Exploring the Pokémon Data set from kaggle.com

Importing data into a dataframe:

pokemon <- read.csv("Pokemon.csv")

Reading data:

head(pokemon)

```
> head(pokemon)
  X.                  Name Type.1 Type.2 Total HP Attack Defense Sp..Atk Sp..Def Speed Generation Legendary
1  1             Bulbasaur  Grass Poison   318 45     49      49      65      65    45          1     False
2  2               Ivysaur  Grass Poison   405 60     62      63      80      80    60          1     False
3  3              Venusaur  Grass Poison   525 80     82      83     100     100    80          1     False
4  3 VenusaurMega Venusaur  Grass Poison   625 80    100     123     122     120    80          1     False
5  4            Charmander   Fire          309 39     52      43      60      50    65          1     False
6  5            Charmeleon   Fire          405 58     64      58      80      65    80          1     False
```

tail(pokemon)

```
> tail(pokemon)
     X.                 Name  Type.1 Type.2 Total  HP Attack Defense Sp..Atk Sp..Def Speed Generation Legendary
795 718    Zygarde50% Forme  Dragon Ground   600 108    100     121      81      95    95          6      True
796 719             Diancie    Rock  Fairy   600  50    100     150     100     150    50          6      True
797 719 DiancieMega Diancie    Rock  Fairy   700  50    160     110     160     110   110          6      True
798 720 HoopaHoopa Confined Psychic  Ghost   600  80    110      60     150     130    70          6      True
799 720  HoopaHoopa Unbound Psychic   Dark   680  80    160      60     170     130    80          6      True
800 721           Volcanion   Fire  Water   600  80    110     120     130      90    70          6      True
```

str(pokemon)

```
> str(pokemon)
'data.frame':   800 obs. of  13 variables:
 $ X.         : int  1 2 3 3 4 5 6 6 6 7 ...
 $ Name       : Factor w/ 800 levels "Abomasnow","AbomasnowMega Abomasnow",..: 81 330 746 747 103 104 100 101 102 666 ...
 $ Type.1     : Factor w/ 18 levels "Bug","Dark","Dragon",..: 10 10 10 10 7 7 7 7 7 18 ...
 $ Type.2     : Factor w/ 19 levels "","Bug","Dark",..: 15 15 15 15 1 1 9 4 9 1 ...
 $ 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 ...
 $ Sp..Atk    : int  65 80 100 122 60 80 109 130 159 50 ...
 $ Sp..Def    : 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 : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
```

Converting 'int' variable Generation to factor / factorizing Generation variable:

pokemon$Generation <- factor(pokemon$Generation)

str(pokemon)

```
> str(pokemon)
'data.frame':   800 obs. of  13 variables:
 $ X.         : int  1 2 3 3 4 5 6 6 6 7 ...
 $ Name       : Factor w/ 800 levels "Abomasnow","AbomasnowMega Abomasnow",..: 81 330 746 747 103 104 100 101 102 666 ...
 $ Type.1     : Factor w/ 18 levels "Bug","Dark","Dragon",..: 10 10 10 10 7 7 7 7 7 18 ...
 $ Type.2     : Factor w/ 19 levels "","Bug","Dark",..: 15 15 15 15 1 1 9 4 9 1 ...
 $ 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 ...
 $ Sp..Atk    : int  65 80 100 122 60 80 109 130 159 50 ...
 $ Sp..Def    : int  65 80 100 120 50 65 85 85 115 64 ...
 $ Speed      : int  45 60 80 80 65 80 100 100 100 43 ...
 $ Generation: Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Legendary : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
>
```

Changing column names:

colnames(pokemon)

```
> colnames(pokemon)
 [1] "X."          "Name"        "Type.1"      "Type.2"      "Total"       "HP"          "Attack"      "Defense"     "Sp..Atk"
[10] "Sp..Def"     "Speed"       "Generation" "Legendary"
```

colnames(pokemon) <- c ("No","Name","Type1","Type2","Total","HP","Attack","Defnse", "SpAttack","SpDefense","Speed","Gen","IsLegendary")

colnames(pokemon)

```
> colnames(pokemon)
 [1] "No"          "Name"        "Type1"       "Type2"       "Total"       "HP"          "Attack"      "Defnse"
 [9] "SpAttack"    "SpDefense"   "Speed"       "Gen"         "IsLegendary"
```

