# Heuristics for 2048 : Data Visulisation

August 2024

## Introduction

This document shows some visulisations of the results of 17 differently weighted bots playing the game "2048". The version of 2048 used to collect this data was originally written by me in Haskell. All the bots work in the same way, namely by picking their next move at random, however we apply weighting to each direction resulting in different bots favouring different directions. Each direction has one of:

- no particular preference (weight = 1)
- a weak preference (weight = 5)
- a strong preference (weight = 25)

Each combination of preferences was repeated 50 times with the score on every turn recorded until the game was finished. In order to do this, I used Python to repeated call the Haskell executable (which runs in terminal).

## Raw Data Summary

Below is the raw data collected after all 17 of the bots had finished their 50 attempts at playing the game. The 7 columns contain the following information:

- Bot id : First character indicates which weight combination is being used, the number indicates it is the xth attempt with this weight combination
- Turn no. : The turn the bot is on in that attempt
- Current Score : The total score on each turn
- Up/Left/Down/Right weight : How much of a preference is given to each direction (numbers explained above)

```
summary(bot_data) %>%
  knitr :: kable("simple",format='latex',booktabs= TRUE) %>%
    kable_styling(latex_options = c("HOLD_position","scale_down") )
```

| Bot id | Turn no. | Current score | up weight | left weight | down weight | right weight |
|---|---|---|---|---|---|---|
| Length:129474 | Min. : 0.00 | Min. : 0.0 | Min. : 1.000 | Min. : 1.000 | Min. : 1.000 | Min. : 1.000 |
| Class :character | 1st Qu.: 38.00 | 1st Qu.: 208.0 | 1st Qu.: 1.000 | 1st Qu.: 1.000 | 1st Qu.: 1.000 | 1st Qu.: 1.000 |
| Mode :character | Median : 76.00 | Median : 556.0 | Median : 1.000 | Median : 1.000 | Median : 1.000 | Median : 1.000 |
| NA | Mean : 84.95 | Mean : 739.4 | Mean : 6.144 | Mean : 6.309 | Mean : 6.331 | Mean : 6.367 |
| NA | 3rd Qu.:121.00 | 3rd Qu.:1092.0 | 3rd Qu.: 5.000 | 3rd Qu.: 5.000 | 3rd Qu.: 5.000 | 3rd Qu.: 5.000 |
| NA | Max. :390.00 | Max. :5176.0 | Max. :25.000 | Max. :25.000 | Max. :25.000 | Max. :25.000 |

## Expanding the table

We can now add a few columns to add some extra information for when we start making charts:

- lastTurn : Bool which says whether that turn is the bots last turn in a game
- weightClass: column containing the character from Bot id (condenses info from up/down/left/right weight)

- prefStrength: Strength of preference (Strong = at least one weight is 25, Weak = largest weighting is 5, Control = all weights are 1)
- Direction: If only one direction has a weighting larger than 1, that direction will appear in this column. If 2 directions have a weighting larger than 1, it's classed as M(ixed). If all directions are 1, it is classed as C(ontrol). The strength of the preference is virtually ignored here.
- comboOrSingle - generalisation of the Direction column, single means only one direction has a preference, combo means multiple directions have a preference

The pairings of the mixed combinations of directions are Up/Left, and Down/Right

```
allCols <- bot_data %>%
      mutate(lastTurn = case_when(lead(`Bot id`) == `Bot id` ~ FALSE,
                            TRUE ~ TRUE)) %>%
      mutate(weightClass = substr(`Bot id`,1,1)) %>%
      mutate( prefStrength = case_when(`up weight`== 25 | `down weight` == 25
                                  | `left weight` == 25 | `right weight` == 25 ~ "Strong",
                                    `up weight`== 5 | `down weight` == 5
                                  | `left weight` == 5 | `right weight` == 5 ~ "Weak",
                                    TRUE ~ "Control")) %>%
      mutate(Direction = case_when(`up weight` > `down weight` & `left weight` == 1 ~ "U",
                              `down weight` > `up weight` & `right weight` == 1 ~ "D",
                              `left weight` > `right weight` & `up weight` == 1 ~ "L",
                              `right weight` > `left weight` & `down weight` == 1 ~ "R",
                              `up weight` != `down weight` ~ "M",
                               TRUE ~ "C")) %>%
      mutate(comboOrSingle = case_when(Direction == 'M' ~ "Combo",
                                      Direction == 'C' ~ "Control",
                                      TRUE ~ "Single"))

knitr :: kable(select(head(allCols,5),c(1:3,8:12)),format='latex',booktabs= TRUE,
               caption="Start of the overall table") %>%
     kable_styling(latex_options = c("HOLD_position","scale_down")) %>%
  add_footnote("Up/Down/Left/Right weight ommited from figure",notation="none")
```

Table 1: Start of the overall table

| Bot id | Turn no. | Current score | lastTurn | weightClass | prefStrength | Direction | comboOrSingle |
|---|---|---|---|---|---|---|---|
| a0 | 0 | 0 | FALSE | a | Control | C | Control |
| a0 | 1 | 0 | FALSE | a | Control | C | Control |
| a0 | 2 | 0 | FALSE | a | Control | C | Control |
| a0 | 3 | 4 | FALSE | a | Control | C | Control |
| a0 | 4 | 8 | FALSE | a | Control | C | Control |

Up/Down/Left/Right weight ommited from figure

We also create a table which only contains the final scores for each bots attempt as a lot of visulisations are primarily interested in this and not the journey to the score.

```
final_scores <- allCols %>%
    filter(lastTurn) %>%
     select(-lastTurn) %>%
      rename(finalScore = `Current score`, Turns = `Turn no.`)
#reorder based on direction column
final_scores$Direction <- factor(final_scores$Direction,levels=direction_order)
```

```r
knitr :: kable(select(head(final_scores,5),c(1:3,8:11)),'latex',booktabs= TRUE,
               caption="Additional table housing final scores",) %>%
   kable_styling(latex_options = c("HOLD_position","scale_down")) %>%
  add_footnote("Up/Down/Left/Right weight ommited from figure",notation="none")
```

Table 2: Additional table housing final scores

| Bot id | Turns | finalScore | weightClass | prefStrength | Direction | comboOrSingle |
|---|---|---|---|---|---|---|
| a0 | 97 | 672 | a | Control | C | Control |
| a1 | 216 | 2408 | a | Control | C | Control |
| a2 | 74 | 432 | a | Control | C | Control |
| a3 | 89 | 644 | a | Control | C | Control |
| a4 | 90 | 624 | a | Control | C | Control |

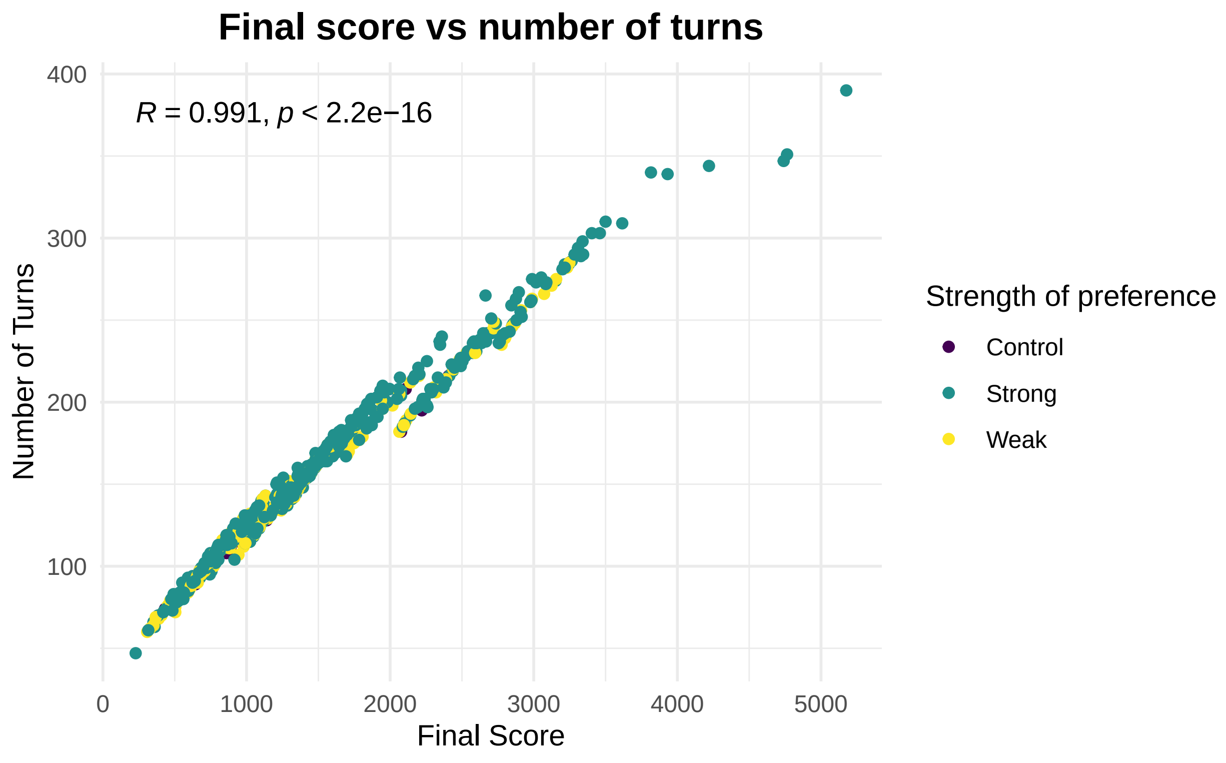
Up/Down/Left/Right weight ommited from figure

```r
# We define some functions to make the plotting of similar graphs easier

make_summary_table <- function(data,col) {
    summarize(data,Mean = mean(.data[[col]]),
            Median = median(.data[[col]]),
            LQ = quantile(.data[[col]],0.25),
            UQ = quantile(.data[[col]],0.75),
            Max = max(.data[[col]]),
            Min = min(.data[[col]]),
            SD  = sd(.data[[col]]))
}

make_boxplot <- function(data,xaxis,yaxis,colour,show_legend = TRUE) {
ggplot(data,aes_string(x=xaxis,y=yaxis)) +
  geom_boxplot(aes_string(color=colour),show.legend = show_legend,lwd=1,fatten=1) +
  theme_minimal() +
  scale_color_viridis(discrete = TRUE,option="D") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14))
}
```

