Elden Ring Weapons Data

2022-07-23

Introduction

Elden Ring is an open-world adventure/fantasy video game which you defeat bosses to claim the title as Elden Lord and claim the Elden Ring. This analysis focuses on the weapons available to the player and their stats. This dataset can be found at https://www.kaggle.com/datasets/l3llff/-elden-ring-weapons

Load the main packages

```
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 1.3.2 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
install.packages("here")
```

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

```
library("here")
```

```
## here() starts at /cloud/project
```

```
install.packages("skimr")
```

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

```
library("skimr")
install.packages("janitor")
```

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

```
library("janitor")
```

```
##
## Attaching package: 'janitor'
##
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library("dplyr")
```

Import and read the csv

```
weapons <- read.csv("elden_ring_weapon.csv", header = TRUE, sep = ",")
```

First look at the data

```
glimpse(weapons)
```

```
## Rows: 307
## Columns: 23
## $ Name                   <chr> "Academy Glintstone Staff", "Alabaster Lord's S~
## $ Type                   <chr> "Glintstone Staff", "Greatsword", "Bow", "Glint~
## $ PhysicalDamage         <int> 43, 313, 200, 29, 240, 308, 43, 404, 347, 58, 2~
## $ MagicalDamage          <int> NA, 93, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ FireDamage             <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ LightDamage            <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ HolyDamage             <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ CriticalDamage         <int> 100, 100, 100, 100, 100, 100, 100, 100, 100, 10~
## $ StaminaUsage           <int> 40, 126, 60, 38, 62, 68, 40, 150, 150, 42, 98, ~
## $ StrengthScaling        <int> 2, 4, 1, 2, 2, NA, 2, 3, 3, 3, 2, 3, 3, 3, 1, 3~
## $ DexterityScaling       <int> NA, 2, 2, NA, 3, NA, NA, 2, 3, NA, 3, 2, 3, 2, ~
## $ IntelligenceScaling    <int> 6, 2, NA, 4, NA, NA, 6, NA, NA, 6, NA, NA, NA, ~
## $ FaithScaling           <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ ArcaneScaling          <int> NA, NA, NA, 6, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ PhysicalBlockingDamage <int> 25, 56, NA, 23, 47, NA, 25, 75, 65, 25, 48, 68,~
## $ MagicalBlockingDamage  <int> 15, 33, NA, 14, 31, NA, 15, 45, 35, 15, 36, 36,~
## $ FireBlockingDamage     <int> 15, 27, NA, 14, 31, NA, 15, 45, 35, 15, 36, 36,~
## $ LightBlockingDamage    <int> 15, 27, NA, 14, 31, NA, 15, 45, 35, 15, 36, 36,~
## $ HolyBlockingDamage     <int> 15, 27, NA, 14, 31, NA, 15, 45, 35, 15, 36, 36,~
## $ Boost                  <int> 15, 39, NA, 14, 25, NA, 15, 50, 42, 15, 36, 52,~
## $ Rest                   <int> 10, 19, NA, 9, 10, NA, 10, 25, 20, 10, 12, 21, ~
## $ WeightOfWeapon         <dbl> 3.0, 8.0, 4.5, 2.5, 3.0, 6.0, 3.0, 18.0, 11.0, ~
## $ Upgrade                <chr> "Smithing Stones", "Somber Smithing Stones", "S~
```

```
str(weapons)
```

```
## 'data.frame':    307 obs. of  23 variables:
##  $ Name                   : chr  "Academy Glintstone Staff" "Alabaster Lord's Sword" "Albinauric Bow"
##  $ Type                   : chr  "Glintstone Staff" "Greatsword" "Bow" "Glintstone Staff" ...
##  $ PhysicalDamage         : int  43 313 200 29 240 308 43 404 347 58 ...
##  $ MagicalDamage          : int  NA 93 NA NA NA NA NA NA NA NA ...
##  $ FireDamage             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ LightDamage            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ HolyDamage             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CriticalDamage         : int  100 100 100 100 100 100 100 100 100 100 ...
##  $ StaminaUsage           : int  40 126 60 38 62 68 40 150 150 42 ...
##  $ StrengthScaling        : int  2 4 1 2 2 NA 2 3 3 3 ...
##  $ DexterityScaling       : int  NA 2 2 NA 3 NA NA 2 3 NA ...
```

```