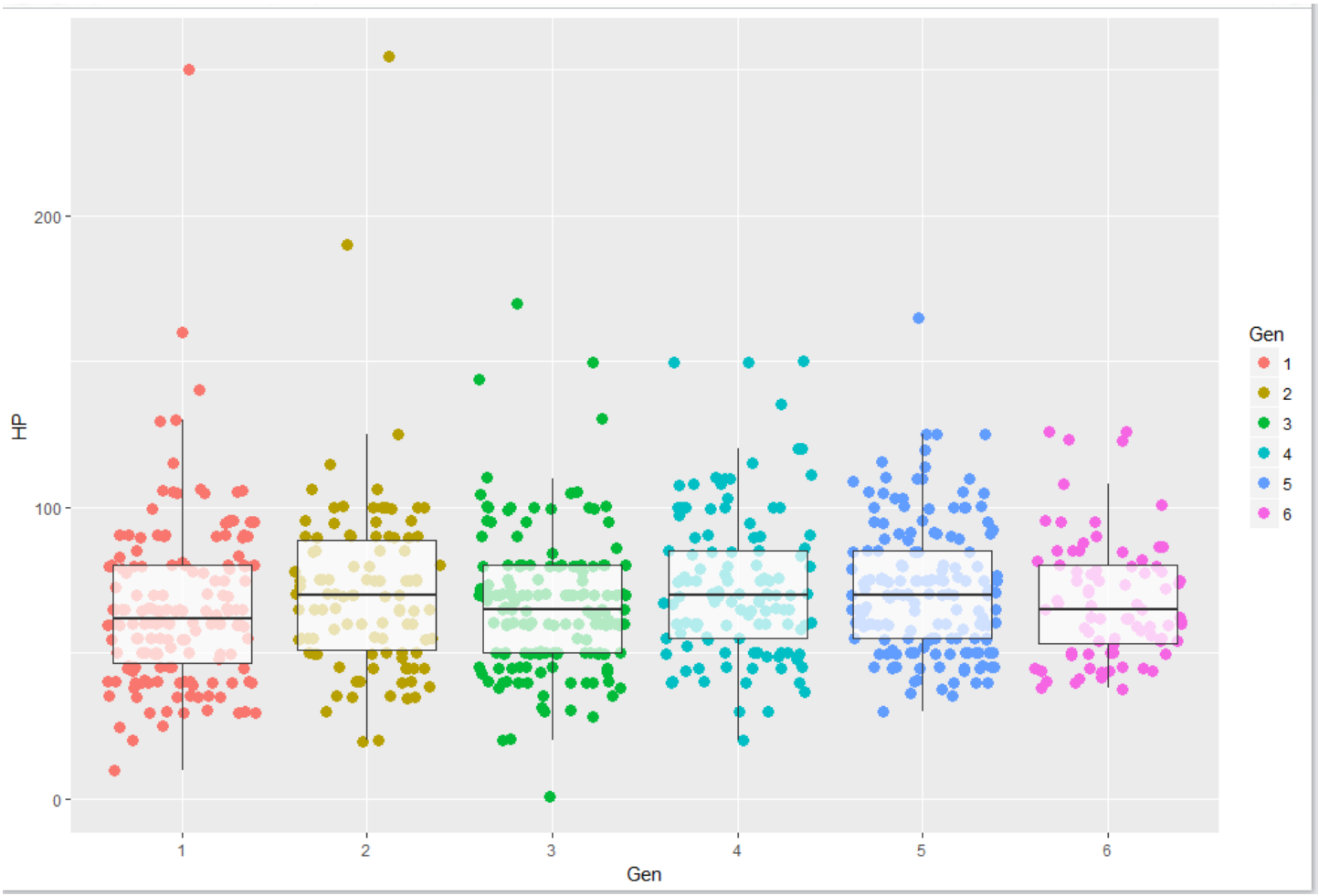
Deriving Insights using ggplot2:

library(ggplot2)

1. Check HP of Pokemon across generations:

a <- ggplot(data = pokemon, aes(x=Gen,y=HP))

a + geom_jitter(aes(color=Gen),size=3) + geom_boxplot(alpha=0.7,outlier.colour = NA)

