VISUALIZATION PROJECT UNIT 1

SHREYAN CHAKRABORTY

2022-10-29

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     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()
```

```
chess_dataset<-read.csv('C:/Users/91974/Desktop/cmi study/Chess games stats.csv')
head(chess_dataset)
```

```
##   X  Game.ID White.Rating Black.Rating Opening.ECO Opening.Ply
## 1 0 J7Xvjkte         1441         1559         C20           4
## 2 1 SSzpi7W1         1258         1567         C46           6
## 3 2 NHpcYm3r         1697         1712         C68           7
## 4 3 Nrzmgzmn         1978         1868         D31           5
## 5 4 xMuFsnC6         2073         1816         A01           2
## 6 5 gcdFqoqK         1944         2052         C44           9
##   White.Centi.pawn.Loss White.s.Number.of.Inaccuracies
## 1                    32                              2
## 2                    43                              0
## 3                    11                              1
## 4                    21                              3
## 5                    35                              3
## 6                    79                              3
##   White.s.Number.of.Mistakes White.s.Number.of.Blunders Black.Centi.pawn.Loss
## 1                          2                           0                    87
## 2                          0                           1                     6
## 3                          0                           0                    42
## 4                          0                           0                    53
## 5                          2                           0                    50
## 6                          0                           2                    62
##   Black.s.Number.of.Inaccuracies Black.s.Number.of.Mistakes
## 1                              2                          4
## 2                              0                          0
## 3                              1                          0
## 4                              6                          0
## 5                              6                          3
## 6                              3                          1
##   Black.s.Number.of.Blunders
```

```
## 1                          1
## 2                          0
## 3                          2
## 4                          1
## 5                          0
## 6                          1
```

```
summary(chess_dataset)
```

```
##        X           Game.ID           White.Rating   Black.Rating
##  Min.   :    0   Length:18637       Min.   : 784   Min.   : 780
##  1st Qu.: 4659   Class :character   1st Qu.:1551   1st Qu.:1552
##  Median : 9318   Mode  :character   Median :1782   Median :1781
##  Mean   : 9318                      Mean   :1775   Mean   :1777
##  3rd Qu.:13977                      3rd Qu.:1991   3rd Qu.:1996
##  Max.   :18636                      Max.   :2997   Max.   :2995
##  Opening.ECO         Opening.Ply     White.Centi.pawn.Loss
##  Length:18637       Min.   : 1.000   Min.   :  0.00
##  Class :character   1st Qu.: 3.000   1st Qu.: 33.00
##  Mode  :character   Median : 5.000   Median : 53.00
##                     Mean   : 5.319   Mean   : 58.81
##                     3rd Qu.: 7.000   3rd Qu.: 77.00
##                     Max.   :28.000   Max.   :517.00
##  White.s.Number.of.Inaccuracies White.s.Number.of.Mistakes
##  Min.   : 0.000                 Min.   : 0.000
##  1st Qu.: 1.000                 1st Qu.: 0.000
##  Median : 3.000                 Median : 1.000
##  Mean   : 3.259                 Mean   : 1.326
##  3rd Qu.: 5.000                 3rd Qu.: 2.000
##  Max.   :19.000                 Max.   :12.000
##  White.s.Number.of.Blunders Black.Centi.pawn.Loss
##  Min.   : 0.000             Min.   :  0.00
##  1st Qu.: 1.000             1st Qu.: 33.00
##  Median : 2.000             Median : 55.00
##  Mean   : 2.116             Mean   : 60.78
##  3rd Qu.: 3.000             3rd Qu.: 80.00
##  Max.   :18.000             Max.   :550.00
##  Black.s.Number.of.Inaccuracies Black.s.Number.of.Mistakes
##  Min.   : 0.000                 Min.   : 0.000
##  1st Qu.: 1.000                 1st Qu.: 0.000
##  Median : 3.000                 Median : 1.000
##  Mean   : 3.283                 Mean   : 1.332
##  3rd Qu.: 5.000                 3rd Qu.: 2.000
##  Max.   :22.000                 Max.   :12.000
##  Black.s.Number.of.Blunders
##  Min.   : 0.000
##  1st Qu.: 1.000
##  Median : 2.000
##  Mean   : 2.118
##  3rd Qu.: 3.000
##  Max.   :22.000
```

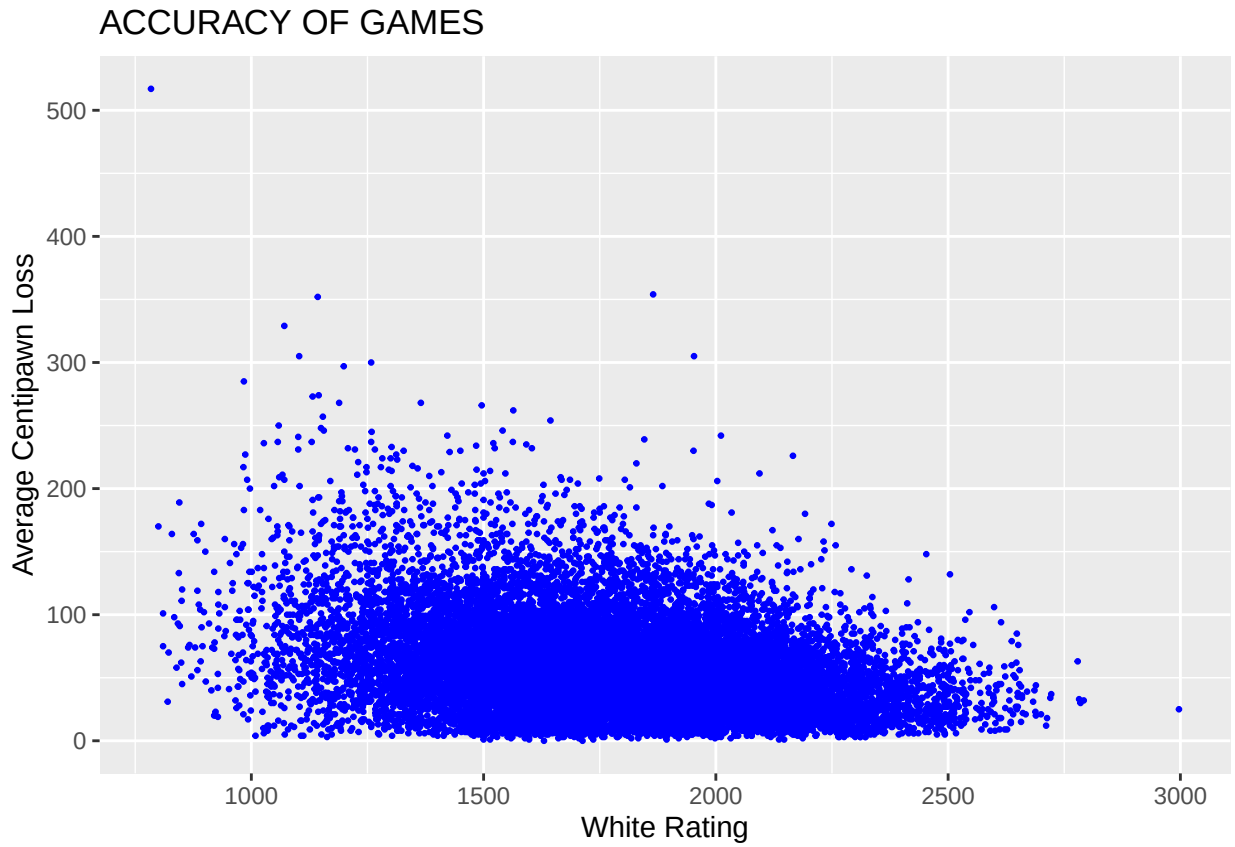
```
attach(chess_dataset)
Op=array(chess_dataset["Opening.Ply"])
counter<-c(rep(0,28))
```

