

DATA607 Project 1 - Elo

David Simbandumwe

Summary

This project starts with a text file of chess tournament results. This file is processed to develop an R Markown file that generates a .CSV file that includes: + Player's Name, + Player's State, + Total Number of Points, + Player's Pre-Rating, + Average Pre Chess Rating of Opponent

Load Files

```
eloTxt <- read_delim( file = "https://raw.githubusercontent.com/dsimband/DATA607/main/Project1/tournament.txt",
  delim = " ",
  col_names = c("rec"),
  trim_ws = TRUE)
```

```
## Rows: 196 Columns: 1
```

```
## -- Column specification -----
## Delimiter: " "
## chr (1): rec
```

```
##
## 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.
```

```
# clean data frame remove extra lines
eloTxt <- eloTxt %>%
  filter(!str_detect(rec,"\\--"))
```

```
## Warning: One or more parsing issues, see 'problems()' for details
```

```
# remove heading rows
eloTxt <- eloTxt %>% slice(-c(1,2))
```

Process Odd Rows

```

eloTxt_odd <- eloTxt %>% filter(row_number() %% 2 == 1)

eloTxt_odd <- eloTxt_odd %>%
  separate(rec,
    into = c("id", "name", "score", "r1", "r2", "r3", "r4", "r5", "r6", "r7", "x" ),
    extra = "merge",
    fill = "left",
    convert = TRUE,
    sep = "\\|"
  ) %>% gather(
    "r1", "r2", "r3", "r4", "r5", "r6", "r7", key = "round", value = "op_data"
  ) %>%
  separate(op_data,
    into = c("r_result", "op_id"),
    convert = TRUE
  ) %>%
  drop_na(op_id)

```

Process Even Rows

```

eloTxt_even <- eloTxt %>% filter(row_number() %% 2 == 0)

eloTxt_even <- eloTxt_even %>%
  separate(rec,
    into = c("state", "score_data", "x" ),
    extra = "merge",
    fill = "left",
    convert = TRUE,
    sep = "\\|" ) %>%
  mutate(id = row_number()) %>%
  separate(score_data,
    into = c("x1", "x2", "x3", "pre_rating", "post_rating"),
    extra = "merge",
    fill = "left",
    convert = TRUE) %>%
  separate(pre_rating,
    into = c("pre_rating", "x4"),
    extra = "merge",
    fill = "right",
    convert = TRUE,
    sep = "P") %>%
  separate(post_rating,
    into = c("post_rating", "x5"),
    extra = "merge",
    fill = "right",
    convert = TRUE,
    sep = "P") %>%
  select(id, state, pre_rating, post_rating)

```

Join All Rows

```
eloTxt_comb <- eloTxt_odd %>%
  select (id, name, score, round, r_result, op_id) %>%
  left_join(eloTxt_even, by = "id") %>%
  left_join(eloTxt_even, by = c("op_id" = "id"), suffix = c("", ".op"))
```

Calculate

```
win_df <- eloTxt_comb %>%
  group_by(id,r_result) %>%
  filter(str_detect(r_result, "W")) %>%
  mutate(
    win_elo = sum(pre_rating.op) + n()*400,
    win_num = n()
  ) %>%
  select(id,r_result,win_elo, win_num) %>%
  distinct()

loss_df <- eloTxt_comb %>% group_by(id,r_result) %>%
  filter(str_detect(r_result, "L")) %>%
  mutate(
    loss_elo = sum(pre_rating.op) - n()*400,
    loss_num = n()
  ) %>%
  select(id,r_result,loss_elo, loss_num) %>%
  distinct()
```

create the summary table

```
eloFinal <- eloTxt_comb %>%
  select (id, name, state, score, round, r_result, op_id, pre_rating, post_rating, pre_rating.op, post_
  left_join(win_df, by = "id") %>%
  left_join(loss_df, by = "id") %>%
  replace_na(list(win_elo = 0, win_num = 0, loss_elo = 0, loss_num = 0 ))

# caculate average / post elo and filter columns
eloFinal <- eloFinal %>%
  group_by(id) %>%
  mutate(
    op_avg_rating = round(mean(pre_rating.op),0),
    num_games = n(),
    elo_post = round((win_elo + loss_elo) / n(),0)
  ) %>%
  select(id , name, score, state, pre_rating, post_rating, op_avg_rating, num_games, elo_post) %>%
  distinct() %>%
  arrange(id)
```

Write Files

The instructions say

When you write your .CSV, do not include a relative path (e.g. “tournament.csv” instead of “C:/Users/geroso/Documents/DATA-607/tournament.csv”), so that your code is more reproducible. (2 points)

this seems a little counter intuitive as I would assume that not everyone would have the same directory structure on their local machine. I based the code on a relative path but i also included the full path per the instructions.

