Ball progression metrics report

Measuring Player Progression via Open Data — Mathis Sedira-Lemaire

Abstract

Using freely accessible FBref data covering 1,369 players with at least 1,000 league minutes across the five major European leagues in the 2024–2025 season, this project develops open and reproducible metrics to quantify how effectively footballers progress play through passing and ball carrying. I propose two position-adjusted indices, the Pass Progression Index and the Carry Progression Index, each combining progressive volume, end-product contribution, and ball security into a single 0–100 score within positional cohorts. The resulting leaderboards are coherent with observed roles and reputations: midfield and wing-back rankings are dominated by highly creative players with strong offensive responsibility in build-up and chance creation, while forward and goalkeeper cohorts highlight wide creators and distribution-oriented keepers. Methodologically, I show that estimating data-driven importance weights for heterogeneous progressive actions yields a more faithful measure of progression capacity than equal-weight percentile averaging, because it rewards specialised high-impact progression modes rather than stylistic versatility per se. Finally, I aggregate passing- and carrying-based information into a global progression score that summarises a player's overall ability to advance possession while remaining interpretable across positions. Cross-checks against extensive match viewing from the same season suggest that these indices provide credible low-cost tools for profiling progression and supporting recruitment decisions, especially in contexts where proprietary event-value models are unavailable. The full pipeline is transparent, reproducible, and intended as a baseline for scouting-oriented applications on less-covered leagues.

Introduction

This project aims to build practical, reproducible metrics that quantify a player's ability to progress the ball for their team, both by passing and by carrying, using freely available data and open tools. The motivation is concrete and personal: I want to augment my visual match-reading with objective measures drawn from advanced football statistics I consult daily, and to leverage my proficiency in R to implement them. The goal is to produce indices that are analytically sound, operationally useful, and transparent about the choices made to construct them.

Why this problem matters

Conventional chance-creation measures such as expected assists and expected goals capture very well a player's capacity to generate shots and clear-cut opportunities. The capacity to progress the ball, however, is conceptually different and empirically fuzzier. Progression can be achieved in multiple ways, by chains of passes, by individual carries, or by combinations of both, and it is rarely captured fully by

one or two single columns in a table. That is why a composite index is necessary: by combining quantity, quality and risk components drawn from multiple complementary statistics we aim to capture a broader, more robust signal of true progression ability. This index is designed to complement, not replace, visual scouting and event-level models; it should help prioritise which players to watch, and to test whether my in-match impressions align with systematic patterns in open data.

Data philosophy and reproducibility

The project is deliberately constrained to free, public data so the workflow remains reproducible and portable. Anyone with access to the same public sources can recreate, validate and extend this work. All transformations and analyses are implemented in R with clear, commented code that documents every step: variable selection, percentile and normalization choices, weighting decisions, PCA or aggregation logic, and final index construction. Transparency and reproducibility are core values; every numeric decision is recorded and justifiable.

Method and validation

In practice there are two distinct progression profiles worth distinguishing: players whose primary contribution is through progressive passing, and players who primarily advance play by carrying the ball (dribblers/carriers). That is why I first construct two separate, interpretable sub-indices, a pass index and a carry index; each designed to summarise quantity, quality and risk for its respective mode of progression.

In addition to those weighted-percentile indices, I also explore a PCA-based approach as a fundamentally different method rather than as a simple aggregation of the two sub-indices. The PCA is performed on the underlying percentile variables themselves, not on the pass index and carry index, and it yields orthogonal components that capture the principal directions of variation in the raw features. This PCA index is conceptually distinct: it produces weights that are determined by the covariance structure of the variables and highlights the dominant shared signal without requiring arbitrary weight choices.

I apply the indices to the five major European leagues that I follow closely so I can verify that players who score highly under each method correspond to my visual assessments and domain knowledge. This helps ensuring me that the metrics capture meaningful on-field behaviour rather than artefacts of data processing, and it informs subsequent choices about thresholds, robustness checks and any post-processing needed.

Practical application and next steps

The indices are intended as a toolset. Using them for operational scouting and discovery of underappreciated talent is the natural next phase of the project, and will be the subject of subsequent work. For now, the emphasis is on building robust, interpretable indices and on validating them on leagues and players I know well. Once validated, the same metrics can be applied to lesser-known competitions to prioritise players for further video scouting and recruitment consideration, focusing on profiles able to make a difference by progressing play either through carries or passes.

What this document contains

This document will provide a clear description of every variable and the reason for its selection, a full account of the data-collection procedures, each code snippet used to compute the indices with line-by-line explanations and the rationale for every choice. It will also present notes on robustness checks and stability thresholds, diagnostic outputs (such as correlations, PCA loadings and ranked player lists), and interpretative commentary that links numerical results to on-field behaviour.
It will further explain the thought process behind weighting, inclusion rules, and any smoothing or shrinkage applied to low-volume players, so that another analyst can follow, critique, or adapt the pipeline.

Project Status Note

This project is a work in progress. The Carry Index and the combined Carry+Pass metric are still under development, so they are not yet included in a finished form. At this stage, the Pass Progression Index is the only component considered sufficiently complete for analysis. That said, even the Pass Progression Index may still evolve: I'm continuing to refine the methodology and explore more efficient modeling choices, which could lead to adjustments in its definition or implementation in future updates.

A final, candid note

I recognise that much more sophisticated models than these indices exist, notably expected-threat (xT) frameworks that estimate the change in scoring opportunity when the ball moves from one zone to another on the pitch. Similarly, commercial products such as StatsBomb's on-ball value (OBV) models provide more granular, spatially-aware estimates of action value. I do not claim to have revolutionised football analytics; these indices are intended as a practical, reproducible and interpretable approach that complements both visual scouting and more advanced spatial models, not as a replacement for them. Note also that those advanced spatial models and proprietary valuations are paid services, whereas this project prioritises methods and data that are freely accessible.

