Sandbox

Eren Akgunduz

25 April 2023

The data, wrangled and cleaned/tidied

```
library(tidyverse)
df <- read.csv("data/all.csv", stringsAsFactors = FALSE)
```

Make sure it's clean

```
# Check for missing values
na_rows <- df[!complete.cases(df), ]
na_rows
```

```
##  [1] Character  Move       onBlock    plnCmd     airmove    followUp
##  [7] projectile moveType   health     stun       vgauge1    vgauge2
## [13] fDash      bDash      fWalk      bWalk      throwHurt  throwRange
## [19] Damage     Stun
## <0 rows> (or 0-length row.names)
```

Validate before we really begin

```
c(nrow(df), ncol(df))
```

```
## [1] 1809   20
```

```
dim(df) # same thing
```

```
## [1] 1809   20
```

```
summary(df)
```

```
##   Character             Move              onBlock           plnCmd
##  Length:1809        Length:1809        Min.   :-95.000   Length:1809
##  Class :character   Class :character   1st Qu.: -8.000   Class :character
##  Mode  :character   Mode  :character   Median : -3.000   Mode  :character
##                                        Mean   : -4.429
##                                        3rd Qu.:  1.000
##                                        Max.   : 48.000
##    airmove            followUp          projectile          moveType
##  Length:1809        Length:1809        Length:1809        Length:1809
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
##      health           stun          vgauge1         vgauge2
##  Min.   : 900.0   Min.   : 900.0   Min.   :300.0   Min.   :600.0
##  1st Qu.: 950.0   1st Qu.: 950.0   1st Qu.:600.0   1st Qu.:600.0
##  Median :1000.0   Median :1000.0   Median :600.0   Median :600.0
##  Mean   : 987.8   Mean   : 997.2   Mean   :713.9   Mean   :663.5
##  3rd Qu.:1025.0   3rd Qu.:1050.0   3rd Qu.:900.0   3rd Qu.:600.0
##  Max.   :1100.0   Max.   :1100.0   Max.   :900.0   Max.   :900.0
##      fDash          bDash           fWalk             bWalk
##  Min.   :15.0   Min.   :21.00   Min.   :0.02200   Min.   :0.02000
##  1st Qu.:16.0   1st Qu.:21.00   1st Qu.:0.04000   1st Qu.:0.03000
##  Median :17.0   Median :23.00   Median :0.04700   Median :0.03200
##  Mean   :17.5   Mean   :22.99   Mean   :0.04377   Mean   :0.03214
##  3rd Qu.:18.0   3rd Qu.:24.00   3rd Qu.:0.05000   3rd Qu.:0.03500
##  Max.   :25.0   Max.   :26.00   Max.   :0.05500   Max.   :0.04700
##    throwHurt        throwRange         Damage           Stun
##  Min.   :0.2500   Min.   :0.8000   Min.   : 10.0   Min.   :  0.0
##  1st Qu.:0.2500   1st Qu.:0.8000   1st Qu.: 60.0   1st Qu.:100.0
##  Median :0.2500   Median :0.8500   Median : 80.0   Median :120.0
##  Mean   :0.2852   Mean   :0.8358   Mean   : 88.2   Mean   :122.2
##  3rd Qu.:0.3000   3rd Qu.:0.8500   3rd Qu.:100.0   3rd Qu.:150.0
##  Max.   :0.4500   Max.   :1.0000   Max.   :400.0   Max.   :400.0
```

```
str(df)
```

```
## 'data.frame':    1809 obs. of  20 variables:
##  $ Character : chr  "Abigail" "Abigail" "Abigail" "Abigail" ...
##  $ Move      : chr  "Stand LP" "Stand MP" "Stand HP" "Stand LK" ...
##  $ onBlock   : int  3 5 -3 3 0 -8 3 -2 -18 2 ...
##  $ plnCmd    : chr  "LP" "MP" "HP" "LK" ...
##  $ airmove   : chr  "False" "False" "False" "False" ...
##  $ followUp  : chr  "False" "False" "False" "False" ...
##  $ projectile: chr  "False" "False" "False" "False" ...
##  $ moveType  : chr  "normal" "normal" "normal" "normal" ...
##  $ health    : int  1100 1100 1100 1100 1100 1100 1100 1100 1100 1100 ...
##  $ stun      : int  1050 1050 1050 1050 1050 1050 1050 1050 1050 1050 ...
##  $ vgauge1   : int  900 900 900 900 900 900 900 900 900 900 ...
##  $ vgauge2   : int  900 900 900 900 900 900 900 900 900 900 ...
##  $ fDash     : int  25 25 25 25 25 25 25 25 25 25 ...
##  $ bDash     : int  25 25 25 25 25 25 25 25 25 25 ...
##  $ fWalk     : num  0.032 0.032 0.032 0.032 0.032 0.032 0.032 0.032 0.032 0.032 ...
##  $ bWalk     : num  0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 ...
##  $ throwHurt : num  0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 ...
##  $ throwRange: num  0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 ...
##  $ Damage    : int  40 70 90 50 80 90 40 70 90 40 ...
##  $ Stun      : int  70 100 150 70 100 150 70 100 150 70 ...
```

Module 1: Information (description)

Descriptive stats, here mainly seeking to answer the question: how do the distributions look?

