

StatsBombR

Claude REN

11/07/2020

Introduction

In this document we are going to work on the package StatsBombR, which is well known for providing free data about football. The package contains various functions that facilitate the extraction of data and those functions are what we are going to study here with the help of the work of R(yo) : <https://ryon7.github.io/2019-08-21-visualize-soccer-statsbomb-part-1/>

We are going to change the dataset used, but all the graphical representation will be extracted from his work in order to show a better looking work for an easier reading. Of course every chunk of code will be explain and you can go to my presentation document on ggplot2 to have further understanding.

So the dataset we are going to study here is the women premier league, and more specifically a match between Arsenal WFC and Everton LFC. We will study expected goal (xG), passes in the final third and the preferred link up by the players.

```
library(tidyverse)
library(StatsBombR)
```

```
## Warning: replacing previous import 'foreach::when' by 'purrr::when' when loading
## 'StatsBombR'
```

```
## Warning: replacing previous import 'jsonlite::flatten' by 'purrr::flatten' when
## loading 'StatsBombR'
```

```
## Warning: replacing previous import 'foreach::accumulate' by 'purrr::accumulate'
## when loading 'StatsBombR'
```

```
library(ggplot2)
library(tinytex)
```

```
comps <- FreeCompetitions()
```

```
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
```

```
FAW_match <- comps %>%
  filter(competition_id == 37) %>%
  FreeMatches()
```

```
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
```

```
FAW <- StatsBombFreeEvents(MatchesDF = FAW_match)
```

```
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
```

```
## Warning in if (MatchesDF == "ALL") {: la condition a une longueur > 1 et seul le  
## premier élément est utilisé
```

```
FAW_clean <- FAW %>%  
  allclean()
```

```
AWFC <- FAW_clean %>%  
  filter(match_id == 19811) %>%  
  mutate(shot.statsbomb_xg = if_else(is.na(shot.statsbomb_xg),  
                                     0, shot.statsbomb_xg))
```

```
AWFC_xg <- AWFC %>%  
  group_by(team.name) %>%  
  summarize(tot_xg = sum(shot.statsbomb_xg) %>% signif(digits = 2)) %>%  
  mutate(team_label = glue::glue("{team.name}: {tot_xg} xG"))
```

```
AWFC <- AWFC %>%  
  left_join(AWFC_xg, by = "team.name") %>%  
  mutate(player_label = case_when(  
    shot.outcome.name == "Goal" ~ glue::glue("{player.name}: {shot.statsbomb_xg %>% signif(digits = 2)}  
    TRUE ~ ""))
```

```
AWFC_xg_timelineplot <- AWFC %>%  
  ggplot() +  
  geom_segment(x = 0, xend = 95,  
              y = 0, yend = 0) +  
  geom_rect(data = AWFC %>% filter(shot.outcome.name == "Goal"),  
            aes(xmin = minute - 2, xmax = minute + 2,  
                ymin = -0.005, ymax = 0.005),  
            alpha = 0.3, fill = "green") +  
  geom_label_repel(data = AWFC %>% filter(shot.outcome.name == "Goal"),  
                  aes(x = minute, y = 0,  
                      color = team.name, label = player_label),  
                  nudge_x = 4, nudge_y = 0.003,  
                  show.legend = FALSE) +  
  geom_point(data = AWFC %>% filter(shot.statsbomb_xg != 0),  
            shape = 21, stroke = 1.5,  
            aes(x = minute, y = 0,  
                size = shot.statsbomb_xg, fill = team.name)) +  
  scale_color_manual(values = c("Arsenal WFC" = "#a50044",  
                                "Everton LFC" = "black")) +  
  scale_fill_manual(values = c("Arsenal WFC" = "#a50044",  
                               "Everton LFC" = "white")) +  
  facet_wrap(vars(team_label), ncol = 1) +  
  scale_x_continuous(breaks = seq(0, 95, by = 5),  
                    labels = c(seq(0, 40, by = 5), "HT",  
                                seq(50, 90, by = 5), "FT"),  
                    limits = c(-3, 95),
```