```
for(i in 1:18637){for(j in 1:28){
    if(Opening.Ply[i]==j){counter[j]<-counter[j]+1}
}}
Mop<-max(Opening.Ply)
CO<-as.data.frame(counter)
N_PLY<-c(seq(1,28,1))
CO["OPENING_PLY"]<-N_PLY
```

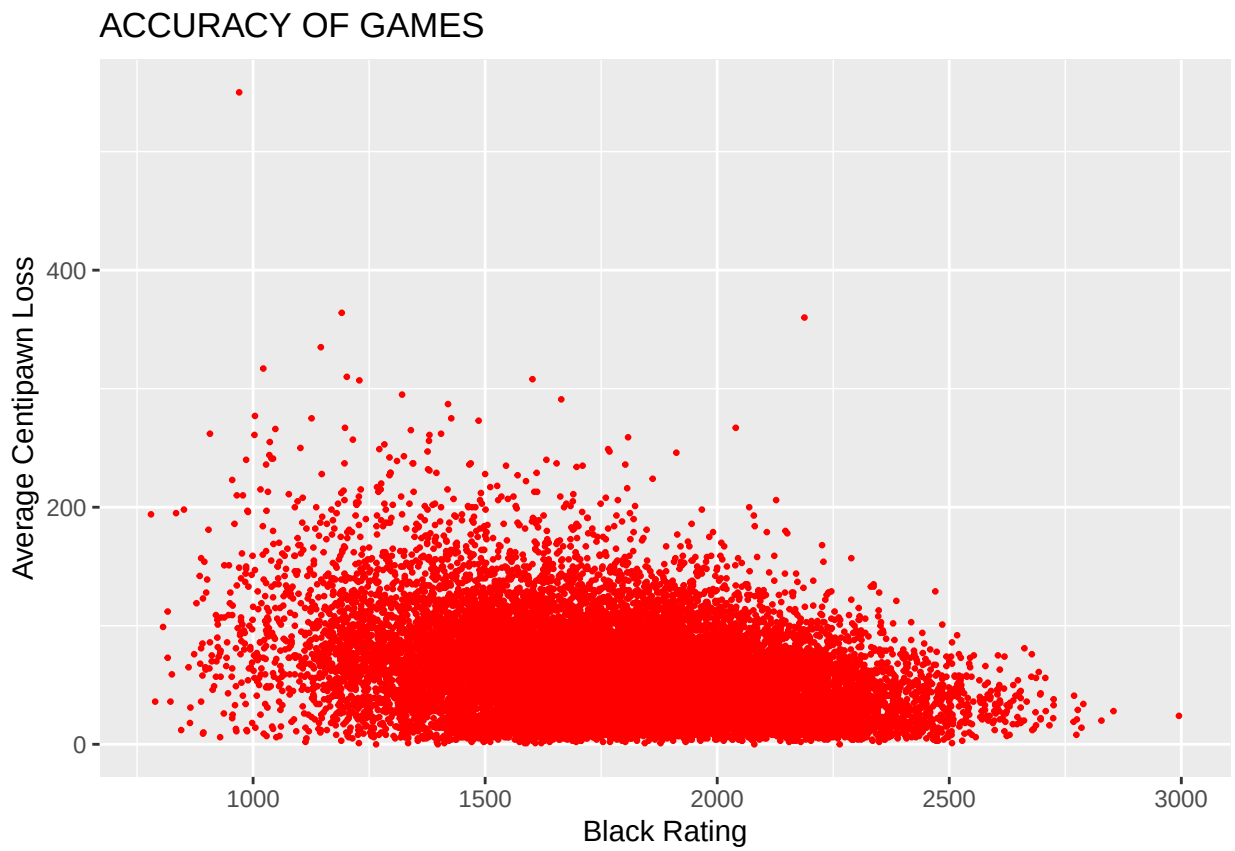
Plots

Some of the useful plots are as follows:

