SpatialSportsR Analysis — Kaleb Coleman

Kaleb Coleman

Abstract

This paper presents a reproducible, end-to-end examination of NBA offensive strategy and player performance. We detail a comprehensive shot dataset compiled via a custom data pipeline spanning the 2014-15 through 2025-26 seasons. Using this data, we develop an expected field goal (xFG) model via logistic regression, achieving 63% accuracy with 0.653 AUC-ROC. Building on this foundation, we deliver three advanced analytics products: (1) Residual Analysis to identify players who over/underperform expectations, (2) a Shot Difficulty Index (SDI) quantifying shot complexity, and (3) role-aware GMM clustering to identify player shot archetypes. We then integrate salary data to contextualize performance via POE per $1M and a complementary FG% residual vs salary view that isolates shot-making independent of free throws. Results highlight Luke Kennard as a top overperformer (+12.3% residual) and surface distinct shot-profile archetypes (e.g., **Paint-Dominant** and **Perimeter-Focused** profiles).

Introduction

The National Basketball Association (NBA) has undergone a profound data revolution, fundamentally altering strategies and analysis. This paper addresses the challenge of translating shot data into actionable insights by leveraging a comprehensive, custom-built NBA shot dataset spanning the 2014-15 through 2025-26 seasons.

Contributions

- A reproducible data pipeline consolidating NBA shot data into an analysis-ready SQLite database.
- An expected field goal (xFG) model using logistic regression with 63% accuracy.
- Three advanced analytics products: Residual Analysis, Shot Difficulty Index, and Player Archetypes.
- A salary-adjusted value layer (POE per $1M) plus FG% residual vs salary to separate shot-making from free-throw effects.
- League-wide shot selection summaries including efficiency metrics and attempt distribution.
- Nonlinear effect summaries from a GAM model (partial dependence plots) to interpret how context changes shot success.

Paper Roadmap

Section 2 reviews relevant background. Section 3 details methods. Section 4 reports results including the xFG model and advanced analytics. Section 5 discusses findings and limitations.

Related Work

Prior research on basketball analytics has emphasized the relationship between shot location and scoring efficiency. This study builds on that work by providing a fully reproducible xFG model and extending it with residual analysis, shot difficulty quantification, and player clustering.

Methods

Data Sources and Pipeline

The dataset was compiled through a custom R package interfacing with the NBA Stats API and ESPN NBA API. Raw JSON data was parsed, cleaned, and stored in a SQLite database (`nba.sqlite`) containing 2.5M+ shots across 12 seasons. (*NBA Stats API*, n.d.; *ESPN NBA API*, n.d.)

Pipeline overview

1. **Collect**: `collect_raw()` downloads game-level JSON from ESPN and NBA Stats with caching and rate-limit handling.
2. **Parse**: `parse_raw()` normalizes raw JSON into standardized tables (games, events, shots, box scores, play-by-play).
3. **Validate**: `validate_tables()` enforces schema and key constraints to ensure referential integrity.
4. **Write**: `write_tables()` and `write_sqlite_from_rds()` persist data to `data/parsed/` and a consolidated SQLite file.

Analysis scripts

- `expected_points_analysis.py`: trains the xFG model and writes `shots_with_xp_*.parquet`
- `advanced_analytics.py`: residuals, SDI, and player archetype clustering
- `player_performance_analysis.py`: POE leaderboards and player shot charts
- `shot_density.py`: league-wide shot density heatmaps
- `salary_collector.py`: player salary data from Basketball-Reference (*Basketball-Reference Player Contracts*, n.d.)
- `value_analysis.py`: POE per $1M salary rankings

xFG Model

The expected field goal model uses logistic regression with the following features:

- **Spatial**: LOC_X, LOC_Y, shot distance, shot angle
- **Shot Type**: layup, dunk, jump shot, hook, floater indicators
- **Context**: period, seconds remaining, clutch indicator
- **Zones**: SHOT_ZONE_BASIC, SHOT_ZONE_AREA (one-hot encoded)

Rolling Window Training (True Out-of-Sample)

To eliminate data leakage and provide rigorous out-of-sample evaluation, we employ a **5-season rolling window** training methodology:

- **Training data**: 2020-21 through 2024-25 (5 consecutive seasons, ~1M shots)
- **Evaluation data**: 2025-26 (current season, true out-of-sample)

This approach captures the **modern 3-point era** shooting patterns while ensuring that all player residuals and POE values are genuine out-of-sample predictions—the model has never seen any 2025-26 shots during training. This is the same methodology a production system would use to predict shot difficulty in real-time.

