

BST 270: Final Project

Dominic DiSanto

Table of contents

Set-Up	1
Preface	1
“Why Some Tennis Matches Take Forever”	2
Table 1: Events	2
Table 2: Players	4
Summary/Analysis	6
“We Watched 906 Foul Balls To Find Out Where The Most Dangerous Ones Land”	7
Figure 2	7
Summary/Analysis	8
Appendix	9
A: Session Information	9
B: Code	10

Set-Up

Preface

Tables were generated using the `kable` and `kableExtra` packages and dat visualizations using `ggplot2` from the Tidyverse. This document was generated using [Quarto](#) in RStudio. Additional details are available in the Appendix (e.g. package versions, R/RStudio versions, etc.)

Each article has a brief section that displays the original table(s) or figure from the table, my recreation of the same results, and a brief summary section commenting on the reproducibility

“Why Some Tennis Matches Take Forever”

From Carl Bialik’s article [“Why Some Tennis Matches Take Forever”](#), we will recreate his two tables (included below).

Table 1: Events

We will first look at the original table from Bialik’s article:

Surface Speeds				
Average time added per point in men’s tennis				
Fastest tournaments				
	TOURNAMENT	SURFACE	YEARS RUNNING	TIME ADDED
1	Wimbledon	Grass	1991-2014	-2.98 s
2	London Olympics	Grass	2012	-2.62
3	Manchester	Grass	1991-94	-2.33
4	Eastbourne	Grass	2009-14	-1.78
5	Birmingham	Carpet	1991	-1.63
6	Queen’s Club	Grass	1991-2014	-1.50
7	Lyon	Hard	2009	-1.35
8	Las Vegas	Hard	2006-08	-1.32
9	Stockholm Masters	Carpet	1991-94	-0.90
10	Nottingham	Grass	1995-2008	-0.82
Slowest tournaments				
	TOURNAMENT	SURFACE	YEARS RUNNING	TIME ADDED
196	Birmingham	Clay	1994	+4.49 s
197	Oporto	Clay	1995-96	+4.69
198	Genova	Clay	1991-93	+4.71
199	Bologna	Clay	1991-98	+4.74
200	Merano	Clay	1999	+4.93
201	Viña del Mar	Clay	2009	+4.96
202	Florence	Clay	1991-94	+5.08
203	Costa do Sauipe	Clay	2004-11	+5.19
204	Maceio	Clay	1992	+5.28
205	Rio Open	Clay	2015	+5.38
Other notable tournaments				
	TOURNAMENT	SURFACE	YEARS RUNNING	TIME ADDED
21	U.S. Open	Hard	1991-2014	-0.17 s
24	Australian Open	Hard	1991-2015	-0.11
54	Roland Garros	Clay	1991-2014	+0.79

The data used to recreate these tables is located in Bialik’s provided `events.csv` file.


Fastest tournaments				
	TOURNAMENT	SURFACE	YEARS RUNNING	TIME ADDED
1	Wimbledon	Grass	1991-2014	-2.98s
2	London Olympics	Grass	2012	-2.62
3	Manchester	Grass	1991-1994	-2.33
4	Eastbourne	Grass	2009-2014	-1.78
5	Birmingham	Carpet	1991	-1.63
6	Queen's Club	Grass	1991-2014	-1.50
7	Lyon	Hard	2009	-1.35
8	Las Vegas	Hard	2006-2008	-1.32
9	Stockholm Masters	Carpet	1991-1994	-0.90
10	Nottingham	Grass	1995-2008	-0.82

Slowest tournaments				
	TOURNAMENT	SURFACE	YEARS RUNNING	TIME ADDED
196	Birmingham	Clay	1994	4.49
197	Oporto	Clay	1995-1996	4.69
198	Genova	Clay	1991-1993	4.71
199	Bologna	Clay	1991-1998	4.74
200	Merano	Clay	1999	4.93
201	Viña del Mar	Clay	2009	4.96
202	Florence	Clay	1991-1994	5.08
203	Costa Do Sauipe	Clay	2004-2011	5.19
204	Maceio	Clay	1992	5.28
205	Rio Open	Clay	2015	5.38s

Other notable tournaments				
	TOURNAMENT	SURFACE	YEARS RUNNING	TIME ADDED
21	US Open	Hard	1991-2014	-0.17s
24	Australian Open	Hard	1991-2015	-0.11
54	Roland Garros	Clay	1991-2014	0.79

Table 2: Players

Again, we first we can review the original table(s) presented in the article:

Player Speeds		
Average time added per point in men's tennis		
Fastest players		Slowest players
1	Dustin Brown	- 6 . 37 s
2	Rohan Bopanna	- 4 . 95
3	Chris Guccione	- 4 . 63
4	Benoit Paire	- 4 . 56
5	Lukas Dlouhy	- 4 . 35
6	Brendan Evans	- 4 . 25
7	Igor Sijsling	- 4 . 19
8	Lukas Rosol	- 4 . 13
9	Alexander Kudryavtsev	- 4 . 05
10	Sam Querrey	- 3 . 99
Other notable players		
22	Goran Ivanisevic	- 3 . 15 s
36	Roger Federer	- 2 . 43
124	Novak Djokovic	+ 2 . 21
141	Andy Murray	+ 2 . 53
191	Pat Cash	+ 3 . 73
202	Ivan Lendl	+ 4 . 35
203	Jim Courier	+ 4 . 51
207	Jimmy Connors	+ 4 . 90
 FIVETHIRTYEIGHT		BASED ON DATA FROM JEFF SACKMANN

Now we can begin attempting to recreate this table, doing our best to mirror the format of the 538 results shown above in structure/formatting.

We generate each of the three tables included in the image above, using the `players_time.csv` table. Player rank (the integer column) was generated as noted below, player name taken from the `player` column, and the added time from the `seconds_added_per_point` column:

Fastest players			Slowest players		
1	Dustin Brown	-6.37s	209	Michael Chang	4.94s
2	Rohan Bopanna	-4.95	210	Joao Cunha Silva	5.10
3	Chris Guccione	-4.63	211	Julian Knowle	5.15
4	Benoit Paire	-4.56	212	John McEnroe	5.22
5	Lukas Dlouhy	-4.35	213	Lucas Arnold Ker	5.35
6	Brendan Evans	-4.25	214	T.J. Middleton	5.60
7	Igor Sijsling	-4.19	215	Martin Stringari	5.68
8	Lukas Rosol	-4.13	216	Rafael Nadal	5.92
9	Alexander Kudryavtsev	-4.05	217	Nicolas Massu	6.21
10	Sam Querrey	-3.99	218	Emanuel Couto	6.35

Other notable players		
22	Goran Ivanisevic	-3.15s
36	Roger Federer	-2.43
124	Novak Djokovic	2.21
141	Andy Murray	2.53
191	Pat Cash	3.73
202	Ivan Lendl	4.35
203	Jim Courier	4.51
207	Jimmy Connors	4.90

We see our results mirror those in Bialik's original article exactly, and we were able to easily identify and match the data to recreate these tables from the provided GitHub repository (even without any code given by the author).

Summary/Analysis

For Bialik's article on the pace of play in modern tennis, the data for the presented tables was immediately available and well-formatted to corroborate these results and easily output tables that are similar to those in Bialik's original article. The data used was sparse (with no additional information), which necessitated little documentation.

However considering the larger article, the two figures are presented without sufficient data to re-created. The provided data does not include any temporal data on player tendencies for the second figure, and the events data has been aggregated so that annual data is not available to create the first figure. The article also discusses regression modelling for pace of play but with no presentation of results or sharing of code to use in evaluating the model.

As an entertainment article, we obviously would not expect the same level of rigor for reproducibility as we might expect for a peer-reviewed publication or scientific article. The visualization and modelling are presented with no available data or thorough discussion of the methods to generate the results, while the tables in Bialik's article are easily reproduced from the shared materials.

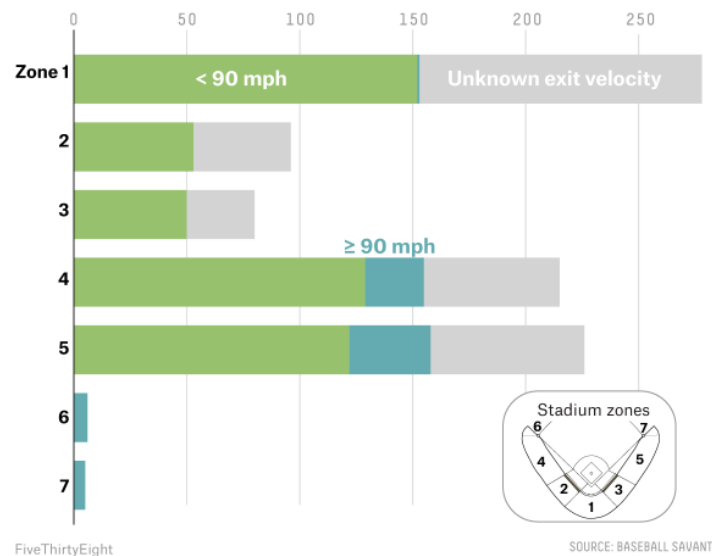
“We Watched 906 Foul Balls To Find Out Where The Most Dangerous Ones Land”