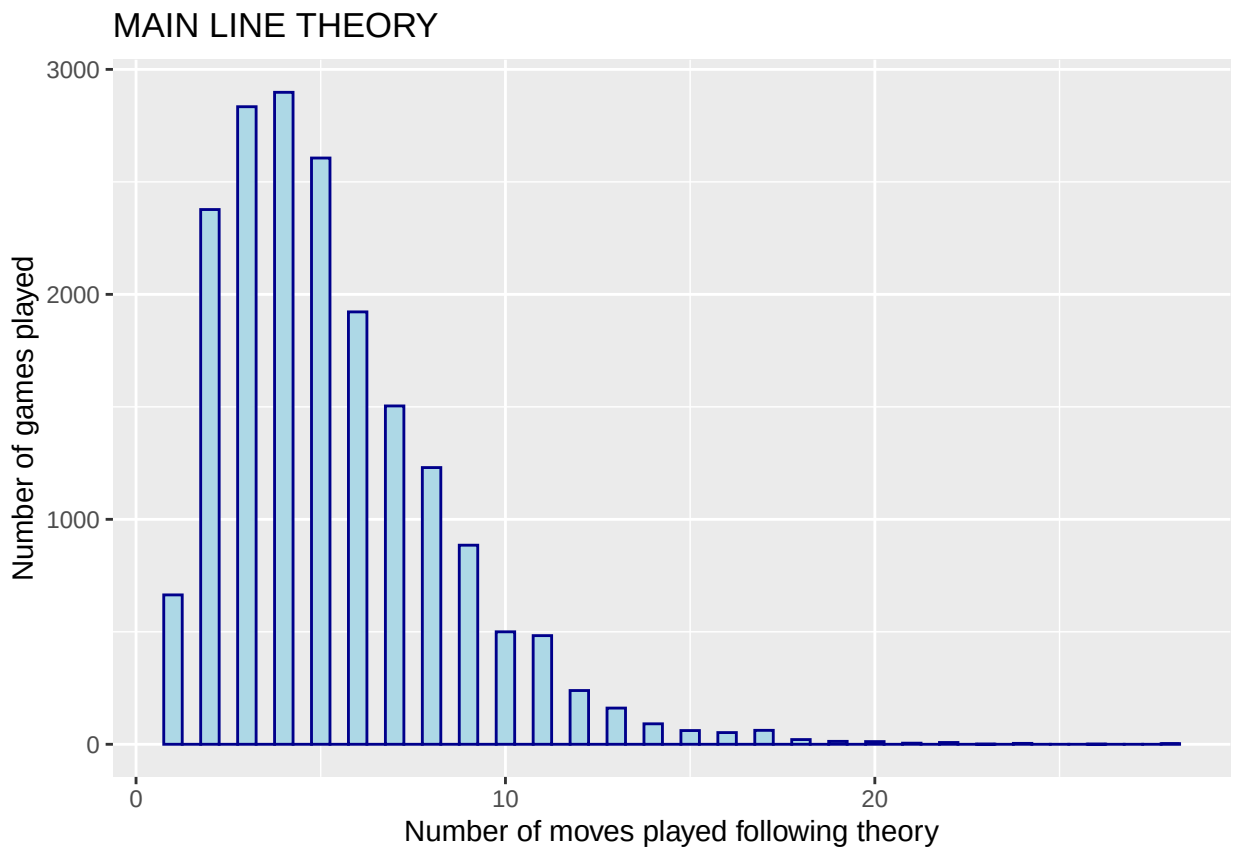
```
ggplot(chess_dataset, aes(x=(White.Rating), y=(White.Centi.pawn.Loss))) +
  geom_point(col="blue",size=0.5)+
     labs(title="ACCURACY OF GAMES")+
     xlab("White Rating")+
    ylab("Average Centipawn Loss")
```



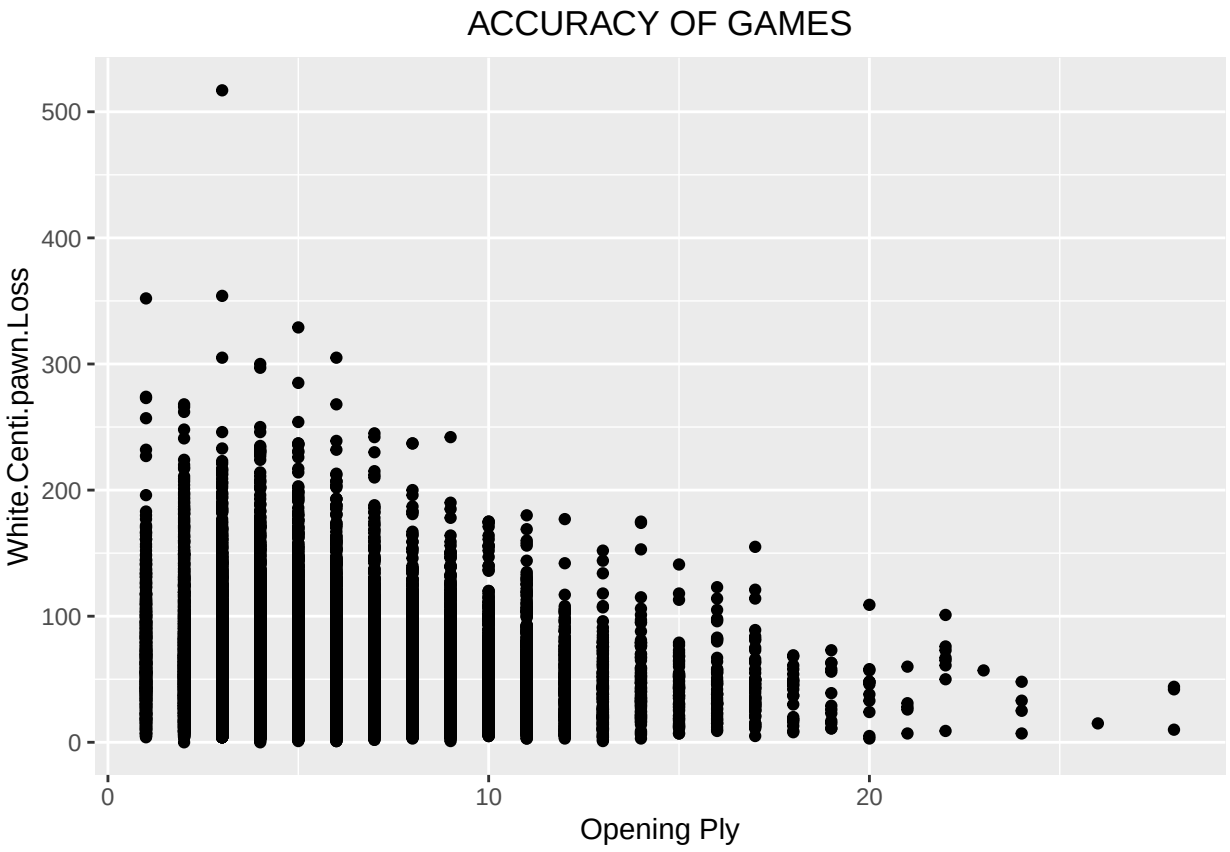

