Model Fitting

Amanda Luby, Shannon Gallagher, Kayla Frisoli

9/10/2018

We initially planned to use points won by an individual as our outcome variable, but found that it may not be an ideal measure of player performance. For instance, players may score few points in a match due to a poor performance, but they may also score few (relative) points if they win a short match. Modeling the number of points won is also complicated by the difference between length of matches for men (best of 5 sets) and women (best of 3 sets). For these reasons, we chose to model whether a player won the match, with the following predictors:

- `ioc_fac`: Country that the player represents
- `tournament`: Which Grand Slam the match was played at
- `late_round`: Indicator for whether the match occured in Round of 16 or later
- `rank`: Rank of player at the time of the match
  - Included on log scale
- `opponent_rank`: Rank of opponent at the time of the match
  - Included on log scale
- `year`: Factor variable with a level for each year included in the dataset (2013-2017)
- `atp`: Indicator that the match was played in the ATP league instead of the WTF league.

We will use a multilevel logistic regression framework. This allows us to include fixed effects, which remain the same for each player, as well as include player-level effects. Since players appear multiple times in the data, the observations are not independent and including a player-level effect allows us to account for this dependence. It also provides a way to examine individual player effects and assess, for instance, whether Nadal is more likely to win at Wimbledon after accounting for other variables. We considered the following models:

- No mixed effects
  - `nocourt_logistic`: $\text{logit}(\pi_i) = \beta X$, X does not include `tournament`
    - If this model fits well, court surface does not have an effect on winning probability
  - `base_logistic`: $\text{logit}(\pi_i) = \beta X$
    - If this model fits better than the mixed-effects models, performance does not depend on individual-level or IOC-level
- IOC-level effect
  - `country_logistic`: $\text{logit}(\pi_i) = \beta X + \beta_{T[i]} T$, $\beta_{T[i]} = \alpha_i C_i$
- Player-level effect
  - `ind_logistic`: $\text{logit}(\pi_i) = \beta X + \beta_{T[i]} T$, $\beta_{T[i]} = \alpha_i P_i$
    - Includes both a fixed-effect term for IOC and a player-level effect for each grand slam
  - `ind_logistic_noioc`: $\text{logit}(\pi_i) = \beta X + \beta_{T[i]} T$, where X does not include IOC and T represents indicators for tournament, $\beta_{T[i]} = \alpha_i P_i$
    - Simpler than the `ind_logistic` model, since we exclude IOC effects.
  - `ind_int_noioc`: $\text{logit}(\pi_i) = \beta X + \beta_{0[i]}$, where X does not include IOC, $\beta_{0[i]} = \alpha_i$
    - Similar to `ind_logistic_noioc`, but includes a single player-level effect rather than a player-level effect for each grand slam.
  - `ind_year_logistic`: $\text{logit}(\pi_i) = \beta X + \beta_{D[i]}$, where X does not include IOC and D represents indicators for years, $\beta_{Y[i]} = \alpha_i Y_i$
    - Rather than including a player-level effect for grand slam, includes a player-level effect for each year in the dataset

```
set.seed(091418)
nocourt_logistic = glm(did_win ~ ioc_fac + late_round + log(rank) + log(opponent_rank) + year_fac + atp
base_logistic = glm(did_win ~ ioc_fac +  tournament + late_round + log(rank) + log(opponent_rank) + yea
country_logistic = glmer(did_win ~ late_round + log(rank) + log(opponent_rank) + year_fac + atp + (0 +
ind_logistic = glmer(did_win ~  ioc_fac + late_round + log(rank) + log(opponent_rank) + year_fac + atp +
ind_logistic_noioc = glmer(did_win ~ late_round + log(rank) + log(opponent_rank) + year_fac + atp + (0 +
ind_int_logistic = glmer(did_win ~ late_round + log(rank) + log(opponent_rank) + year_fac + atp + tourna
ind_year_logistic = glmer(did_win ~ late_round + log(rank) + log(opponent_rank) + atp + tournament + (0
```