##  $ IntelligenceScaling   : int  6 2 NA 4 NA NA 6 NA NA 6 ...
##  $ FaithScaling          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ArcaneScaling         : int  NA NA NA 6 NA NA NA NA NA NA ...
##  $ PhysicalBlockingDamage: int  25 56 NA 23 47 NA 25 75 65 25 ...
##  $ MagicalBlockingDamage : int  15 33 NA 14 31 NA 15 45 35 15 ...
##  $ FireBlockingDamage    : int  15 27 NA 14 31 NA 15 45 35 15 ...
##  $ LightBlockingDamage   : int  15 27 NA 14 31 NA 15 45 35 15 ...
##  $ HolyBlockingDamage    : int  15 27 NA 14 31 NA 15 45 35 15 ...
##  $ Boost                 : int  15 39 NA 14 25 NA 15 50 42 15 ...
##  $ Rest                  : int  10 19 NA 9 10 NA 10 25 20 10 ...
##  $ WeightOfWeapon        : num  3 8 4.5 2.5 3 6 3 18 11 4 ...
##  $ Upgrade               : chr  "Smithing Stones" "Somber Smithing Stones" "Smithing Stones" "Smithi
```

```
summary(weapons)
```

```
##      Name               Type           PhysicalDamage  MagicalDamage
##  Length:307         Length:307         Min.   :  0.0   Min.   : 58.0
##  Class :character   Class :character   1st Qu.:203.0   1st Qu.:137.0
##  Mode  :character   Mode  :character   Median :267.0   Median :166.0
##                                        Mean   :250.2   Mean   :161.9
##                                        3rd Qu.:303.0   3rd Qu.:196.0
##                                        Max.   :672.0   Max.   :328.0
##                                                        NA's   :270
##    FireDamage     LightDamage      HolyDamage    CriticalDamage
##  Min.   : 75.0   Min.   :129.0   Min.   : 36.0   Min.   :100.0
##  1st Qu.:151.0   1st Qu.:140.2   1st Qu.:164.8   1st Qu.:100.0
##  Median :176.0   Median :149.0   Median :191.0   Median :100.0
##  Mean   :176.2   Mean   :154.5   Mean   :195.7   Mean   :101.2
##  3rd Qu.:191.0   3rd Qu.:163.2   3rd Qu.:224.2   3rd Qu.:100.0
##  Max.   :267.0   Max.   :191.0   Max.   :301.0   Max.   :140.0
##  NA's   :286     NA's   :303     NA's   :275
##   StaminaUsage   StrengthScaling DexterityScaling IntelligenceScaling
##  Min.   : 38.0   Min.   :1.000   Min.   :1.000    Min.   :1.000
##  1st Qu.: 80.0   1st Qu.:2.000   1st Qu.:2.000    1st Qu.:2.000
##  Median :100.0   Median :3.000   Median :2.000    Median :3.000
##  Mean   :105.3   Mean   :2.608   Mean   :2.586    Mean   :3.686
##  3rd Qu.:128.0   3rd Qu.:3.000   3rd Qu.:3.000    3rd Qu.:5.500
##  Max.   :224.0   Max.   :6.000   Max.   :4.000    Max.   :6.000
##                  NA's   :16      NA's   :44       NA's   :256
##   FaithScaling   ArcaneScaling   PhysicalBlockingDamage MagicalBlockingDamage
##  Min.   :2.000   Min.   :2.000   Min.   : 9.00          Min.   :10.0
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:36.00          1st Qu.:25.0
##  Median :3.000   Median :3.000   Median :47.00          Median :33.0
##  Mean   :3.448   Mean   :3.667   Mean   :48.15          Mean   :31.5
##  3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.:61.00          3rd Qu.:36.0
##  Max.   :6.000   Max.   :6.000   Max.   :88.00          Max.   :63.0
##  NA's   :249     NA's   :292     NA's   :25             NA's   :25
##  FireBlockingDamage LightBlockingDamage HolyBlockingDamage     Boost
##  Min.   :13.00      Min.   :10.00       Min.   :10.0       Min.   :14.00
##  1st Qu.:25.00      1st Qu.:24.25       1st Qu.:25.0       1st Qu.:25.00
##  Median :31.00      Median :31.00       Median :31.0       Median :36.00
##  Mean   :30.57      Mean   :29.81       Mean   :31.6       Mean   :35.77
##  3rd Qu.:36.00      3rd Qu.:36.00       3rd Qu.:36.0       3rd Qu.:47.00
##  Max.   :54.00      Max.   :52.00       Max.   :72.0       Max.   :69.00
##  NA's   :25         NA's   :25          NA's   :25         NA's   :25
```

```
##       Rest       WeightOfWeapon     Upgrade
##  Min.   : 9.00   Min.   : 0.000   Length:307
##  1st Qu.:11.00   1st Qu.: 3.000   Class :character
##  Median :15.00   Median : 5.500   Mode  :character
##  Mean   :15.98   Mean   : 7.124
##  3rd Qu.:20.00   3rd Qu.:10.000
##  Max.   :29.00   Max.   :26.500
##  NA's   :25
```