```
ggplot(chess_dataset, aes(x=(Black.Rating), y=(Black.Centi.pawn.Loss))) +
  geom_point(col="red",size=0.5)+
     labs(title="ACCURACY OF GAMES")+
     xlab("Black Rating")+
    ylab("Average Centipawn Loss")
```




```
ggplot(chess_dataset,aes(x=Opening.Ply)) +
  geom_histogram(bins=28,binwidth=0.5,color="darkblue",fill="lightblue")+
     labs(title="MAIN LINE THEORY")+
     xlab("Number of moves played following theory")+
    ylab("Number of games played")
```



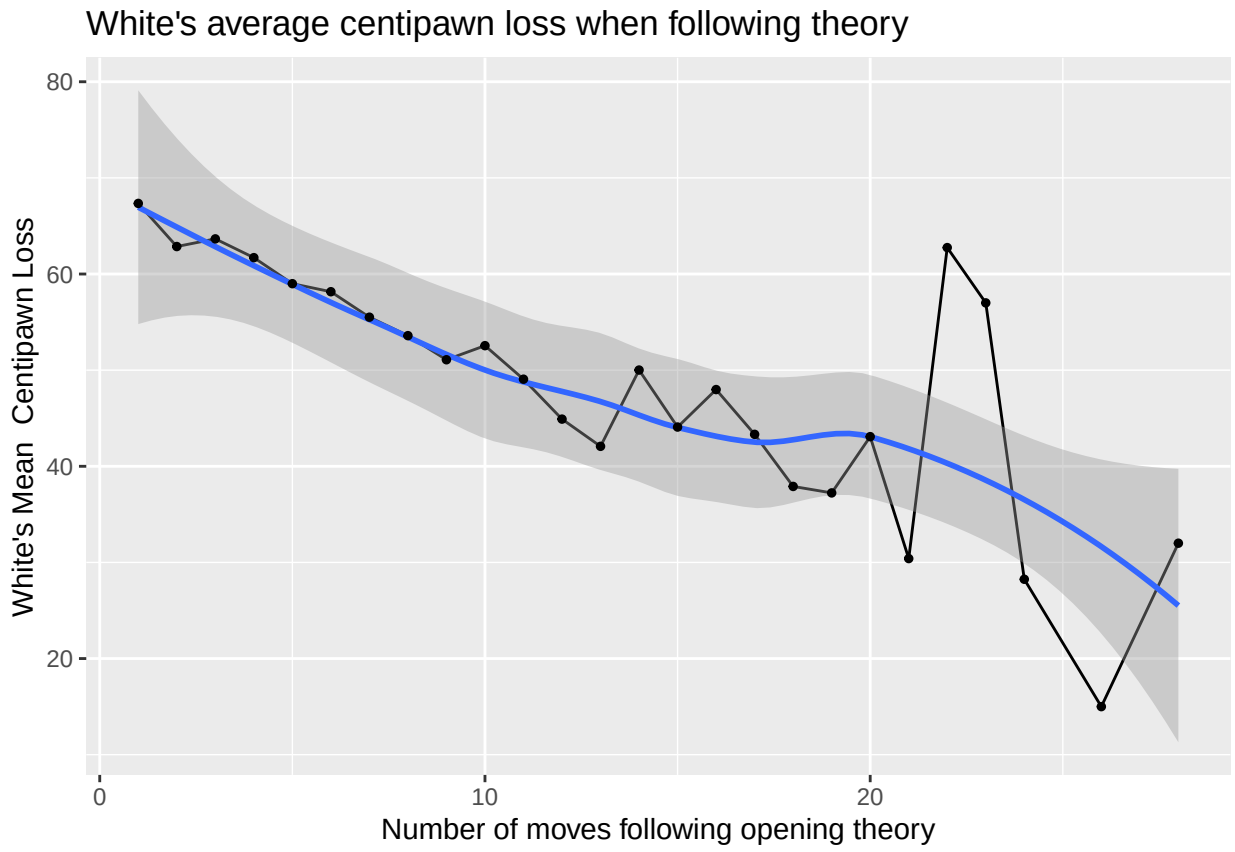

```
ggplot(chess_dataset, aes(x=Opening.Ply, y=White.Centi.pawn.Loss))+
    geom_point()+
    labs(title="ACCURACY OF GAMES",x="Opening Ply")+
    theme(plot.title = element_text(hjust = 0.5))
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
chess_dataset %>%
    select(Opening.Ply, White.Centi.pawn.Loss) %>%
    group_by(Opening.Ply) %>%
    summarise(mean_wcpl = mean(White.Centi.pawn.Loss)) %>%
    ggplot(aes(Opening.Ply, mean_wcpl)) +
    geom_line()+
    geom_smooth() +
    geom_point(size = 1) +