Based on AIC, the `ind_logistic_noioc` model performed best after accounting for the number of variables in the model.

```
model_names = c("nocourt_logistic", "base_logistic", "country_logistic", "ind_logistic",
                "ind_logistic_noioc", "ind_int_logistic", "ind_year_logistic")
model.sums = data.frame(model = model_names, aic = rep(NA, length(model_names)), edf = rep(NA,length(mod
model.ests = data.frame()
ran.effects = data.frame()
for(mod in 1:nrow(model.sums)){
  name = model_names[mod]
  model.sums$aic[mod] = extractAIC(get(name))[2]
  model.sums$edf[mod] = extractAIC(get(name))[1]
  betas = tidy(get(name))
  betas$model = name
  if("group" %in% colnames(betas)){
  model.ests = rbind(model.ests, filter(betas, group == "fixed") %>%
    select(., -"group"))
  }
  else model.ests = rbind(model.ests, betas)
}
```

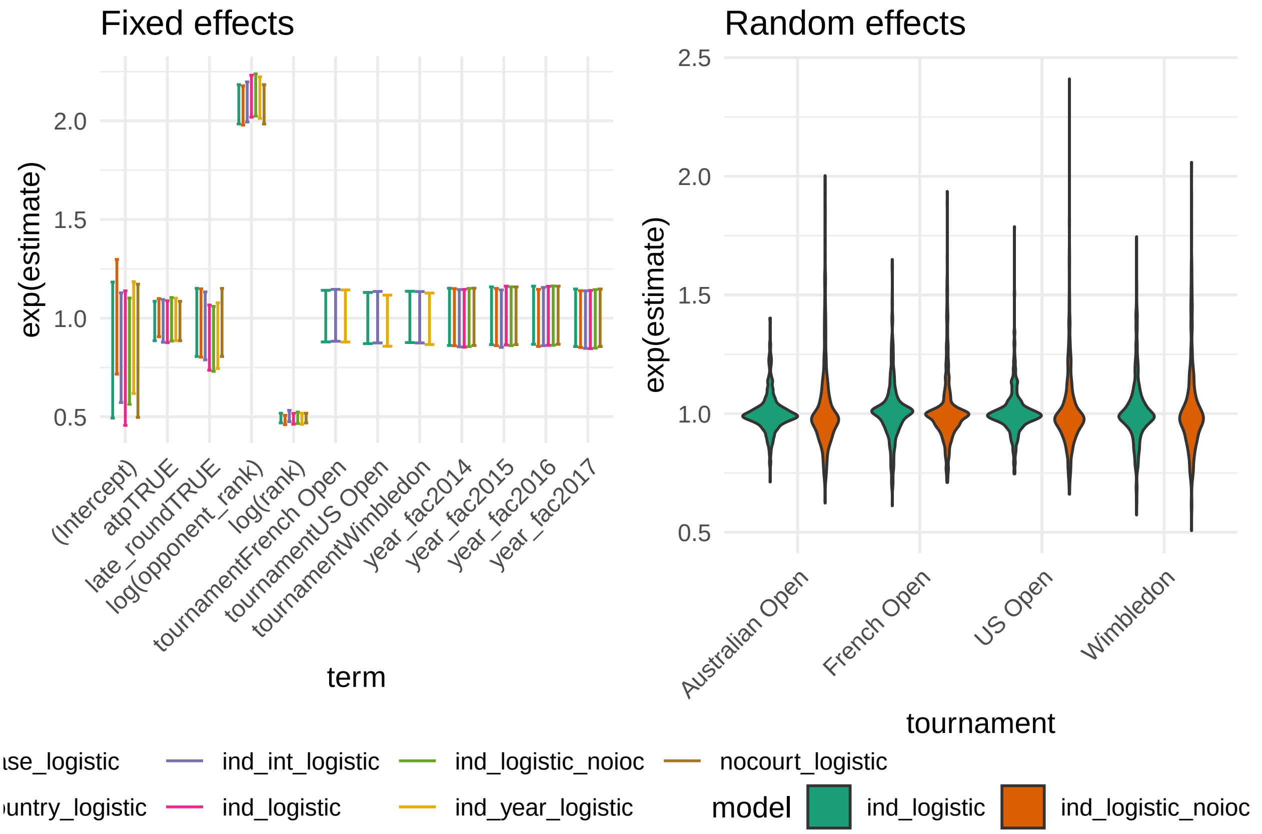
```
## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
kable(model.sums)
```

model	aic	edf
nocourt_logistic	11398.51	69
base_logistic	11404.48	72
country_logistic	11374.86	19
ind_logistic	11395.69	79
ind_logistic_noioc	11354.20	19
ind_int_logistic	11365.07	13
ind_year_logistic	11366.68	23