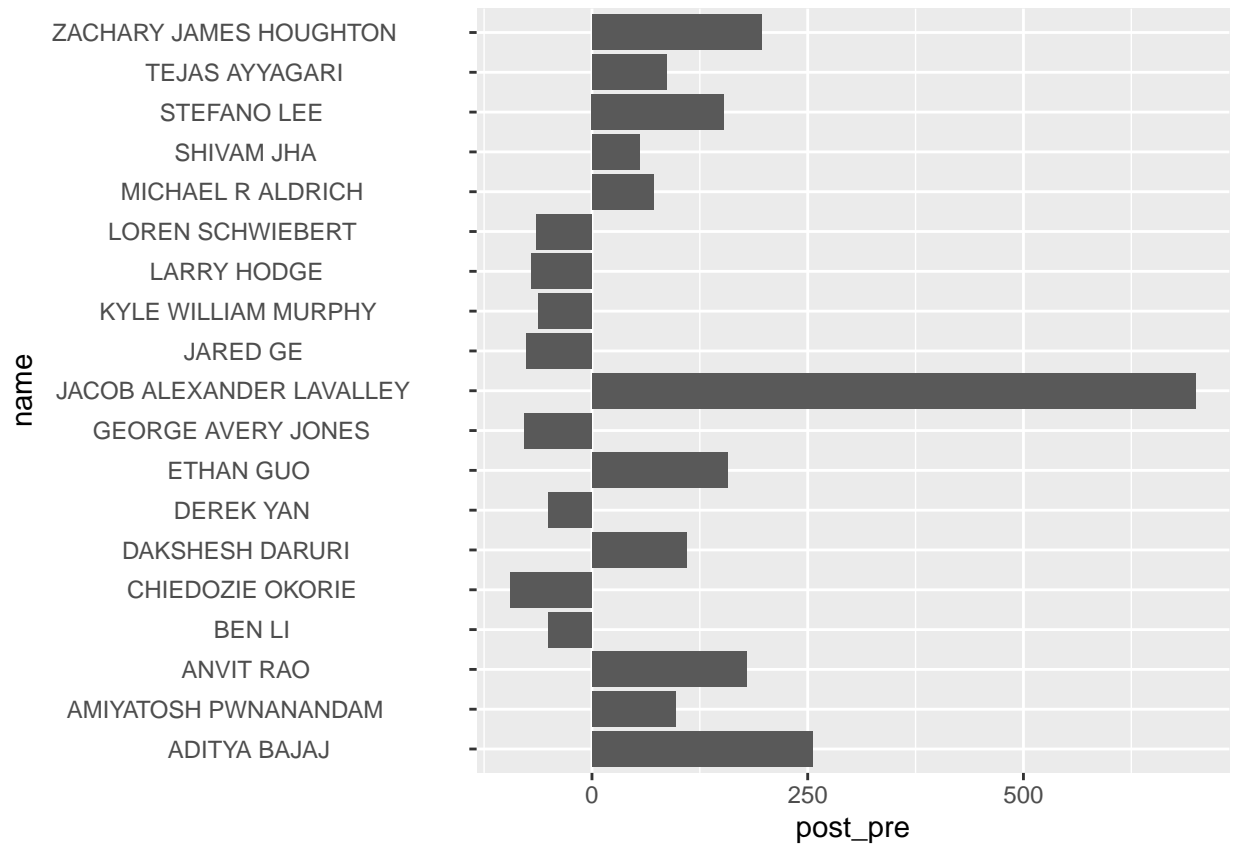
```
# write the results out to disk
write.csv(eloFinal, "tournamentinfo.csv", row.names=FALSE)

# full path
write.csv(eloFinal, "/Users/dsimbandumwe/dev/cuny/data_607/DATA607/Project1/tournamentinfo.csv", row.names=FALSE)
```