    labs(title = "White's average centipawn loss when following theory",
         x = "Number of moves following opening theory",
         y = "White's Mean  Centipawn Loss")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
chess_dataset %>%
    select(Opening.Ply, White.Centi.pawn.Loss) %>%
    group_by(Opening.Ply) %>%
    summarise(mean_wcpl = mean(White.Centi.pawn.Loss),
              sd_wcpl = sd(White.Centi.pawn.Loss)) %>%
    filter(!is.na(sd_wcpl))
```

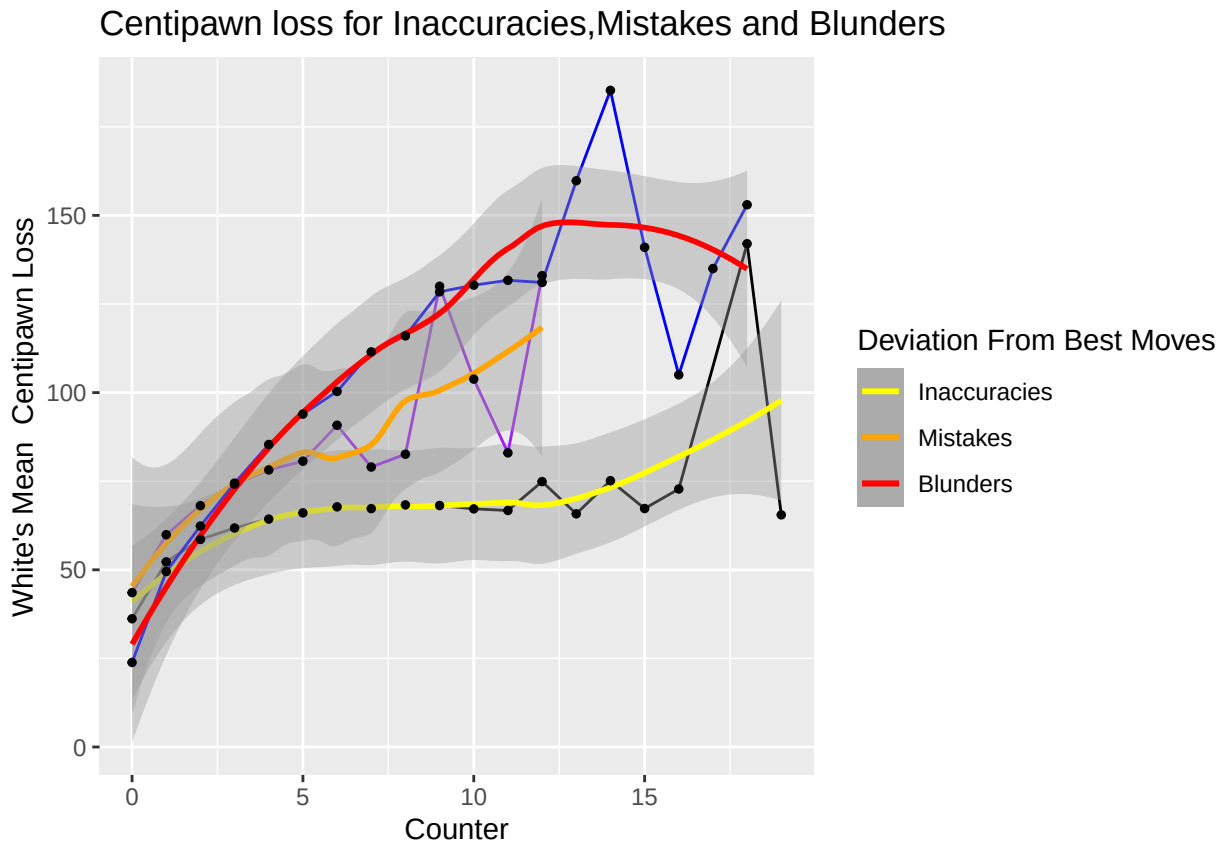
```
## # A tibble: 24 x 3
##    Opening.Ply mean_wcpl sd_wcpl
##          <int>     <dbl>   <dbl>
##  1           1      67.3    41.6
##  2           2      62.9    37.0
##  3           3      63.7    38.0
##  4           4      61.7    37.9
##  5           5      59.0    37.6
##  6           6      58.2    36.7
##  7           7      55.5    34.5
##  8           8      53.6    32.4
##  9           9      51.1    32.2
## 10          10      52.5    32.5
## # ... with 14 more rows
## # i Use `print(n = ...)` to see more rows
```