GAM Model (Nonlinear Effects)

We additionally fit a Generalized Additive Model (GAM) to capture nonlinear relationships between shot success and spatial/context features. This model is not used for the main xFG predictions, but it provides interpretable partial dependence plots (PDPs) that clarify how shot probability changes across distance, angle, time, and location.

Shot Difficulty Index (SDI)

SDI quantifies shot difficulty using a weighted combination:

$$SDI = 0.30 \times distance + 0.20 \times clock + 0.20 \times type + 0.15 \times zone + 0.15 \times angle$$

Higher SDI indicates a more difficult shot.

Player Clustering

Role-aware Gaussian Mixture Models (GMMs) were applied to player-level features, with the number of clusters per role chosen by minimum BIC (k=2..6).

Player features include:

- Shot diet composition (rim, mid-range, corner 3, above-the-break 3)
- Distance distribution (std, IQR, entropy)
- Shot-type rates (pull-ups, layups, dunks, hooks, floaters)
- Difficulty/efficiency (avg SDI, avg xFG, actual FG%)
- Usage and volume (usage%, attempts per game)

Players are first grouped into coarse roles (Big/Wing/Guard) using shot-diet thresholds (rim share, 3PT share, and distance), then clustered within those roles. SDI is explicitly included in the clustering feature set and is given extra weight to emphasize shot-difficulty separation. PCA is used **only** for 2D visualization; it preserves variance, not cluster semantics, so spatial proximity in the plot may not reflect why players were clustered.

Archetype names are assigned *post-hoc* using cluster averages (rim%, 3PT%, mid-range share, SDI, usage). Labels are based on **season-relative quantiles** (`q70` for rim/3PT/midrange dominance) rather than fixed absolute cutoffs. We use spatial-descriptive labels (e.g., "Paint-Dominant", "Perimeter-Focused") rather than evaluative terms to clarify that these are shot profiles, not skill assessments.

SDI-Based Split: For players with mixed (multi-zone) profiles, we split the cluster by Shot Difficulty Index (SDI): - **Shot Creator (Mixed)**: Higher SDI than the season median (difficult, self-created shots). - **Role Player (Mixed)**: SDI at/below the season median (easier, scheme-generated looks).

Usage does not determine creator vs role-player; it only adds suffixes such as `(High Usage)` and `(Low Usage)` using top/bottom 30% cut points.

> **Disclaimer**: These archetypes describe shot location and difficulty patterns only. They do NOT imply offensive skill, versatility, or overall player quality.

Clustering variables and SQL sources

- `pct_rim`: Share of shots in Restricted Area + In The Paint (Non-RA).
  Source: `nba_stats_shots.SHOT_ZONE_BASIC` aggregated per player.
- `pct_midrange`: Share of shots in Mid-Range.
  Source: `nba_stats_shots.SHOT_ZONE_BASIC`.
- `pct_3pt`: Share of 3PT shots (Above the Break + both corners).
  Source: `nba_stats_shots.SHOT_ZONE_BASIC`.
- `pct_corner_3`: Share of corner threes (Left/Right Corner 3).
  Source: `nba_stats_shots.SHOT_ZONE_BASIC`.
- `distance_std`: Std. dev. of shot distance (feet).
  Source: `nba_stats_shots.SHOT_DISTANCE` with fallback to `LOC_X/LOC_Y`.
- `distance_entropy`: Entropy of shot-distance bins `[0, 3, 10, 16, 24, 30, 40]`.
  Source: `nba_stats_shots.SHOT_DISTANCE`.
- `avg_sdi`: Avg. Shot Difficulty Index (SDI).
  Source: derived from `SHOT_DISTANCE`, `MINUTES_REMAINING`, `SECONDS_REMAINING`, `ACTION_TYPE`, `SHOT_ZONE_BASIC`, and `LOC_X/LOC_Y` (angle).
- `pullup_rate`: Share of shots tagged as jump shots/pull-ups.
  Source: `nba_stats_shots.ACTION_TYPE` text rules.

- `usage_pct`: Minutes-weighted usage%.
  Source: `nba_stats_player_box_usage.usagePercentage`.
- `attempts_per_game`: Total attempts ÷ games played.
  Source: `nba_stats_shots` grouped by `PLAYER_ID` and `GAME_ID`.

Note: `xP_prob` (xFG) is produced by the logistic model and is **not** a clustering feature in the current setup; SDI is included and given extra weight.