Annette Choi's article

Figure 2

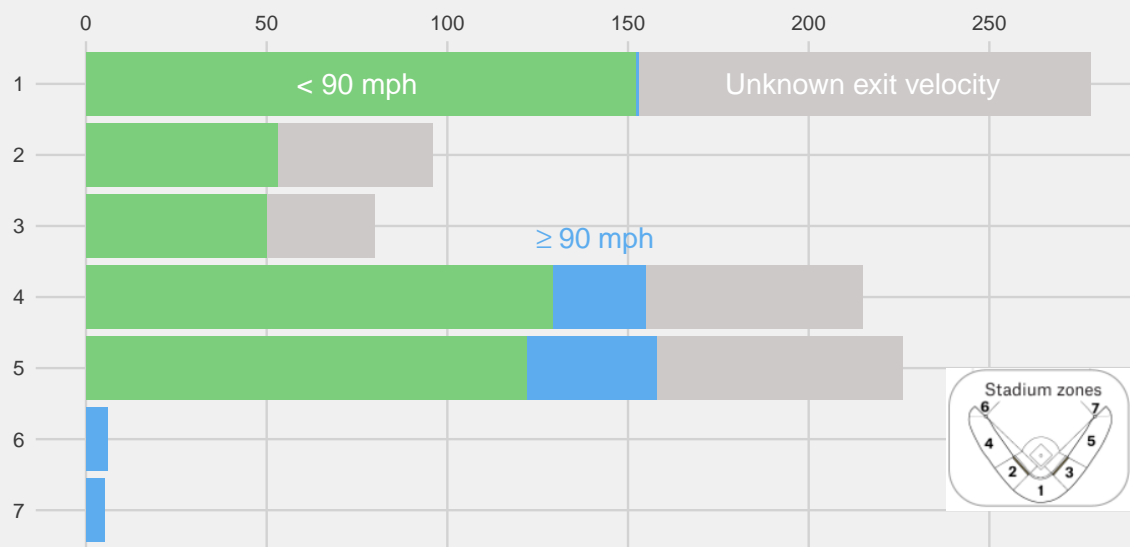
The hardest-hit fouls seem to land in unprotected areas

Foul balls by the stadium zone they landed in and their exit velocity, among 906 fouls hit this season in the most foul-heavy day at the 10 MLB stadiums that produced the most fouls as of June 5



The hardest hit foul-balls seem to land in unprotected areas

Foul balls by the stadium zone they landed in and their exit velocity, among 906 foul balls hit this season in the most foul-heavy day at the 10 MLB stadiums that produced the most fouls as of June 5



Summary/Analysis

Appendix

A: Session Information

R version 4.2.1 (2022-06-23 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22621)

Matrix products: default

locale:

[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:

[1] grid stats graphics grDevices utils datasets methods
[8] base

other attached packages:

[1] png_0.1-7 gtExtras_0.4.5 gt_0.8.0 knitr_1.40
[5] kableExtra_1.3.4 stringr_1.4.1 dplyr_1.0.10 ggplot2_3.4.0
[9] here_1.0.1

loaded via a namespace (and not attached):

[1] tidyselect_1.2.0 xfun_0.33 purrr_0.3.5 rematch2_2.1.2
[5] ggthemes_4.2.4 paletteer_1.5.0 V8_4.2.2 colorspace_2.0-3
[9] vctrs_0.5.1 generics_0.1.3 htmltools_0.5.3 viridisLite_0.4.1
[13] yaml_2.3.5 utf8_1.2.2 rlang_1.0.6 pillar_1.8.1
[17] glue_1.6.2 juicyjuice_0.1.0 withr_2.5.0 DBI_1.1.3
[21] lifecycle_1.0.3 commonmark_1.8.1 munsell_0.5.0 gtable_0.3.1
[25] rvest_1.0.3 fontawesome_0.4.0 evaluate_0.17 fastmap_1.1.0
[29] curl_4.3.3 fansi_1.0.3 Rcpp_1.0.9 scales_1.2.1
[33] webshot_0.5.4 jsonlite_1.8.2 farver_2.1.1 systemfonts_1.0.4
[37] digest_0.6.29 stringi_1.7.8 rprojroot_2.0.3 cli_3.4.1
[41] tools_4.2.1 magrittr_2.0.3 sass_0.4.2 tibble_3.1.8
[45] pkgconfig_2.0.3 xml2_1.3.3 assertthat_0.2.1 rmarkdown_2.17
[49] svglite_2.1.0 httr_1.4.4 rstudioapi_0.14 R6_2.5.1
[53] compiler_4.2.1