Glossary of Variables and Calculation Methods

I selected all available variables that measure both the quantity and the quality of progressive actions to construct the progression indices. The included variables are "prpasses90", "prcarries90", "prrecep90", "cpa90", "passesfinalthird90", "crspa90", "tbpass90", "pertes90", "kp90", "xa90", "passesratees90", "carriesratees90", "scato90" and "db90".

I use "xa90" rather than raw assist counts to isolate the passer contribution. Expected assists estimate the quality of the chances a player creates. They do not depend on whether teammates convert those chances. Raw assists combine chance creation with teammates finishing ability. That conflation would reward players whose teammates overperform their expected goals. Using "xa90" removes that source of variance and better reflects the passer's own contribution.

For dribbling progression I prefer "scato90" over "gcato90" for the same reason. "scato90" counts dribbles that lead to a shot. "gcato90" counts dribbles that lead to a goal. Goals are rarer events and are

strongly influenced by teammates finishing quality. Including “gcato90” would import that external effect into the progression index. “scato90” therefore better captures the player specific ability to advance play by beating opponents and generating shooting opportunities.

All variables used in the analyses are expressed per 90 minutes of playing time. Normalising raw counts to a common minute exposure produces rates that are directly comparable across players. This practice limits bias from unequal playing time and reduces heteroscedasticity in downstream modelling. It therefore improves the statistical stability and interpretability of the indices derived from these metrics.

The field `Pos.x` contains the full set of positions listed by FBref for each player. FBref’s position taxonomy is deliberately coarse and uses only four categories: `GK`, `DF`, `MF` and `FW`. As a consequence, full-backs, wing-backs and central defenders are always grouped as `DF`, wingers and centre-forwards are always grouped as `FW` and defensive and attacking midfielders under `MF`. Players may be listed at multiple positions; for comparisons we create a separate variable `Pos` that retains the first position reported by FBref and treat that as the player’s primary position. Because positional groups on FBref are broad, percentile comparisons by position will sometimes compare players with markedly different on-field roles and responsibilities.

The following definitions refer either to the original FBref field or to variables derived in the pipeline. The exact code used to create every derived variable will be provided in the next section.

`Player`
Player name as recorded by FBref.

`Squad.x`
Club or squad as recorded by FBref.

`Pos.x`
All positions listed by FBref for the player, in reported order.

`Pos`
Primary position retained for analysis, equal to the first position listed in `Pos.x`.

`Min_Playing`
Total minutes played by the player during the season.

`prpasses90` (Progressive passes per 90) A progressive pass is a completed pass that advances the ball by at least ten yards or that enters the opponent’s penalty area. This variable is computed as the FBref progressive-pass count divided by (Min_Playing / 90).

`prcarries90` (Progressive carries per 90)
A progressive carry advances play toward the opponent’s goal by at least ten yards or results in entry into the penalty area. This variable is computed as the FBref progressive-carry count divided by (Min_Playing / 90).

`prrecep90` (Progressive receptions per 90)
A progressive reception is a received pass that advances play toward the opponent’s goal by at least ten

yards or results in entry into the penalty area. This variable is computed as the FBref progressive-reception count divided by (Min_Playing / 90).

`cpa90` (Carries into the penalty area per 90) Carries that end with the player carrying the ball into the opponent's penalty area, expressed per 90 minutes.

`passesfinalthird90` (Passes into the final third per 90)
Passes that enter the final third of the pitch, normalised to a 90-minute basis.

`crspa90` (Crosses into the penalty area per 90)
Crosses directed into the opponent's penalty area per 90 minutes, corresponding to FBref's crosses-to-penalty-area counts.

`tbpass90` (Through balls per 90)
Through-ball attempts per 90 minutes. Through balls are passes that play a teammate into space behind the defensive line.

`pertes90` (Losses of possession per 90)
Constructed turnover rate per 90 minutes. `Ballons_perdus` is the sum of `Dis_Carries`, `Mis_Carries` and `Passes_ratees`. `pertes90` equals `Ballons_perdus` divided by (Min_Playing / 90). On FBref, `Dis_Carries` records occasions where a player is dispossessed while carrying the ball. `Mis_Carries` counts failed or miscontrolled carries. `Passes_ratees` is computed as attempted passes minus completed passes and represents unsuccessful pass completions.

`kp90` (Key passes per 90) Key passes are final passes that lead directly to a teammate's shot.

`xa90` (Expected assists per 90)
xA measures the probability that a given pass will become an assist; the season total xA is divided by (Min_Playing / 90). It is a metric that captures the quality of the situations created by a player's passing by measuring how effectively their deliveries generates advantageous opportunities for teammates.

`passesratees90` (Unsuccessful passes per 90)
Unsuccessful pass completions per 90 minutes, computed as (Att_Total − Cmp_Total) divided by (Min_Playing / 90).

`carriesratees90` (Failed carries per 90)
Failed carry events per 90 minutes, computed as (Dis_Carries + Mis_Carries) divided by (Min_Playing / 90).

`scato90` (Shot-creating actions take-ons per 90)
A shot-creating take-on is a successful dribble or take-on that directly leads to a shot. The variable `scato90` records the number of such actions per 90 minutes by dividing the FBref shot-creating-take-on count by (Min_Playing / 90).

`db90` (Successful take-ons per 90)
Successful take-ons or dribbles per 90 minutes, based on FBref's `Succ` field and normalised by minutes played.