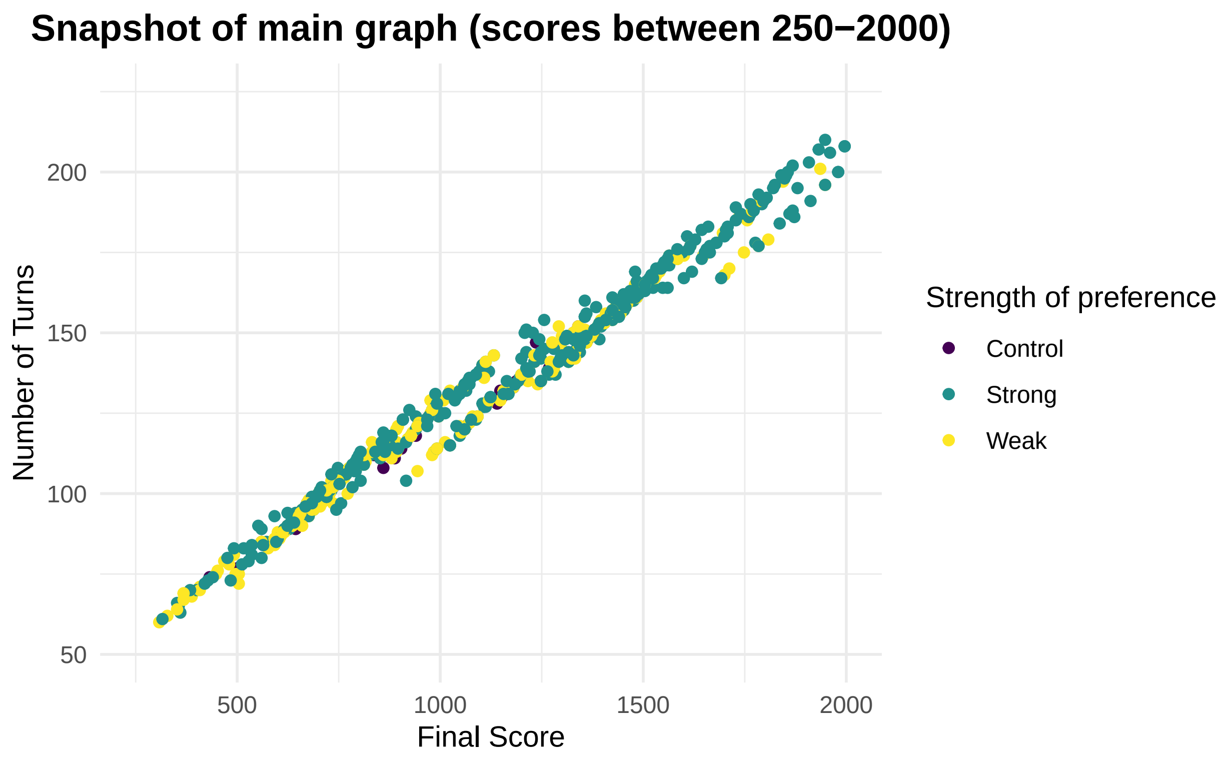
## General observations

Firstly, we just look a general pattern in the data between final score and the number of turns

```r
finalScoreVsTurns <- ggplot(final_scores,aes(x=finalScore,y=Turns)) +
  geom_point(aes(color=prefStrength)) + theme_minimal() +
  labs(x="Final Score",
       y= "Number of Turns",
       color="Strength of preference") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14)) +
  scale_colour_viridis_d()
finalScoreVsTurns + stat_cor(digits=3) +labs(title="Final score vs number of turns")
```



```
finalScoreVsTurns + xlim(250,2000) + ylim(50,225) +
  labs(title="Snapshot of main graph (scores between 250-2000)")
```



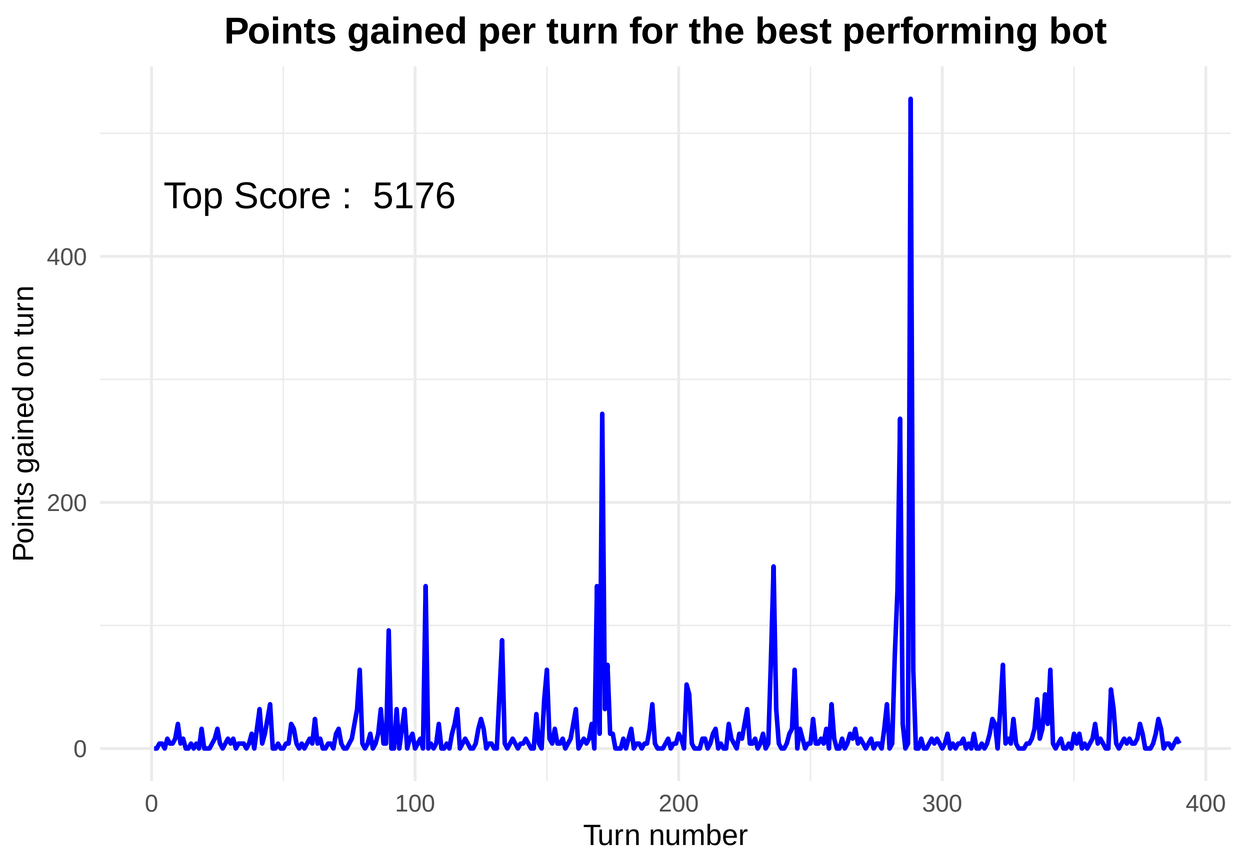
As we would expect, there is a very strong linear relationship between score and number of turns since the bots need to keep scoring in order to keep the board empty and keep playing. However, it is interesting to note that for some scores there can be very wide range of the number of turns it took to get there. For example for scores just past 2000 it can take the bots anywhere between ~180-215 turns to get there.

For some of the highest scores, it's particularly interesting to see that roughly the same number of turns (in this case ~350) can lead to a fairly wild range of final scores, in this case from ~3800-4750.