Correlation Statistics

Initially, I'm interested in the relationships between the basic weapons stats.

```
cor(weapons[c("PhysicalDamage","StaminaUsage")])
```

```
##                PhysicalDamage StaminaUsage
## PhysicalDamage      1.0000000    0.7338257
## StaminaUsage        0.7338257    1.0000000
```

```
cor(weapons[c("PhysicalDamage","StaminaUsage","CriticalDamage","WeightOfWeapon")])
```

```
##                PhysicalDamage StaminaUsage CriticalDamage WeightOfWeapon
## PhysicalDamage      1.0000000    0.7338257     -0.0495295      0.6478566
## StaminaUsage        0.7338257    1.0000000     -0.1808315      0.8092308
## CriticalDamage     -0.0495295   -0.1808315      1.0000000     -0.1702511
## WeightOfWeapon      0.6478566    0.8092308     -0.1702511      1.0000000
```

We see a strong positive correlation between physical damage (how much damage the weapon does to an enemy), the stamina used per use, and the weight of the weapon. This makes sense - heavier weapons logically create more damage, but use lots of stamina to move them. However, when we factor in critical damage (how much damage is done with a critical hit), we don't see the same trend. For the vast majority of weapons available, critical damage is the same (100 damage).

Correlation between Damage and Blocking Damage

Next, I'm looking at the relationships between the amount of damage weapons can inflict vs. the damage the player receives when blocking with that weapon (i.e., you hit vs. you parry), broken down by damage type.

```
cor(weapons[c("PhysicalDamage","PhysicalBlockingDamage")], use = "complete")
```

```
##                        PhysicalDamage PhysicalBlockingDamage
## PhysicalDamage              1.0000000              0.8320234
## PhysicalBlockingDamage      0.8320234              1.0000000
```

```
cor(weapons[c("MagicalDamage","MagicalBlockingDamage")], use = "complete")
```

```
##                        MagicalDamage MagicalBlockingDamage
## MagicalDamage              1.000000              0.773124
## MagicalBlockingDamage      0.773124              1.000000
```

```
cor(weapons[c("FireDamage","FireBlockingDamage")], use = "complete")
```

```
##                     FireDamage FireBlockingDamage
## FireDamage           1.0000000         -0.1574888
## FireBlockingDamage -0.1574888          1.0000000
```

```
cor(weapons[c("LightDamage","LightBlockingDamage")], use = "complete")
```

```
##                      LightDamage LightBlockingDamage
```

```
## LightDamage           1.0000000           0.8037503
## LightBlockingDamage   0.8037503           1.0000000
```

```
cor(weapons[c("HolyDamage","HolyBlockingDamage")], use = "complete")
```

```
##                    HolyDamage HolyBlockingDamage
## HolyDamage          1.0000000          0.8096308
## HolyBlockingDamage  0.8096308          1.0000000
```

Results: Fire is the only damage type that shows a negative correlation. Why is this important? A negative correlation in this relationship means that you deal a high amount of damage, but you take low damage when parrying with these weapons. This could be a reason why some of the weapons that have fire damage/fire blocking damage are considered "very powerful" in the gaming community, including Rivers of Blood (a katana).