Data Processing and Indices Methodology

Packages needed to run the code :

```
#packages needed
install.packages(c("tidyverse","readxl","worldfootballR","janitor","missMDa","ggplot2","ggrepl'
library(worldfootballR)
library(readxl)
library(tidyverse)
library(janitor)
library(missMDA)
library(ggplot2)
library(ggrepel)
library(gt)
library(webshot2)
```

The opening section of the script imports comprehensive player statistics from FBref for the five major European leagues. During ingestion the Shot-Creating Actions (SCA) and Carries into the Penalty Area (CPA) fields, however, exhibited an unusually high proportion of missing values, so these two metrics were extracted manually and saved as Excel workbooks in a separate folder to ensure completeness and reproducibility. The supplemental Excel files are read into the pipeline with the following lines (please update the paths to match your environment before running): `df_sample8 <- read_excel("~/M2/Recherche Stage M2/Scouting dashboard/bdd_scaclean2.xlsx")` and `df_sample9 <- read_excel("~/M2/Recherche Stage M2/Scouting dashboard/indices finaux/CPA.xlsx")`. Finally, all FBref tables and the Excel supplements are normalized as required, most importantly by standardizing player names, and merged on `player_name` to yield a single master dataset containing the full set of statistics used to compute the indices.

```
#data collection
season_end_year <- 2025

stat_types <- c("standard", "shooting", "passing", "passing_types", "passing_outcomes",
                "gca", "possession", "misc", "defense")

list_stats <- map(stat_types, ~{
  tryCatch({
    df <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                             stat_type = .x,
                                             team_or_player = "player")
    df$stat_type_source <- .x
    df
  }, error = function(e) {
    message("stat_type ", .x, " non dispo : ", e$message)
    NULL
  })
})
```

```
df_sample1 <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                                 stat_type = "possession",
                                                 team_or_player = "player")

df_sample2 <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                                 stat_type = "standard",
                                                 team_or_player = "player")
df_sample3 <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                                 stat_type = "shooting",
                                                 team_or_player = "player")
df_sample4 <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                                 stat_type = "passing",
                                                 team_or_player = "player")
df_sample5 <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                                 stat_type = "passing_types",
                                                 team_or_player = "player")
df_sample6 <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                                 stat_type = "misc",
                                                 team_or_player = "player")
df_sample7 <- load_fb_big5_advanced_season_stats(season_end_year = season_end_year,
                                                 stat_type = "defense",
                                                 team_or_player = "player")
df_sample8 <- read_excel("~/M2/Recherche Stage M2/Scouting dashboard/bdd_scaclean2.xlsx")

df_sample9 <- read_excel("~/M2/Recherche Stage M2/Scouting dashboard/indices finaux/CPA.xlsx")

df_merged <- Reduce(function(x, y) merge(x, y, by = "Player", all = TRUE),
                    list(df_sample1, df_sample2, df_sample3, df_sample4,
                         df_sample5, df_sample6, df_sample7, df_sample8, df_sample9))
```

The merged dataset is reduced to a single row per player by creating a composite key that combines club and player name (`paste(Squad.x, Player, sep = "|")`) and by removing duplicated keys. Duplicate columns are then eliminated with `df_merged <- df_merged[, !duplicated(names(df_merged))]` to avoid redundancy. I compute a set of per 90 minutes variables (for example `prcarries90 = PrgC_Progression / (Min_Playing / 90)`, `kp90`, `xa90`, `pertes90`) to normalise event counts by minutes of exposure. I then create `df1` to retain only the required identifiers and indicators. `Min_Playing` is coerced to numeric and the dataset is filtered to include players with at least 1000 minutes in the previous season. This threshold, roughly equivalent to eleven full 90 minute appearances, ensures that estimates rely on meaningful playing time and are statistically more stable. Finally, I inspect missing values, note 75 missing `pertes90` entries, identify the affected players and remove those rows to produce the final cleaned sample `df3_clean`.

```
#data cleaning
df_merged <- df_merged[, !duplicated(names(df_merged))]
key <- paste(df_merged$Squad.x, df_merged$Player, sep = "|")
df_merged <- df_merged[!duplicated(key), ]
```

```
colnames(df_merged)

#variables creation
df_merged$prcarries90 <- df_merged$PrgC_Progression / (df_merged$Min_Playing / 90)
df_merged$prpasses90 <- df_merged$PrgP_Progression / (df_merged$Min_Playing / 90)
df_merged$prrecep90 <- df_merged$PrgR_Progression / (df_merged$Min_Playing / 90)
df_merged$kp90 <- df_merged$KP / (df_merged$Min_Playing / 90)
df_merged$Passes_ratees <- df_merged$Att_Total - df_merged$Cmp_Total
df_merged$passesratees90 <- df_merged$Passes_ratees / (df_merged$Min_Playing / 90)
df_merged$carriesratees <- df_merged$Dis_Carries + df_merged$Mis_Carries
df_merged$carriesratees90 <- df_merged$carriesratees / (df_merged$Min_Playing / 90)
df_merged$Ballons_perdus <- df_merged$Dis_Carries + df_merged$Mis_Carries + df_merged$Passes_ra
df_merged$pertes90 <- df_merged$Ballons_perdus / (df_merged$Min_Playing / 90)
df_merged$passesfinalthird90 <- df_merged$Final_Third / (df_merged$Min_Playing / 90)
df_merged$crspa90 <- df_merged$CrsPA / (df_merged$Min_Playing / 90)
df_merged$tbpass90 <- df_merged$TB_Pass / (df_merged$Min_Playing / 90)
df_merged$cpa90 <- df_merged$cpa / (df_merged$Min_Playing / 90) #
df_merged$xa90 <- df_merged$xA_Expected / (df_merged$Min_Playing / 90)
df_merged$scato90 <- df_merged$scato / (df_merged$Min_Playing / 90)
df_merged$db90 <- df_merged$Succ / (df_merged$Min_Playing / 90)
df_merged$prgdist90 <- df_merged$PrgDist_Total / (df_merged$Min_Playing / 90)

df1 <- df_merged[, c("Player","Squad.x","Comp.x","Nation.x","Pos.x","Born.x","Age.x","MP_Playir

df3 <- df1 %>%
  mutate(Min_Playing = as.numeric(gsub(",", ".", as.character(Min_Playing)))) %>%
  filter(!is.na(Min_Playing) & Min_Playing >= 1000)

na_count <- sum(is.na(df3$pertes90))
print(na_count)#74 only variable with missing datas, 5% of the df (and not best players) -> car
unique(as.character(df3[is.na(df3$pertes90), 1]))
df4 <- df3 %>%
  filter(!is.na(pertes90))
```