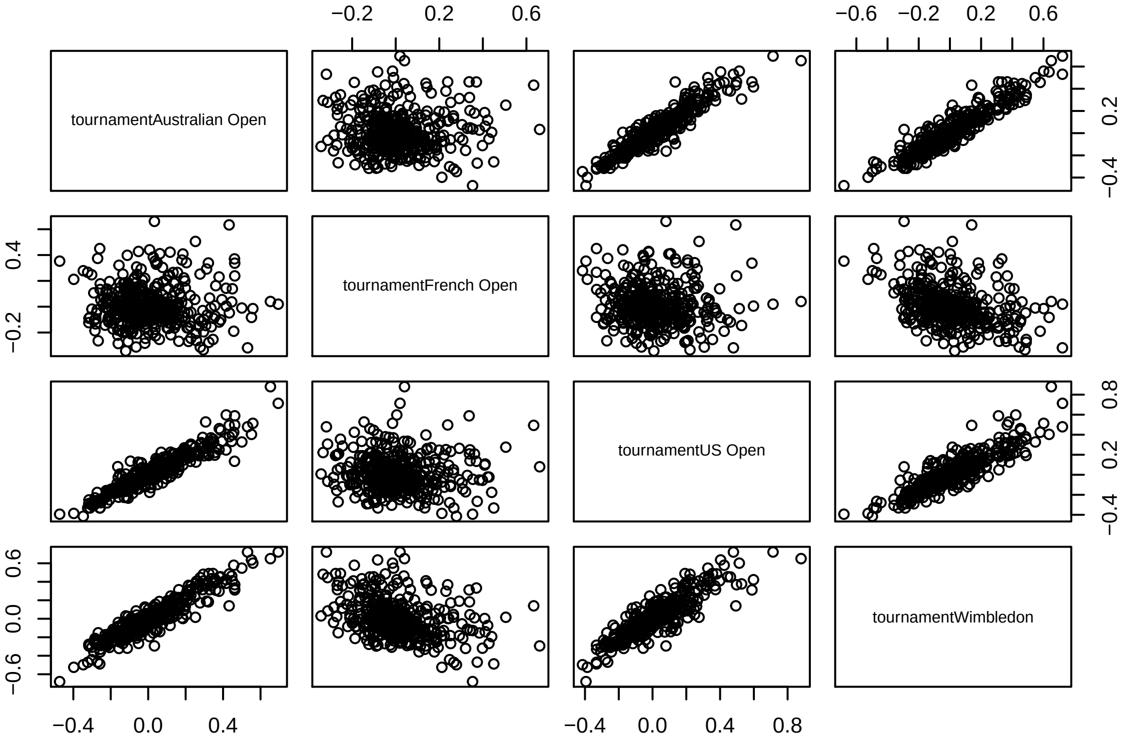
```
fixed.effects.plot = model.ests %>%
  filter(., !grepl("ioc", term)) %>%
  ggplot(., aes(x = term, y = exp(estimate), col = model, ymin = exp(estimate - 2*std.error), ymax = exp
  geom_errorbar(width = .7, position = position_dodge()) +
  theme_minimal() +
  theme(axis.text.x=element_text(angle = 45, hjust = 1)) +
  scale_color_brewer(palette = "Dark2") +
  ggtitle("Fixed effects") +
  theme(legend.position = "bottom")
```

```
ran_tourn = rbind(ranef(ind_logistic)$name_int, ranef(ind_logistic_noioc)$name_int)
ran_tourn$model = c(rep("ind_logistic", nrow(ranef(ind_logistic)$name_int)), rep("ind_logistic_noioc", r

random.effects.plot = ran_tourn %>%
  gather(., tourn, estimate, -model) %>%
  mutate(., tournament = gsub("tournament", "", tourn)) %>%
  ggplot(., aes(x = tournament, y = exp(estimate), fill = model)) +
  geom_violin() +
  theme_minimal() +
  scale_fill_brewer(palette = "Dark2") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ggtitle("Random effects") +
  theme(legend.position = "bottom")

plot_grid(fixed.effects.plot, random.effects.plot)
```



