Duke Sports Analytics Club: Technical Workshop

Jack Lichtenstein

2021-04-19

Loading libraries

```
library(tidyverse)
theme_set(theme_light()) # setting a theme
```

nflfastR

Let's take a look at NFL play by play data using the `nflfastR` package!

```
# load package
library(nflfastR)

# load in some play-by-play data
nfl_pbp <- nflfastR::load_pbp(seasons = 2014:2020)
```

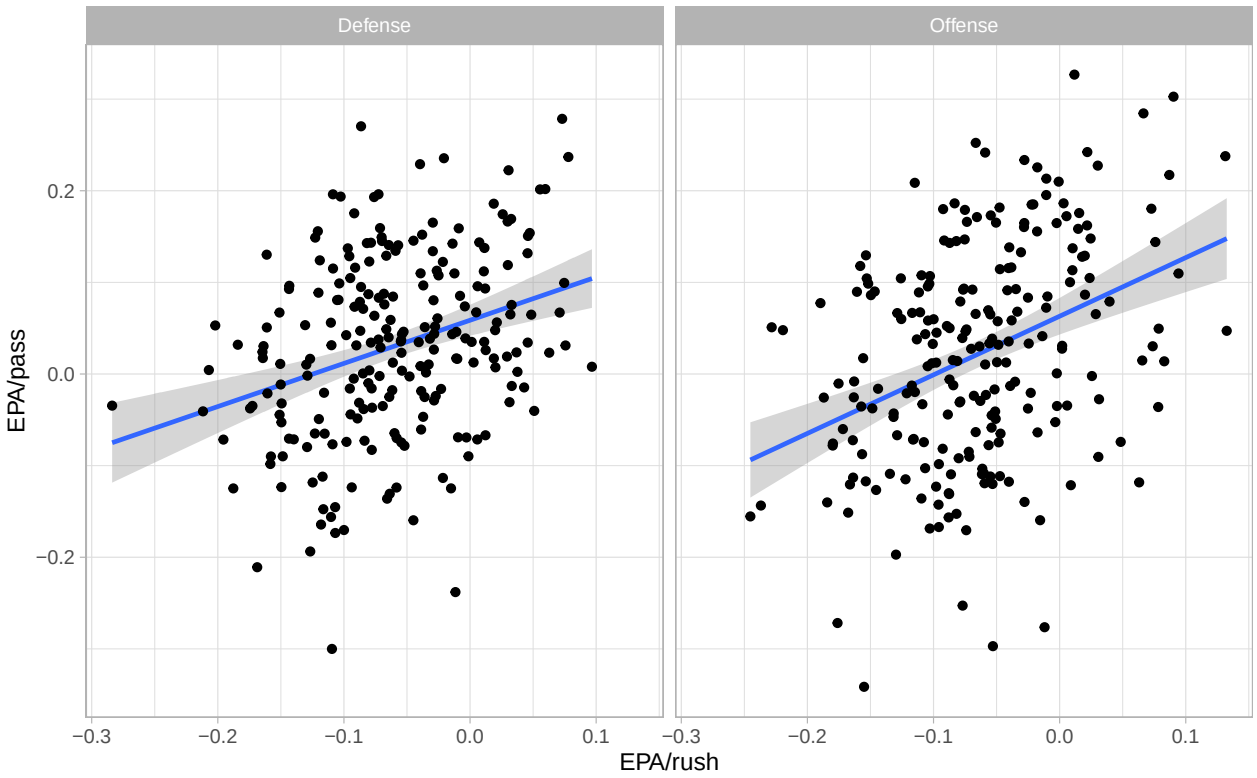
```
# find offensive EPA by team for a given season
off_epa <- nfl_pbp %>%
  filter(
    !is.na(posteam),
    !is.na(epa),
    !is.na(down),
    penalty == 0,
    play_type %in% c("pass", "run")
  ) %>%
  group_by(season, posteam) %>%
  summarise(
    plays = n(),
    passes = sum(play_type == "pass"),
    rushes = sum(play_type == "run"),
    epa_pass = mean(epa[play_type == "pass"]),
    epa_run = mean(epa[play_type == "run"])
  )

# find defensive EPA by team for a given season
def_epa <- nfl_pbp %>%
  filter(
    !is.na(posteam),
    !is.na(epa),
    !is.na(down),
```

```
    penalty == 0,
    play_type %in% c("pass", "run")
  ) %>%
  group_by(season, defteam) %>%
  summarise(
    plays = n(),
    passes = sum(play_type == "pass"),
    rushes = sum(play_type == "run"),
    epa_pass = mean(epa[play_type == "pass"]),
    epa_run = mean(epa[play_type == "run"])
  )

# bind together
epa <- bind_rows(
  off_epa %>%
    mutate(type = "Offense") %>%
    rename(team = posteam),
  def_epa %>%
    mutate(type = "Defense") %>%
    rename(team = defteam)
)
```