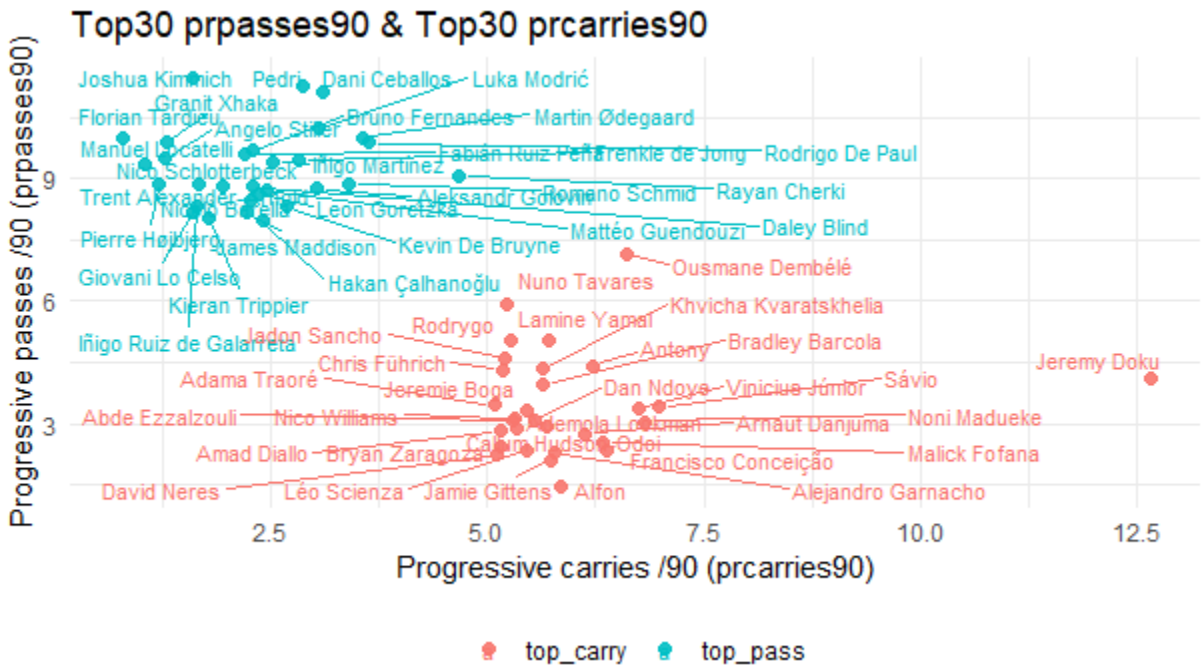
I wanted to ensure that the statistics properly reflect the existence of different player profiles. Some players progress the ball mainly through dribbling and others mainly through passing. To verify this distinction, I plotted progressive carries on the x-axis and progressive passes on the y-axis. The objective was to explore whether players tend to specialise in one behaviour or whether some combine both at a high level.

I decided to include only the thirty players with the highest volume of progressive carries and the thirty players with the highest volume of progressive passes. These two variables are strong indicators of how often a player advances the ball by dribbling or by passing, so they provide a meaningful picture of progression styles. The graph clearly shows two dominant profiles. Some players rely much more on dribbling to break lines, while others mainly progress play with their passing. There are a few notable exceptions who show high volume in both dimensions, such as Dembélé, Yamal and Nuno Tavares.

This observation leads to a structured approach for building the indices. First, I will construct one index that measures progression through passing and another that measures progression through dribbling. Only after defining those two dimensions will I attempt to design a combined index that captures total progressive impact. Players like the ones mentioned above, who contribute significantly through both methods, should then be rewarded accordingly in the final comprehensive metric.

```
top_pass  <- df4 %>% filter(!is.na(prpasses90))  %>% arrange(desc(prpasses90))  %>% slice_head(
top_carry <- df4 %>% filter(!is.na(prcarries90)) %>% arrange(desc(prcarries90)) %>% slice_head(


plot_df <- df4 %>%
  filter(Player %in% union(top_pass, top_carry)) %>%
  mutate(group = case_when(
    Player %in% top_pass & Player %in% top_carry ~ "both",
    Player %in% top_pass ~ "top_pass",
    TRUE ~ "top_carry"
  ))

ggplot(plot_df, aes(x = prcarries90, y = prpasses90, color = group)) +
  geom_point(size = 2, alpha = 0.9) +
  geom_text_repel(aes(label = Player),
                  size = 3,
                  max.overlaps = 200,
                  box.padding = 0.25,
                  point.padding = 0.3,
                  segment.size = 0.3) +
  labs(x = "Progressive carries /90 (prcarries90)",
       y = "Progressive passes /90 (prpasses90)",
       title = "Top30 prpasses90 & Top30 prcarries90") +
  theme_minimal() +
  theme(legend.position = "bottom", legend.title = element_blank())

top_pass  <- df4 %>% filter(!is.na(xa90))  %>% arrange(desc(xa90))  %>% slice_head(n = 30) %>%
top_carry <- df4 %>% filter(!is.na(sca90)) %>% arrange(desc(sca90)) %>% slice_head(n = 30) %>%


plot_df <- df4 %>%
  filter(Player %in% union(top_pass, top_carry)) %>%
  mutate(group = case_when(
    Player %in% top_pass & Player %in% top_carry ~ "both",
    Player %in% top_pass ~ "top_pass",
    TRUE ~ "top_carry"
  ))

ggplot(plot_df, aes(x = sca90, y = xa90, color = group)) +
  geom_point(size = 2, alpha = 0.9) +
  geom_text_repel(aes(label = Player),
                  size = 3,
                  max.overlaps = 200,
```

```
                box.padding = 0.25,
                point.padding = 0.3,
                segment.size = 0.3) +
  labs(x = "Shot-creating actions /90 (sca90)",
       y = "Expected assists /90 (xa90)",
       title = "Top30 xa90 & Top30 sca90") +
  theme_minimal() +
  theme(legend.position = "bottom", legend.title = element_blank())
```