```
df1 <- chess_dataset %>%
    select(White.s.Number.of.Inaccuracies, White.Centi.pawn.Loss)   %>%
    group_by(White.s.Number.of.Inaccuracies) %>%
    summarise(mean_wcpl1 = mean(White.Centi.pawn.Loss))
```

```
df2 <- chess_dataset %>%
    select(White.s.Number.of.Mistakes, White.Centi.pawn.Loss) %>%
    group_by(White.s.Number.of.Mistakes) %>%
    summarise(mean_wcpl2 = mean(White.Centi.pawn.Loss))
df3 <- chess_dataset %>%
    select(White.s.Number.of.Blunders, White.Centi.pawn.Loss) %>%
    group_by(White.s.Number.of.Blunders) %>%
    summarise(mean_wcpl3 = mean(White.Centi.pawn.Loss))
 ggplot(data=df1,aes(White.s.Number.of.Inaccuracies,mean_wcpl1)) +
    geom_line()+
    geom_smooth(aes(color="Yellow")) +
     scale_color_identity(guide = "legend")+
     geom_line(data=df2,aes(White.s.Number.of.Mistakes,mean_wcpl2),col="purple")+
     geom_smooth(data=df2,aes(White.s.Number.of.Mistakes,mean_wcpl2,color="Orange"))+
     scale_color_identity(guide = "legend")+
     geom_line(data=df3,aes(White.s.Number.of.Blunders,mean_wcpl3),col="blue")+
     geom_smooth(data=df3,aes(White.s.Number.of.Blunders,mean_wcpl3,color="Red"))+
     scale_color_identity(breaks = c("Yellow", "Orange", "Red"),
                          labels = c("Inaccuracies", "Mistakes", "Blunders"),guide = "legend")+
    geom_point(size = 1) +
    geom_point(data=df2,aes(White.s.Number.of.Mistakes,mean_wcpl2),size=1)+
         geom_point(data=df3,aes(White.s.Number.of.Blunders,mean_wcpl3),size=1)+
     labs(title = "Centipawn loss for Inaccuracies,Mistakes and Blunders",
         x = "Counter",
         y = "White's Mean  Centipawn Loss",
         color="Deviation From Best Moves")
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggsave(filename = "graph1.png",
       path = "C:/Users/91974/Desktop/cmi study/Graphs/",
       dpi = 300,
       device = "png")
```

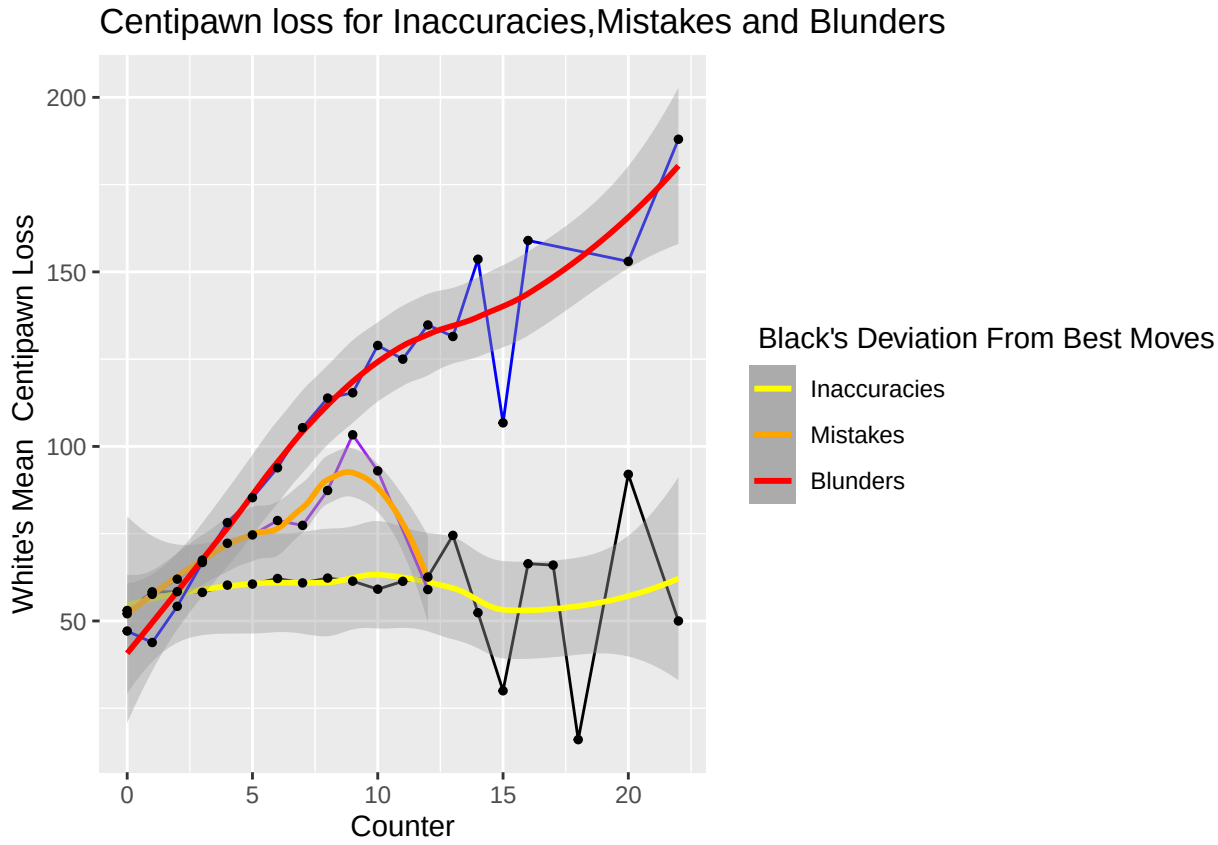
```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
dF1 <- chess_dataset %>%
    select(Black.s.Number.of.Inaccuracies, White.Centi.pawn.Loss)   %>%
    group_by(Black.s.Number.of.Inaccuracies) %>%
    summarise(mean_wBcpl1 = mean(White.Centi.pawn.Loss))
dF2 <- chess_dataset %>%
    select(Black.s.Number.of.Mistakes, White.Centi.pawn.Loss)   %>%
    group_by(Black.s.Number.of.Mistakes) %>%
    summarise(mean_wBcpl2 = mean(White.Centi.pawn.Loss))
dF3 <- chess_dataset %>%
    select(Black.s.Number.of.Blunders, White.Centi.pawn.Loss)   %>%
    group_by(Black.s.Number.of.Blunders) %>%
    summarise(mean_wBcpl3 = mean(White.Centi.pawn.Loss))
ggplot(data=dF1,aes(Black.s.Number.of.Inaccuracies,mean_wBcpl1)) +
    geom_line()+
    geom_smooth(aes(color="Yellow")) +
     scale_color_identity(guide = "legend")+
     geom_line(data=dF2,aes(Black.s.Number.of.Mistakes,mean_wBcpl2),col="purple")+
```