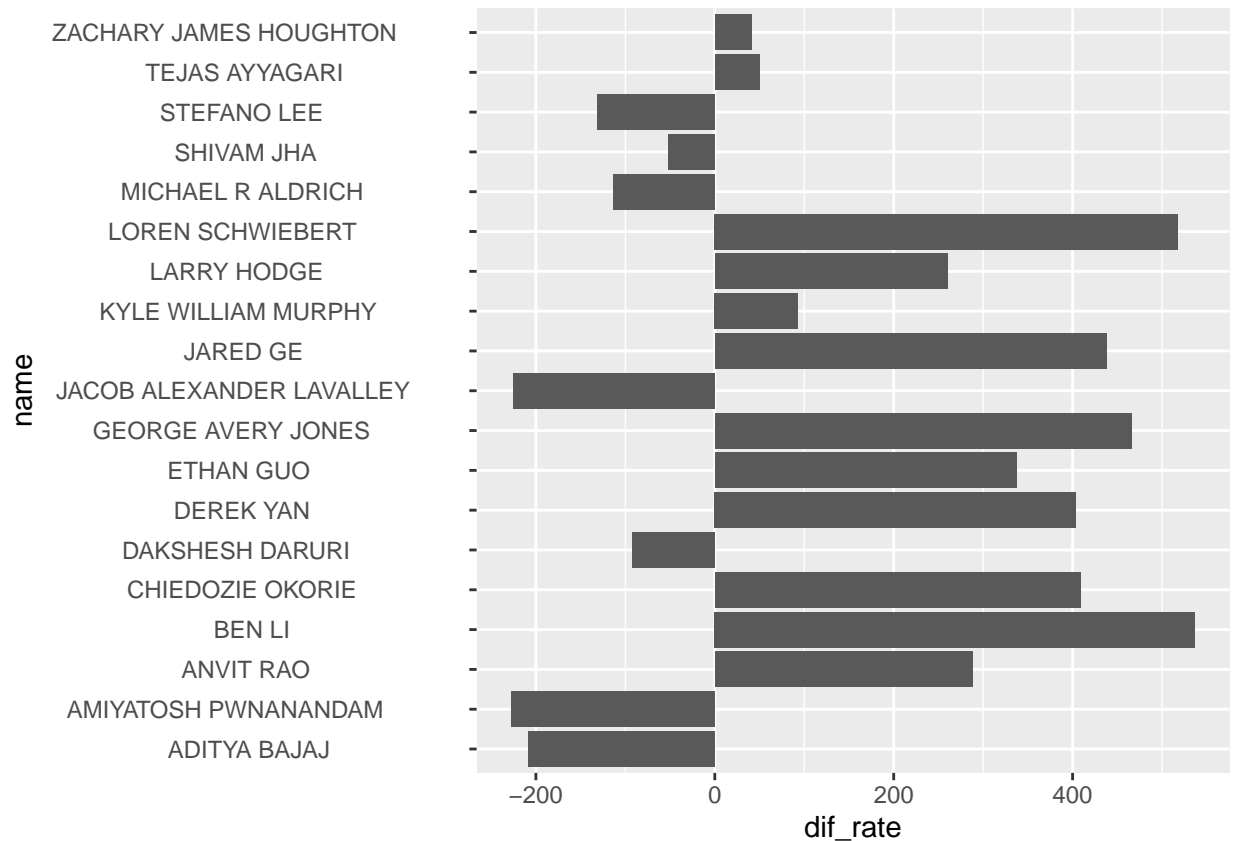
Results

Reviewing the results of the tournament there are several big movers in both the positive and the negative directions. The largest positive jump can be attributed to Jacob not having a high ELO rating at the start of the tournament but managing to win games against higher ranked players. The negative movement were much smaller and were caused by losses to players with lower ELO rankings

```
eloFinal %>%
  mutate(
    dif_rate = post_rating - elo_post,
    post_pre = post_rating - pre_rating
  ) %>%
  filter (post_pre > 50 | post_pre < -50) %>%
  ggplot(aes(x=name, y=post_pre)) + geom_bar(stat="identity") + coord_flip()
```