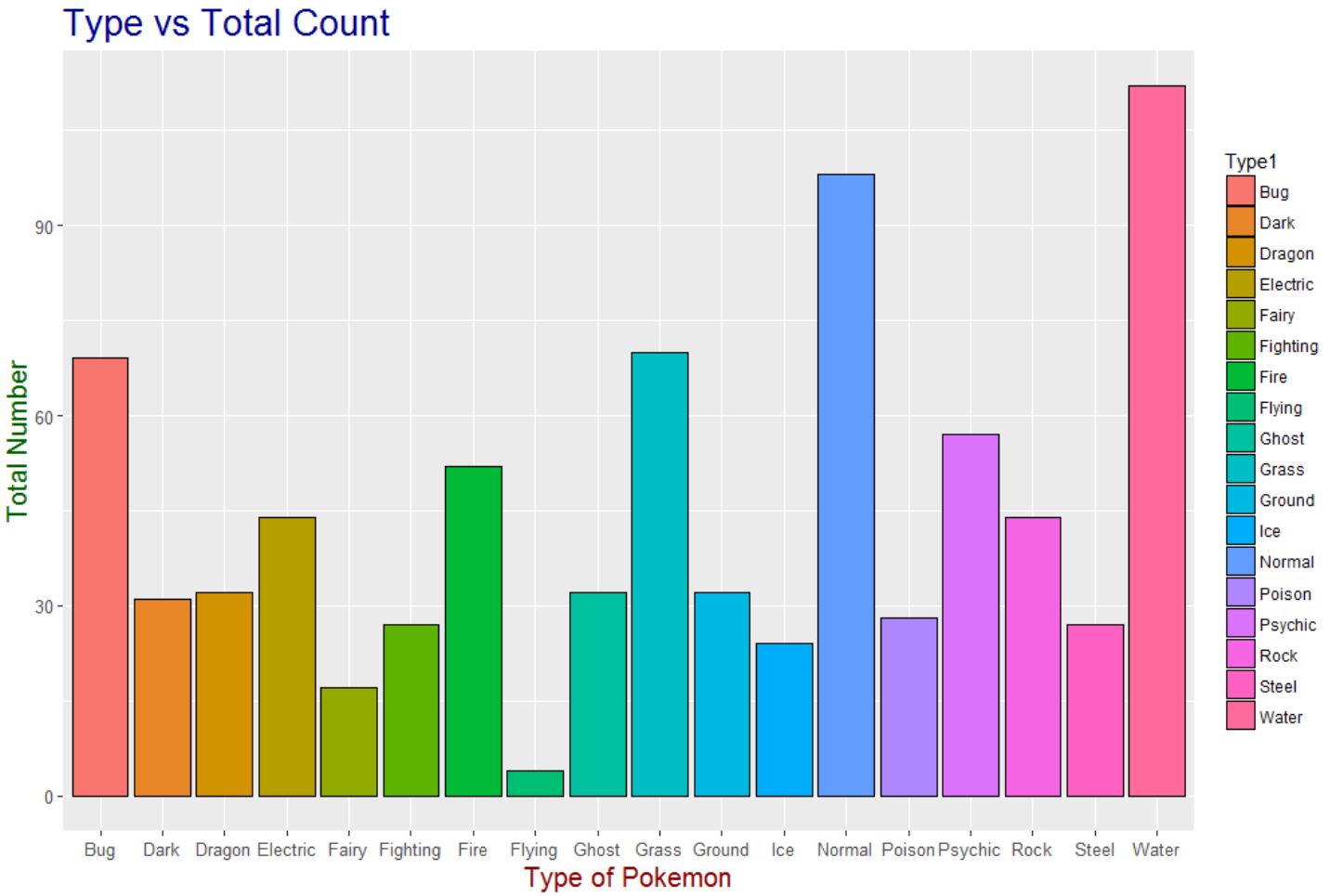


Result: Gen 2 Pokemon have the highest HP, followed by Gen 4, and Gen 5. Gen 1 have the lowest HP.

2. Check the most common types of Pokemon (only type1 considered):

```
b <- ggplot(data = pokemon, aes(x=Type1))

b + geom_histogram(stat="count", aes(fill=Type1), color="Black") + ggtitle("Type vs Total Count") +
  xlab("Type of Pokemon") + ylab("Total Number") +  theme(axis.title.x = element_text(colour = "DarkRed",size = 15),
      axis.title.y = element_text(colour = "DarkGreen",size = 15), axis.text.x = element_text(size = 10),
      axis.text.y = element_text(size = 10), plot.title = element_text(colour = "DarkBlue",size = 20))
```

