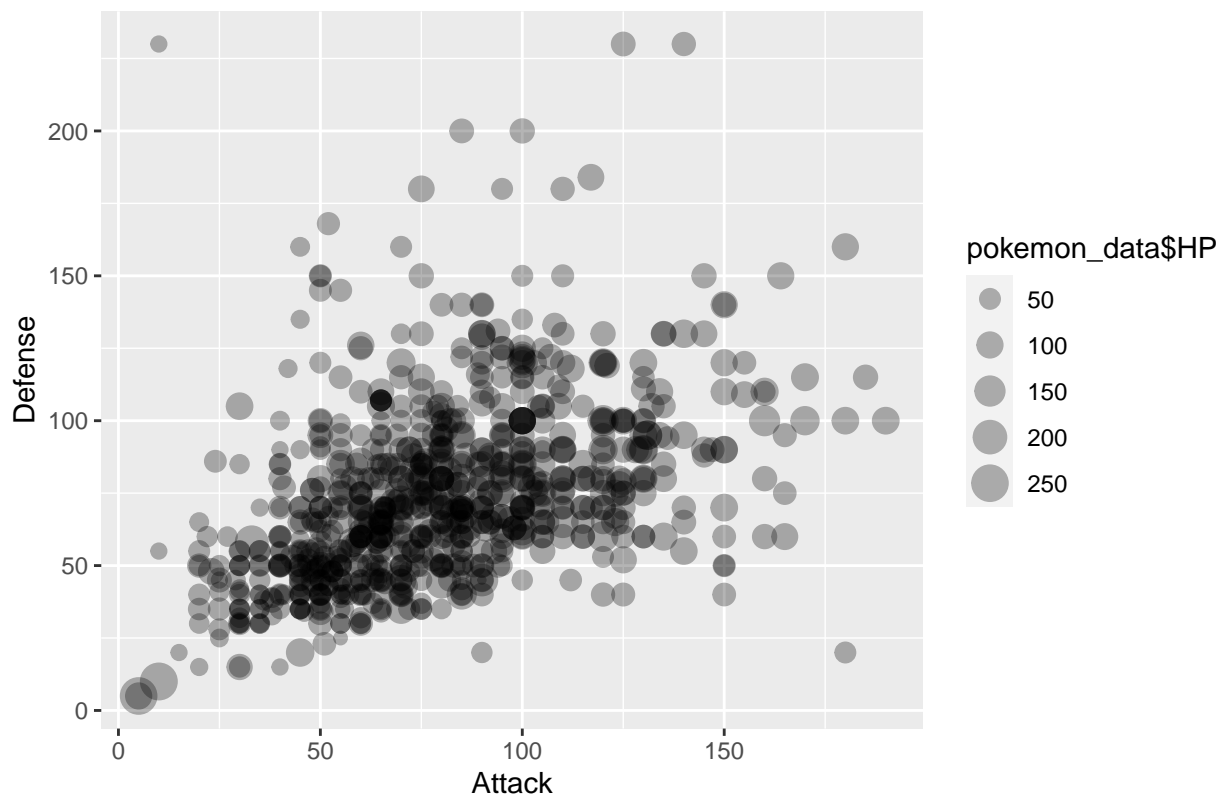


```

pokemonScatterPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"Attack",
                                                y=pokemon_data$"Defense",
                                                size=pokemon_data$"HP")) +
  geom_point(alpha=0.3) +
  labs(title = "Pokemon HP Values vs. Pokemon Attack", x="Attack",
        y = "Defense")
pokemonScatterPlot

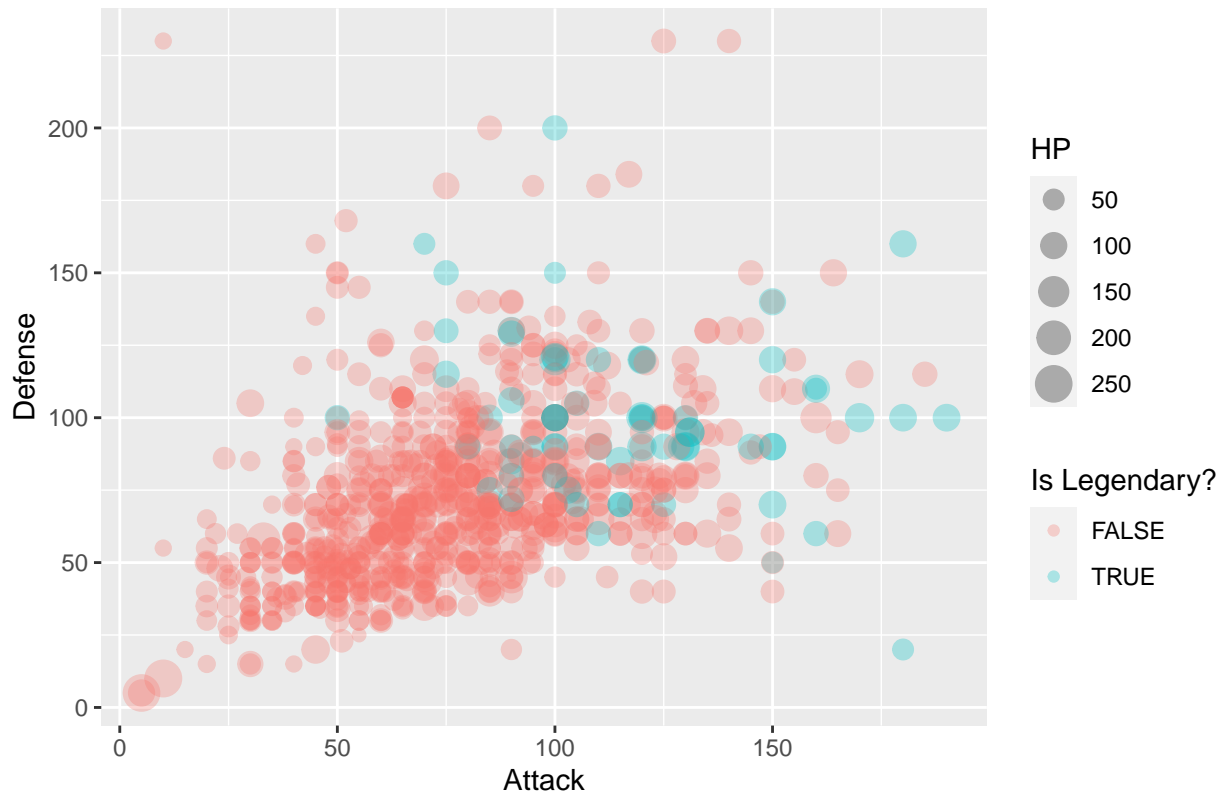
```

Pokemon HP Values vs. Pokemon Attack



```
pokemonScatterPlot <- ggplot(pokemon_data, aes(x=pokemon_data$"Attack",  
                                                y=pokemon_data$"Defense",  
                                                size=pokemon_data$"HP",  
                                                color=pokemon_data$"Legendary")) +  
  geom_point(alpha=0.3) +  
  labs(title = "Pokemon HP Values vs. Pokemon Attack",  
       x="Attack", y = "Defense", color = "Is Legendary?",  
       size = 'HP')  
pokemonScatterPlot
```

Pokemon HP Values vs. Pokemon Attack



```
pokemon_data$"Generation" <- as.character(pokemon_data$"Generation")

pokemonGenVerticalBar <- ggplot(pokemon_data, aes(x = "",
                                                    y = pokemon_data$"Generation",
                                                    fill = pokemon_data$"Generation")) +
  geom_bar(stat="identity")

pokemonGenPie <- pokemonGenVerticalBar + coord_polar("y", start=0, direction = -1) +
  labs(title = "Pokemon Generation Pie Chart", x = "", y = "",
        fill = "Pokemon Generation") +
  theme(axis.text.x=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid = element_blank())

pokemonGenPie
```

Pokemon Generation Pie Chart

