Analytical visualisations provided by soccerlytics

LZ

11 4 2023

Load data

The visualistions will be carried out based on the publicly availabe event data set from the 2012/2013 Champion's League Final. The data is made available by StatsBomb.

```
# load Data
FreeComp <- StatsBombR::FreeCompetitions()
```

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

```
FreeMatch <-  StatsBombR::FreeMatches(FreeComp %>% filter(season_name == "2012/2013"))
```

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

```
# filter match with id 18240 (CL final)
FreeMatch <- FreeMatch %>% filter(match_id == 18240)
eventData <- StatsBombR::get.matchFree(FreeMatch)
```

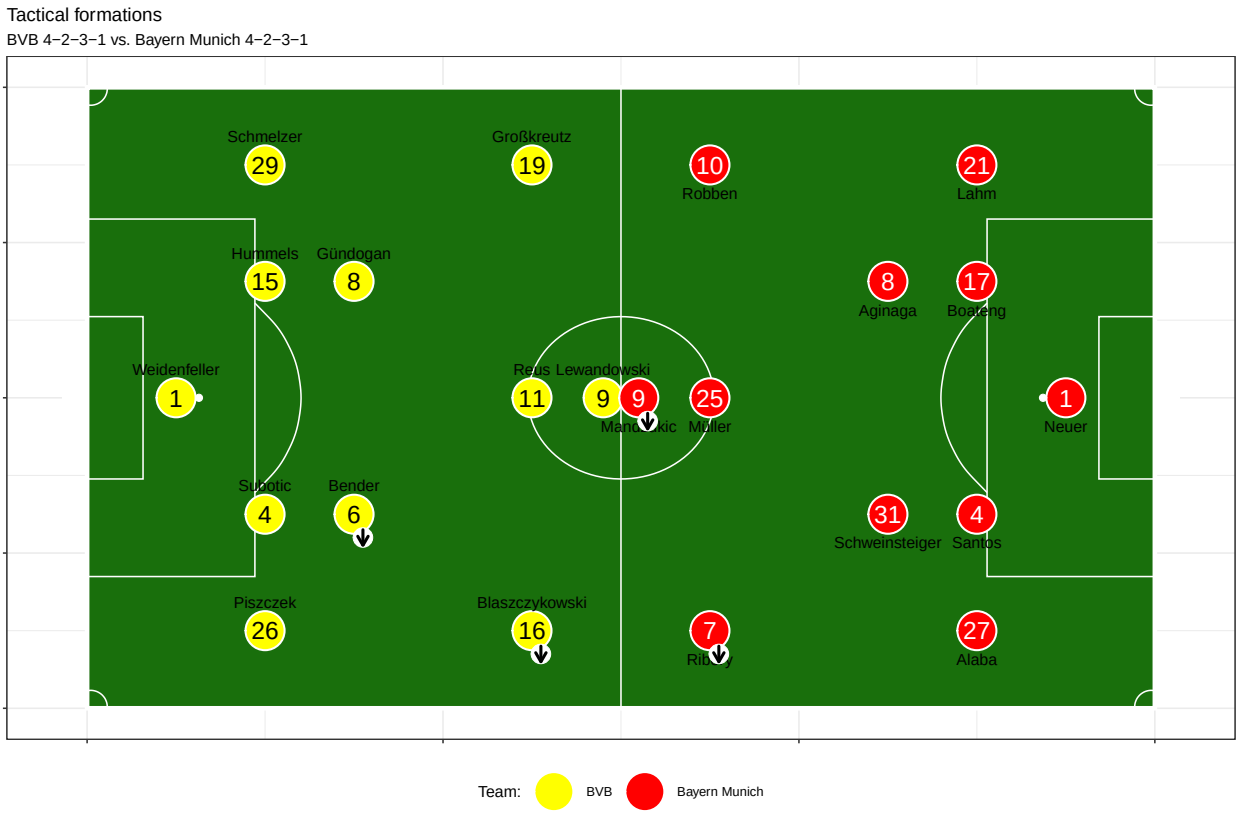
```
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
```

```
## clean eventData (unnest several columns)
eventData <- StatsBombR::allclean(eventData)
```