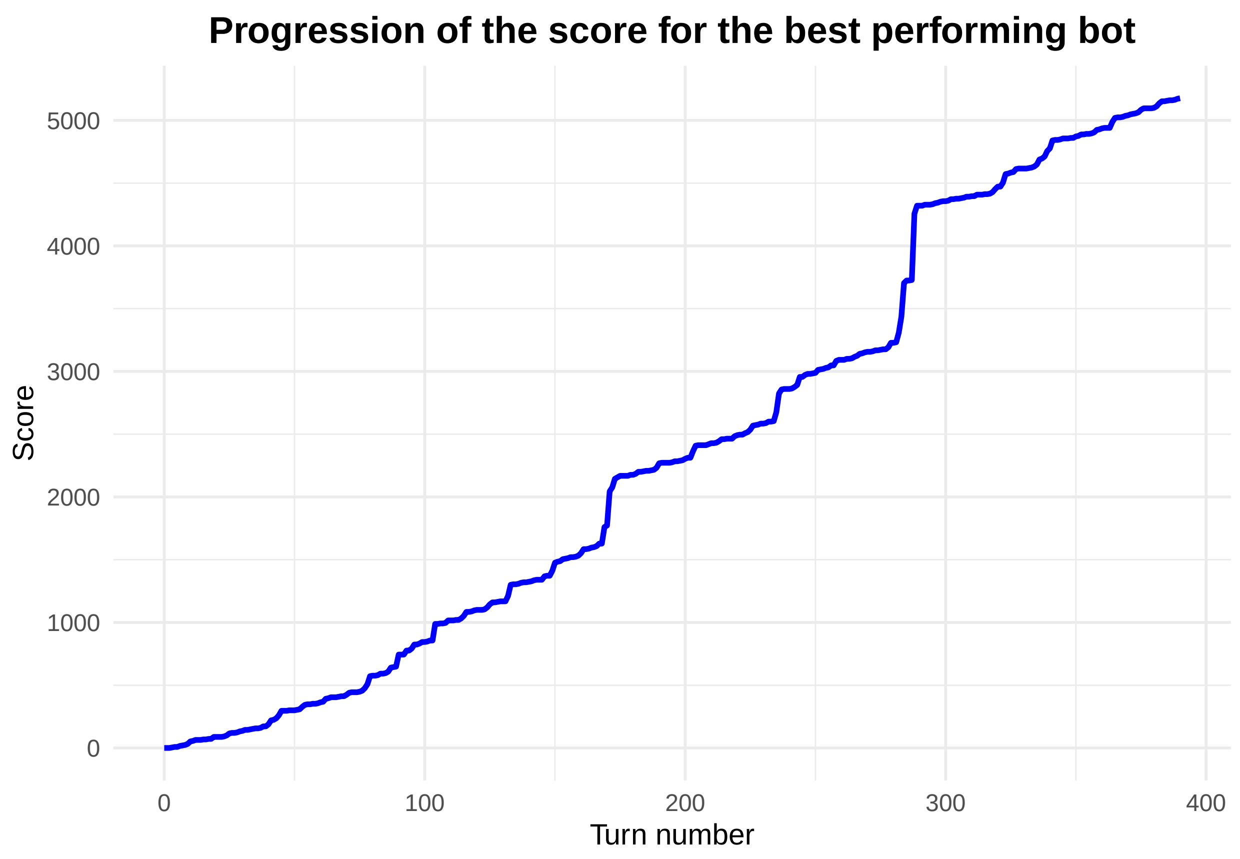
From these scatters we can also see that it tends to be that the bots with a strong preference for some direction have the highest scores. Even in scores between 250-2000, the top end is dominated by the bots with strong preference. However it should be noted that there is a large spread in scores no matter the preference, with some of the lowest scores achieved by a bot with a strong preference. This will be looked at further in another section.

We can also look at the progression of the score for the best performing bot, which achieved a final score of 5176 in 390 turns. This bot was given a strong preference for both going down and right.

```r
best_score <- top_n(final_scores,1,finalScore)
top_scorer <- best_score$`Bot id`
top_score <- best_score$finalScore

best_perf <- allCols %>%  filter(,`Bot id` == top_scorer) %>%
  mutate(score_gained_on_move = `Current score` - lag(`Current score`))

ggplot(best_perf,aes(x=`Turn no.`,y=score_gained_on_move)) +
  geom_line(colour="blue",lwd=0.8) + theme_minimal() +
  labs(x="Turn number",y="Points gained on turn",
       title="Points gained per turn for the best performing bot") +
theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14)) +
  annotate("text",x=60,y=450,label = paste("Top Score : ",top_score),size=5)
```



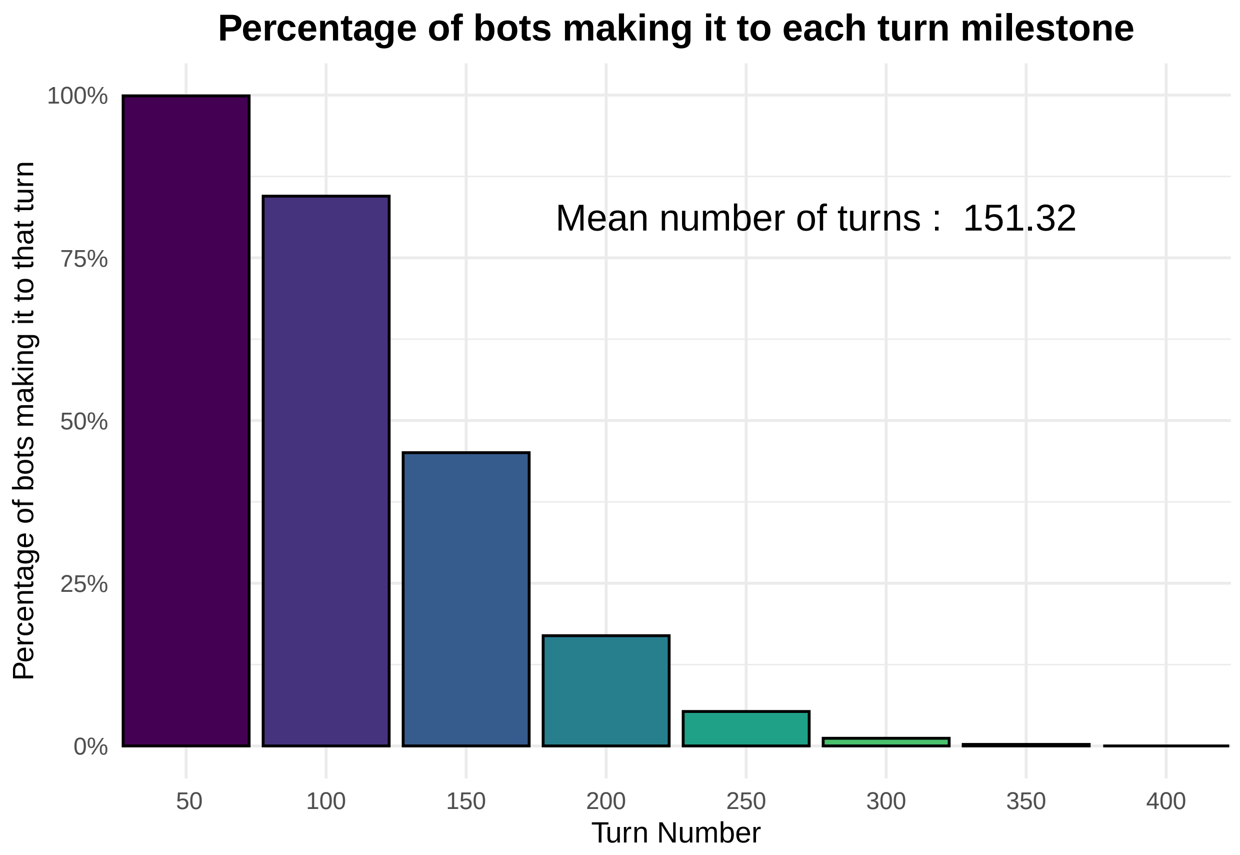
```r
ggplot(best_perf, aes(x = `Turn no.`, y=`Current score`)) +
  geom_line(colour="blue",lwd=1) +  theme_minimal() +
  labs(x = "Turn number", y = "Score",
       title = "Progression of the score for the best performing bot") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14))
```



Using these graphs we can spot where some high scoring tiles were combined, with the largest peak/jump being 512 (the largest tile obtained by the bot), and the two other large peaks both being 256. The spread of large advancements is fairly even, as would be expected since the bot needs to build up to them. We can also see that the rate of progression in the flatter sections is roughly the same throughout.