B: Code

```
events <- read.csv(here("Data", "tennis-time", "events_time.csv"), header=T) %>%
  mutate(Rank=row_number())

players <- read.csv(here("Data", "tennis-time", "players_time.csv")) %>%
  mutate(Rank=row_number()) # adding the "Rank" variable (the integers in each table above)
#####
### Table 1 ###
#####

knitr::include_graphics(here("Images", "tennis-time", "bialik-tennis-time-table21.png"))

events_10f <- events %>%
  arrange(seconds_added_per_point) %>%
  head(10) %>%
  mutate(seconds = # reformatting as string to include the "s" for the first observation,
           case_when(row_number()==1 ~ paste0(sprintf("%0.2f", seconds_added_per_point)
                                                    , "s")
                     , T ~ sprintf("%0.2f", seconds_added_per_point)
                     )
           ) %>%
  select(Rank, tournament, surface, years, seconds)

events_10s <- events %>%
  mutate(Rank=row_number()) %>%
  arrange(desc(seconds_added_per_point)) %>%
  head(10) %>%
  mutate(seconds = # reformatting as string to include the "s" for the first observation,
           case_when(row_number()==1 ~ paste0(sprintf("%0.2f", seconds_added_per_point)
                                                    , "s")
                     , T ~ sprintf("%0.2f", seconds_added_per_point)
                     )
           ) %>%
  arrange(seconds_added_per_point) %>%
  select(Rank, tournament, surface, years, seconds)

events_ntbl <- events %>%
  filter(str_detect(tournament, "US Op|Australian Open|Roland Garr")) %>%
  mutate(seconds = # reformatting as string to include the "s" for the first observation,
           case_when(row_number()==1 ~ paste0(sprintf("%0.2f", seconds_added_per_point)
```

```

                                , "s")
                                , T ~ sprintf("%.2f", seconds_added_per_point)
                                )
                                ) %>%
select(Rank, tournament, surface, years, seconds)

event_cols <- c("", "TOURNAMENT", "SURFACE", "YEARS RUNNING", "TIME ADDED")

events_10f %>%
  kable(col.names=event_cols, booktabs=T) %>%
  kable_styling(latex_options = c("striped", "HOLD_position")) %>%
  add_header_above(c("Fastest tournaments"=5)) %>%
  row_spec(0, align="l")
events_10s %>%
  kable(col.names=event_cols, booktabs=T) %>%
  kable_styling(latex_options = c("striped", "HOLD_position")) %>%
  add_header_above(c("Slowest tournaments"=5)) %>%
  row_spec(0, align="l")
events_ntbl %>%
  kable(col.names=event_cols, booktabs=T) %>%
  kable_styling(latex_options = c("striped", "HOLD_position")) %>%
  add_header_above(c("Other notable tournaments"=5)) %>%
  row_spec(0, align="l")
#####
### Table 2 ###
#####

knitr::include_graphics(here("Images", "tennis-time", "bialik-tennis-time-table1.png"))

# Top 10 Fastest Players (lower [or more negative] seconds_added_per_point is faster pace
t10_f <- players %>%
  arrange(seconds_added_per_point) %>% # sort by ascending order (lower seconds added = fa
  head(10) %>% # top 10
  mutate(seconds = # reformatting as string to include the "s" for the first observation,
    case_when(row_number()==1 ~ paste0(sprintf("%.2f", seconds_added_per_point)
                                          , "s")
              , T ~ sprintf("%.2f", seconds_added_per_point)
              )
    ) %>%
  select(Rank, player, seconds)