Before constructing the progression metrics, I test the pairwise correlations among the variables intended for use. The code builds a numeric matrix from the selected fields, computes a pairwise-complete correlation matrix, and prints any variable pairs with absolute correlation greater than 0.85. This automatic check flags potential multicollinearity before indices are formed.

Two strong relationships were observed and are expected on theoretical grounds. `xa90` and `kp90` correlate at approximately 0.90. This is sensible because both capture aspects of chance creation by passing: key passes count the final passes that lead directly to a teammate's shot, while expected assists (xA) quantify the shot-creating quality of passes by assigning each pass a probability of becoming an assist. Players who produce many key passes therefore tend also to accumulate high xA, which explains the high correlation. `pertes90` and `passesratees90` correlate at roughly 0.99. This near-perfect correlation is unsurprising because `pertes90` is constructed by aggregating `passesratees90` together with other turnover components; when a constructed variable contains another as a primary component, a very high correlation is inevitable.

These findings have practical implications for index construction. Highly collinear variables may inflate variance in composite scores or multivariate models, so I exclude variables exhibiting excessive collinearity when constructing indices as weighted averages to avoid double-counting redundant information. When exclusion would discard materially distinct signals, I apply principal component analysis and use the resulting component scores in place of the original variables (this will be the case

for my final metric). This preserves interpretability while ensuring that each index combines information that is orthogonal and statistically stable.

```
#checking for correlation
mat <- df4 %>% select(all_of(vars_use)) %>% mutate(across(everything(), as.numeric))
cor_mat <- cor(mat, use = "pairwise.complete.obs")
print(round(cor_mat, 2))

high_cor <- which(abs(cor_mat) > 0.85 & abs(cor_mat) < 1, arr.ind = TRUE)
if (nrow(high_cor) > 0) {
  cat("Paires très corrélées (>0.85) :\n")
  apply(high_cor, 1, function(i) {
    cat(rownames(cor_mat)[i[1]], " <-> ", colnames(cor_mat)[i[2]],
        " = ", round(cor_mat[i[1], i[2]], 2), "\n")
  })
} else cat("Aucune paire > 0.85 détectée.\n")
```

Pass Progresion Index

This section describes the construction of the pass-progression index. The index combines three complementary dimensions of passing: the volume of progressive actions (“PassQuantity”), the ability of those actions to generate genuine chances (“PassQuality”), and the degree of control with which they are executed (“PassAccuracy”). The core passing variables retained are $\mathrm{prpasses90}$, $\mathrm{passesfinalthird90}$, $\mathrm{crspa90}$ and $\mathrm{tbpass90}$, which represent progressive passes, passes into the final third, crosses into the penalty area and through balls. These four variables capture distinct modalities by which a player can advance possession by passing. A naïve way to measure “quantity” would be to convert each of the four variables into positional percentiles and then average those percentiles with equal weights. Such a construction would primarily reward versatility: players who use several different types of progressive passes would score better than players who rely heavily on a single type. For this project, however, the intention is to measure volume of progression rather than versatility of patterns. If a player produces an extremely high number of through balls but rarely attempts crosses into the box, he should still be rated as providing a very large amount of progression by passing. For that reason, PassQuantity is built from the raw per-90 volumes, combined through data-driven weights and only then converted to a percentile. I chose not to include metres gained by pass variables because they are largely collinear with the progressive pass modalities already modelled and add little incremental information on chance creation: conditional on the weighted progressive-pass volumes, their marginal association with xA is negligible, so incorporating them would primarily increase measurement noise rather than improve signal. Moreover, a raw metres-gained metric is not valuation-neutra, gains in a team’s own half are not equivalent to gains in advanced zones. So using it properly would require a complex, zone-dependent weighting scheme; the current framework already captures “dangerous” progression through empirically calibrated weights tied to xA, making an additional distance term both redundant and potentially destabilising for the index.