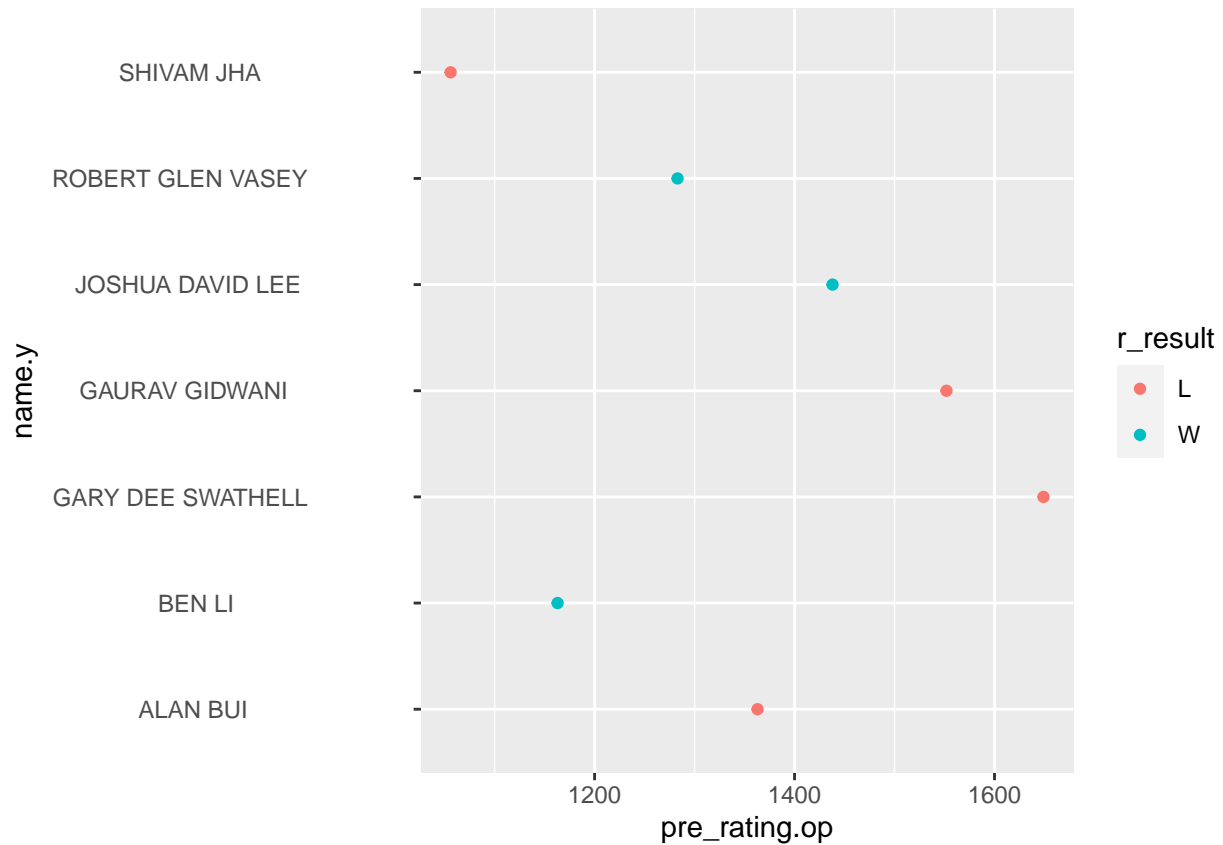
```
eloFinal %>%
mutate(
  dif_rate = post_rating - elo_post,
  post_pre = post_rating - pre_rating
) %>%
filter (post_pre > 50 | post_pre < -50) %>%
ggplot(aes(x=name, y=dif_rate)) + geom_bar(stat="identity") + coord_flip()
```



46 JACOB ALEXANDER LAVALLEY

Jacob started with a low ELO rating and he competed against players with much higher ratings and he was able to defeat several players. The average ELO rating of his opponent was 1358. He had a solid tournament if you calculated his ELO rating based on his tournament performance and the average pre tournament ratings of his competitors his ELO rating would have been 1301 (ignoring his pre tournament ELO rating).