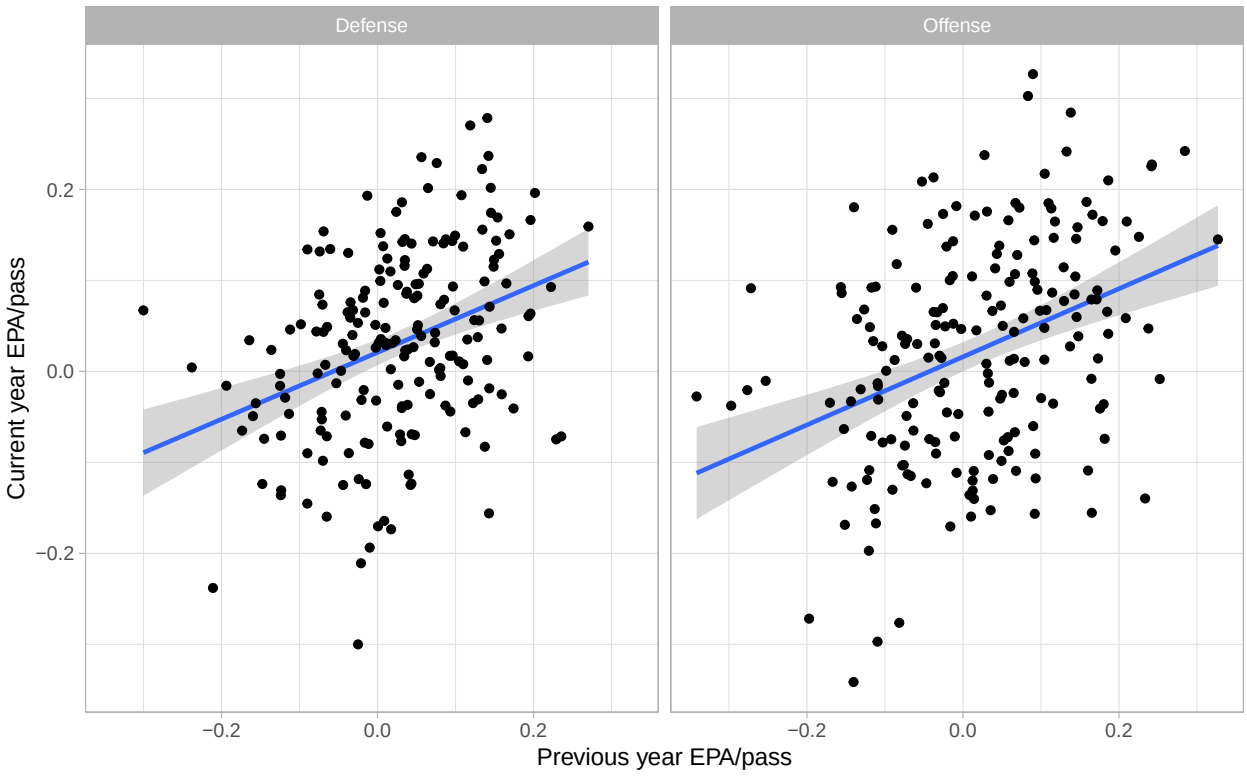
```
# how are offensive/defensive EPA rush/pass related
# are good offenses good at both rush/pass?
# are good defenses good at both rush/pass?
epa %>%
  ggplot(aes(epa_run, epa_pass)) +
  geom_smooth(method = "lm") +
  geom_point() +
  facet_wrap(~type) +
  labs(
    x = "EPA/rush",
    y = "EPA/pass"
  )
```