```

```

# "Top" 10 slowest players (largest values)
t10_s <- players %>%
  arrange(desc(seconds_added_per_point)) %>%
  head(10) %>%
  arrange(seconds_added_per_point) %>%
  mutate(seconds =
    case_when(row_number()==1 ~ paste0(sprintf("%.2f", seconds_added_per_point), "s"),
              , T ~ sprintf("%.2f", seconds_added_per_point))
  ) %>%
  select(Rank, player, seconds)

# Specific players ("notable") taken from the full data set
tbl_pls <- players %>%
  filter(str_detect(player, "Goran|Roger Fed|Novak Djok|Andy Murray|Pat Cash|Ivan Lend|Jim"))
  arrange(seconds_added_per_point) %>%
  mutate(seconds =
    case_when(row_number()==1 ~ paste0(sprintf("%.2f", seconds_added_per_point), "s"),
              , T ~ sprintf("%.2f", seconds_added_per_point))
  ) %>%
  select(Rank, player, seconds)

t10_f %>%
  kable(col.names=NULL, booktabs=T) %>%
  kable_styling(latex_options = c("striped", "HOLD_position")) %>%
  add_header_above(c("Fastest players"=3)) %>%
  row_spec(0, align="l")

t10_s %>%
  kable(col.names=NULL, booktabs=T) %>%
  kable_styling(latex_options = c("striped", "HOLD_position")) %>%
  add_header_above(c("Slowest players"=3)) %>%
  row_spec(0, align="l")
tbl_pls %>%
  kable(col.names=NULL, booktabs=T) %>%
  kable_styling(latex_options = c("striped", "HOLD_position")) %>%
  add_header_above(c("Other notable players"=3)) %>%

```

```

  row_spec(0, align="l")
# Initially used the gt package to create prettier HTML tables
# The assignment prompt asks for a PDF, so keeping this code for reference
# but not including it in the final, knitted document

library(gt)
library(gtExtras)

gt_fast <- t10_f %>%
  gt() %>%
  gt::tab_header("Fastest players") %>%
  gt::tab_options(column_labels.hidden = T) %>%
  gt_theme_538() %>%
  gt::as_raw_html()

gt_slow <- t10_s %>%
  gt() %>%
  gt::tab_header("Slowest players") %>%
  gt::tab_options(column_labels.hidden = T) %>%
  gt_theme_538() %>%
  gt::as_raw_html()

gt_ntbl <- tbl_pls %>%
  gt() %>%
  gt::tab_header("Other notable players") %>%
  gt::tab_options(column_labels.hidden = T) %>%
  gt_theme_538() %>%
  gt::as_raw_html()

rbind(data.frame(t1=gt_fast, t2=gt_slow)
      , data.frame(t1=gt_ntbl, t2="")) %>%
# data.frame(gtf=c(gt_fast, gt_ntbl), gts=c(gt_slow, '')) %>%
  gt() %>%
  gt::tab_header(title = "Player Speeds"
                , subtitle="Average time added per point in men's tennis"
                ) %>%
  opt_align_table_header(align = "left") %>%
  fmt_markdown(columns=everything()) %>%
  gt::tab_options(column_labels.hidden=T) %>%
  gtExtras::gt_theme_538() %>%

```

```

gt::tab_style(style = list(cell_text(weight="bold"))
              , location=cells_body(columns=1) )
knitr::include_graphics(here("Images", "foul-balls", "choi-foul-0625-2-1.png"))
foul_balls <- read.csv(here("Data", "foul-balls", "foul-balls.csv"))
library(png); library(grid)

img <- readPNG(here("Images", "foul-balls", "stadium_zones.png"))
g <- rasterGrob(img, interpolate=TRUE)

foul_balls %>%
  mutate(Velocity_Cat =
    case_when(exit_velocity<90 ~ 1
              , exit_velocity>=90 ~ 2
              , TRUE ~ 3)) %>%
  mutate(fvc = factor(Velocity_Cat, levels=c(3, 2, 1))) %>%
  ggplot(aes(x=reorder(as.factor(used_zone), -used_zone), fill=fvc)) +
  geom_bar() +
  scale_y_continuous(breaks=seq(0, 250, 50)
                    , position="right") +
  scale_fill_manual(values=c("snow3", "steelblue2", "palegreen3")) +
  coord_flip() +
  ggthemes::theme_fivethirtyeight() +
  ggtitle(label="The hardest hit foul-balls seem to land in unprotected areas"
          , subtitle = "Foul balls by the stadium zone they landed in and their exit velocity")
  theme(legend.position="none", plot.title = element_text(size=16)) +
  annotation_custom(g, xmin=1, xmax=3, ymin=210, ymax=320) +
  annotate("text", x = 7, y = 75, label="< 90 mph", col="white", size=5) +
  annotate("text", x = 7, y = 215, label="Unknown exit velocity", col="white", size=5) +
  annotate("text", x = 4.8, y = 140, label="''>= 90 ~ 'mph'", col="steelblue2", size=5,

sessionInfo()

```