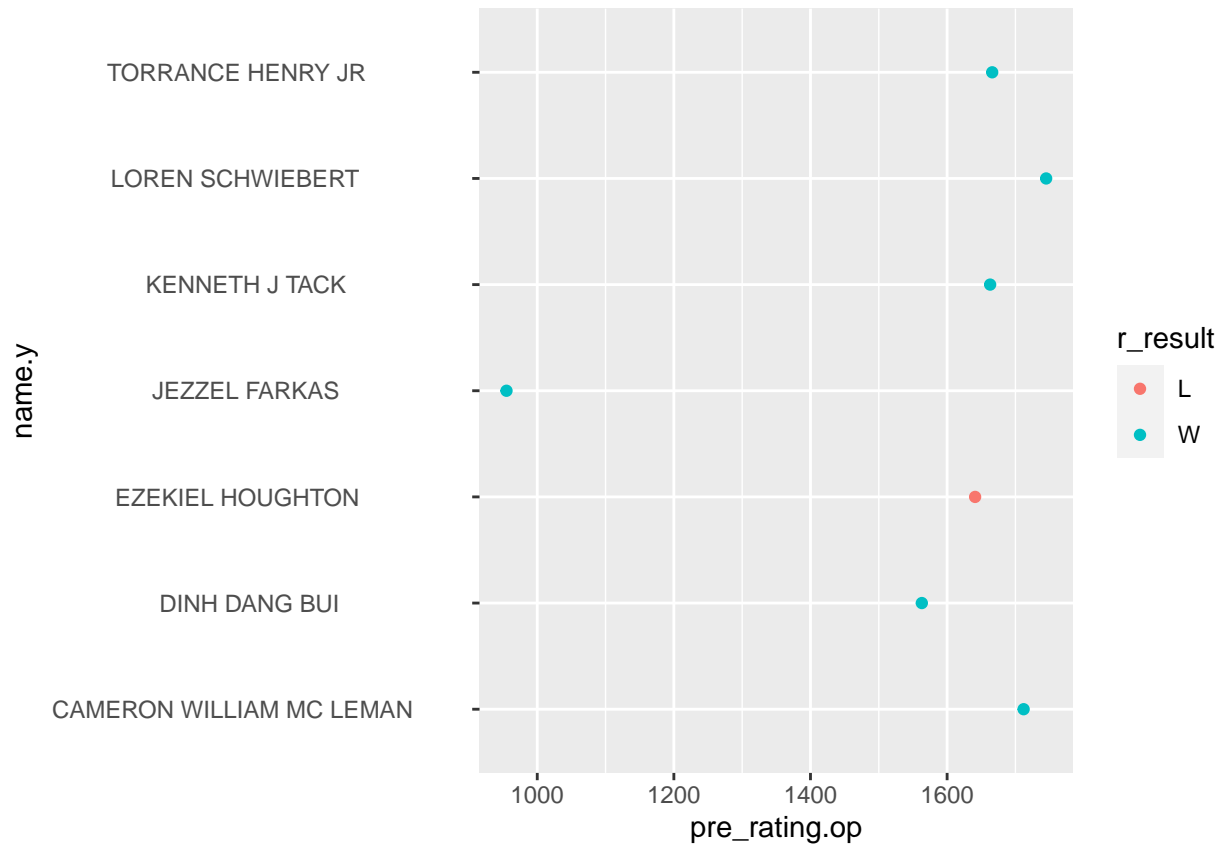
```
eloTxt_comb %>%
  left_join(eloFinal, by = c("op_id" = "id")) %>%
  filter (id == 46) %>%
  select(id, name.x, pre_rating.op, r_result, name.y) %>%
  distinct() %>%
  ggplot(aes(y=name.y, x=pre_rating.op, color = r_result)) + geom_point()
```



3 ADITYA BAJAJ

Aditya performed well against players with higher rankings that accounted for the movement in his ELO score. He had a solid tournament if you calculated his ELO rating based on his tournament performance and the average pre tournament ratings of his competitors his ELO rating would have been 1849 (ignoring his pre tournament ELO rating)