Let's take a look at the stability of EPA from year to year, both on offense and defense as well as by rush and pass.

```
lagged_epa <- epa %>%
  group_by(team, type) %>%
  # find previous year epa
  mutate(
    prev_epa_pass = lag(epa_pass),
    prev_epa_run = lag(epa_run)
  ) %>%
  ungroup()

lagged_epa %>%
  ggplot(aes(prev_epa_pass, epa_pass)) +
  geom_smooth(method = "lm") +
  geom_point() +
  facet_wrap(~type) +
  labs(
    x = "Previous year EPA/pass",
    y = "Current year EPA/pass"
  )
```




```
# 0.138 r-squared
# pass offense from year-to-year is stable!
lagged_epa %>%
  filter(type == "Offense") %>%
  lm(epa_pass ~ prev_epa_pass, data = .) %>%
  broom::glance() %>%
  pull(r.squared)
```

```
## [1] 0.13827
```

```
# look at stability of all metrics
lagged_epa %>%
  select(-c(plays:rushes)) %>%
  pivot_longer(
    cols = c(epa_pass, epa_run),
    names_to = "current",
    names_prefix = "epa_"
  ) %>%
  pivot_longer(
    cols = c(prev_epa_pass, prev_epa_run),
    names_to = "prev",
    values_to = "prev_value",
    names_prefix = "prev_epa_"
  ) %>%
  # filter(name == prev) %>%
  # filter(name == "pass") %>%
  # lm(value ~ prev_value, data = .) %>%
  # broom::glance()
```

```
  nest(data = c(season, team, value, prev_value)) %>%
  mutate(
    lm = map(data, ~ lm(value ~ prev_value, data = .)),
    glanced = map(lm, broom::glance)
  ) %>%
  hoist(glanced,
    r_2 = "r.squared",
    p_value = "p.value"
  ) %>%
  transmute(type,
    target = current,
    predictor = paste0("Previous year ", prev),
    r_2, p_value
  ) %>%
  mutate(across(target:predictor, ~ paste0(., " EPA"))) %>%
  arrange(desc(r_2)) %>%
  kableExtra::kable(format = "markdown")
```

type	target	predictor	r_2	p_value
Offense	pass EPA	Previous year pass EPA	0.1382700	0.0000001
Defense	pass EPA	Previous year pass EPA	0.1246626	0.0000005
Offense	run EPA	Previous year run EPA	0.1210136	0.0000008
Defense	run EPA	Previous year run EPA	0.0585490	0.0007214
Defense	pass EPA	Previous year run EPA	0.0543926	0.0011318
Offense	pass EPA	Previous year run EPA	0.0460221	0.0028078
Offense	run EPA	Previous year pass EPA	0.0160587	0.0798553
Defense	run EPA	Previous year pass EPA	0.0087545	0.1967552

Passing offense is most stable year to year! This is followed by pass defense

Web scraping

Let's try some web-scraping! We will look to scrape NFL draft big boards for the upcoming 2021 NFL Draft.

- CBS
- Walter Football
- Drafttek

```
# load library
library(rvest)

# CBS
url <- "https://www.cbssports.com/nfl/draft/prospect-rankings/"

html <- read_html(url)

# html_table is great
cbs <- html %>%
  html_table() %>%
  pluck(1) %>%
```

```
  janitor::clean_names() %>%
  filter(row_number() %% 2 == 1) %>%
  select(-c(x:x_2)) %>%
  mutate(across(
    c(rk, pos_rk, wt),
    as.numeric
  )) %>%
  rename(
    ovr_rk = rk,
    player_name = player
  )
```

```
# Walter football
url <- "https://walterfootball.com/nfldraftbigboard/"

html <- read_html(url)

# pull out the text
text <- html %>%
  html_elements(".divPlayerRanking") %>%
  html_text() %>%
  str_squish()

# put text in tibble and extract information
wf <- tibble(text = text) %>%
  mutate(
    ovr_rk = str_extract(text, "\\d{1,3}\\."),
    ovr_rk = readr::parse_number(ovr_rk)
  ) %>%
  filter(!is.na(ovr_rk)) %>%
  tidyr::separate(text,
    into = c(
      "player_name",
      "pos_abbr",
      "rest"
    ),
    sep = ", ",
    extra = "merge"
  ) %>%
  mutate(player_name = str_remove(player_name, "\\d{1,3}\\. "))
```