We select the number of components (k) per role using Bayesian Information Criterion (BIC), evaluating k=2..6 and choosing the minimum BIC. Lower BIC indicates a better balance of fit and parsimony; the chart below shows the role-specific BIC curves and highlights the selected k for each group. `cluster_confidence` is reported as the in-sample GMM posterior assignment probability and should be interpreted as relative assignment certainty, not calibrated uncertainty.

Analysis Dataset

```
shots <- read_shots_parquet("../data/shots_with_xp_2025-26.parquet")
residuals <- read.csv("../data/player_residuals.csv")
clusters <- read.csv("../data/player_clusters.csv")
```

Results

xFG Model Performance

```
metrics_files <- c("../data/model_metrics_xfg.csv", "../data/model_metrics_gam.csv")
metrics_list <- lapply(metrics_files, function(path) {
  if (file.exists(path)) {
    read.csv(path, stringsAsFactors = FALSE)
  } else {
    NULL
  }
})
  metrics <- dplyr::bind_rows(metrics_list)

  if (nrow(metrics) > 0) {
    # Ensure columns exist to handle mixed schemas
    if (!"train_seasons" %in% names(metrics)) metrics$train_seasons <- NA
    if (!"eval_season" %in% names(metrics)) metrics$eval_season <- NA
    if (!"season" %in% names(metrics)) metrics$season <- NA  # Fallback for old schema

    # Handle both old and new column formats
    metrics_display <- metrics |>
    transmute(
      Model = model,
      `Train Seasons` = if_else(!is.na(train_seasons), train_seasons, as.character(season)),
      `Eval Season` = if_else(!is.na(eval_season), eval_season, as.character(season)),
      Accuracy = sprintf("%.1f%%", accuracy * 100),
      `AUC-ROC` = sprintf("%.3f", auc_roc),
      `Log Loss` = sprintf("%.3f", log_loss),
      `Brier Score` = if_else(!is.na(brier_score), sprintf("%.4f", brier_score), "-")
    )
  kable(metrics_display, caption = "Model Performance (Out-of-Sample Evaluation)")
} else {
```

```
  kable(data.frame(Note = "Model metrics files not found. Run analysis scripts to generate metrics."),
        caption = "Model Performance")
}
```

Table 1: Model Performance (Out-of-Sample Evaluation)

Model	Train Seasons	Eval Season	Accuracy	AUC-ROC	Log Loss	Brier Score
xFG (Logistic Regression)	2020-21 to 2024-25	2025-26	62.7%	0.649	0.646	0.2280
GAM (PyGAM)	2020-21 to 2024-25	2025-26	62.7%	0.654	0.643	0.2268

Summary Metrics

```
summary_metrics <- shots |>
  mutate(
    shot_value = if_else(SHOT_TYPE == "3PT Field Goal", 3L, 2L),
    made = SHOT_MADE_FLAG == 1L
  ) |>
  summarise(
    attempts = n(),
    fg_pct = mean(made),
    fg2_pct = mean(made[SHOT_TYPE == "2PT Field Goal"]),
    fg3_pct = mean(made[SHOT_TYPE == "3PT Field Goal"]),
    avg_distance = mean(shot_distance_feet, na.rm = TRUE)
  )

kable(
  summary_metrics,
  digits = 3,
  caption = "League-wide shooting summary for the 2025-26 regular season."
)
```

Table 2: League-wide shooting summary for the 2025-26 regular season.

attempts	fg_pct	fg2_pct	fg3_pct	avg_distance
140676	0.469	0.547	0.359	13.843

Correlation Matrix

The correlation matrix summarizes relationships across player-level metrics (shot selection, efficiency, and volume) to highlight how shot profile dimensions move together. A clear cluster emerges in the top-left: **average distance, SDI, and pull-up rate are strongly correlated with each other and strongly *negatively* correlated with average xFG**, reflecting the tradeoff between shot difficulty/creation and expected efficiency. This is a descriptive signal of shot-profile structure; if these features are used together in modeling, collinearity should be handled explicitly.