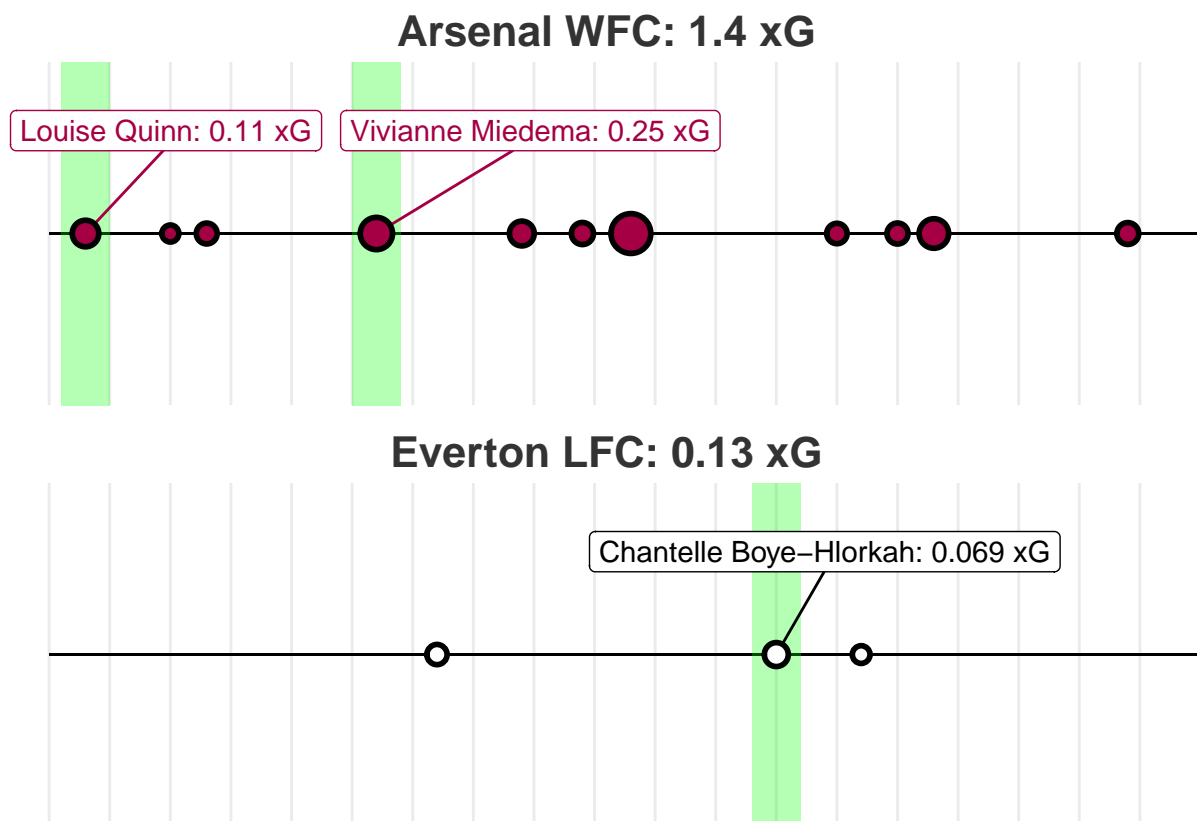
```

        expand = c(0.01, 0)) +
scale_y_continuous(limits = c(-0.005, 0.005),
        expand = c(0, 0)) +
scale_size(range = c(2, 6)) +
theme_minimal() +
theme(legend.position = "none",
      strip.text = element_text(size = 16,
                                face = "bold", color = "grey20"),
      plot.caption = element_text(color = "grey20",
                                  hjust = 0),

      axis.title = element_blank(),
      axis.text = element_blank(),
      panel.grid.minor = element_blank(),
      panel.grid.major.y = element_blank())

```

AWFC_xg_timelineplot



```

AWFC_rollsum <- AWFC %>%
  group_by(minute, team.name, period) %>%
  summarize(sumxg = sum(shot.statsbomb_xg)) %>%
  ungroup() %>%
  group_by(team.name) %>%
  mutate(rollsum = lag(cumsum(sumxg)),
         rollsum = if_else(is.na(rollsum), 0, rollsum)) %>%
  select(team.name, minute, rollsum, sumxg) %>%