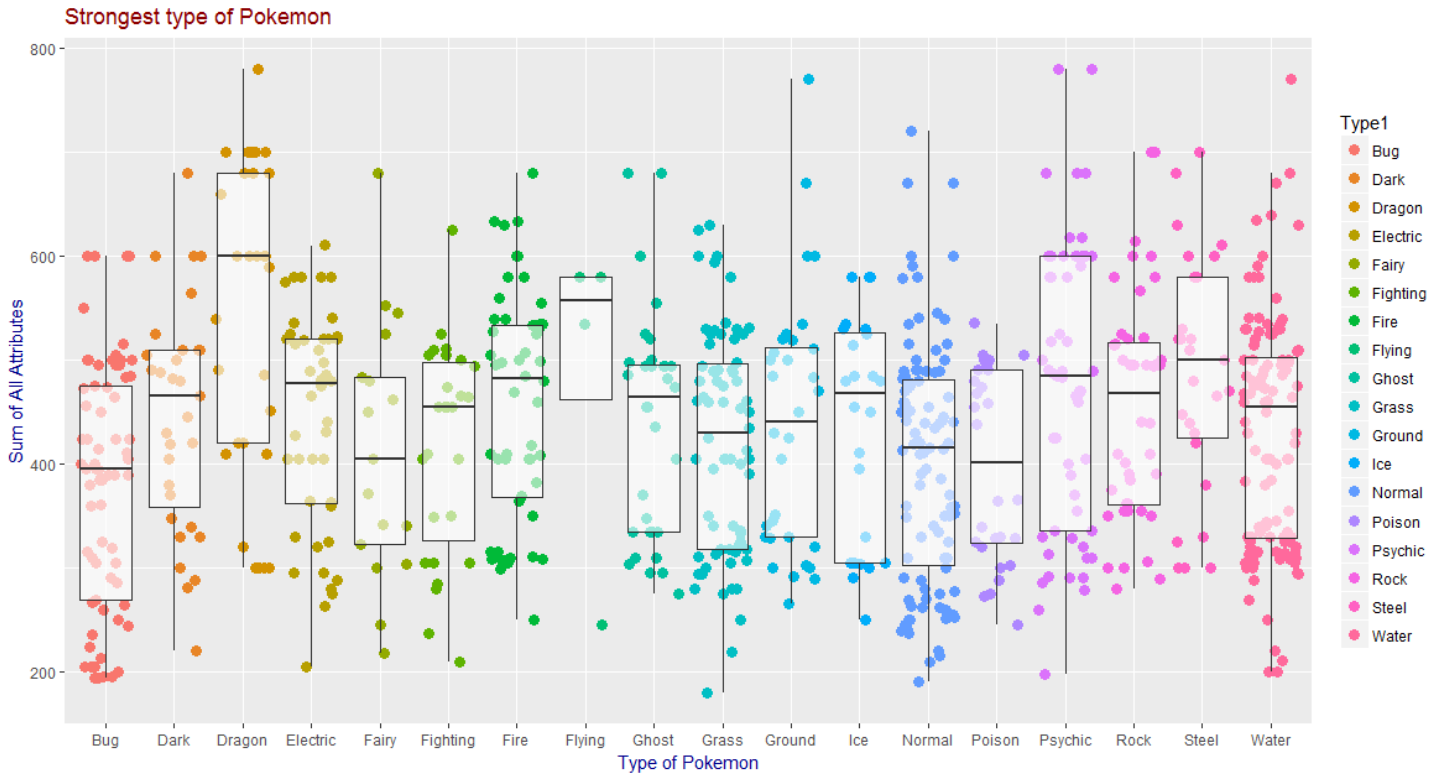


Result: Water Pokemon are the most common type of Pokemon while Flying are the least common type.

3. What type of Pokemon are the strongest?

```
c <- ggplot(data = pokemon, aes(x=Type1, y=Total))

c + geom_jitter(aes(color=Type1),size=3) +  geom_boxplot(alpha=0.6,outlier.colour = NA) + ggtitle("Strongest type of Pokemon") + xlab("Type of Pokemon") + ylab("Sum of All Attributes") + theme(axis.title.x = element_text(color = "DarkBlue"), axis.title.y = element_text(color = "DarkBlue"), plot.title = element_text(color = "DarkRed"))
```



Result: Dragon Pokemon are the Strongest Type.

4. Do Pokemon get stronger over generation?

d <- ggplot(data = pokemon, aes(x=Gen, y=Total))

d + geom_point(aes(color=Gen)) + geom_boxplot(alpha=0.6,outlier.colour = NA) + ggtitle("Pokemon Strength across Generations") + xlab("Generations") + ylab("Sum of Attributes")