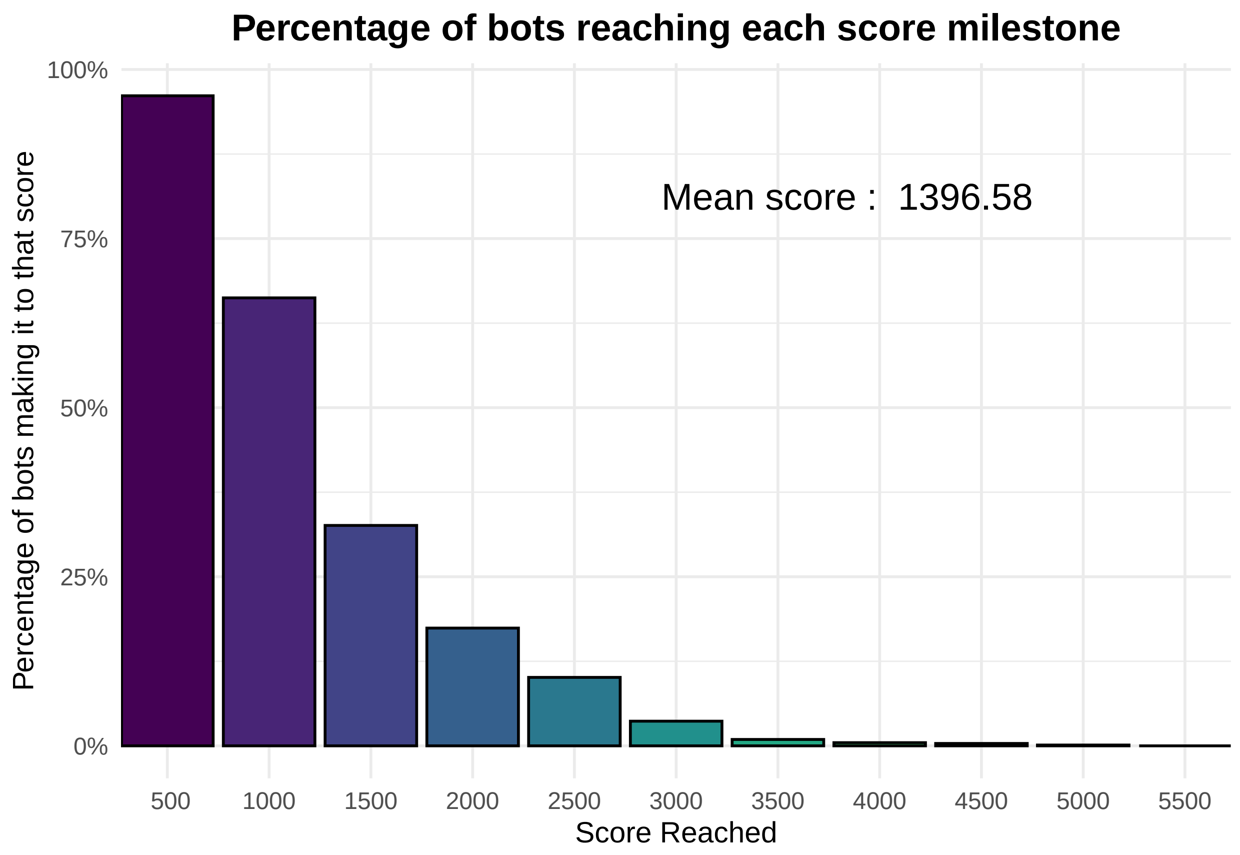
Now we look at the performance of the bots as a collective, and how effective they were at reaching certain milestones in terms of numbers of turns and score.

```r
counts = seq(from=50,to=400,by=50)

total_bots = allCols$`Bot id` %>%
  unique() %>%  length()

mean_num_of_turns <- mean(final_scores$Turns) %>% round(2)

inter <- allCols %>%
  group_by(`Turn no.`) %>%
  summarize(count = n()) %>% ungroup() %>%
   filter(`Turn no.` %in% counts) %>%
   right_join(data.frame("Turn no." = counts,
                         check.names = FALSE),by="Turn no.") %>%
   replace(is.na(.),0) %>% mutate(as_percent = 100*(count / total_bots))

figure <- ggplot(inter,aes(x=`Turn no.`,y=as_percent)) +
  geom_col(aes(fill=as.factor(`Turn no.`)),colour="black",show.legend=FALSE) +
   theme_minimal() + xlim(0,400) +
  labs(x="Turn Number",y="Percentage of bots making it to that turn",
       title = "Percentage of bots making it to each turn milestone") +
  annotate("text", x=275,y=81.25,
           label=paste("Mean number of turns : ",mean_num_of_turns),size=5) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14))

# to remove warning about replacing scale
figure$scales$scales <- list()
figure +  scale_x_discrete(limits=counts) +
  scale_fill_viridis(discrete = TRUE, option = "D") +
   scale_y_continuous(labels = function(x) paste0(x,"%"))
```



From this bar chart we can see the drop-off in the percentage of bots getting past a certain number of turns. The greatest drop-off occurs between 100 and 150 turns, and 150 and 200 turns. This aligns with what we would expect as the mean number of turns is 151.3. This chart reinforces the fact that very few of the bots managed to make it past 300 turns.

```r
score_counts = seq(from=500,to=5500,by=500)
mean_score <- mean(final_scores$finalScore) %>% round(2)

score_bracketing <- final_scores$finalScore %>%
  findInterval(vec=c(0,score_counts)) %>% table()
total_bracketing <- tail(rev(cumsum(rev(score_bracketing))),-1)
length(total_bracketing) <- length(score_counts)
x <- data.frame(total_bracketing,score_counts) %>% replace(is.na(.),0) %>%
  mutate(as_percents = 100 * (total_bracketing / total_bots))

figure <- ggplot(x,aes(x=score_counts,y=as_percents)) +
  geom_col(aes(fill=as.factor(`score_counts`)),colour="black",show.legend=FALSE) +
   theme_minimal() + xlim(0,5500) +
  labs(x="Score Reached",y="Percentage of bots making it to that score",
       title = "Percentage of bots reaching each score milestone") +
  scale_y_continuous(labels = function(x) paste0(x,"%")) +
  annotate("text", x=3840,y=81.25,
           label=paste("Mean score : ",mean_score),size=5) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14))

# to remove waning message about replacing scale
figure$scales$scales <- list()
figure + scale_x_discrete(limits=score_counts) +
  scale_fill_viridis(discrete = TRUE, option = "D") +
    scale_y_continuous(labels = function(x) paste0(x,"%"))
```



96% of the bots successfully made it to 500, after which the next two scores have a fairly similar drop-off of around 30%. 10% of bots made it past 2500, but by 3000 this had fallen to just 3%, and then less than 1% by the time we reached a score of 3500. The mean score is fairly low, likely dragged down by a lot of bots only achieving small scores outweighing the effects of the few bots achieving a larger score. Given that all the bots have been grouped together, some of the poorer performing bots may have also weighed down the mean.

## Directional Preference

In this section we see if direction had an effect on the success of the bots (here we generally do not worry about the strength of the preference in each direction). We would expect that the single directions should all have roughly equivalent performance, since the board is symmetric.

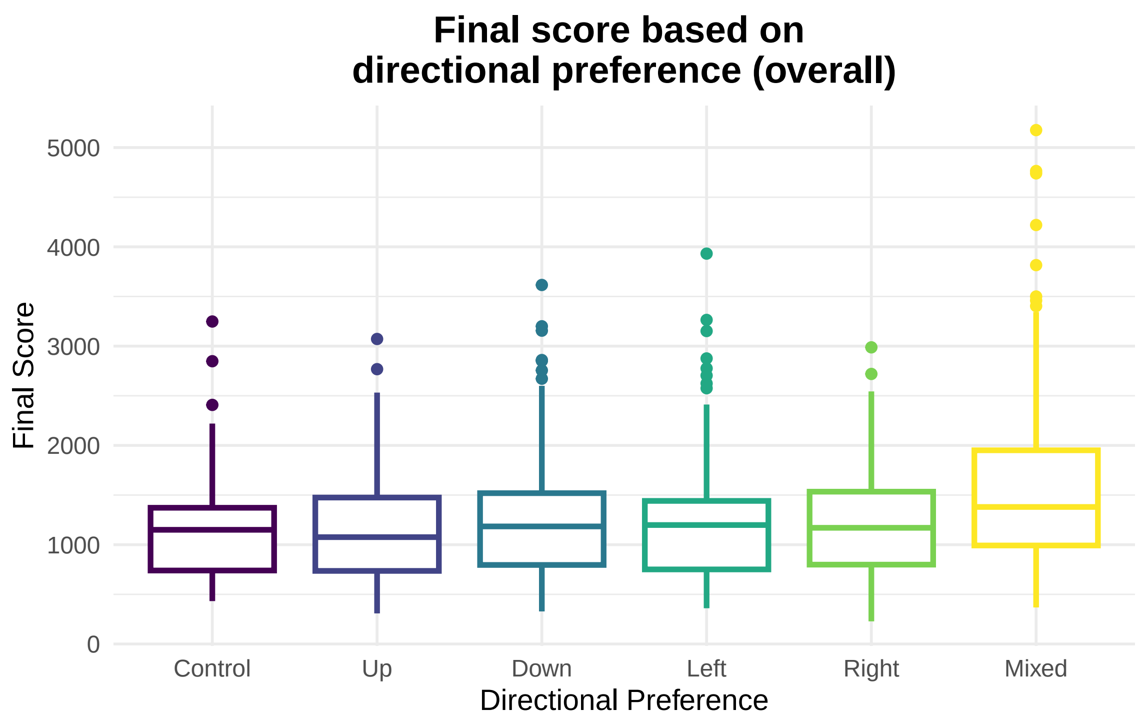
```
dir_finalscore_summary <- final_scores %>%
  group_by(Direction) %>%
   make_summary_table("finalScore")%>%
  arrange(match(Direction,direction_order)) %>%
    mutate(Direction = direction_labs)

knitr :: kable(dir_finalscore_summary,"simple",digits=2,
               caption="Summary stats based on final score")
```