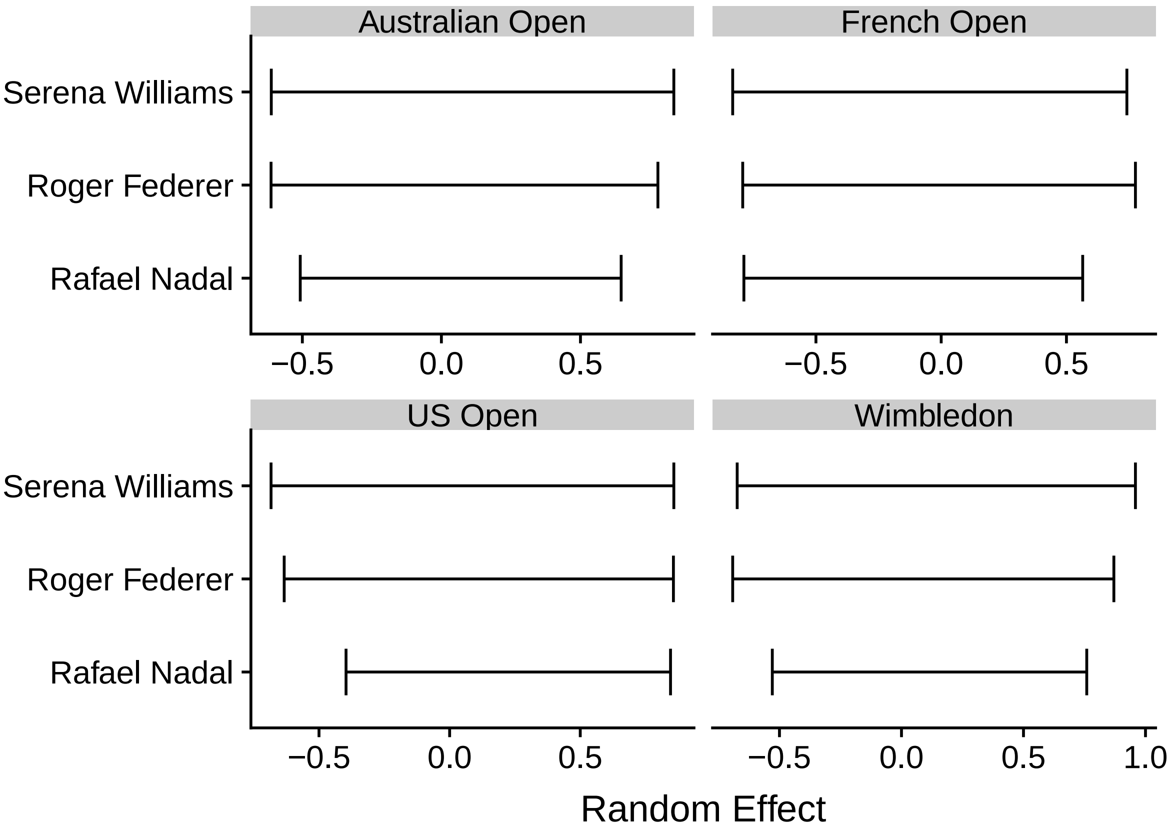
Looking at the `ind_logistic_noioc` random effects more in-depth, we see that the effects for the Austrialian Open, US Open, and Wimbledon are all quite correlated. This suggests that, after accounting for additional variables such as rank and opponent rank, differences in win probability are not detectable between the two types of hard court and grass.

```
pairs(ranef(ind_logistic_noioc)$name_int, pch = 21)
```



If we look at our big three, their individual effects are not very informative (especially after taking uncertainty into consideration).