Rather than assigning arbitrary weights to the four pass types, I estimate how strongly each one is associated with expected assists per 90 ($\mathrm{xa90}$). For each player i, let xA_i denote $\mathrm{xa90}_i$ and let X_{ik} be the standardised value of pass-type $k \in \{1, \ldots, 4\}$ (progressive passes, passes into the final third,

the standardised value of pass type $k \in \{1, \ldots, 4\}$ (progressive passes, passes into the final third, crosses into the box and through balls). I estimate an ordinary least squares (OLS) regression.

$$xA_i = \alpha + \sum_{k=1}^{4} \beta_k X_{ik} + \varepsilon_i.$$

OLS chooses the coefficients α,β1,...,β4 that minimise the sum of squared residuals

$$\sum_i \varepsilon_i^2 \;=\; \sum_i \Big(xA_i - \alpha - \sum_{k=1}^{4} \beta_k X_{ik} \Big)^2$$

The absolute coefficients are then normalised to sum to one and interpreted as importance weights

$$w_k = \frac{|\beta_k|}{\sum_{j=1}^{4} |\beta_j|}, \qquad k = 1, \ldots, 4,$$

so that a pass type which tends to generate more xA across the sample receives a higher weight in the quantity measure. In the present dataset this yields approximate weights

$$w_{\text{prpasses90}} \approx 0.26, \quad w_{\text{passesfinalthird90}} \approx 0.20, \quad w_{\text{crspa90}} \approx 0.26, \quad w_{\text{tbpass90}} \approx 0.29,$$

which reflect the idea that through balls and crosses into the box are, on average, somewhat more productive for chance creation than simple progressive passes or generic passes into the final third. The raw quantity score for player i is the weighted sum

$$\text{PassQuantity}_i^{\text{raw}} = \sum_{k=1}^{4} w_k P_{ik},$$

where P_{ik} is the per-90 count of pass-type k in original units. This provides an empirically grounded "conversion rate" between the four pass types and turns them into a single notion of progressive passing volume.

For the quality component I deliberately retain expected assists per 90, xa90, rather than working with residual xA. At an intermediate stage I constructed a "pure" quality term by regressing xA on $\text{PassQuantity}_i^{\text{raw}}$ and taking residuals, so that quality captured only the portion of xA not explained by volume. This residual specification is statistically neat, but it has an undesirable side-effect for an index intended to measure progression by passing: archetypal high-volume playmakers whose xA is roughly proportional to their involvement are partly discounted, while more marginal profiles with modest involvement and volatile xA per 90 tend to be pushed up. Since the purpose of the index is precisely to surface players who consistently move the ball forward and generate good chances for their team, it is more appropriate to reward the total level of expected assists. In the final specification, PassQuality is therefore defined as xA per 90, expressed as a positional percentile. Misplaced passes per 90 (passesratees90) are used to form the PassAccuracy component: within each positional group I compute the percentile of missed passes and invert it so that a higher PassAccuracy_pct corresponds to a lower rate of failed passes.

All three components are then put on a common 0–100 scale by computing positional percentiles. Formally, for any sample $\{x_k\}_{k=1}^{n}$ used to compute percentiles within a given positional group, the rank of observation x_i is its ordinal position in the sorted (ascending) sample:

$$\mathrm{rank}(x_i) = 1 \text{ for the smallest value}, \qquad \mathrm{rank}(x_i) = n \text{ for the largest value}.$$

The empirical percentile-rank (PR) and its 0–100 counterpart are then defined by

$$\mathrm{PR}(x_i) = \frac{\mathrm{rank}(x_i) - 1}{n - 1} \in [0, 1], \qquad \mathrm{PR\%}(x_i) = 100 \cdot \mathrm{PR}(x_i).$$

For each player i in positional group $\mathcal{P} = \mathrm{Pos}(i)$, I denote by

$$\tilde{Q}_i = \mathrm{PR\%}\big(\mathrm{PassQuantity}_i^{\mathrm{raw}}\big), \qquad \tilde{R}_i = \mathrm{PR\%}\big(\mathrm{xa90}_i\big), \qquad \tilde{A}_i = 100 - \mathrm{PR\%}\big(\mathrm{passesratees90}_i\big),$$

the corresponding positional percentiles of progressive passing volume, expected-assist output, and (inverted) misplaced-pass rate. These three scores represent, respectively, how much progressive passing volume a player provides compared to his peers at the same position, how much expected-assist output he generates, and how well he controls the ball while attempting these passes.

To inform the relative weighting of these three components, I estimate a diagnostic model in which expected assists are regressed on three orthogonalised dimensions: a quantity percentile, a residual-quality percentile and an accuracy percentile. Concretely, I first regress xA on $\mathrm{PassQuantity}_i^{\mathrm{raw}}$ and form residuals xA_i^{res}; I then compute positional percentiles Q_i, R_i and A_i of the quantity, residual-quality and accuracy terms, and fit

$$xA_i = \gamma_0 + \gamma_Q Q_i + \gamma_R R_i + \gamma_A A_i + \eta_i.$$

Here OLS again chooses

$$\gamma_0, \gamma_Q, \gamma_R, \gamma_A$$

to minimise the sum of squared residuals, so the fitted linear combination of Qi,Ri,Ai provides the best least-squares approximation to xA using these three diagnostic components. Normalising the absolute coefficients,

$$\omega_Q = \frac{|\gamma_Q|}{|\gamma_Q| + |\gamma_R| + |\gamma_A|}, \quad \omega_R = \frac{|\gamma_R|}{|\gamma_Q| + |\gamma_R| + |\gamma_A|}, \quad \omega_A = \frac{|\gamma_A|}{|\gamma_Q| + |\gamma_R| + |\gamma_A|},$$

yields approximate shares $\omega_Q \approx 0.33$, $\omega_R \approx 0.62$ and $\omega_A \approx 0.04$. Statistically, this procedure asks: "if we try to explain variation in xA using these three dimensions, how much weight does each one naturally receive in the best linear approximation?". Using this regression as a calibration step is pertinent for a progression index, because it anchors the weights in the empirical link between each component and a concrete measure of valuable attacking output. At the same time, the index itself is not meant to be a pure xA predictor, but a balanced summary of progression by passing.