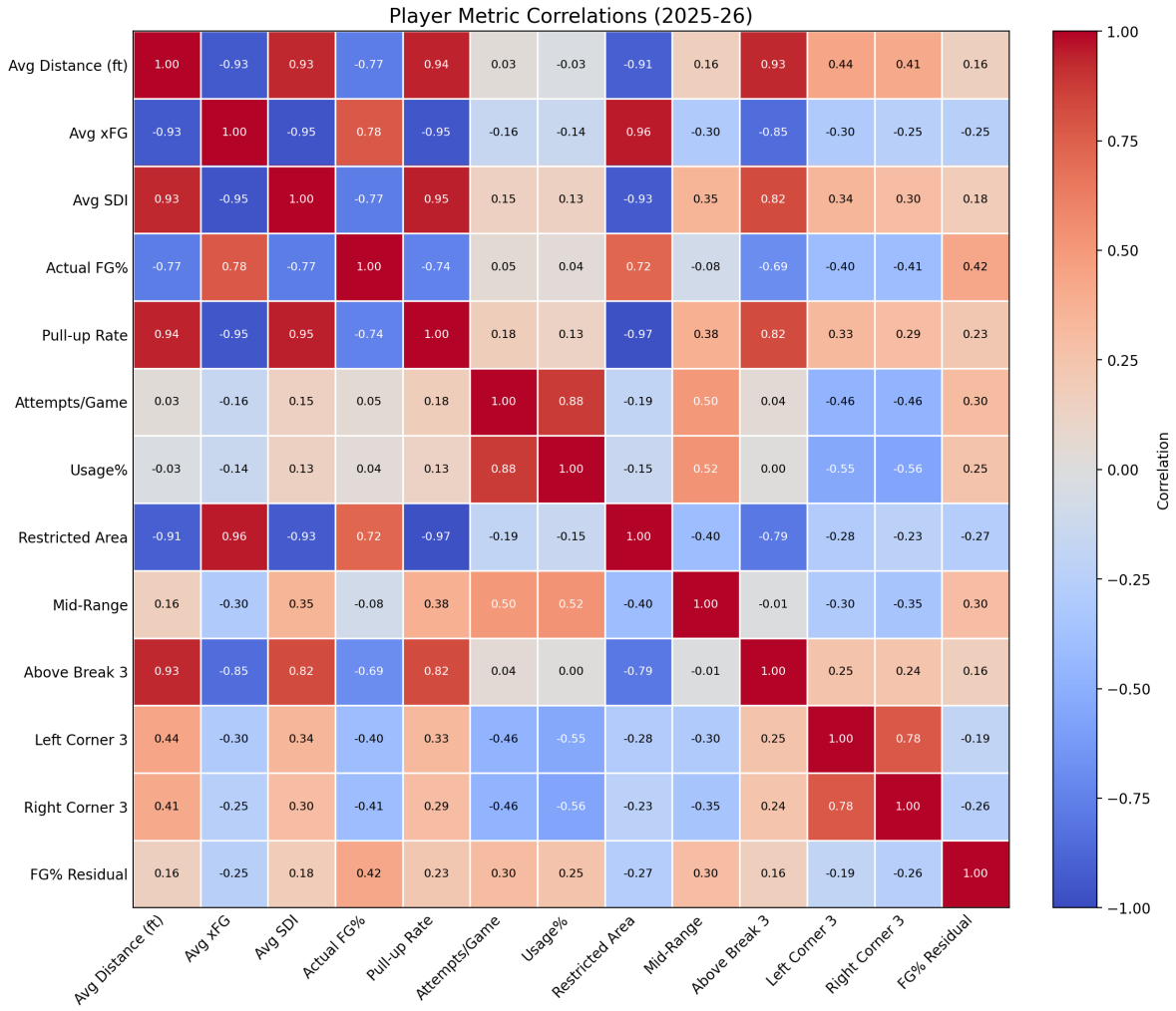


Figure 1: Correlation matrix across player-level metrics for the 2025-26 regular season.

Data: Player-level aggregates from shots_with_xp_2025-26.parquet plus usage from nba.sqlite (when available).
Filters: Players with >=200 shots; only numeric columns with variance retained for correlation.
Exclusions: Low-volume players; columns that are all-NA or constant.
Why: Summarize relationships among shot profile, volume, and efficiency metrics to diagnose collinearity.

Shot Distribution

The shot chart and distance distribution contextualize league-wide shot selection. As expected in the modern NBA, attempts are concentrated at the rim and beyond the three-point line, while mid-range volume is comparatively low.