```
     geom_smooth(data=dF2,aes(Black.s.Number.of.Mistakes,mean_wBcpl2,color="Orange"))+
     scale_color_identity(guide = "legend")+
     geom_line(data=dF3,aes(Black.s.Number.of.Blunders,mean_wBcpl3),col="blue")+
     geom_smooth(data=dF3,aes(Black.s.Number.of.Blunders,mean_wBcpl3,color="Red"))+
     scale_color_identity(breaks = c("Yellow", "Orange", "Red"),
                          labels = c("Inaccuracies", "Mistakes", "Blunders"),guide = "legend")+
    geom_point(size = 1) +
    geom_point(data=dF2,aes(Black.s.Number.of.Mistakes,mean_wBcpl2),size=1)+
         geom_point(data=dF3,aes(Black.s.Number.of.Blunders,mean_wBcpl3),size=1)+
     labs(title = "Centipawn loss for Inaccuracies,Mistakes and Blunders",
         x = "Counter",
         y = "White's Mean  Centipawn Loss",
         color=" Black's Deviation From Best Moves")
```

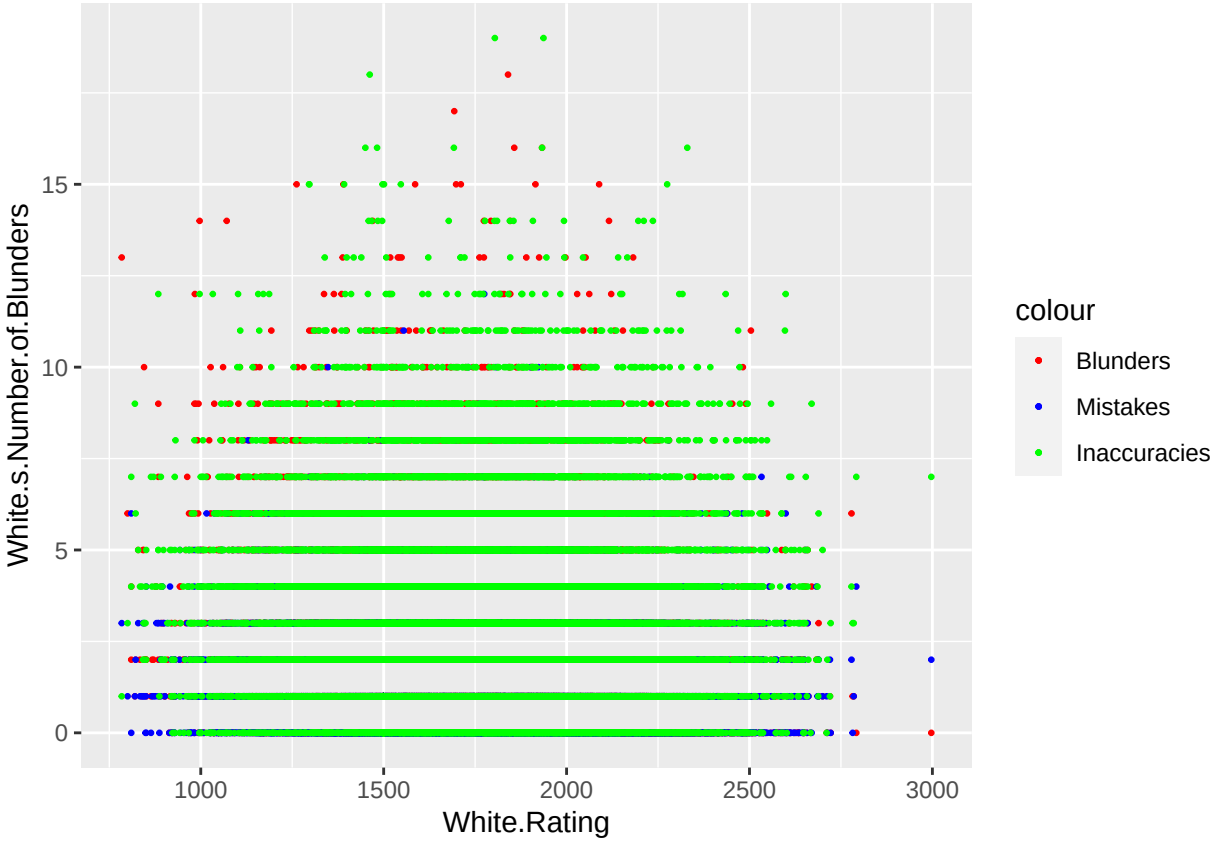
```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
dt1<-chess_dataset %>%
    select(White.Rating,White.s.Number.of.Blunders)
dt2<-chess_dataset %>%
    select(White.Rating,White.s.Number.of.Mistakes)
```

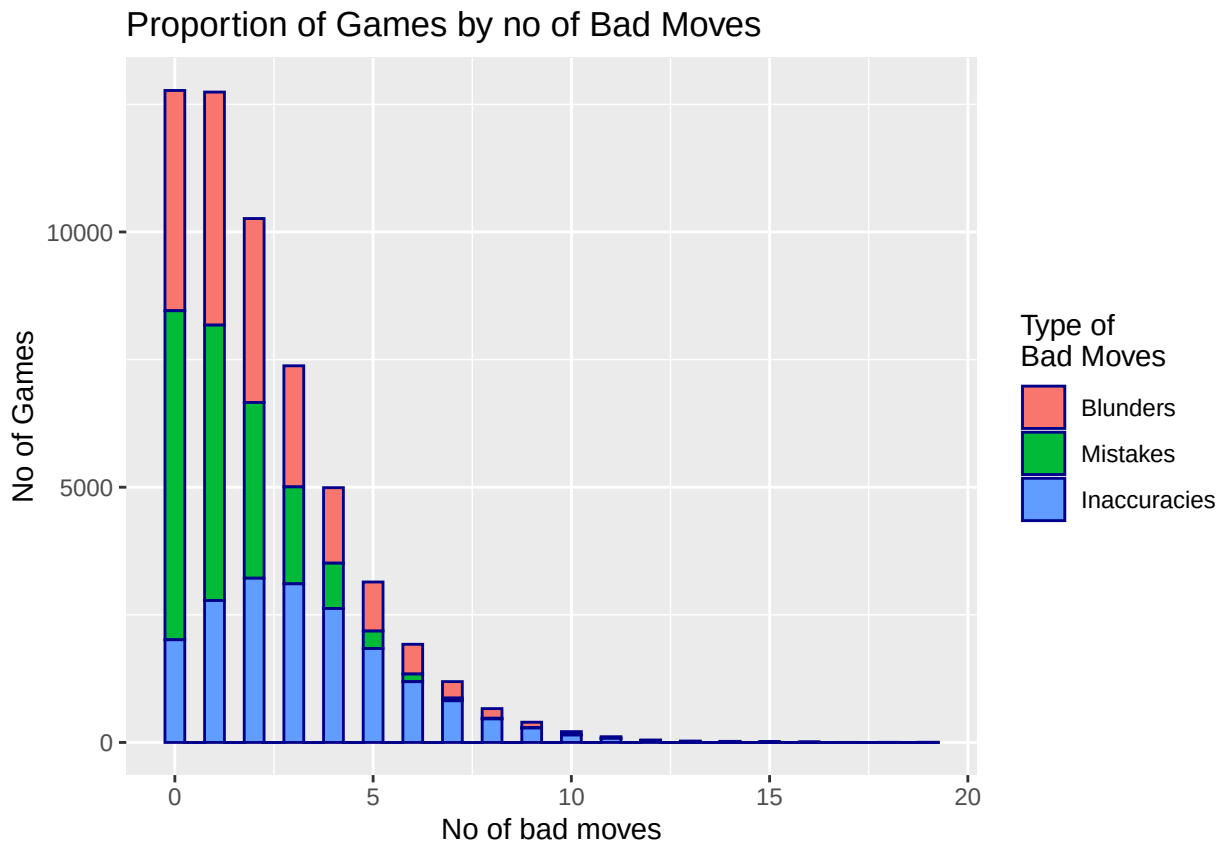
```
dt3<-chess_dataset %>%
    select(White.Rating,White.s.Number.of.Inaccuracies)
ggplot(data=dt1,aes(White.Rating,White.s.Number.of.Blunders))+
    geom_point(aes(color="Red"),size=0.5)+
    geom_point(data=dt2,aes(White.Rating,White.s.Number.of.Mistakes,color="Blue"),size=0.5)+
    geom_point(data=dt3,aes(White.Rating,White.s.Number.of.Inaccuracies,color="Green"),size=0.5)+
    scale_color_identity(breaks = c("Red", "Blue", "Green"),
                         labels = c("Blunders", "Mistakes", "Inaccuracies"),guide = "legend")
```