All characters

```
library(tidyverse)
df <- read.csv("data/all.csv", stringsAsFactors = FALSE)
```

Character numerical features summary (averages)

```
colMeans(df[9:18])
```

```
##       health         stun      vgauge1      vgauge2        fDash        bDash
## 987.83858485 997.23604201 713.93034826 663.51575456  17.50082919  22.99281371
##        fWalk        bWalk    throwHurt   throwRange
##   0.04376739   0.03213941   0.28518159   0.83583223
```

Attack categorical features summary tables

```
# define the function to create each summary table
summary_tb <- function(feature) {
  df %>%
    group_by({{ feature }}) %>%
    summarise(
      count = n(),
      average_damage = mean(Damage),
      average_stun = mean(Stun)
    ) %>%
    arrange(desc(average_damage))
}
```

```
# call for each feature with its name as argument
summary_tb(plnCmd)
```

```
## # A tibble: 545 x 4
##    plnCmd                    count average_damage average_stun
##    <chr>                     <int>          <dbl>        <dbl>
##  1 Hold & Release PP/KK Max      1            370          400
##  2 qcf,qcf+P (can hold)          3            353.           0
##  3 qcb,qcb+K                     2            345            0
##  4 qcf,qcf+K                     8            339.          44.5
##  5 qcb,qcb+P                     1            330            0
##  6 qcf,qcf+P                    24            330            0
##  7 qcf,qcf+P or qcb,qcb+P        1            330            0
##  8 b,f,b,f+P                     1            320            0
##  9 Hold & Release PP/KK lvl9     1            290          300
## 10 Hold & Release PP/KK lvl8     1            260          300
## # i 535 more rows
```

```
summary_tb(moveType)
```

```
## # A tibble: 11 x 4
##    moveType         count average_damage average_stun
##    <chr>            <int>          <dbl>        <dbl>
##  1 super               40          334            8.9
##  2 command-grab        18          176.         207.
##  3 throw               22          131.         168.
##  4 vtrigger            38          106.         124.
```

```
##  5 special            701           95.8        140.
##  6 movement-special     2           90          100
##  7 vskill             165           81.1        119.
##  8 taunt                1           70          100
##  9 normal             757           68.1        119.
## 10 vbreak              41           60            0
## 11 alpha               24           55            0
```

```
summary_tb(airmove)
```

```
## # A tibble: 2 x 4
##   airmove count average_damage average_stun
##   <chr>   <int>          <dbl>        <dbl>
## 1 True      178           90.3         137.
## 2 False    1631           88.0         121.
```

```
summary_tb(followUp)
```