##Load some additional packages for further exploration (mosaic)

I want to use the tally function and favstats, so let's load mosaic.

```
install.packages("mosaic")
```

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

```
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Attaching package: 'mosaic'
## The following object is masked from 'package:Matrix':
##
##     mean
## The following object is masked from 'package:skimr':
##
##     n_missing
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
## The following object is masked from 'package:purrr':
##
##     cross
## The following object is masked from 'package:ggplot2':
##
##     stat
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

Stat scaling breakdown

Scaling for each weapon stat shows how quickly you can level up that stat. For ease of use, the scale has been swapped to a numerical format: (S, A, B, C, D, E -> 6, 5, 4, 3, 2, 1).

```
tally(~StrengthScaling, data=weapons, margins=TRUE)
```

```
## StrengthScaling
##     1     2     3     4     5     6  <NA> Total
##    31   110    98    47     4     1    16   307
```

```
tally(~DexterityScaling, data=weapons, margins=TRUE)
```

```
## DexterityScaling
##     1     2     3     4  <NA> Total
##    24   110    80    49    44   307
```

```
tally(~IntelligenceScaling, data=weapons, margins=TRUE)
```

```
## IntelligenceScaling
##     1     2     3     4     5     6  <NA> Total
##     1    16     9    10     2    13   256   307
```

```
tally(~FaithScaling, data=weapons, margins=TRUE)
```

```
## FaithScaling
##     2     3     4     5     6  <NA> Total
##    16    16    16     4     6   249   307
```

```
tally(~ArcaneScaling, data=weapons, margins=TRUE)
```

```
## ArcaneScaling
##     2     3     4     6  <NA> Total
##     5     3     3     4   292   307
```

Let's swap this to percentages.

```
tally(~StrengthScaling, data=weapons, margins=TRUE, format = "perc")
```

```
## StrengthScaling
##           1           2           3           4           5           6
##  10.0977199  35.8306189  31.9218241  15.3094463   1.3029316   0.3257329
##        <NA>       Total
##   5.2117264 100.0000000
```

```
tally(~DexterityScaling, data=weapons, margins=TRUE, format = "perc")
```

```
## DexterityScaling
##         1         2         3         4      <NA>     Total
##   7.81759  35.83062  26.05863  15.96091  14.33225 100.00000
```

```
tally(~IntelligenceScaling, data=weapons, margins=TRUE, format = "perc")
```

```
## IntelligenceScaling
##           1           2           3           4           5           6
##   0.3257329   5.2117264   2.9315961   3.2573290   0.6514658   4.2345277
```

```
##        <NA>       Total
##  83.3876221 100.0000000
```

```
tally(~FaithScaling, data=weapons, margins=TRUE, format = "perc")
```

```
## FaithScaling
##          2          3          4          5          6       <NA>      Total
##   5.211726   5.211726   5.211726   1.302932   1.954397  81.107492 100.000000
```

```
tally(~ArcaneScaling, data=weapons, margins=TRUE, format = "perc")
```

```
## ArcaneScaling
##          2          3          4          6       <NA>       Total
##   1.6286645  0.9771987  0.9771987  1.3029316 95.1140065 100.0000000
```

Some stats don't use the whole range - Faith and Arcane scaling starts at 2, and no weapon has Dexterity scaling over 4.

Also, this shows that weapons are more likely to have Strength or Dexterity scaling capabilties rather than Intelligence, Faith, or Arcane scaling. If a player wants to build their character around the latter stats, their weapon choice is more limited.

Further statistical analysis

Use favstats to look at further statistics of the basic weapons stats.

```
favstats(~PhysicalDamage, data=weapons)
```

```
##  min  Q1 median  Q3 max     mean       sd   n missing
##    0 203    267 303 672 250.2052 92.62755 307       0
```

```
favstats(~CriticalDamage, data=weapons)
```

```
##  min  Q1 median  Q3 max     mean       sd   n missing
##  100 100    100 100 140 101.1694 4.421129 307       0
```

```
favstats(~StaminaUsage, data=weapons)
```

```
##  min Q1 median  Q3 max     mean       sd   n missing
##   38 80    100 128 224 105.3355 41.60978 307       0
```