For that reason I do not plug $\omega_Q, \omega_R, \omega_A$ directly into the index. Instead I adopt rounded, interpretable weights

-

$$\lambda_Q = 0.4, \qquad \lambda_R = 0.5, \qquad \lambda_A = 0.1,$$

which preserve the ordering suggested by the data (quality somewhat more important than quantity, accuracy less so) while ensuring that the accuracy dimension retains a meaningful influence. The raw pass-progression index for player i is then

$$\mathrm{PassIndex}_i^{\mathrm{raw}} = \lambda_Q \, \tilde{Q}_i + \lambda_R \, \tilde{R}_i + \lambda_A \, \tilde{A}_i,$$

and the final index $\mathrm{PassIndex}_i^{\mathrm{pct}}$ is obtained by rank-standardising $\mathrm{PassIndex}_i^{\mathrm{raw}}$ within the player's positional group to a 0–100 percentile scale, that is

$$\mathrm{PassIndex}_i^{\mathrm{pct}} = \mathrm{PR}\%\big(\mathrm{PassIndex}_i^{\mathrm{raw}}\big).$$

In light of the results, the index appears robust in practice. Among midfielders, players such as Joshua Kimmich, Luka Modrić and Pedri appear at the very top of the distribution, which coheres with their widely recognised status as central playmaking hubs and elite progressive passers. Among forwards, Lee Kang-in reaches the 100th percentile within his cohort; even if it is not considered as an absolute top player, this can be understood in light of his occasional deeper deployment at PSG, a substantial share of minutes accumulated late in matches against tired opposition, and a relatively high proportion of minutes in comparatively favourable Ligue 1 fixtures, all of which facilitate strong per-90 passing numbers. He is closely accompanied by Ousmane Dembélé and Michael Olise, whose creative profiles and tendency to supply dangerous passes from wide or half-space zones are well established. In the defender group, Konrad Laimer and Leon Goretzka rank highly chiefly because they are classified as defenders in the underlying data despite occupying de facto midfield roles, while attacking full-backs and wing-backs such as Trent Alexander-Arnold, Achraf Hakimi, Nuno Mendes, Raphaël Guerreiro and Alphonso Davies also feature prominently, in line with their reputations as major outlets for progression from wide areas. For goalkeepers the index is of secondary interest, but it still captures distributional quality: Oliver Baumann, Ederson and Manuel Neuer all rate very highly, reflecting their roles in teams that actively use the goalkeeper to initiate play and break opposition lines rather than confining them to pure shot-stopping duties.

```
#pass index
df5 <- df4
pass_vars <- c("prpasses90","passesfinalthird90","crspa90","tbpass90")
xa_var    <- "xa90"
miss_var  <- "passesratees90"

mod_df <- df5 %>%
  select(all_of(c(xa_var, pass_vars))) %>%
  na.omit()

X <- scale(mod_df[, pass_vars])
y <- mod_df[[xa_var]]

mod <- lm(y ~ X)
```

```r
coefs <- coef(mod)[-1]
names(coefs) <- pass_vars

if (all(is.na(coefs))) {
  weights_quantity <- rep(1 / length(pass_vars), length(pass_vars))
  names(weights_quantity) <- pass_vars
} else {
  coefs_abs <- abs(coefs)
  if (sum(coefs_abs, na.rm = TRUE) == 0) {
    weights_quantity <- rep(1 / length(pass_vars), length(pass_vars))
    names(weights_quantity) <- pass_vars
  } else {
    weights_quantity <- coefs_abs / sum(coefs_abs, na.rm = TRUE)
  }
}

barplot(
  heights <- weights_quantity,
  names.arg = names(weights_quantity),
  main = "Weights for PassQuantity (learned from xa90)",
  ylab = "Weight",
  las = 2
)

df5 <- df5 %>%
  mutate(
    PassQuantity_raw = as.numeric(as.matrix(select(., all_of(pass_vars))) %*% weights_quantity)
  )

df5 <- df5 %>%
  mutate(Pos = sapply(strsplit(as.character(Pos.x), "[, /-]"), `[`, 1)) %>%
  group_by(Pos) %>%
  mutate(
    PassQuantity_pct = percent_rank(PassQuantity_raw) * 100,
    PassQuality_pct  = percent_rank(.data[[xa_var]]) * 100,
    PassAccuracy_pct = (1 - percent_rank(.data[[miss_var]])) * 100
  ) %>%
  ungroup()
xa_var   <- "xa90"
miss_var <- "passesratees90"

mod_q_diag <- lm(as.formula(paste(xa_var, "~ PassQuantity_raw")), data = df5)
df5$xa_resid_diag <- resid(mod_q_diag)

df5_diag <- df5 %>%
  group_by(Pos) %>%
  mutate(
    PassQuantity_pct_diag      = percent_rank(PassQuantity_raw) * 100,
    PassQuality_resid_pct_diag = percent_rank(xa_resid_diag) * 100,
    PassAccuracy_pct_diag      = (1 - percent_rank(.data[[miss_var]])) * 100
```

```
  ) %>%
  ungroup()

mod_w_diag <- df5_diag %>%
  select(all_of(c(xa_var,
                  "PassQuantity_pct_diag",
                  "PassQuality_resid_pct_diag",
                  "PassAccuracy_pct_diag"))) %>%
  na.omit()

Z_diag <- scale(mod_w_diag[, c("PassQuantity_pct_diag",
                               "PassQuality_resid_pct_diag",
                               "PassAccuracy_pct_diag")])
y_diag <- mod_w_diag[[xa_var]]

mod_index_diag <- lm(y_diag ~ Z_diag)

beta_diag <- coef(mod_index_diag)[-1]
names(beta_diag) <- c("quantity_resid", "quality_resid", "accuracy_resid")

beta_diag_abs <- abs(beta_diag)
weights_index_diag_resid <- beta_diag_abs / sum(beta_diag_abs)

print(beta_diag)
print(weights_index_diag_resid)

weights_index_final <- c(quantity = 0.4, quality = 0.5, accuracy = 0.1)

df5 <- df5 %>%
  group_by(Pos) %>%
  mutate(
    PassIndex_weighted_raw =
      weights_index_final["quantity"] * PassQuantity_pct +
      weights_index_final["quality"]  * PassQuality_pct +
      weights_index_final["accuracy"] * PassAccuracy_pct,
    PassIndex_weighted_pct = round(percent_rank(PassIndex_weighted_raw) * 100, 2)
  ) %>%
  ungroup()

export <- df5 %>%
  select(Player, Squad.x, Pos.x, Pos, Comp.x, Min_Playing, PassQuantity_raw, xa90, xa_resid_dia
         PassQuantity_pct, PassQuality_pct, PassAccuracy_pct, PassIndex_weighted_raw,
         PassIndex_weighted_pct) %>%
  arrange(desc(PassIndex_weighted_pct))

write.csv2(export, "PassIndex.csv",
           row.names = FALSE, fileEncoding = "UTF-8")

top5_passindex <- df5 %>%
  mutate(Pos = factor(Pos, levels = c("GK", "DF", "MF", "FW"))) %>%
```

```
  group_by(Pos) %>%
  slice_max(PassIndex_weighted_pct, n = 5, with_ties = FALSE) %>%
  arrange(Pos, desc(PassIndex_weighted_pct)) %>%
  ungroup() %>%
  select(
    Pos,
    Player,
    Squad = Squad.x,
    PassIndex = PassIndex_weighted_pct
  )
top5_passindex %>%
  gt() %>%
  tab_header(
    title = "Top 5 players by PassIndex for each position group"
  )
top5_passindex %>%
  gt() %>%
  tab_header(
    title = "Top 5 players by PassIndex for each position group"
  )

gt_tbl <- top5_passindex %>%
  gt() %>%
  tab_header(
    title = "Top 5 players by PassIndex for each position group"
  )
dir.create("C:/Users/liloy/Documents/M2/Recherche Stage M2/Scouting dashboard", showWarnings =
gtsave(gt_tbl, filename = "C:/Users/liloy/Documents/M2/Recherche Stage M2/Scouting dashboard/tc
```