```

```

mutate(rollsum = case_when(
  row_number() == n() & sumxg != 0 ~ rollsum + sumxg,
  TRUE ~ rollsum
))

AWFC_rollsum <- AWFC_rollsum %>%
  left_join(AWFC %>% filter(shot.outcome.name == "Goal") %>% select(minute, shot.outcome.name, team.name)
            by = c("minute", "team.name")) %>%
  mutate(rollsum_goal = rollsum + sumxg,
         minute_goal = minute + 1,
         player_label = case_when(
           shot.outcome.name == "Goal" ~ glue::glue("{player.name}: {sumxg %>% signif(digits = 2)} xG")
           TRUE ~ ""))

```

```

tot_awfc_df <- AWFC_xg %>%
  pull(tot_xg)

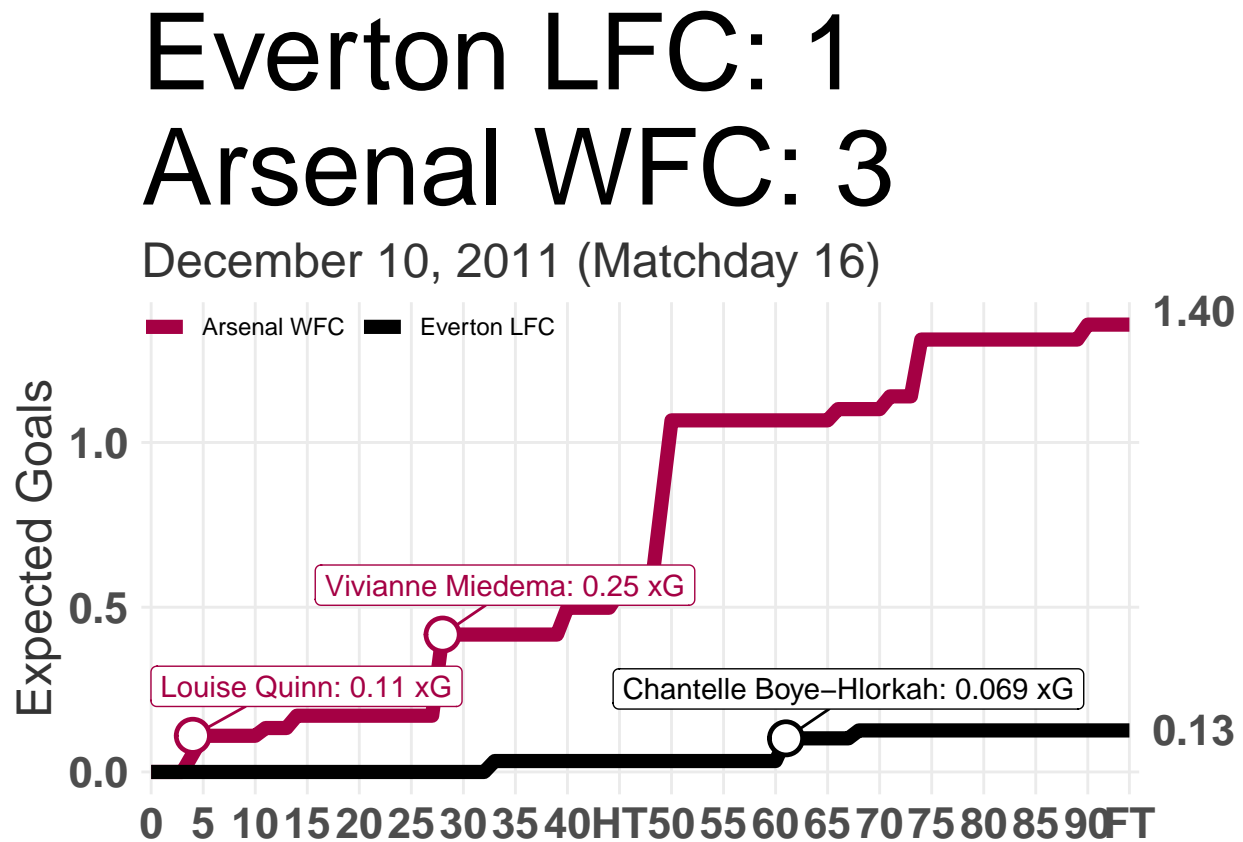
AWFC_rollsumxg_plot <- AWFC_rollsum %>%
  ggplot(aes(x = minute, y = rollsum,
             group = team.name, color = team.name)) +
  geom_line(size = 2.5) +
  geom_label_repel(data = AWFC_rollsum %>% filter(shot.outcome.name == "Goal"),
                  aes(x = minute_goal, y = rollsum_goal,
                      color = team.name, label = player_label),
                  nudge_x = 6, nudge_y = 0.15,
                  show.legend = FALSE) +
  geom_point(data = AWFC_rollsum %>% filter(shot.outcome.name == "Goal"),
            aes(x = minute_goal, y = rollsum_goal, color = team.name), show.legend = FALSE,
            size = 5, shape = 21, fill = "white", stroke = 1.25) +
  scale_color_manual(values = c("Arsenal WFC" = "#a50044",
                                "Everton LFC" = "#000000"),
                    labels = c("Arsenal WFC",
                                "Everton LFC")) +
  scale_fill_manual(values = c("Arsenal WFC" = "#a50044",
                                "Everton LFC" = "#000000")) +
  scale_x_continuous(breaks = c(seq(0, 90, by = 5), 94),
                    labels = c(seq(0, 40, by = 5), "HT",
                                    seq(50, 90, by = 5), "FT"),
                    expand = c(0.01, 0),
                    limits = c(0, 94)) +
  scale_y_continuous(sec.axis = sec_axis(~ ., breaks = tot_awfc_df)) +
  labs(title = "Everton LFC: 1 \nArsenal WFC: 3",
       subtitle = "December 10, 2011 (Matchday 16)",
       x = NULL,
       y = "Expected Goals") +
  theme_minimal() +
  theme(plot.title = element_text(size = 40),
        plot.subtitle = element_text(size = 18, color = "grey20"),
        axis.title = element_text(size = 18, color = "grey20"),
        axis.text = element_text(size = 16, face = "bold"),
        panel.grid.minor = element_blank(),
        legend.position = c(0.2, 0.95),
        legend.direction = "horizontal",