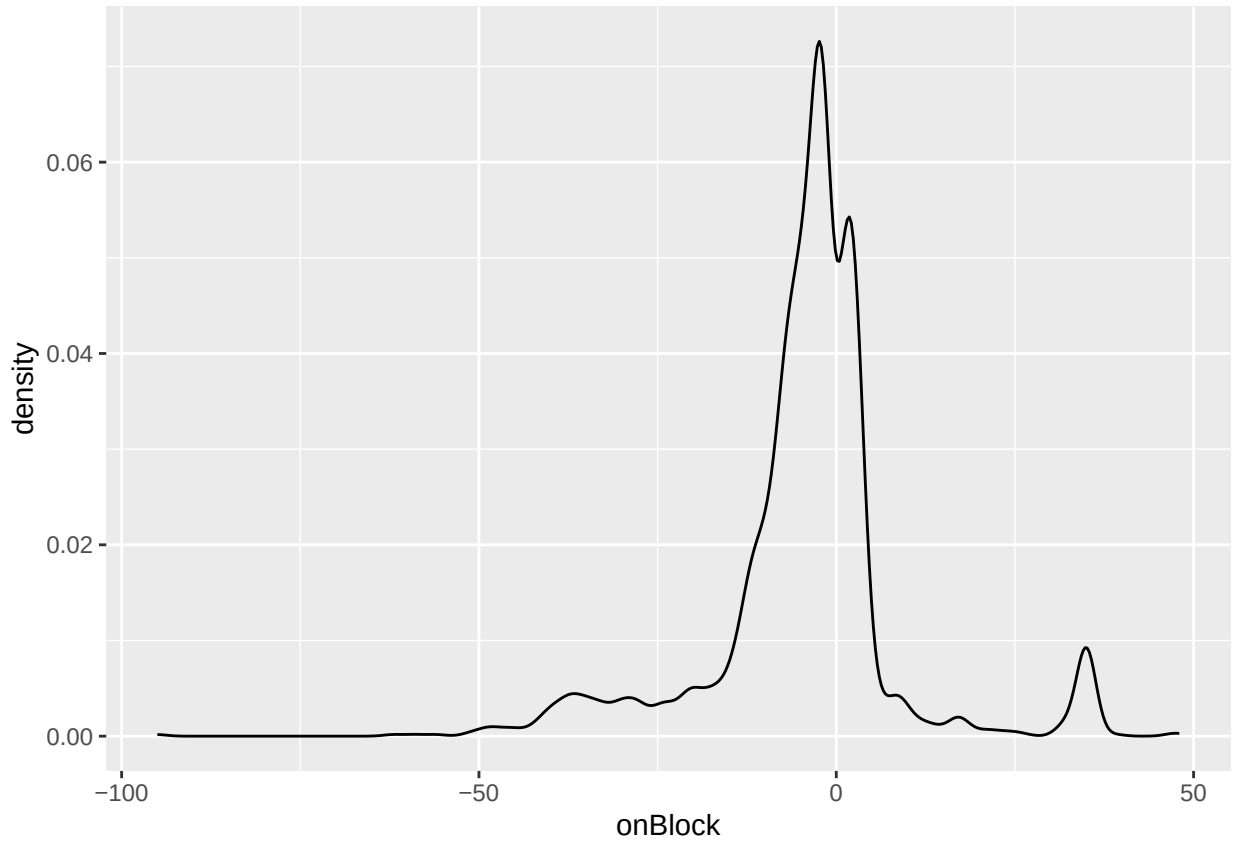
```
## # A tibble: 2 x 4
##   followUp count average_damage average_stun
##   <chr>    <int>          <dbl>        <dbl>
## 1 True       258           96.6         135.
## 2 False     1551           86.8         120.
```

```
summary_tb(projectile)
```

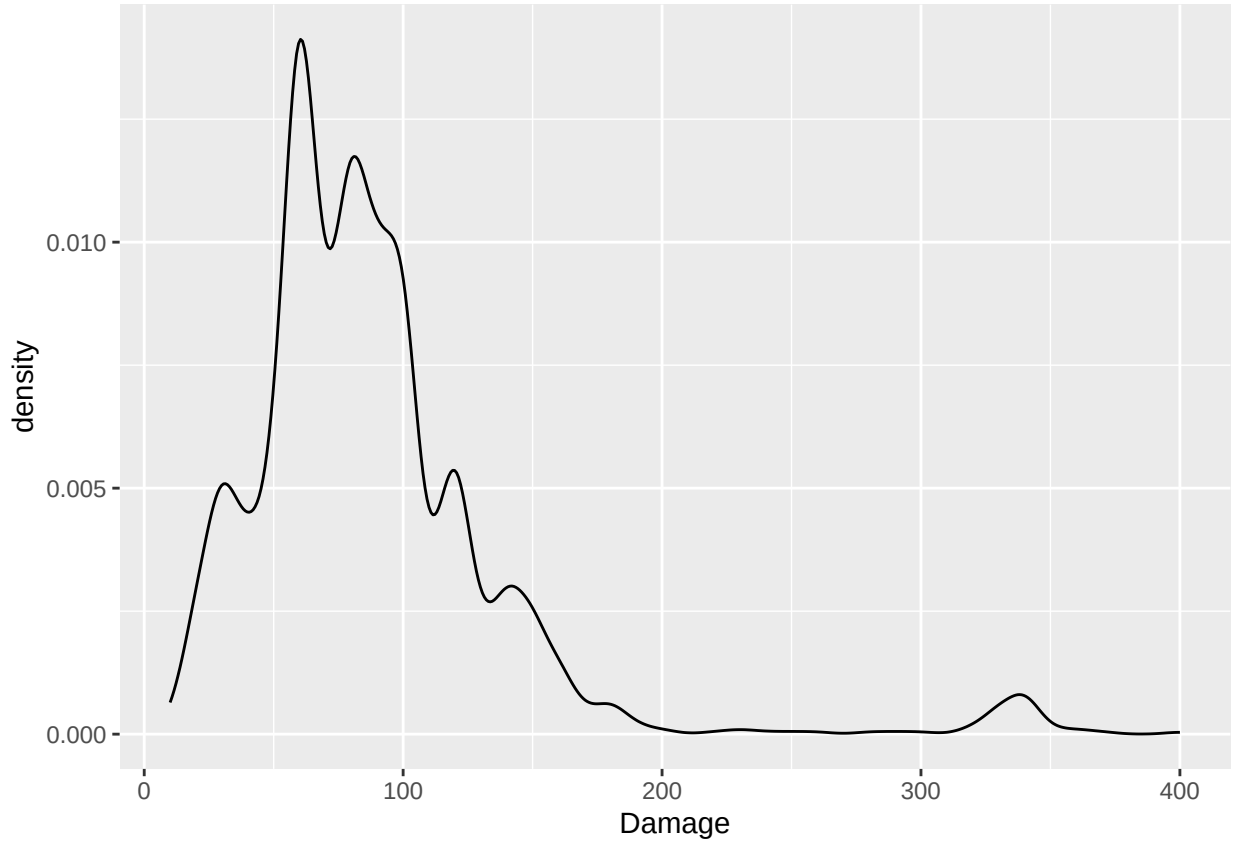
```
## # A tibble: 2 x 4
##   projectile count average_damage average_stun
##   <chr>      <int>          <dbl>        <dbl>
## 1 False       1549           88.9         125.
## 2 True         260           83.9         105.
```

