Data Cleaning using Pokemon Data

Perebibowei Azazi

2022-06-21

##Install Packages and load data ## Load the data you can do this in mutiple ways

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
Pokemon <- read_csv("Pokemon.csv")
```

```
## Rows: 1168 Columns: 10
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): #, Name, Type
## dbl (7): Total, HP, Attack, Defense, Special Attack, Special Defense, Speed
##
## 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.
```

Next View the Data to make sure it reads in properly

```
tibble(Pokemon)
```

```
## # A tibble: 1,168 × 10
##    `#`         Name          Type   Total    HP Attack Defense `Special Attack`
##    <chr>       <chr>         <chr>  <dbl> <dbl>  <dbl>   <dbl>            <dbl>
##  1 "\xa0001"   Bulbasaur     GRASS    318    45     49      49               65
##  2 "\xa0001"   Bulbasaur     POISON   318    45     49      49               65
##  3 "\xa0002"   Ivysaur       GRASS    405    60     62      63               80
##  4 "\xa0002"   Ivysaur       POISON   405    60     62      63               80
##  5 "\xa0003"   Venusaur      GRASS    525    80     82      83              100
##  6 "\xa0003"   Venusaur      POISON   525    80     82      83              100
##  7 "\xa0003.1" Mega Venusaur GRASS    625    80    100     123              122
##  8 "\xa0003.1" Mega Venusaur POISON   625    80    100     123              122
##  9 "\xa0004"   Charmander    FIRE     309    39     52      43               60
## 10 "\xa0005"   Charmeleon    FIRE     405    58     64      58               80
## # … with 1,158 more rows, and 2 more variables: `Special Defense` <dbl>,
## #   Speed <dbl>
```

now lets check for nulls

```
sum(is.na(Pokemon))
```

```
## [1] 0
```

so it looks like we have zero nulls that is good

##Incase we did we can use this code to drop nulls

```
Pokemon <-na.omit(Pokemon)
```

Next we need to find any duplicate values

```
Pokemon <- distinct(Pokemon)
```

Now it is time for us to explore the data to see what the data is telling us

```
## Here we are filtering down the data
Pokemon %>% filter(Type=="FIRE",Total>600)
```

```
## # A tibble: 5 × 10
##   `#`   Name  Type  Total    HP Attack Defense `Special Attack` `Special Defen…`
##   <chr> <chr> <chr> <dbl> <dbl>  <dbl>   <dbl>            <dbl>            <dbl>
## 1 "\xa… Mega… FIRE    634    78    130     111              130               85
## 2 "\xa… Mega… FIRE    634    78    104      78              159              115
## 3 "\xa… Ho-oh FIRE    680   106    130      90              110              154
## 4 "\xa… Mega… FIRE    630    80    160      80              130               80
## 5 "\xa… Resh… FIRE    680   100    120     100              150              120
## # … with 1 more variable: Speed <dbl>
```

```
Pokemon %>% filter(HP>120, Attack>150)
```

```
## # A tibble: 3 × 10
##   `#`   Name  Type  Total    HP Attack Defense `Special Attack` `Special Defen…`
##   <chr> <chr> <chr> <dbl> <dbl>  <dbl>   <dbl>            <dbl>            <dbl>
## 1 "\xa… Slak… NORM…   670   150    160     100               95               65
## 2 "\xa… Blac… DRAG…   700   125    170     100              120               90
## 3 "\xa… Blac… ICE     700   125    170     100              120               90
## # … with 1 more variable: Speed <dbl>
```

Find the Pokemon with the Higest Total stats

```
strongest <- Pokemon %>%
  filter(Total==max(Pokemon$Total))
head(strongest)
```

```
## # A tibble: 3 × 10
##   `#`   Name  Type  Total    HP Attack Defense `Special Attack` `Special Defen…`
##   <chr> <chr> <chr> <dbl> <dbl>  <dbl>   <dbl>            <dbl>            <dbl>
## 1 "\xa… Mega… PSYC…   780   106    190     100              154              100
## 2 "\xa… Mega… FIGH…   780   106    190     100              154              100
## 3 "\xa… Mega… PSYC…   780   106    150      70              194              120
## # … with 1 more variable: Speed <dbl>
```

Which type has the highest total on average

```
group_type <- aggregate(Pokemon$Total, list(Pokemon$Type), mean) %>% arrange(desc(x))
group_type
```