Table 3: Summary stats based on final score

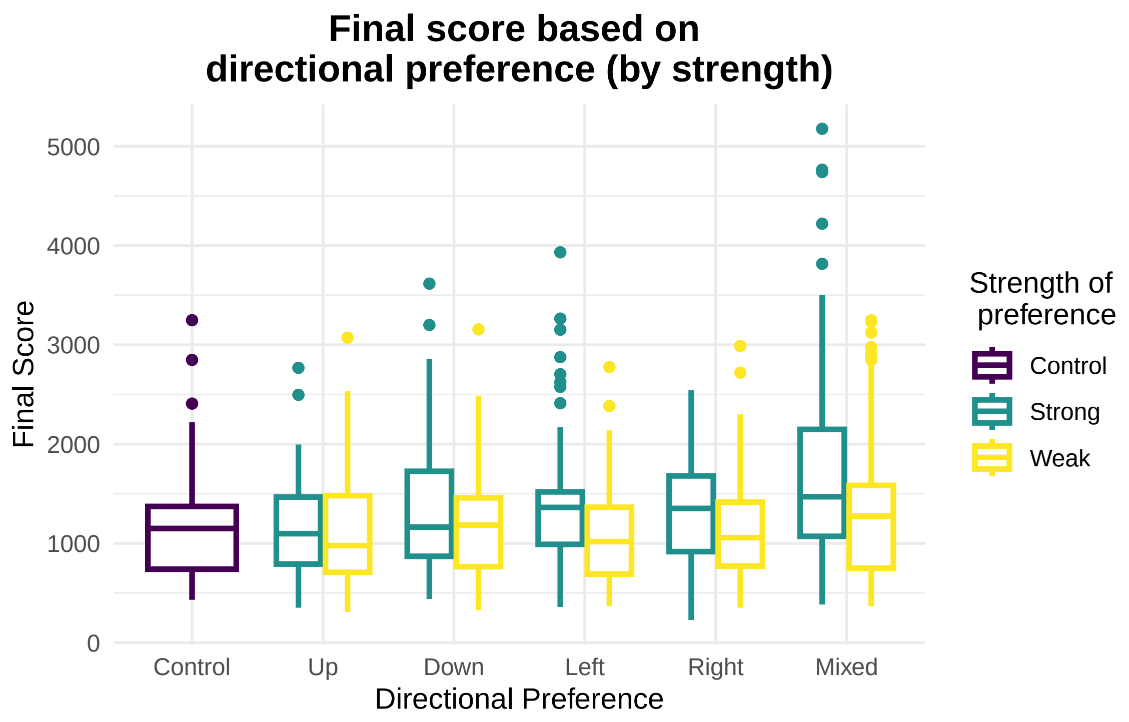
| Direction | Mean | Median | LQ | UQ | Max | Min | SD |
|---|---|---|---|---|---|---|---|
| Control | 1187.12 | 1150 | 740 | 1372 | 3248 | 432 | 589.18 |
| Up | 1157.28 | 1076 | 736 | 1475 | 3072 | 308 | 514.58 |
| Down | 1297.16 | 1184 | 796 | 1519 | 3616 | 328 | 668.43 |
| Left | 1267.32 | 1198 | 751 | 1441 | 3932 | 360 | 687.74 |
| Right | 1262.64 | 1170 | 799 | 1534 | 2988 | 228 | 582.58 |
| Mixed | 1573.25 | 1380 | 992 | 1951 | 5176 | 368 | 807.98 |

Just by looking at the summary table we can see that the four solo directions were fairly similar, with Up being the weakest. Down and Left also have fairly high standard deviations compared to Right and Up. However in most respects these four behaved as expected and were fairly similar, something which is even more apparent in the boxplots below. The mixed directions clearly far outperformed all the other combinations.

```
dirVsFinalScore0 <- make_boxplot(final_scores,"Direction","finalScore",
                                 "Direction",show_legend = FALSE)
dirVsFinalScore0 +
      labs(x="Directional Preference", y = "Final Score",
       title = "Final score based on \ndirectional preference (overall)") +
  scale_x_discrete(labels = direction_labs)
```



```
dirVsFinalScoreS <- make_boxplot(final_scores,"Direction","finalScore",
                                  "prefStrength")
dirVsFinalScoreS +
    labs(x="Directional Preference", y = "Final Score",
       title="Final score based on \ndirectional preference (by strength)",
       color="Strength of \n preference") +
  scale_x_discrete(labels = direction_labs)
```



In these two graphs we see even more clearly that the performance of the four solo directions is very similar, with their boxplots looking almost the same. The control bot where all 4 directions had equal weighting seemingly tended to do the worst, whilst mixed combinations outperformed all other bots including with the most high-valued outliers.

Looking at the second graph, we see that no matter the direction chosen, the bots with a stronger preference tended to do much better than those with a weaker preference. Interestingly though, Left and Right were better by a large margin, whilst Up and Down were only slightly better on a stronger preference.

Now we move to see the effect of each directional preference on the number of turns the bots lasted. Although this is not the metric we particularly care about (a long, low scoring game is not as good as a short, fast scoring game), from above we see that there is a strong correlation between number of turns and score, so a game with lots of turns is likely to have been very successful.

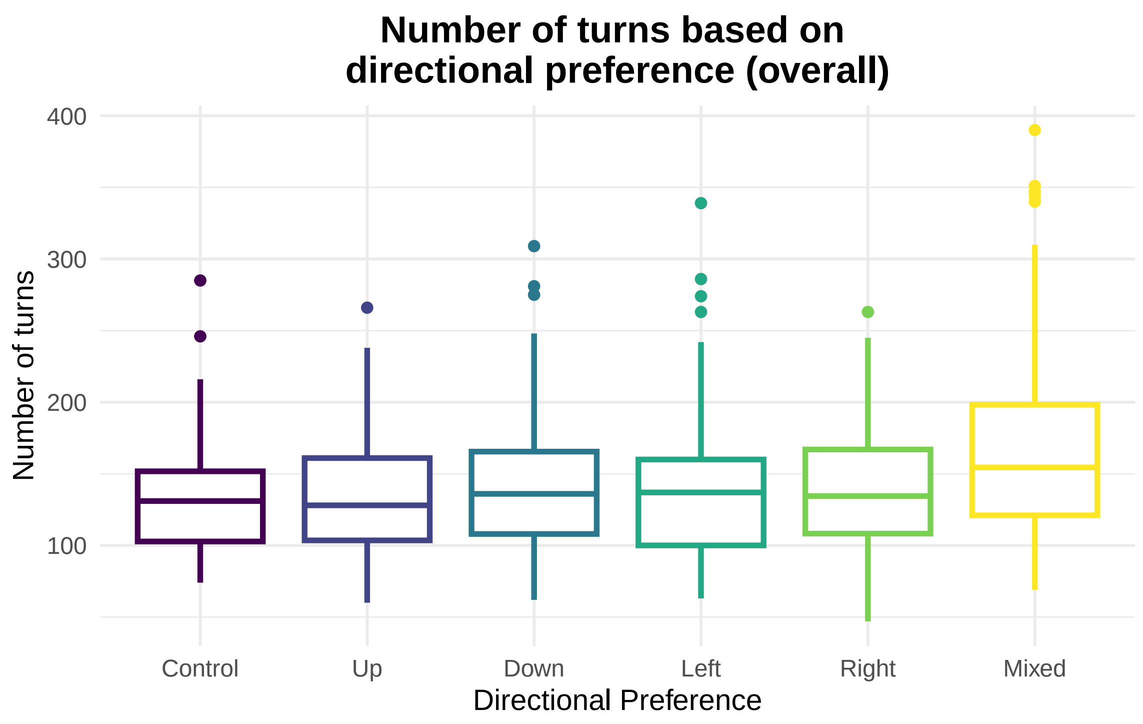
```
dir_turns_summary <- final_scores %>%
  group_by(Direction) %>%
   make_summary_table("Turns") %>%
  arrange(match(Direction,direction_order)) %>%
    mutate(Direction = direction_labs)

knitr :: kable(dir_turns_summary,"simple",digits=2,
               caption="Summary stats based on number of turns")
```

Table 4: Summary stats based on number of turns

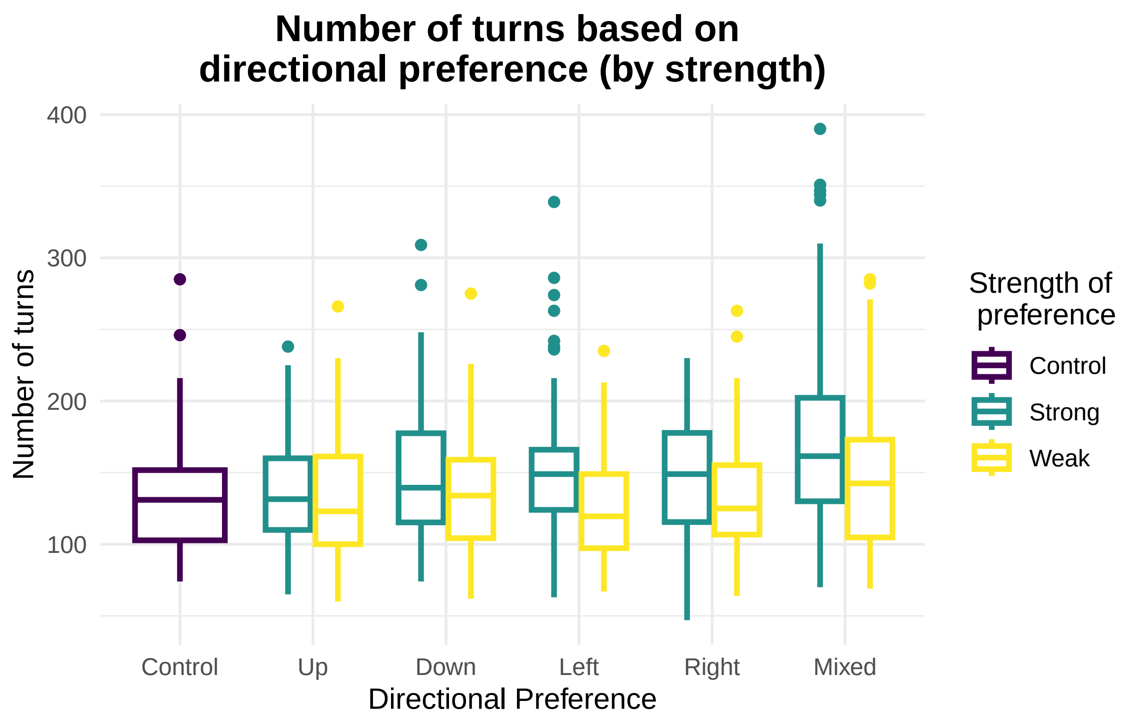
| Direction | Mean | Median | LQ | UQ | Max | Min | SD |
|---|---|---|---|---|---|---|---|
| Control | 134.18 | 131.0 | 102.75 | 151.75 | 285 | 74 | 42.68 |
| Up | 133.97 | 128.0 | 103.50 | 161.00 | 266 | 60 | 39.41 |
| Down | 144.26 | 136.0 | 108.00 | 165.50 | 309 | 62 | 48.73 |
| Left | 140.80 | 137.0 | 100.00 | 160.00 | 339 | 63 | 51.25 |
| Right | 140.77 | 134.5 | 108.25 | 167.00 | 263 | 47 | 43.79 |
| Mixed | 164.84 | 154.5 | 121.00 | 198.25 | 390 | 69 | 57.79 |