```

```
legend.title = element_blank())
```

```
AWFC_rollsumxg_plot
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



```
roll_final_pass <- AWFC %>%
  group_by(team.name, minute) %>%
  mutate(count = case_when(
    type.name == "Pass" & location.x >= 80 ~ 1L,
    TRUE ~ 0L
  )) %>%
  select(team.name, minute, count) %>%
  ungroup()
```

```
first_min <- AWFC$minute %>% unique() %>% first()
last_min <- AWFC$minute %>% unique() %>% last()
minute <- c(first_min:last_min)
team.name <- c("Everton LFC", "Arsenal WFC")

crossing(minute, team.name) %>% slice(26:32)
```

```
## # A tibble: 7 x 2
```

```
## minute team.name
## <int> <chr>
## 1 12 Everton LFC
## 2 13 Arsenal WFC
## 3 13 Everton LFC
## 4 14 Arsenal WFC
## 5 14 Everton LFC
## 6 15 Arsenal WFC
## 7 15 Everton LFC
```

```
rolling_sum <- tibbletime::rollify(.f = sum, window = 5)

roll_awfc_pass <- crossing(minute, team.name) %>%
  left_join(roll_final_pass, by = c("minute", "team.name")) %>%
  group_by(team.name, minute) %>%
  summarize_all(sum) %>%
  ungroup() %>%
  mutate(count = ifelse(is.na(count), 0, count)) %>%
  group_by(team.name) %>%
  mutate(rollsum = rolling_sum(count),
         rollsum = ifelse(is.na(rollsum), 0, rollsum)) %>%
  group_by(team.name) %>%
  select(-count) %>%
  filter(row_number() %% 5 == 1 | row_number() == n())