```
##     Group.1        x
## 1    DRAGON 522.4545
## 2     STEEL 481.5000
## 3  FIGHTING 464.8627
## 4       ICE 464.4054
## 5   PSYCHIC 461.8780
## 6      FIRE 458.4426
## 7      DARK 453.6667
## 8    FLYING 446.1020
## 9  ELECTRIC 444.0400
## 10     ROCK 441.6071
## 11    GHOST 432.1818
## 12   GROUND 427.4677
## 13    WATER 420.5041
## 14    GRASS 414.9355
## 15   NORMAL 396.4343
## 16    FAIRY 395.6471
## 17   POISON 394.9508
## 18      BUG 377.1972
```

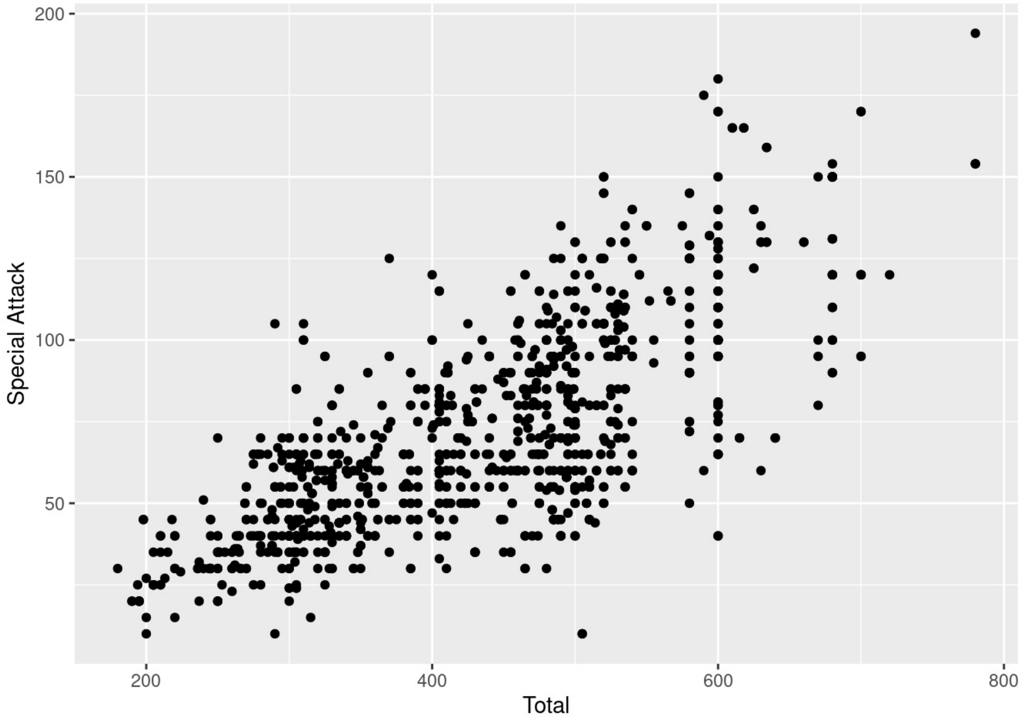
Now we will find out what Type has the the lowest HP on avg