As in the case for the final score, here again we see than the four solo directions were all fairly similar in terms of their performance, with Up still being the weakest. Mixed combinations also came out on top in this metric, although its standard deviation was more similar to the other categories this time.

```
dirVsTurns0 <- make_boxplot(final_scores,"Direction","Turns",
                            "Direction",show_legend = FALSE)
dirVsTurns0 +
  labs(x="Directional Preference", y = "Number of turns",
       title = "Number of turns based on \ndirectional preference (overall)") +
  scale_x_discrete(labels = direction_labs)
```



```
dirVsTurnsS <- make_boxplot(final_scores,"Direction","Turns","prefStrength")
dirVsTurnsS +
  labs(x="Directional Preference", y = "Number of turns",
       title = "Number of turns based on \ndirectional preference (by strength)",
       colour= 'Strength of \n preference') +
  scale_x_discrete(labels = direction_labs)
```



We can see that the graphs for the number of turns and final score are very similar visually, with the difference between them only really clear when you look at the scale. This obviously follows from there being a very strong linear correlation between these two variables, as mentioned earlier. We will also see this pattern repeating the section looking at the effect of the strength of the preference.

## Strength Preference

In this section we change our focus to see if the strength of the preference has an impact on the score. We mainly compare a weak preference (max 5 in any direction) with a strong preference(25 in some direction), alongside the completely random bot.

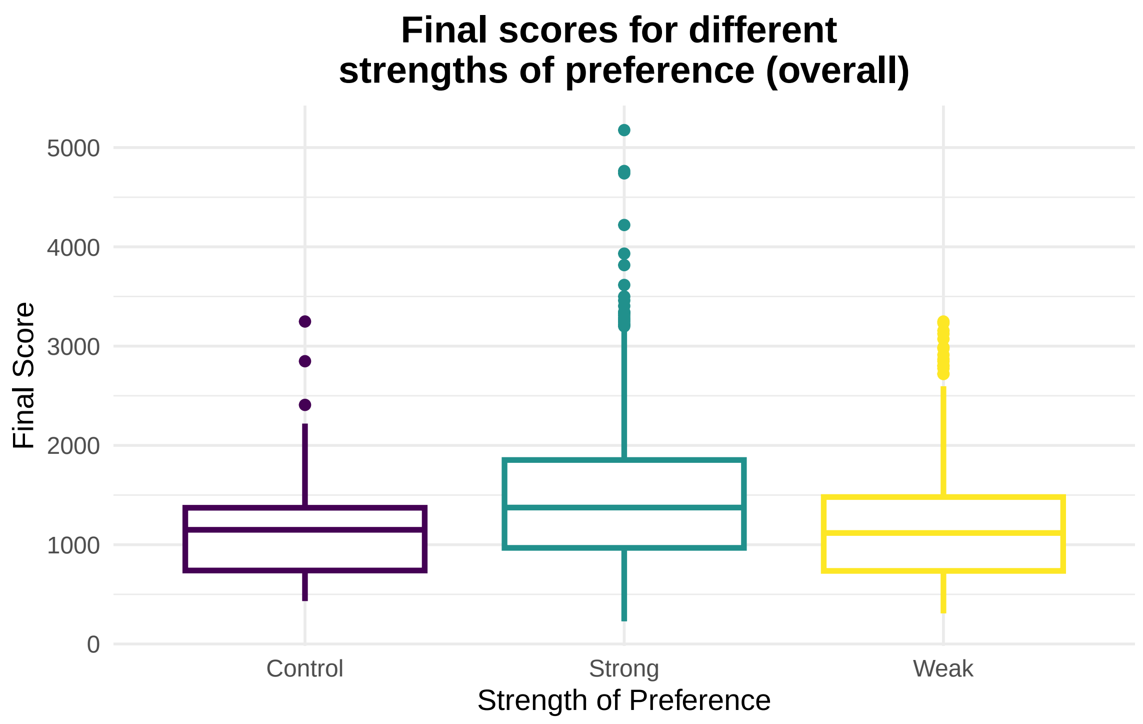
```
strength_finalscore_summary <- final_scores %>%
  group_by(prefStrength) %>%
  make_summary_table("finalScore")  %>%
  rename(`Strength of Preference` =prefStrength)

knitr :: kable(strength_finalscore_summary,"simple",digits=2,
               caption="Summary stats based on final score")
```

Table 5: Summary stats based on final score

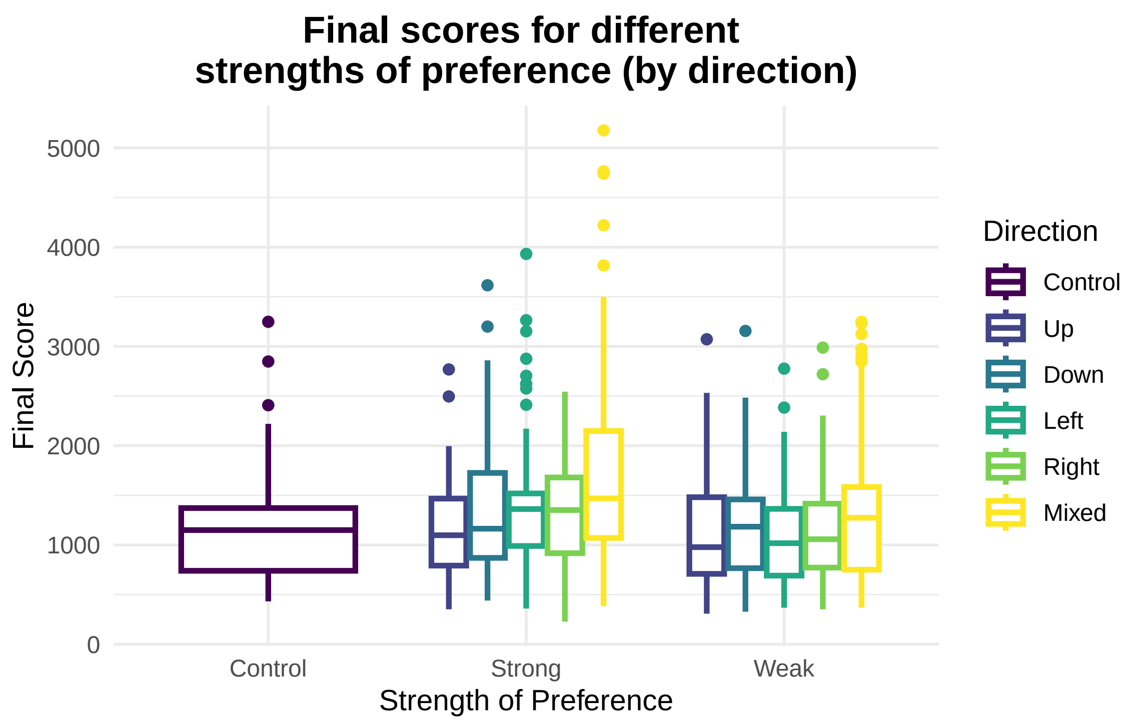
| Strength of Preference | Mean | Median | LQ | UQ | Max | Min | SD |
|---|---|---|---|---|---|---|---|
| Control | 1187.12 | 1150 | 740 | 1372 | 3248 | 432 | 589.18 |
| Strong | 1523.86 | 1374 | 968 | 1853 | 5176 | 228 | 777.08 |
| Weak | 1219.37 | 1118 | 736 | 1480 | 3248 | 308 | 620.64 |

From this table we see that a strong preference comes out superior in virtually every metric, although it does have a much larger spread of values. It's mean score is over 300 points larger than that of the bots with a weak preference, although this is probably helped by a couple of very large scores achieved by the strong preference bots. Interestingly though, the strong preference bots also achieved the lowest score of just 228.

```
strengthVsScore0 <- make_boxplot(final_scores,"prefStrength","finalScore","prefStrength",show_legend = 
strengthVsScore0 + labs(x="Strength of Preference", y = "Final Score",
                        title ="Final scores for different \nstrengths of preference (overall)")
```



```
strengthVsScoreD <- make_boxplot(final_scores,"prefStrength","finalScore","Direction") +
           labs(x="Strength of Preference", y = "Final Score",
                        title ="Final scores for different \nstrengths of preference (by direction)")

strengthVsScoreD$scales$scales <- list()
strengthVsScoreD +  scale_colour_viridis(discrete=TRUE,option='D',labels=direction_labs)
```



From these boxplots we see that on average,the weak bots weren't much better than the control bots, although we see they were far more likely to actually reach higher scores from the number of outliers. Clearly though, on average the strong preference bots were head and shoulders above the others and achieved the vast majority of large scores as seen in all the high outliers.

From the second graph we see that the best performing bots tended to be the bots with a strong, mixed preference for direction.

Similarly to direction, we will also now look at a comparison of the number of turns achieved in the games by bots with different strengths of preferences.