```
favstats(~WeightOfWeapon, data=weapons)
```

```
##  min Q1 median Q3  max     mean       sd   n missing
##    0  3    5.5 10 26.5 7.123779 5.194516 307       0
```

Physical Damage and Stamina Usage have the largest standard deviations. Critical Damage's min-Q3 are the same and the mean is slightly higher, showing that the vast majority of Critical Damage data points are the same. This verifies what we discovered earlier - the vast majority of weapons have a critical damage of 100.

Visualizations

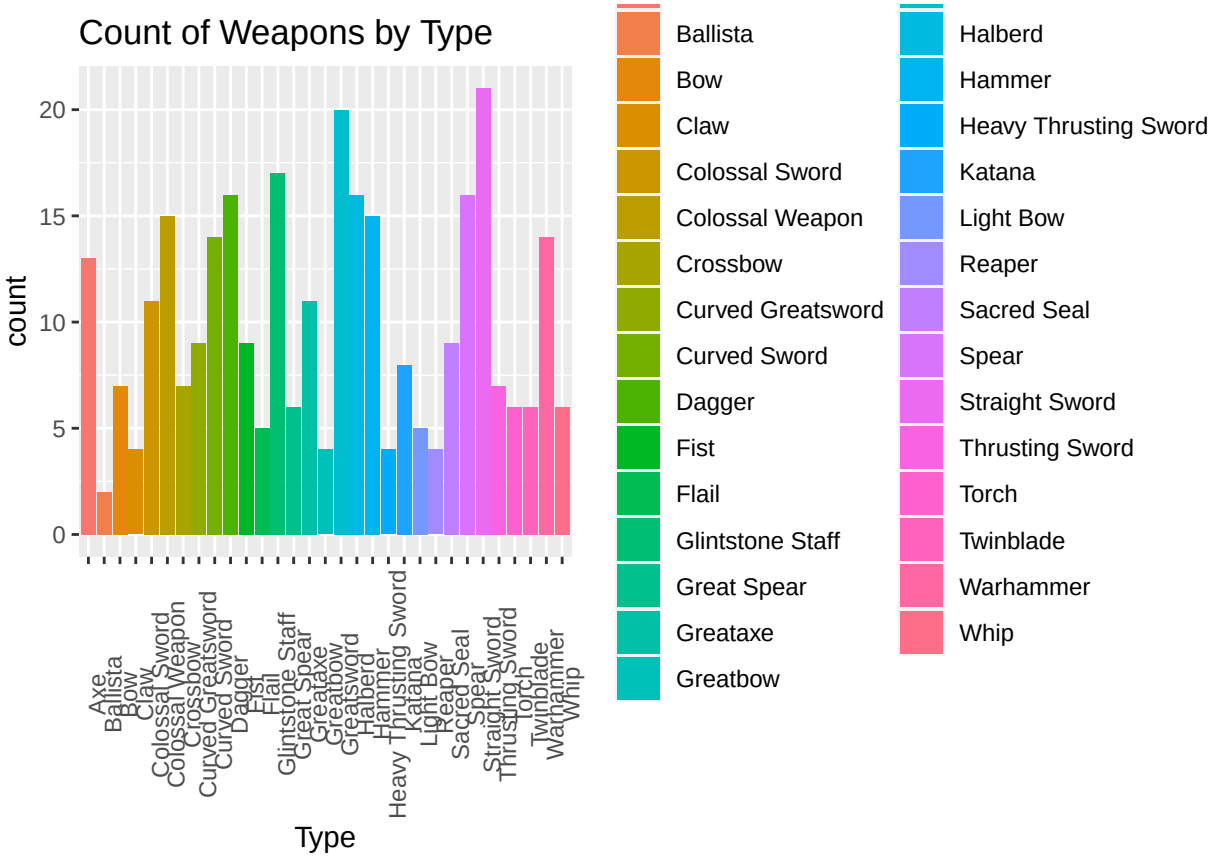
Let's load ggplot2.

```
library("ggplot2")
```

I want to see the distribution of different types of weapons available in the game.