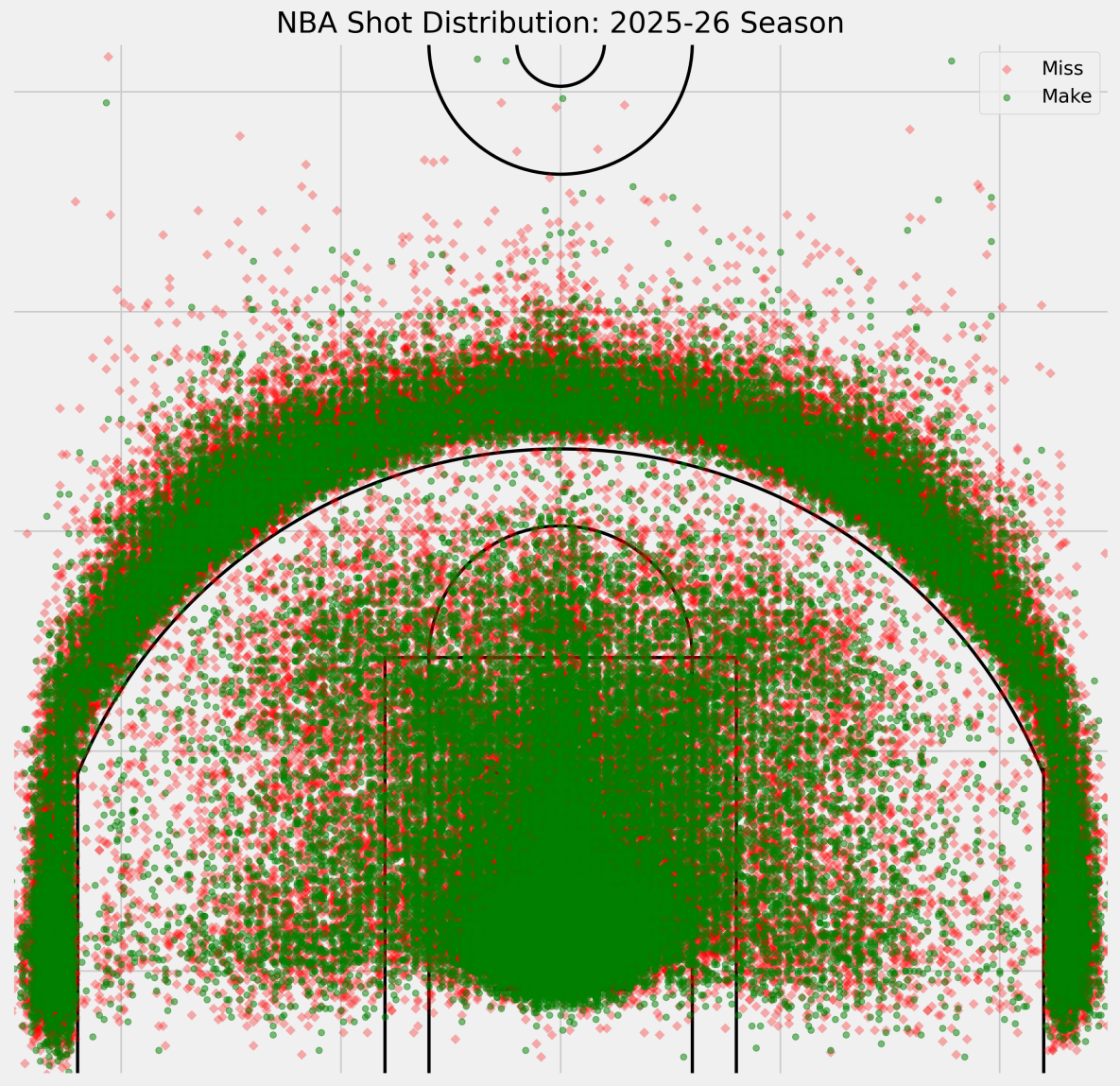


Figure 2: Shot distribution for the 2025-26 regular season.

Data: 2025-26 regular-season shots from nba_stats_shots (SQLite).
Filters: SHOT_ATTEMPTED_FLAG=1; valid LOC_X/LOC_Y and ACTION_TYPE.
Exclusions: Shots with missing location/action metadata.
Why: Visualize spatial distribution of makes vs misses.