roll_awfc_pass %>% head(5)
```

```
## # A tibble: 5 x 3
## # Groups:   team.name [1]
## team.name minute rollsum
## <chr> <int> <dbl>
## 1 Arsenal WFC 0 0
## 2 Arsenal WFC 5 4
## 3 Arsenal WFC 10 1
## 4 Arsenal WFC 15 5
## 5 Arsenal WFC 20 3
```

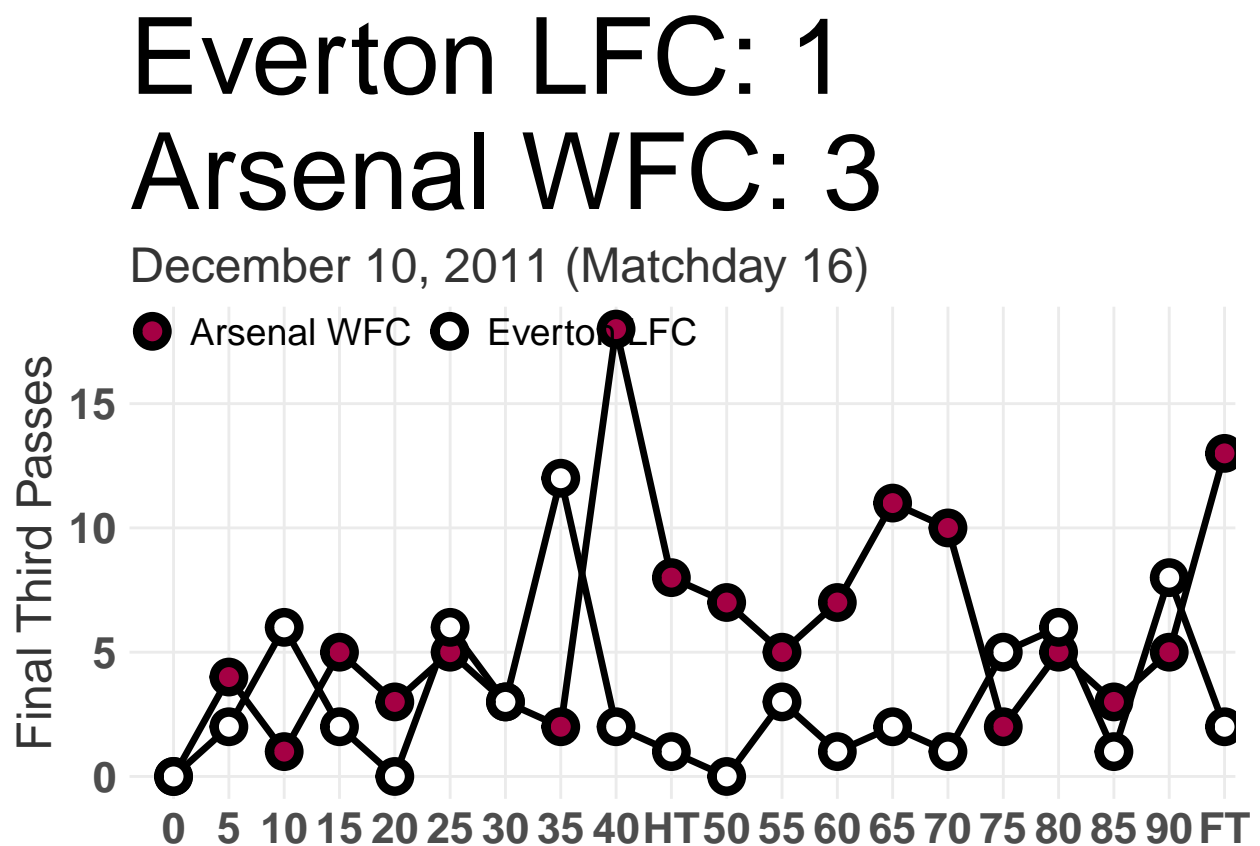
```
finalthird_rollingplot <- roll_awfc_pass %>%
  ggplot(aes(x = minute, y = rollsum,
             group = team.name)) +
  geom_line(data = roll_awfc_pass,
            size = 1.2) +
  geom_point(data = roll_awfc_pass,
             aes(fill = team.name),
             size = 3.5, shape = 21, stroke = 2.5) +
  scale_x_continuous(breaks = seq(0, 95, by = 5),
                     labels = c(seq(0, 40, by = 5), "HT",
                                   seq(50, 90, by = 5), "FT"),
                     limits = c(-3, 95),
                     expand = c(0.01, 0)) +
  scale_y_continuous(breaks = seq(0, 30, by = 5),
                     labels = seq(0, 30, by = 5)) +
  scale_fill_manual(values = c("Arsenal WFC" = "#a50044",
```

```

      "Everton LFC" = "white"),
    labels = c("Arsenal WFC",
               "Everton LFC")) +
labs(title = "Everton LFC: 1 \nArsenal WFC: 3",
     subtitle = "December 10, 2011 (Matchday 16)",
     x = NULL,
     y = "Final Third Passes") +
theme_minimal() +
theme(plot.title = element_text(size = 40),
      plot.subtitle = element_text(size = 18, color = "grey20"),
      axis.title = element_text(size = 18, color = "grey20"),
      axis.text = element_text(size = 16, face = "bold"),
      panel.grid.minor = element_blank(),
      legend.text = element_text(size = 14),
      legend.position = c(0.25, 0.95),
      legend.direction = "horizontal",
      legend.title = element_blank())

```

finalthird_rollingplot