Kernel density estimates (KDEs)

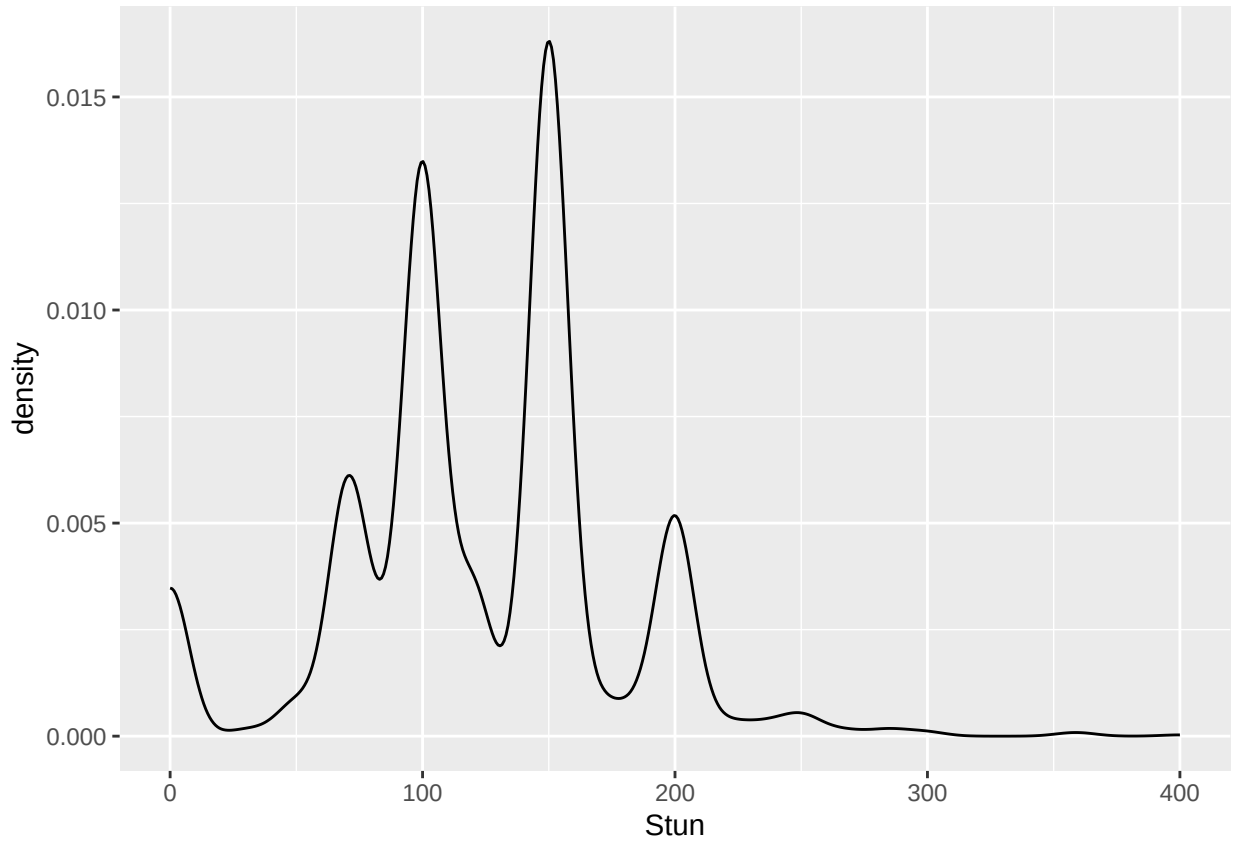
```
# user will be able to adjust bandwidth with slider
ggplot(df, aes(onBlock)) +
  geom_density(adjust = 1)
```




```
ggplot(df, aes(Damage)) +
  geom_density(adjust = 1)
```




```
ggplot(df, aes(Stun)) +
  geom_density(adjust = 1)
```



Individual character