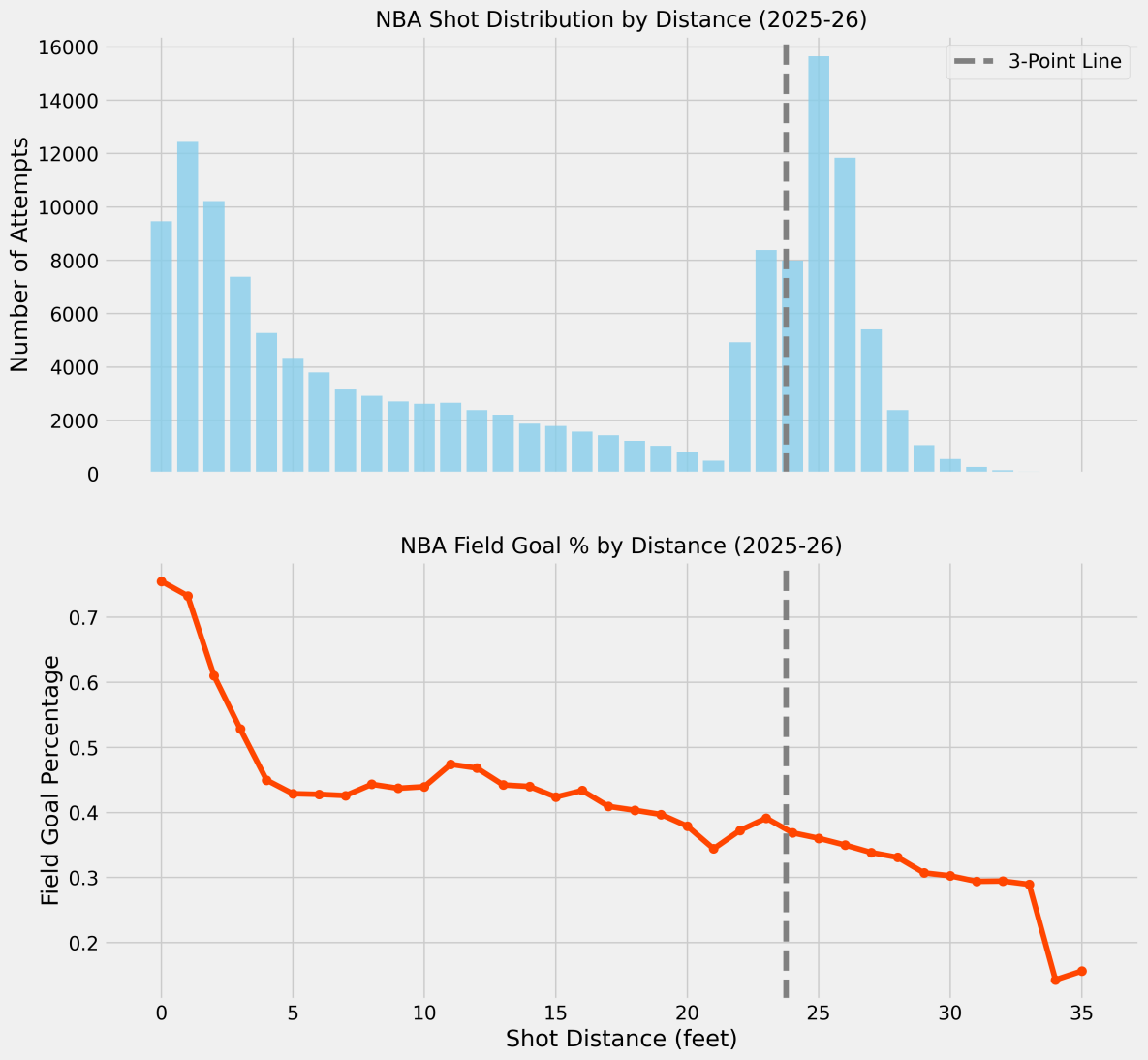


Figure 3: Shot distance distribution for the 2025-26 regular season.

Data: Same 2025-26 regular-season shot sample as the shot chart.
Filters: Distances rounded to nearest foot; display limited to <=35 ft.
Exclusions: Long heaves beyond 35 ft.
Why: Stabilize the distance distribution within typical shooting range.

Residual Analysis

```
top_over <- head(residuals, 10)
kable(
  top_over[, c("player", "avg_xFG", "actual_fg_pct", "residual_fg_pct", "attempts")],
  digits = 3,
  caption = "Top 10 Overperformers (Positive Residuals)"
)
```

Table 3: Top 10 Overperformers (Positive Residuals)

player	avg_xFG	actual_fg_pct	residual_fg_pct	attempts
Luke Kennard	0.415	0.538	0.123	247
Nikola Jokić	0.472	0.594	0.122	638
Luka Garza	0.490	0.585	0.096	217
Shai Gilgeous-Alexander	0.459	0.554	0.095	964
Deandre Ayton	0.583	0.676	0.093	407
Cam Spencer	0.397	0.489	0.092	393
T.J. McConnell	0.455	0.536	0.081	321
Kevin Durant	0.432	0.510	0.078	857
Sam Merrill	0.388	0.464	0.076	267
Austin Reaves	0.439	0.511	0.072	409