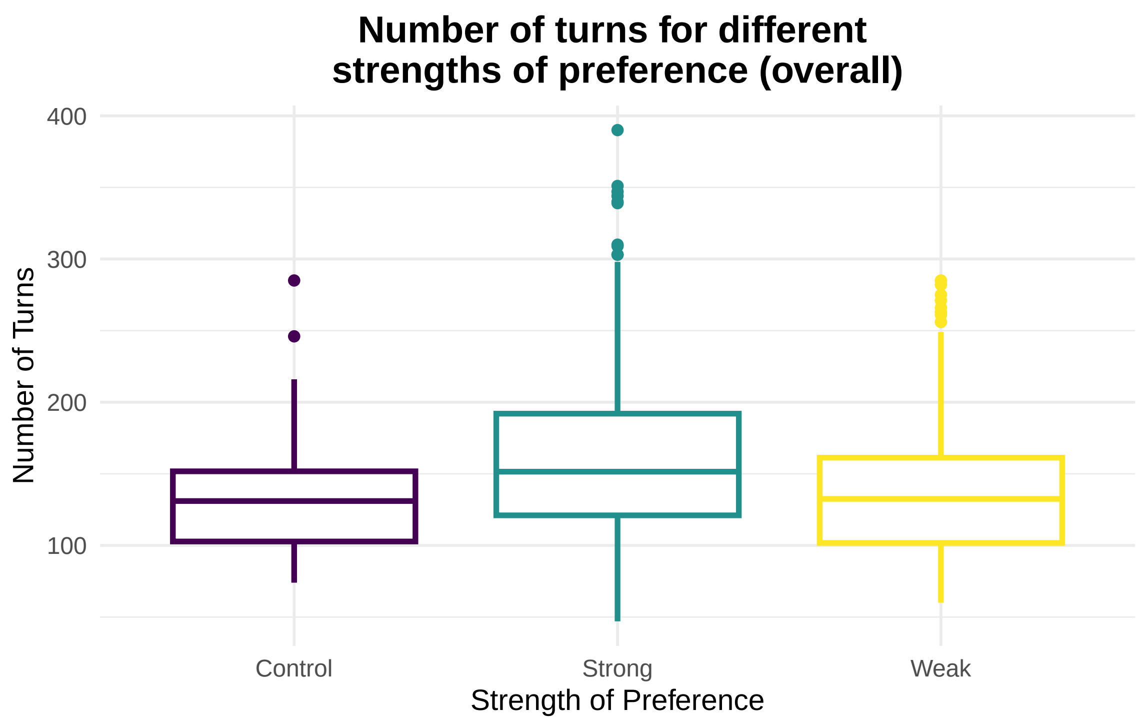
```r
strength_turns_summary <- final_scores %>%
  group_by(prefStrength) %>%
  make_summary_table("Turns")

knitr :: kable(strength_turns_summary,"simple",digits=2,
               caption="Summary stats based on number of turns")
```

Table 6: Summary stats based on number of turns

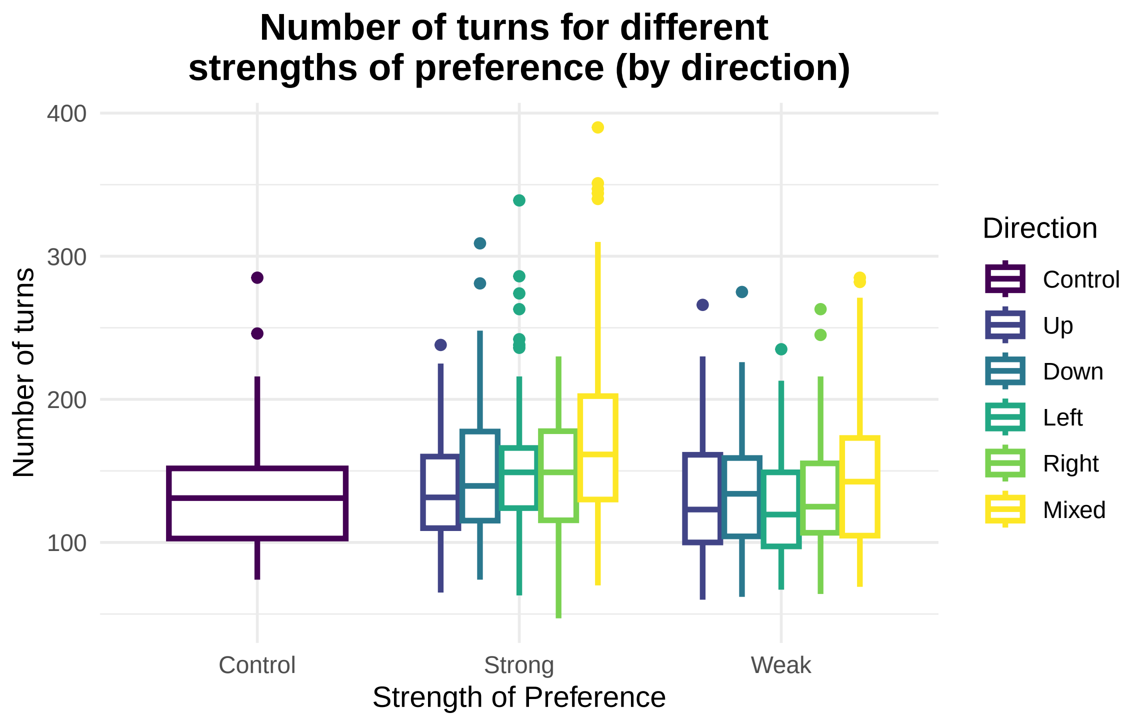
| prefStrength | Mean | Median | LQ | UQ | Max | Min | SD |
|---|---|---|---|---|---|---|---|
| Control | 134.18 | 131.0 | 102.75 | 151.75 | 285 | 74 | 42.68 |
| Strong | 161.38 | 151.5 | 121.00 | 192.00 | 390 | 47 | 55.93 |
| Weak | 137.42 | 132.5 | 101.75 | 161.25 | 285 | 60 | 45.98 |

Echoing the summary from earlier, strong preference bots have outperformed the other bots by a fairly larger margin, with a mean number of turns 30 greater than either of the other two. However their spread of how many turns each game took was much larger, and a strong preference bot also had the shortest game of just 47 turns.

```r
strengthVsTurnsO <- make_boxplot(final_scores,"prefStrength","Turns","prefStrength",show_legend = FALSE
strengthVsTurnsO +labs(x="Strength of Preference", y = "Number of Turns",
                       title ="Number of turns for different \nstrengths of preference (overall)")
```



```
strengthVsTurnsD <- make_boxplot(final_scores,"prefStrength","Turns","Direction") +
                    labs(x="Strength of Preference", y = "Number of turns",
                        title ="Number of turns for different \nstrengths of preference (by direction)"
strengthVsTurnsD$scales$scales <- list()
strengthVsTurnsD + scale_colour_viridis(discrete=TRUE,option='D',labels=direction_labs)
```



As noted earlier these box plots are very similar to those for final score, with the only noticeable difference being in the scale. Mixed combination bots prevail in both strong and weak preferences. We can see that there is a bit of a spread between the different solo directions, with some seeming to perform better than others. Interestingly, Up and Down seem to be the worst when it comes to strong preference bots, but then some of the best on the weak preference bots. However this may just be something that would be ironed out if we ran the bots for more than 50 times each.

## Combination vs Solo preference

Finally we compare the performance of the bots which had a preference for only one direction with the bots which had a preference for two directions (the pairings being up/left and down/right).

```
combo_finalscore_summary <- final_scores %>%
  group_by(comboOrSingle) %>%
  make_summary_table("finalScore") %>%
   slice(2,1,3) %>% rename(`Button combination` = comboOrSingle )

knitr :: kable(combo_finalscore_summary,"simple",digits = 2,
               caption="Summary stats based on final score")
```

Table 7: Summary stats based on final score

| Button combination | Mean | Median | LQ | UQ | Max | Min | SD |
|---|---|---|---|---|---|---|---|
| Control | 1187.12 | 1150 | 740 | 1372 | 3248 | 432 | 589.18 |
| Combo | 1573.25 | 1380 | 992 | 1951 | 5176 | 368 | 807.98 |
| Single | 1246.10 | 1168 | 768 | 1489 | 3932 | 228 | 617.21 |