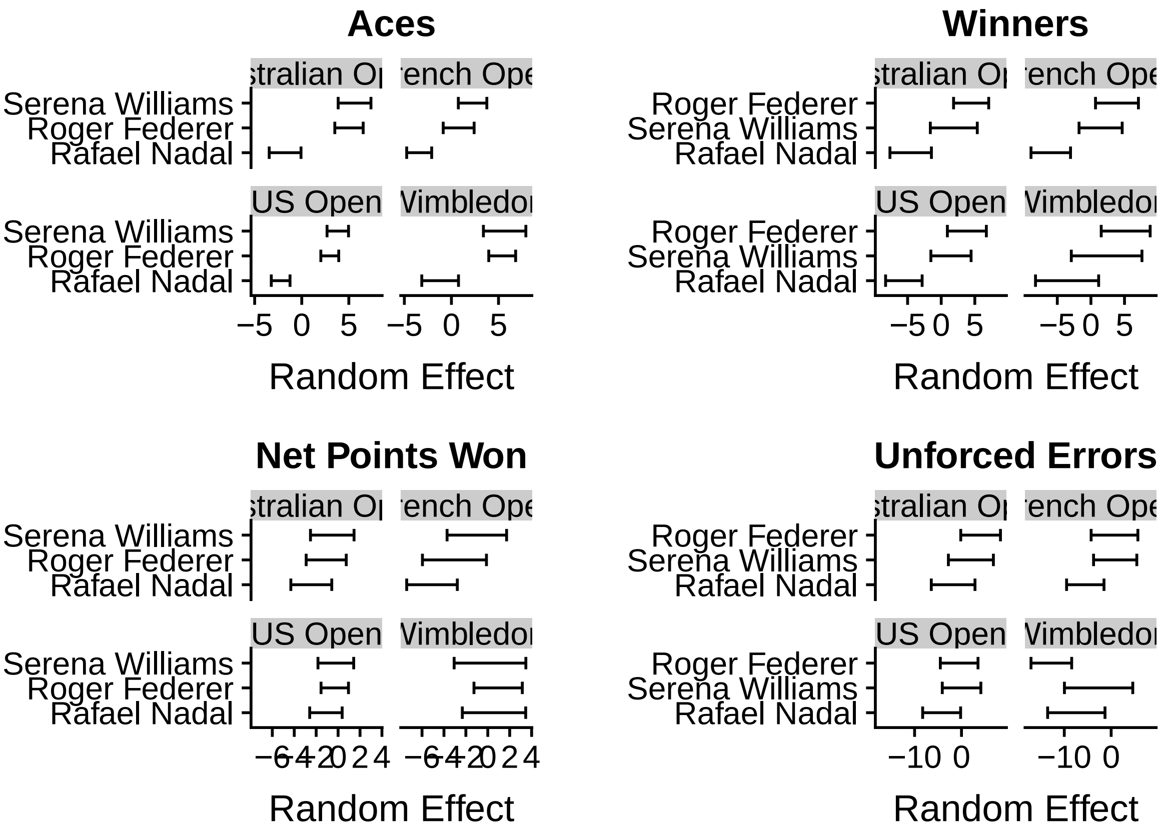
```r
name_lookup = unique(cbind(gs_players$name, gs_players$name_int))
name_lookup = name_lookup[order(as.integer(name_lookup[,2])),]
as.data.frame(ranef(ind_logistic_noioc, condVar = TRUE)) %>%
  mutate(tournament = gsub("tournament", "", term)) %>%
  mutate(name = name_lookup[,1][as.integer(grp)]) %>%
  filter(., name == "Serena Williams" | name == "Rafael Nadal" | name == "Roger Federer") %>%
  ggplot(., aes(y = name, x = condval, xmin = condval- 2*condsd, xmax = condval + 2*condsd)) +
  facet_wrap(~tournament, scales = "free_x") +
  geom_errorbarh(height = .5) +
  xlab("Random Effect") +
  ylab("")
```