```
x1<-chess_dataset %>%
    select(White.s.Number.of.Blunders)
x1["Counter"]<-(c(rep(1,18637)))
x3<-chess_dataset %>%
    select(White.s.Number.of.Mistakes)
x3["Counter"]<-(c(rep(2,18637)))
x5<-chess_dataset %>%
    select(White.s.Number.of.Inaccuracies)
x5["Counter"]<-(c(rep(3,18637)))
colnames(x1)<-c("Value","Counter")
colnames(x3)<-c("Value","Counter")
colnames(x5)<-c("Value","Counter")
x2<-rbind(x1,x3)
x4<-rbind(x2,x5)

x4 %>%
    ggplot(aes(Value,group=Counter,fill=as.factor(Counter)))+
```

```
    geom_histogram(bins=28,binwidth=0.5,color="darkblue")+
    guides(color = guide_legend(title = "Deviations from Best Move"))+
    scale_fill_discrete(labels=c("Blunders", "Mistakes","Inaccuracies")) +
    labs(y="No of Games",x="No of bad moves",fill="Type of\nBad Moves",title="Proportion of Games by no
```




```
    ## scale_x_discrete(limits=c("Blunders", "Mistakes", "Inaccuracies"))
##x4 %>%
  ##  ggplot(aes(x="",y=Value,fill=as.factor(Counter)))
    ##geom_bar(stat="identity", width=1) +
  #coord_polar("y", start=0)
```

```
y1<-chess_dataset %>%
    filter(str_detect(Opening.ECO, "C0") | str_detect(Opening.ECO, "C1")) %>%
    group_by(Opening.ECO) %>%
    summarise(avg_wcpl = mean(White.Centi.pawn.Loss))
ctm<-array(0,6)
ctm[1]<-mean(y1$avg_wcpl)
y2<-chess_dataset %>%
    filter(str_detect(Opening.ECO, "C3")) %>%
    group_by(Opening.ECO) %>%
    summarise(avg_wcpl1 = mean(White.Centi.pawn.Loss))
ctm[2]<-mean(y2$avg_wcpl1)

y3<-chess_dataset %>%
    filter(str_detect(Opening.ECO, "C5")) %>%
    group_by(Opening.ECO) %>%
```

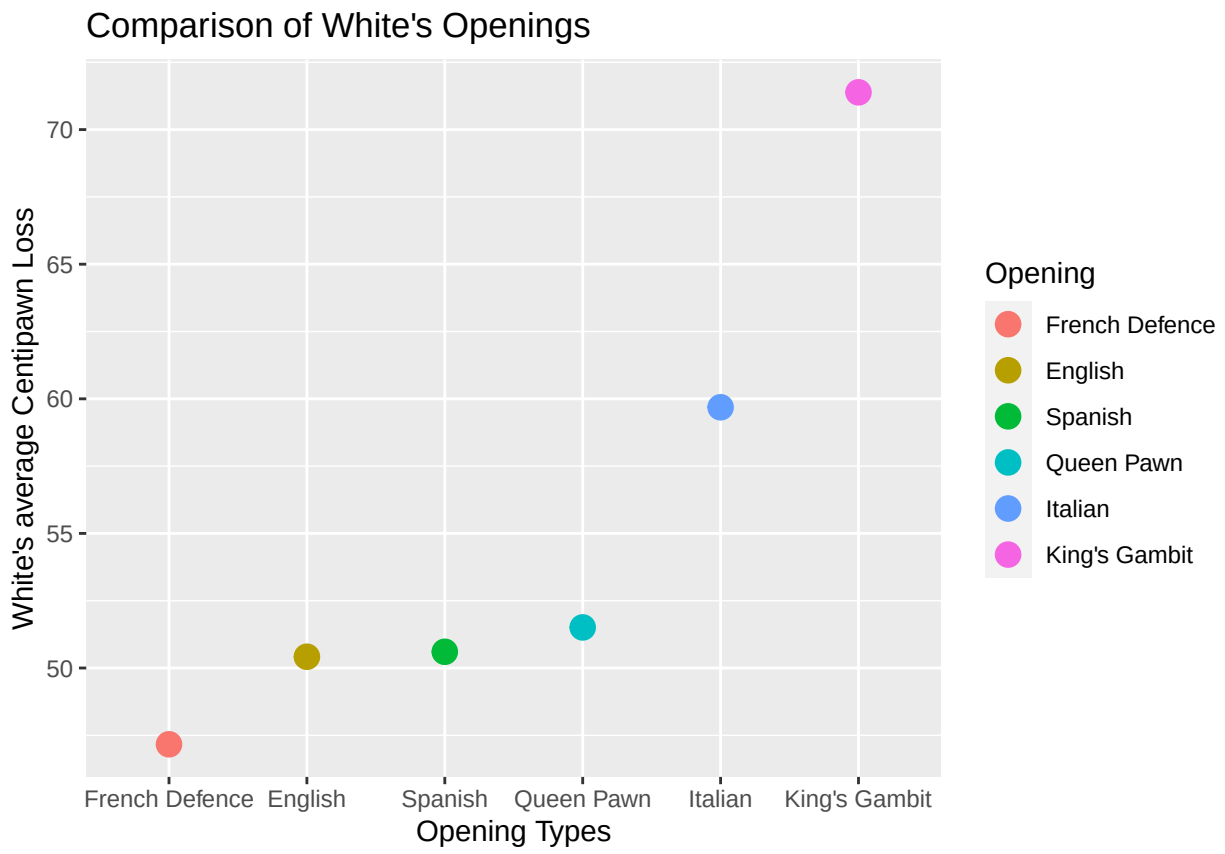
```
    summarise(avg_wcpl2 = mean(White.Centi.pawn.Loss))
ctm[3]<-mean(y3$avg_wcpl2)

y4<-chess_dataset %>%
    filter(str_detect(Opening.ECO, "C6")) %>%
    group_by(Opening.ECO) %>%
    summarise(avg_wcpl3 = mean(White.Centi.pawn.Loss))
ctm[4]<-mean(y4$avg_wcpl3)

y5<-chess_dataset %>%
    filter(str_detect(Opening.ECO, "A1") | str_detect(Opening.ECO, "A2")|str_detect(Opening.ECO, "A3"))
    group_by(Opening.ECO) %>%
    summarise(avg_wcpl4 = mean(White.Centi.pawn.Loss))
ctm[5]<-mean(y5$avg_wcpl4)

y6<-chess_dataset %>%
    filter(str_detect(Opening.ECO, "A4") | str_detect(Opening.ECO, "A5")|str_detect(Opening.ECO, "A6")|s
    group_by(Opening.ECO) %>%
    summarise(avg_wcpl5 = mean(White.Centi.pawn.Loss))
ctm[6]<-mean(y6$avg_wcpl5)
BOX<-data.frame(ctm)
BOX["Opening"]<-c("French Defence","King's Gambit","Italian","Spanish","English","Queen Pawn")
colnames(BOX)<-c("White's average Centipawn Loss","Opening")

BOX %>%
    arrange(`White's average Centipawn Loss`) %>%
    mutate(Opening = fct_reorder(Opening, `White's average Centipawn Loss`)) %>%
    ggplot(aes(Opening,`White's average Centipawn Loss`,color=Opening))+
    geom_point(size=4)+
    labs(x = "Opening Types",title="Comparison of White's Openings")
```




```
x = c("C123", "D234")
str_detect(x, "French Defence")
```

```
## [1] FALSE FALSE
```