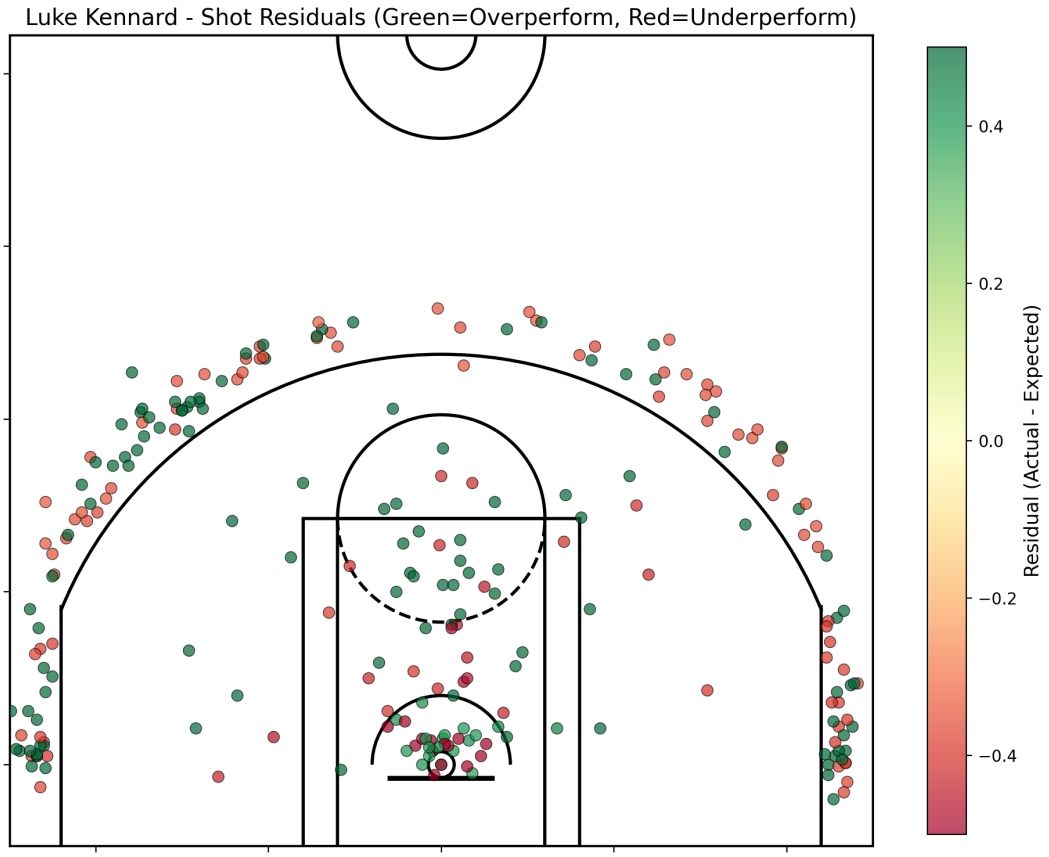


Figure 4: Residual heatmap for top overperformer (Kennard).

Data: shots_with_xp_2025-26.parquet residuals for Luke Kennard (top overperformer).
Filters: Player-specific; requires >=50 shots to render heatmap.
Exclusions: Players below the shot threshold.
Why: Show spatial pockets of over/underperformance vs xFG.