Tactical line up

To plot the tactical line up, we first extract the lineup dataframes within the event dataset and use the preparation function `prep_formation_data` which can then be passed to the `plot_formation` function, along with the additional information of which player was substituted, here provided in the dataframe `subPlayers`.

```
# prepare the dataframes of tactical line ups for both teams and bind the resulting dfs
dfForm <- prep_formation_data(eventData$tactics.lineup[[1]], team.name = "BVB") %>%
  bind_rows(prep_formation_data(eventData$tactics.lineup[[2]], team.name = "Bayern Munich"))
# get information about players that were substituted
subPlayers <- eventData %>% filter(type.name == "Substitution") %>%
  select(team.name, player.name, minute, second, starts_with("Sub"))
# visualise the tactical formations
plot_formation(dfFormation = dfForm, colourHome = "yellow", colourHome2 = "black",
               colourAway = "red", hometeam = "BVB",
               awayteam = "Bayern Munich", subPlayers = subPlayers,
               formHome = "4-2-3-1", formAway = "4-2-3-1")
```

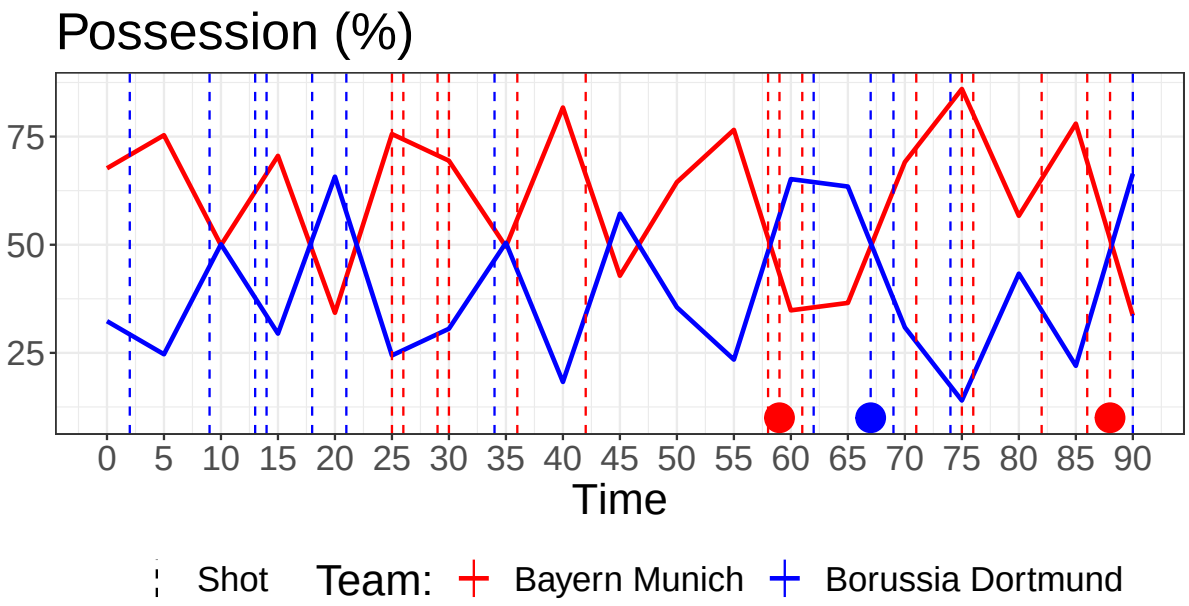


Match dynamics

The function `match_dynamics` can visualise the developement of several statistics over the duration of the match. You can choose the time intervals within which the statistic of interest (such as Pass accuracy, ball possession in %, or pressure events adjusted for opponent ball possession) is computed, to see how it changes over time. Further, vertical lines with Shots that were taken and goals that were scored are shown. Here, one can filter for shots with outcomes that are considered relevant, e.g. goals and saved shots only.