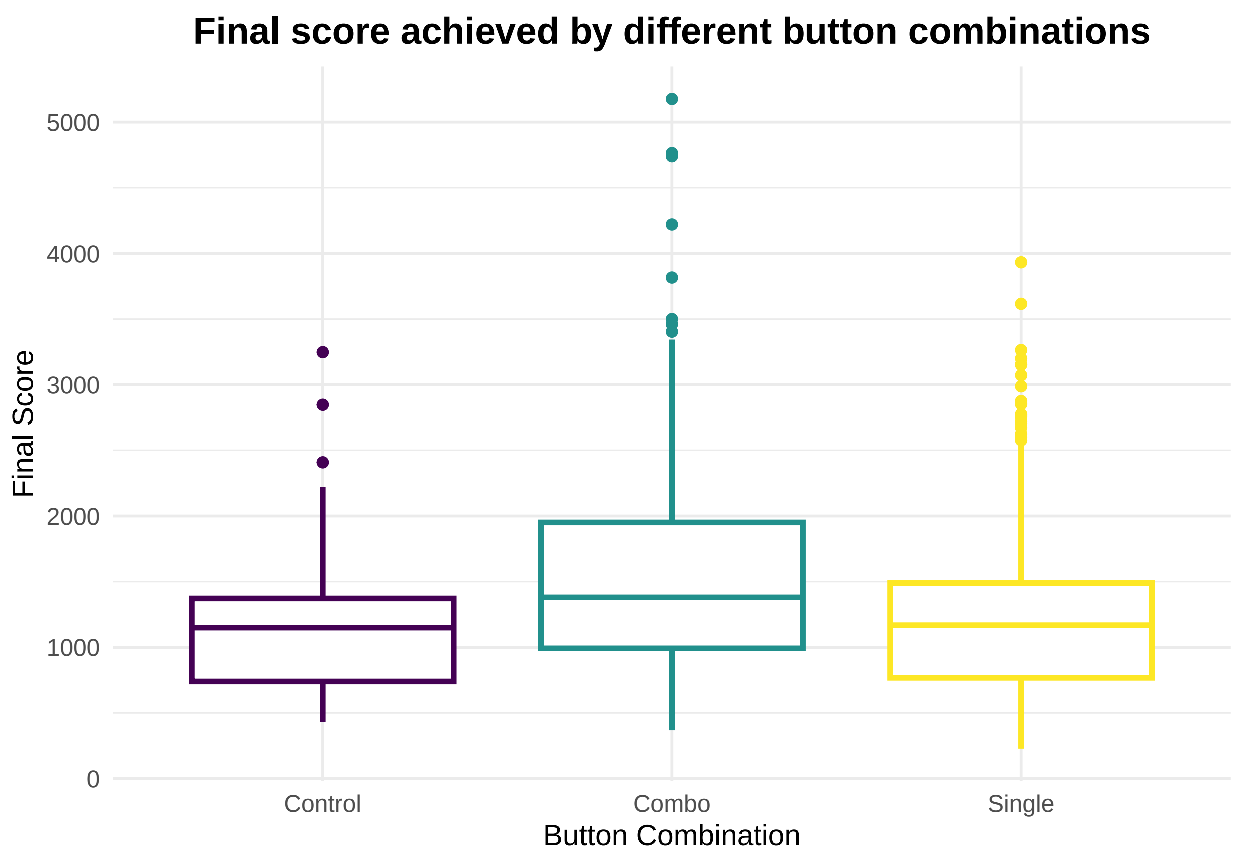
```
combo_turns_summary <- final_scores %>%
  group_by(comboOrSingle) %>%
  make_summary_table("Turns") %>%
  slice(2,1,3) %>% rename(`Button combination` = comboOrSingle )

knitr :: kable(combo_turns_summary,"simple",digits = 2,
               caption="Summary stats based on number of turns")
```

Table 8: Summary stats based on number of turns

| Button combination | Mean | Median | LQ | UQ | Max | Min | SD |
|---|---|---|---|---|---|---|---|
| Control | 134.18 | 131.0 | 102.75 | 151.75 | 285 | 74 | 42.68 |
| Combo | 164.84 | 154.5 | 121.00 | 198.25 | 390 | 69 | 57.79 |
| Single | 139.95 | 134.5 | 106.00 | 163.00 | 339 | 47 | 46.00 |

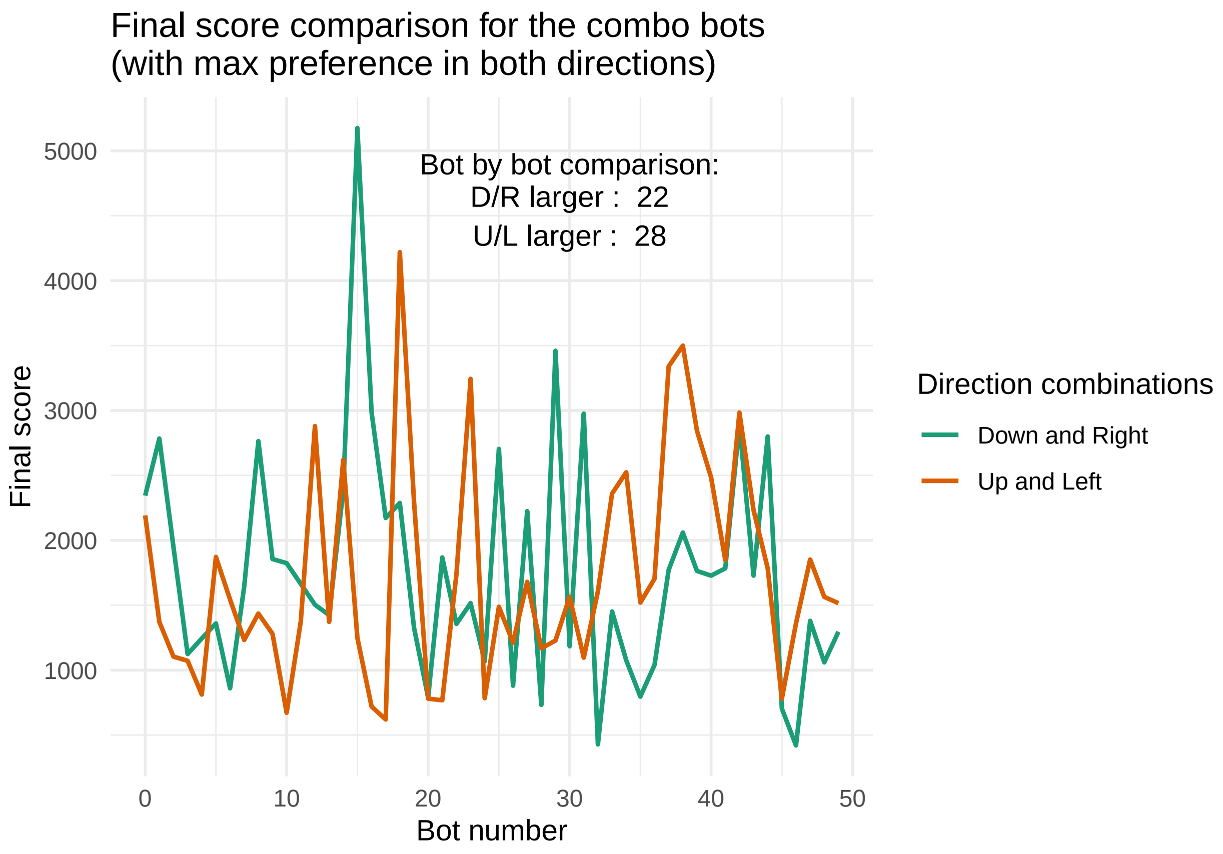
From the summary tables we see that the bots with a combination of preferences generally outperform the other bots in both number of turns and final score. The tables here are extremely similar to those for strength of preference, with nearly identical means for combo/strong preferences and single/weak preferences. (The group of control bots are in fact the same)

```
final_scores$comboOrSingle <- factor(final_scores$comboOrSingle,levels=c("Control","Combo","Single"))

comboVsScore <- make_boxplot(final_scores,"comboOrSingle","finalScore","comboOrSingle",show_legend = FA
comboVsScore +
  labs(x="Button Combination",y = "Final Score",
       title ="Final score achieved by different button combinations")
```



Practically all the scores over 3250 were achieved by bots with a combination of preferences. We can easily see here that the bots with just a single directional preference were not much better than the completely random bots on average, although were better at more consistently achieving a score greater than 2500. The greatest spread of results appears in the combo bots, with their worst score lower than that of the control bots.

```r
max_strength <- final_scores %>%
  filter(comboOrSingle == "Combo" & ((`up weight` == 25 & `left weight` == 25) |
                                       (`down weight` == 25 & `right weight` == 25))) %>%
  mutate(dir = case_when(`up weight` != 1 ~ "U/L",
                         `down weight` != 1 ~ "D/R")) %>%
  mutate(bot_num = as.numeric(substr(`Bot id`,start=2,stop=nchar(`Bot id`))))

comparison <- left_join(filter(max_strength, dir == "D/R"),filter(max_strength, dir == "U/L"),by="bot_nu
  mutate(DRBigger = finalScore.x >= finalScore.y )

value_counts <- table(comparison$DRBigger)
DRBigger <- value_counts['TRUE']
notDRBigger <- value_counts['FALSE']

ggplot(max_strength,aes(x=bot_num,y=finalScore,colour=dir)) + geom_line(lwd=0.8) +
  labs(x="Bot number",
       y = "Final score",
       title = "Final score comparison for the combo bots \n(with max preference in both directions)",
       color="Direction combinations",) + theme_minimal() +
  scale_colour_brewer(palette="Dark2", labels=c("Down and Right", "Up and Left")) +
  annotate("text",x=c(30,30,30),y = c(4900,4650,4350),label=c("Bot by bot comparison:",
                                                              paste("D/R larger : ", DRBigger),
                                                              paste("U/L larger : ", notDRBigger)))
```



Here we can see a comparison of the two different types of combo bots we used, with the combination of Down and Right in green, and Up and Left in red. Their results are fairly similar, as we would expect since they should just be symmetric versions of one another. Both achieve a few high points, and their fair share of lower scores too. On the whole, U/L was slightly better on a bot by bot basis, but this mainly depends on the performance of that numbered bot, rather than any general indications as to trends.

## Conclusion

On the whole, we can see that the optimal bots seem to have strong preferences, and a combination of preferences. It seems that having stronger preferences and using a combination of preferences are equally important, as the means when looking at these categories individually are extremely similar. Having any kind of preference, in any direction, is preferable to complete pseudo-randomness indicating that trying to collect the tiles in a specific location is a strategy which has some chance at success (and indeed is the strategy often used by human players looking to get a good score).

These results could definitely be improved if I were to redo the project, for example by running each bot more times, since 50 is a fairly small sample size. I could also experiment more with different weightings to the preferences. More exploration could also be done, looking at which is the most consistent configuration, for example, since the mixed combinations can perform very well, but also had a few shockingly bad performances as mentioned above.

Thanks for reading, any feedback/advice/improvements would be welcomed. My version of 2048 is available to play yourself and can be found on my github page along with the csv file all these visulisations were created from, and the scripts to run the bots.