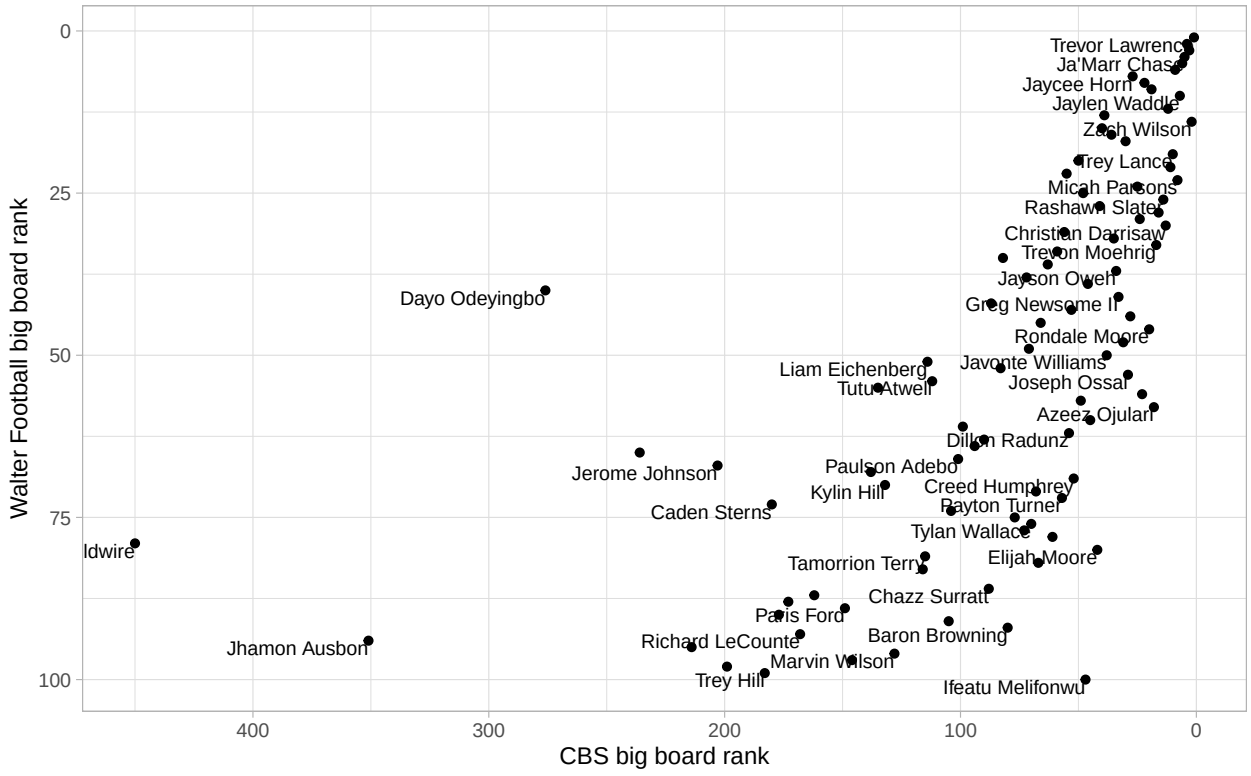
Now that we have two big boards, let's compare!

```
consensus_bb <- bind_rows(
  cbs %>% mutate(type = "CBS"),
  wf %>% mutate(type = "WF")
)

# join in to compare ranks
joined_bb <- cbs %>%
  left_join(wf,
    by = "player_name",
    suffix = c("_cbs", "_wf")
```