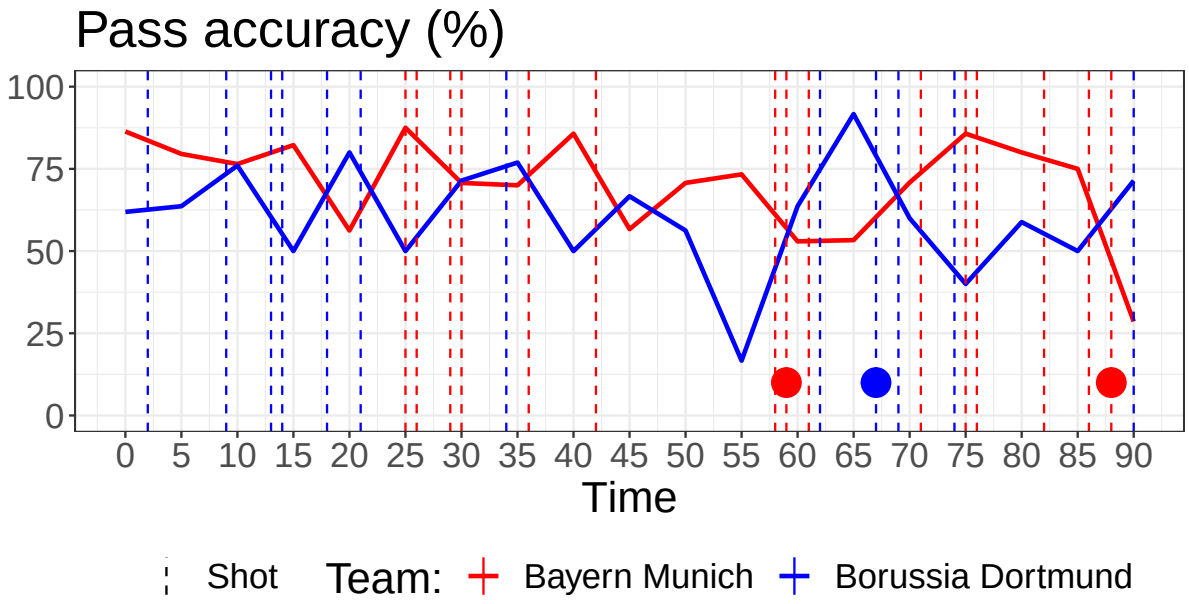
We start with ball possession in %, calculated every 5 minutes.

```
match_dynamics(eventData, binsize = 5, type  = "Possession")
```



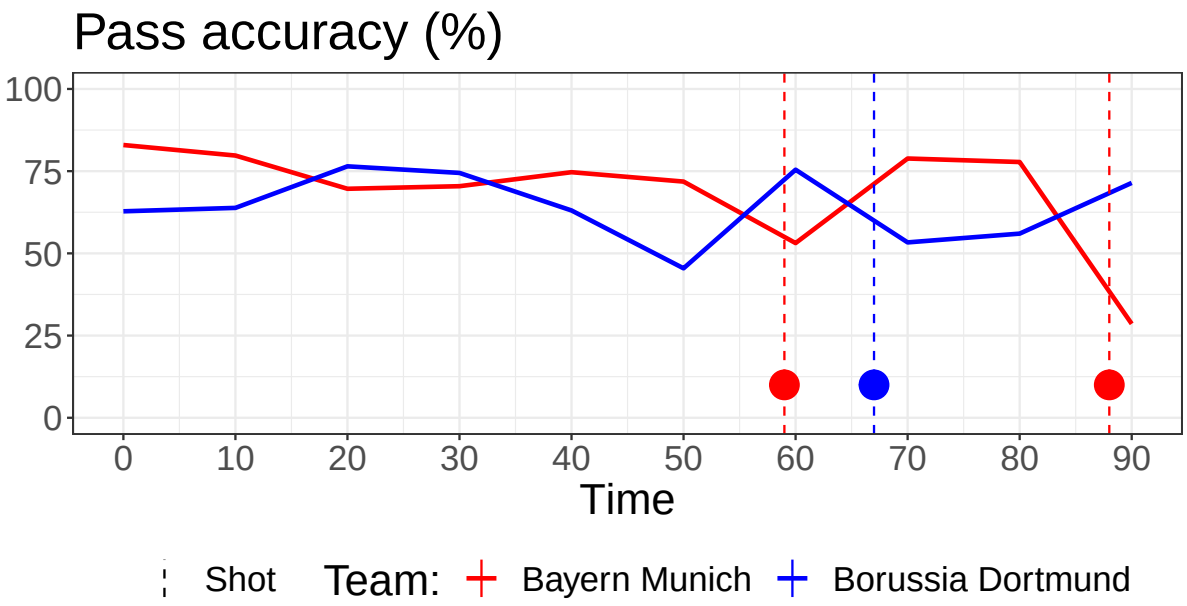
Now we visualise pass accuracy within 5 minute intervals.

```
match_dynamics(eventData, binsize = 5, type  = "Pass")
```



Compare the latter to pass accurracy within 10 minute intervals, and plot only shots that were goals.

```
match_dynamics(eventData, binsize = 10, type  = "Pass", outcome_names = list(pass = NA, shot = "Goal"))
```



We can also visualise cumulated xG values.

```
match_dynamics(eventData, binsize = 5, type  = "xg")
```