```

AWFC_clean <- FAW_clean %>%
  left_join(comps %>% select(season_id, season_name), by = "season_id")

```

```

pass_received_all_box <- AWFC_clean %>%
  mutate(pass.outcome.name = fct_explicit_na(pass.outcome.name, "Complete")) %>%
  filter(type.name == "Pass",
         team.name == "Arsenal WFC",
         pass.outcome.name == "Complete",
         ## Only passes from open play
         !play_pattern.name %in% c("From Corner", "From Free Kick",
                                   "From Throw In"),
         ## Only passes that ended up inside the box:
         pass.end_location.x >= 102 & pass.end_location.y <= 62 &
         pass.end_location.y >= 18) %>%
  select(player.name, pass.recipient.name,
         season_id, season_name,
         position.name, position.id,
         location.x, location.y,
         pass.end_location.x, pass.end_location.y,
         contains("pass")) %>%
  group_by(season_name) %>%
  add_count(player.name, pass.recipient.name, name = "pass_num") %>%
  ungroup() %>%
  mutate(player.name = glue::glue("{player.name}: {pass_num}")) %>%
  mutate(pass_duo = map2(player.name, pass.recipient.name, ~c(.x, .y))) %>%
  select(player.name, pass.recipient.name, pass_num,
         pass_duo, season_name)

```

```

pass_received_all_box %>%
  group_by(season_name) %>%
  nest()

```

```

## # A tibble: 2 x 2
## # Groups:   season_name [2]
##   season_name data
##   <chr>      <list>
## 1 2018/2019 <tibble [609 x 4]>
## 2 2019/2020 <tibble [270 x 4]>

```

```

library(ggplot2)
all_pass_nested_box <- pass_received_all_box %>%
  group_by(season_name) %>%
  nest() %>%
  mutate(plot = map2(
    .x = data, .y = season_name,
    ~ ggplot(data = .x, aes(x = pass_duo)) +
      geom_bar(fill = "#a70042") +
      scale_x_upset(n_intersections = 10,
                    expand = c(0.01, 0.01)) +
      scale_y_continuous(expand = c(0.04, 0.04)) +
      labs(title = glue::glue("
                                Total Completed Passes Into The Box
                                Between All Players ({.y})"),
           subtitle = "'Name: Number' = Passer, 'No Number' = Pass Receiver",
           x = NULL, y = "Number of Passes") +
      theme_combmatrix(

```



```

text = element_text(
  color = "#004c99"),
plot.title = element_text( size = 20,
  color = "#a70042"),
plot.subtitle = element_text( size = 16,
  color = "#004c99"),
axis.title = element_text( size = 14,
  color = "#004c99"),
axis.text.x = element_text( size = 12,
  color = "#004c99"),
axis.text.y = element_text( size = 12,
  color = "#004c99"),
panel.background = element_rect(fill = "white"),
combmatrix.panel.point.size = 4,
combmatrix.panel.point.color.fill = "#a70042",
combmatrix.panel.line.color = "#a70042",
panel.grid = element_line(color = "black"),
panel.grid.major.x = element_blank(),
axis.ticks = element_blank()))

```

```
glimpse(all_pass_nested_box)
```

```

## Rows: 2
## Columns: 3
## Groups: season_name [2]
## $ season_name <chr> "2018/2019", "2019/2020"
## $ data          <list> [<tbl_df[609 x 4]>, <tbl_df[270 x 4]>]
## $ plot          <list> [<Danielle van de Donk: 42, Danielle van de Donk: 42, ...

```

```

all_pass_nested_1112 <- all_pass_nested_box$plot[[2]] +
  scale_y_continuous(labels = seq(0, 15, by = 5),
    breaks = seq(0, 15, by = 5),
    limits = c(0, 15))

```

```
all_pass_nested_1112
```

```
## Warning: Removed 160 rows containing non-finite values (stat_count).
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```

Total Completed Passes Into The Box Between All Players (2019/2020)

'Name: Number' = Passer, 'No Number' = Passer

Number of Passes