Figure 1

Top 5 players by PassIndex for each position group

Pos	Player	Squad	PassIndex
GK	Oliver Baumann	Hoffenheim	100.00
GK	Ederson	Manchester City	98.95
GK	Vanja Milinković-Savić	Torino	97.89
GK	Manuel Neuer	Bayern Munich	96.84

GK	Robin Zentner	Mainz 05	95.79
DF	Konrad Laimer	Bayern Munich	100.00
DF	Leon Goretzka	Bayern Munich	99.81
DF	Achraf Hakimi	Paris S-G	99.62
DF	Nuno Mendes	Paris S-G	99.43
DF	Trent Alexander-Arnold	Liverpool	99.24
MF	Joshua Kimmich	Bayern Munich	100.00
MF	Luka Modrić	Real Madrid	99.78
MF	Pedri	Barcelona	99.55
MF	Fabián Ruiz Peña	Paris S-G	99.33
MF	Angelo Stiller	Stuttgart	99.10
FW	Lee Kang-in	Paris S-G	100.00
FW	Ousmane Dembélé	Paris S-G	99.66
FW	Michael Olise	Bayern Munich	99.33
FW	Paulo Dybala	Roma	98.99
FW	Désiré Doué	Paris S-G	98.65

Scope, Assumptions, and Limitations of the Pass Progression Index

Despite its practical robustness, the pass-progression index has several structural limitations that should

guide interpretation.

First, the index is strongly exposure-driven: per-90 volumes and xA are mechanically higher in high-possession, high-territory teams, so player scores remain partially confounded by team possession share and tactical environment; a possession-adjusted variant (e.g., progression per fixed number of team possessions or passes) would provide a complementary “danger per unit of possession” view.

Second, anchoring pass-type weights and component calibration to xA may understate valuable progression that restructures play without immediately generating shots (tempo-setting switches, field-tilting circulation, pre-assist entries), which can bias against possession hubs whose role is primarily territorial or connective rather than final-action creation.

Third, the linear OLS weighting and additive aggregation assume constant marginal returns and separability between Quantity, Quality, and Accuracy; in reality, progression value is likely non-linear (e.g., certain high-leverage passes carry disproportionate payoff) and interactive (high volume only converts to value above execution thresholds), so the index may smooth over these effects for interpretability. Further work specifically focused on modelling the mapping between progressive passing actions and game progression would be valuable, as it could justify the use of non-linear, interaction-aware approaches that better reflect football reality than linear relationships when estimating passing progression and constructing future indices.

Fourth, per-90 normalisation measures total contribution over playing time rather than threat per action; a per-pass or per-50-team-pass would answer a different question about passing intent and immediacy of threat and would reorder some profiles, especially among forwards.

Fifth, the four progressive-pass inputs are not mutually exclusive: one action can be both progressive and into the final third, or a cross into the box, so PassQuantity involves partial double counting, which the empirical weights mitigate but cannot fully eliminate.

Sixth, positional percentiles rely on broad cohorts that can mix structurally different roles, notably wing-backs being evaluated within defender pools dominated by centre-backs; this can mechanically advantage high-involvement wide profiles relative to peers whose role is less progression-oriented.

Seventh, cross-league comparability is imperfect because baseline difficulty, tempo, and defensive intensity differ across the observed competitions (e.g., Ligue 1 vs. Premier League), meaning that a given percentile does not necessarily represent the same underlying value across leagues.

Finally, the index does not capture structurally progressive backward or lateral passes that enable third-man combinations and subsequent line-breaking actions; this limitation is common to most event-based value models, including standard expected threats implementations, which typically reward only the pass that directly advances location rather than the preparatory pass that makes progression possible.