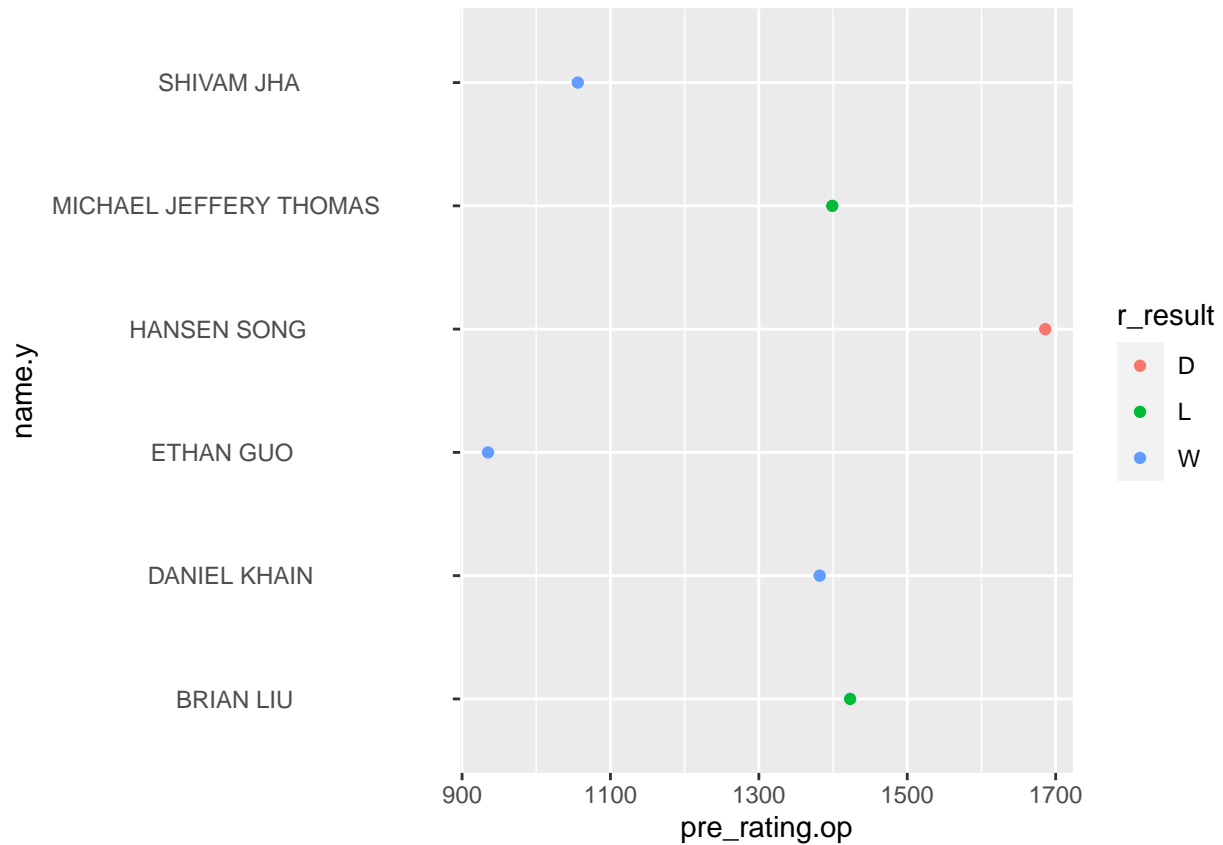
```
eloTxt_comb %>%
  left_join(eloFinal, by = c("op_id" = "id")) %>%
  filter (id == 3) %>%
  select(id, name.x, pre_rating.op, r_result, name.y) %>%
  distinct() %>%
  ggplot(aes(y=name.y, x=pre_rating.op, color = r_result)) + geom_point()
```



29 CHIEDOZIE OKORIE

Chiedoxie lost 2 games. He had a poor tournament if you calculated his ELO rating based on his tournament performance and the average pre tournament ratings of his competitors his ELO rating would have been 1099 (ignoring his pre tournament ELO rating)

```
eloTxt_comb %>%
  left_join(eloFinal, by = c("op_id" = "id")) %>%
  filter (id == 29) %>%
  select(id, name.x, pre_rating.op, r_result, name.y) %>%
  distinct() %>%
  ggplot(aes(y=name.y, x=pre_rating.op, color = r_result)) + geom_point()
```

30 GEORGE AVERY JONES

George lost 2 games to players of with lower ELO rankings. He had a poor tournament if you calculated his ELO rating based on his tournament performance and the average pre tournament ratings of his competitors his ELO rating would have been 978 (ignoring his pre tournament ELO rating)

```
eloTxt_comb %>%
  left_join(eloFinal, by = c("op_id" = "id")) %>%
  filter (id == 30) %>%
  select(id, name.x, pre_rating.op, r_result, name.y) %>%
  distinct() %>%
  ggplot(aes(y=name.y, x=pre_rating.op, color = r_result)) + geom_point()
```