```
group_type2 <- aggregate(Pokemon$HP,list(Pokemon$Type),mean) %>%
  arrange(x)
group_type2
```

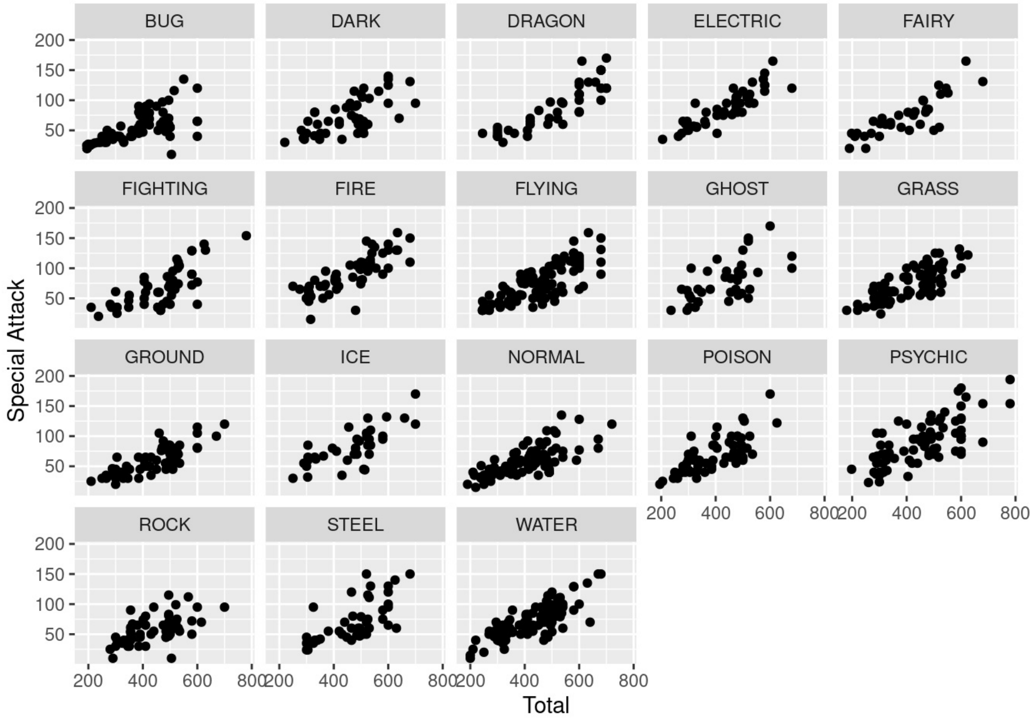
```
##     Group.1        x
## 1       BUG 56.61972
## 2    POISON 62.55738
## 3     GHOST 62.72727
## 4  ELECTRIC 63.20000
## 5     STEEL 64.52174
## 6     GRASS 66.07527
## 7      ROCK 66.58929
## 8     FAIRY 69.44118
## 9      FIRE 69.50820
## 10    WATER 70.28099
## 11  PSYCHIC 70.39024
## 12     DARK 70.45833
## 13   FLYING 70.66327
## 14 FIGHTING 74.88235
## 15   GROUND 75.14516
## 16   NORMAL 76.52525
## 17      ICE 78.59459
## 18   DRAGON 82.72727
```

Now lets graph some data!

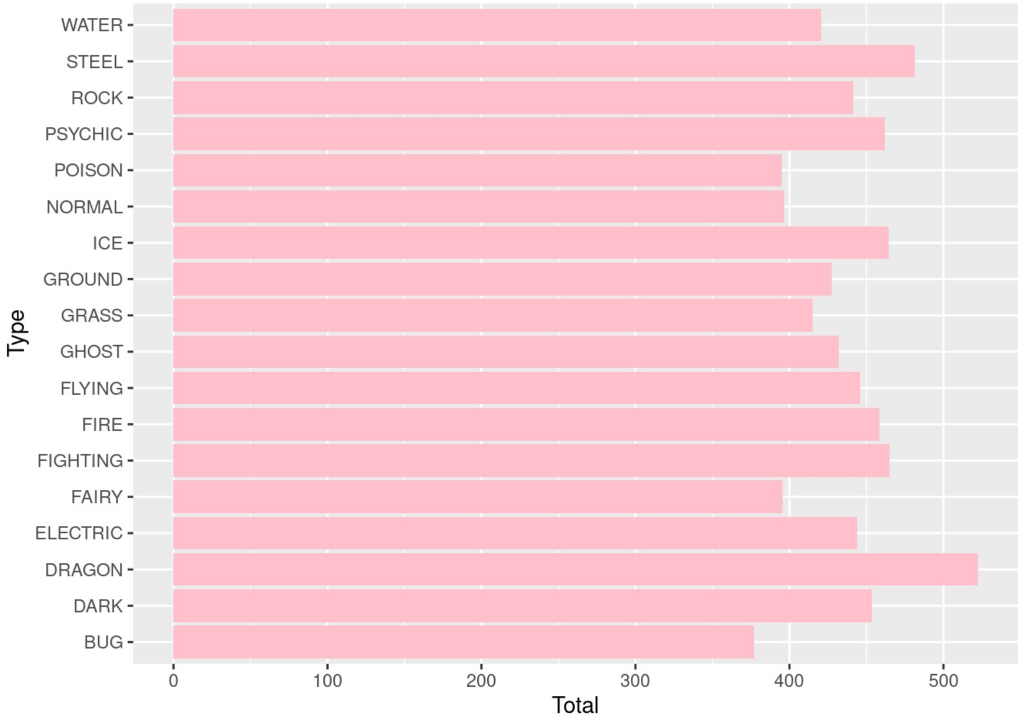
```
## now lets see the corelations between sp.attack and Total stats
ggplot(Pokemon, mapping = aes(x=Total,y=`Special Attack`)) + geom_point()
```



```
##now lets see the same chart but by Pokemon  type
ggplot(Pokemon, mapping = aes(x=Total,y=`Special Attack`)) + geom_point() + facet_wrap('Type')
```



```
## lastly let plot total stats by type
ggplot(Pokemon,mapping =  aes(x = Total, y = Type)) +
  stat_summary(fun = "mean", geom = "bar", fill= 'Pink')
```


WATER
STEEL
ROCK
PSYCHIC
POISON
NORMAL
ICE
GROUND
GRASS
GHOST
FLYING
FIRE
FIGHTING
FAIRY
ELECTRIC
DRAGON
DARK
BUG
Type
0
100
200
300
400
500
Total