Rank and opponent's rank clearly have the largest effect on predicting the winner. Looking at the individual effects for Serena, Nadal, and Federer; they actually fall near the average for all four grand slams. Including rank as an indicator (seeded/not-seeded) substantially decreases the fit of the model, even when individual effects are included. We now look at a similar model, but instead of using 'win' as the outcome variable, we'll look at game statistics that may be more sensitive to court surface – aces, winners, and net points won – and see if individual differences are detectable.

```
load("../../data/gs_partial_players.rda")
gs_partial_players$atp = gs_partial_players$Tour == "atp"
gs_partial_players$year_fac = as.factor(gs_partial_players$year)
gs_partial_players$late_round = gs_partial_players$round >= "R16"
gs_partial_players$seeded = gs_partial_players$rank <= 32
gs_partial_players$opponent_seeded = gs_partial_players$opponent_rank <= 32
n_aces_mod = lmer(n_aces ~ late_round + log(rank) + log(opponent_rank) + year_fac + atp + (0 + tournamen
n_winners_mod = lmer(n_winners ~ late_round + log(rank) + log(opponent_rank) + year_fac + atp + (0 + tou
n_net_mod = lmer(n_netpt_w ~ late_round + log(rank) + log(opponent_rank) + year_fac + atp + (0 + tournam
n_ue_mod = lmer(n_ue ~ late_round + log(rank) + log(opponent_rank) + year_fac + atp + (0 + tournament |

aces_ranef = as.data.frame(ranef(n_aces_mod, condVar = TRUE)) %>%
  mutate(tournament = gsub("tournament", "", term)) %>%
  filter(., grp == "Serena Williams" | grp == "Rafael Nadal" | grp == "Roger Federer") %>%
  ggplot(., aes(y = grp, x = condval, xmin = condval- 2*condsd, xmax = condval + 2*condsd)) +
  facet_wrap(~tournament) +
  geom_errorbarh(height = .5) +
  xlab("Random Effect") +
  ylab("") +
  ggtitle("Aces")
```

```r
winners_ranef = as.data.frame(ranef(n_winners_mod, condVar = TRUE)) %>%
  mutate(tournament = gsub("tournament", "", term)) %>%
  filter(., grp == "Serena Williams" | grp == "Rafael Nadal" | grp == "Roger Federer") %>%
  ggplot(., aes(y = grp, x = condval, xmin = condval- 2*condsd, xmax = condval + 2*condsd)) +
  facet_wrap(~tournament) +
  geom_errorbarh(height = .5) +
  xlab("Random Effect") +
  ylab("") +
  ggtitle("Winners")

nets_ranef = as.data.frame(ranef(n_net_mod, condVar = TRUE)) %>%
  mutate(tournament = gsub("tournament", "", term)) %>%
  filter(., grp == "Serena Williams" | grp == "Rafael Nadal" | grp == "Roger Federer") %>%
  ggplot(., aes(y = grp, x = condval, xmin = condval- 2*condsd, xmax = condval + 2*condsd)) +
  facet_wrap(~tournament) +
  geom_errorbarh(height = .5) +
  xlab("Random Effect") +
  ylab("") +
  ggtitle("Net Points Won")

ue_ranef = as.data.frame(ranef(n_ue_mod, condVar = TRUE)) %>%
  mutate(tournament = gsub("tournament", "", term)) %>%
  filter(., grp == "Serena Williams" | grp == "Rafael Nadal" | grp == "Roger Federer") %>%
  ggplot(., aes(y = grp, x = condval, xmin = condval- 2*condsd, xmax = condval + 2*condsd)) +
  facet_wrap(~tournament) +
  geom_errorbarh(height = .5) +
  xlab("Random Effect") +
  ylab("") +
  ggtitle("Unforced Errors")

plot_grid(aces_ranef, winners_ranef, nets_ranef, ue_ranef)
```


Aces
stralian Op
rench Ope
US Open
Vimbledon
Serena Williams
Roger Federer
Rafael Nadal
−5 0 5
Random Effect
Winners
Roger Federer
Serena Williams
Rafael Nadal
−5 0 5
Random Effect
Net Points Won
−6 −4 −2 0 2 4
Random Effect
Unforced Errors
−10 0
Random Effect