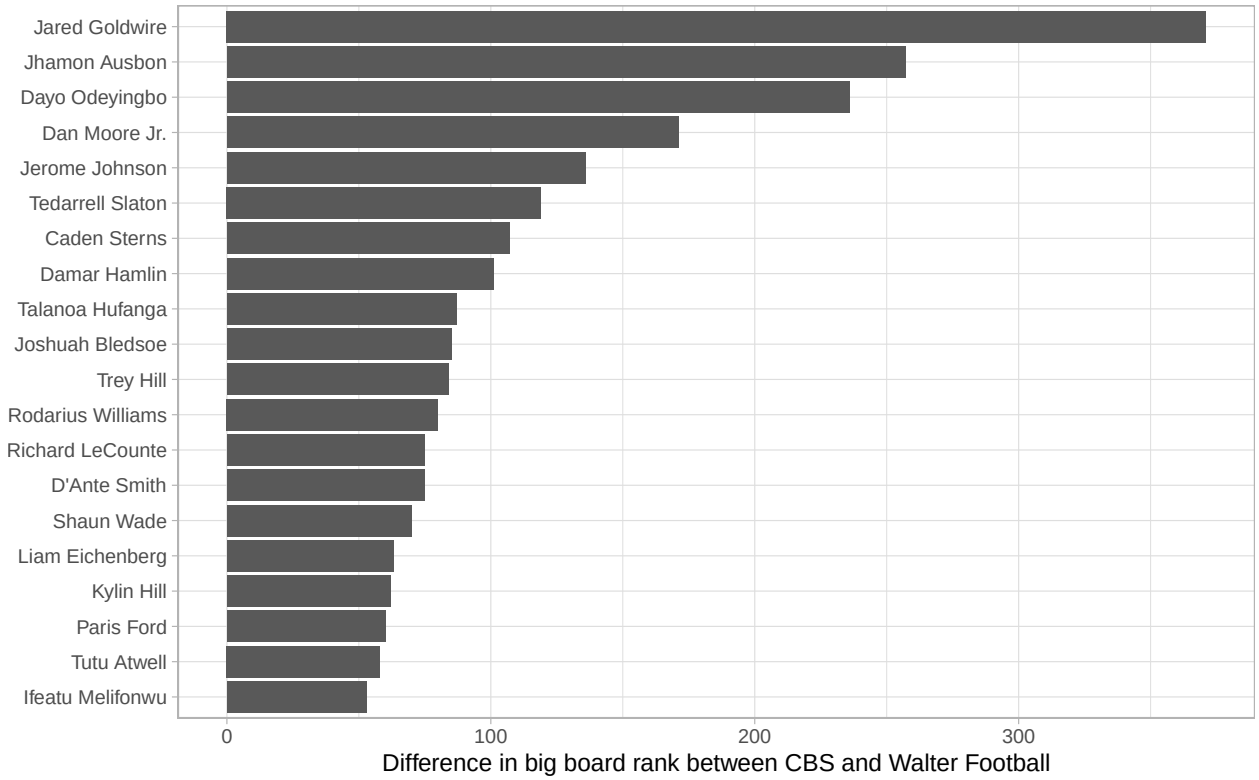
```
  )

# how do these big boards compare?
joined_bb %>%
  ggplot(aes(ovr_rk_cbs, ovr_rk_wf)) +
  geom_text(aes(label = player_name),
    check_overlap = T, size = 3,
    hjust = 1, vjust = 1
  ) +
  geom_point() +
  scale_x_reverse() +
  scale_y_reverse() +
  labs(
    x = "CBS big board rank",
    y = "Walter Football big board rank"
  )
```




```
# who has the largest difference in rank?
joined_bb %>%
  mutate(diff_rk = abs(ovr_rk_cbs - ovr_rk_wf)) %>%
  arrange(desc(diff_rk)) %>%
  head(20) %>%
  select(player_name, ovr_rk_cbs, ovr_rk_wf, diff_rk) %>%
  mutate(player_name = fct_reorder(player_name, diff_rk)) %>%
  ggplot(aes(diff_rk, player_name)) +
  geom_col() +
  labs(
    x = "Difference in big board rank between CBS and Walter Football",
```

```
    y = NULL
  )
```



Jared Goldwire has an enormous difference in big board ranks!

College basketball data sources

Let's explore a little bit of college basketball!

```
# ncaahoopR and kenpomR are both wrappers for ESPN
# play-by-play
library(ncaahoopR)
library(kenpomR)

# these didn't work for some reason
# game_ids <- ncaahoopR::get_game_ids("Duke", season = 2020)
# espn_schedule <- kenpomR::cbb_espn_scoreboard(2020)

# play-by-play of Duke vs. Louisville
espn_pbp <- kenpomR::cbb_espn_game_all("401300969")

head(espn_pbp$Plays)
```

```
## # A tibble: 6 x 19
##   shootingPlay sequenceNumber homeScore scoringPlay awayScore id      text
##   <lgl>        <chr>              <int> <lgl>           <int> <chr>   <chr>
## 1 FALSE        101799901              0 FALSE               0 401300~ Jump Ball~
```

```
## 2 TRUE         101804301              0 FALSE               0 401300~ DJ Stewar~
## 3 FALSE        101804302              0 FALSE               0 401300~ Samuell W~
## 4 TRUE         101806201              0 FALSE               0 401300~ Jae'Lyn W~
## 5 FALSE        101806202              0 FALSE               0 401300~ Wendell M~
## 6 TRUE         101807201              0 TRUE                3 401300~ Matthew H~
## # ... with 12 more variables: scoreValue <int>, period.displayValue <chr>,
## #   period.number <int>, coordinate.x <int>, coordinate.y <int>,
## #   clock.displayValue <chr>, team.id <chr>, type.id <chr>, type.text <chr>,
## #   play.id <chr>, athlete1.id <chr>, athlete2.id <chr>
```

How about gamezoneR? Can find it here: https://jacklich10.github.io/gamezoneR/index.html

```
library(gamezoneR)

schedule <- gamezoneR::get_team_schedule("Duke")

pbp_duke <- gamezoneR::gamezone_cbb_pbp(schedule$game_id[30])

gamezoneR::base_court +
  geom_point(
    data = pbp_duke,
    aes(loc_x, loc_y,
      color = shot_outcome
    )
  )
```