```
ggplot(data=weapons)+
  geom_bar(mapping = aes(x=Type, fill=Type))+
```

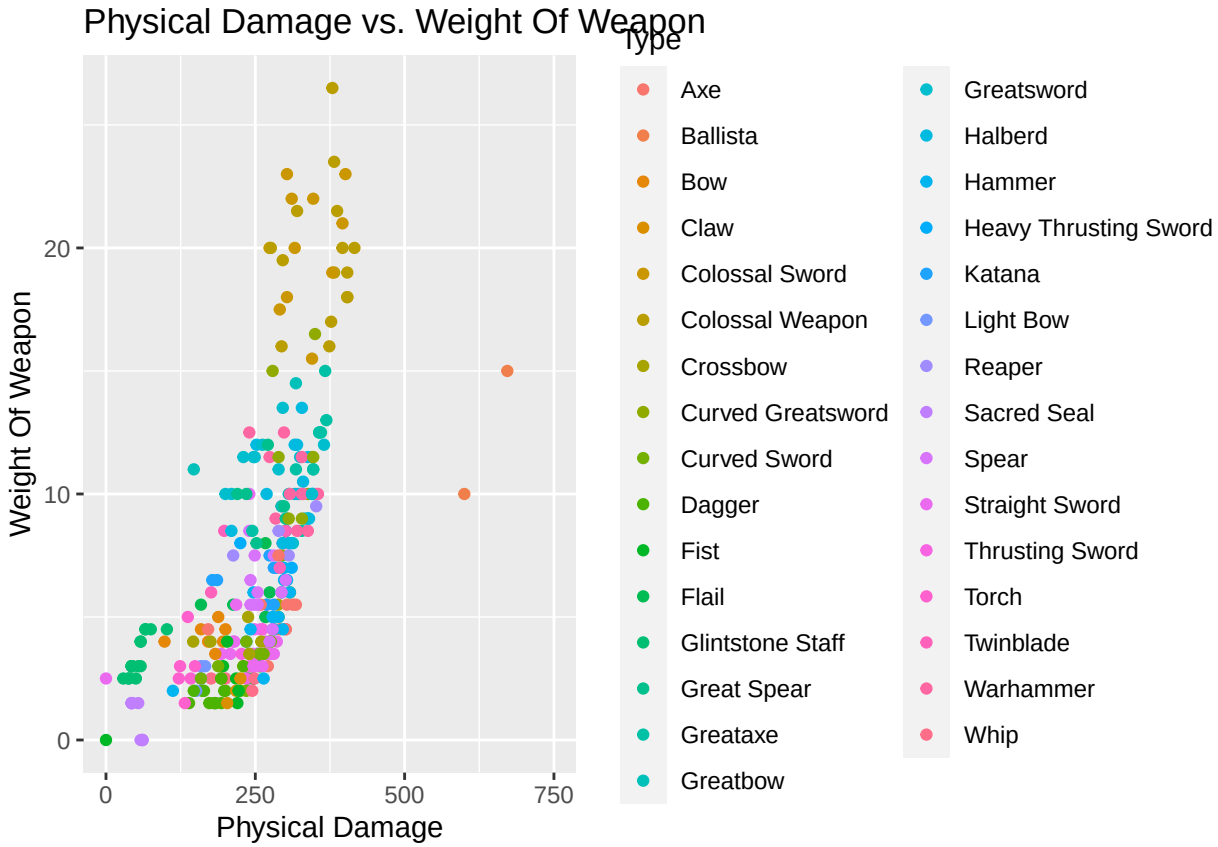
```
  ggtitle("Count of Weapons by Type")+
  theme(axis.text.x = element_text(angle = 90))
```



Straight Swords and Great Swords are the most common weapons available in the game.

Next, let's make a plot of Physical Damage vs. Weight of Weapon.

```
ggplot(data=weapons)+
  geom_point(mapping = aes(x=as.numeric(PhysicalDamage), y=WeightOfWeapon, color=Type))+
  scale_x_continuous(name = "Physical Damage", limits = c(0, 750), breaks = c(0, 250, 500, 750))+
  ggtitle("Physical Damage vs. Weight Of Weapon")+
  ylab("Weight Of Weapon")
```



Colossal Swords and Weapons are the heaviest, but Ballistas inflict the most physical damage for their weight.

Conclusion/tl;dr:

Heavy weapons can inflict great damage but use more stamina. Weapons with Fire damage/Fire blocking damage are better for parrying. Players who make specialty builds (Intelligence, Faith, or Arcane-based) have fewer options for weapons scaling. The vast majority of weapons have the same Critical damage.

Thanks for reading! - Delaney