Shot Difficulty Index

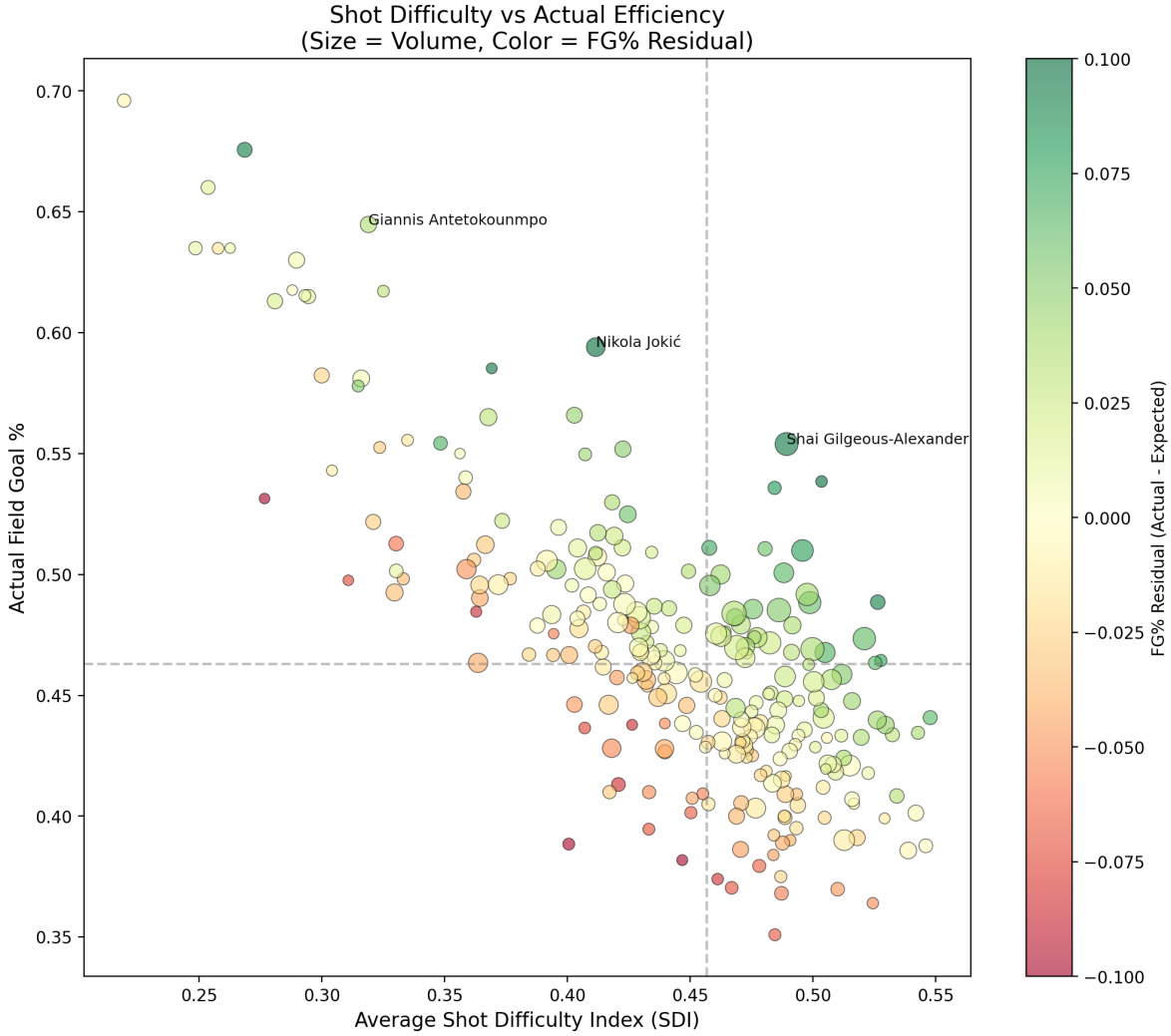


Figure 5: SDI vs xFG scatter plot. Size indicates volume, color indicates actual FG%.

Data: 2025-26 shots with xFG and SDI; player-level averages.
Filters: Players with >=200 attempts; point size scaled by volume.
Exclusions: Low-volume players.
Why: Relate shot difficulty to efficiency and residual performance.

Player Archetypes

```
cluster_summary <- clusters |>
  group_by(role, archetype) |>
  summarise(
    count = n(),
    avg_rim = mean(pct_rim, na.rm = TRUE),
    avg_3pt = mean(pct_3pt, na.rm = TRUE),
    avg_entropy = mean(distance_entropy, na.rm = TRUE),
    avg_usage = mean(usage_pct, na.rm = TRUE),
```

```
    avg_apg = mean(attempts_per_game, na.rm = TRUE),
    .groups = "drop"
  )
kable(cluster_summary, digits = 3, caption = "Player Archetype Summary")
```

Table 4: Player Archetype Summary

role	archetype	count	avg_rim	avg_3pt	avg_entropy	avg_usage	avg_apg
Big	Big – Paint-Dominant (High Usage)	6	0.798	0.090	1.309	0.237	11.077
Big	Big – Paint-Dominant (Low Usage)	21	0.900	0.068	1.030	0.162	7.458
Guard	Guard – Perimeter-Focused	23	0.283	0.607	1.409	0.200	10.324
Guard	Guard – Perimeter-Focused (Low Usage)	18	0.203	0.727	1.251	0.156	8.128
Wing	Wing – Mid-Range Specialist (High Usage)	25	0.498	0.289	1.542	0.269	15.071
Wing	Wing – Role Player (Mixed)	122	0.468	0.424	1.519	0.214	11.604
Wing	Wing – Role Player (Mixed) (Low Usage)	51	0.496	0.464	1.385	0.170	8.338

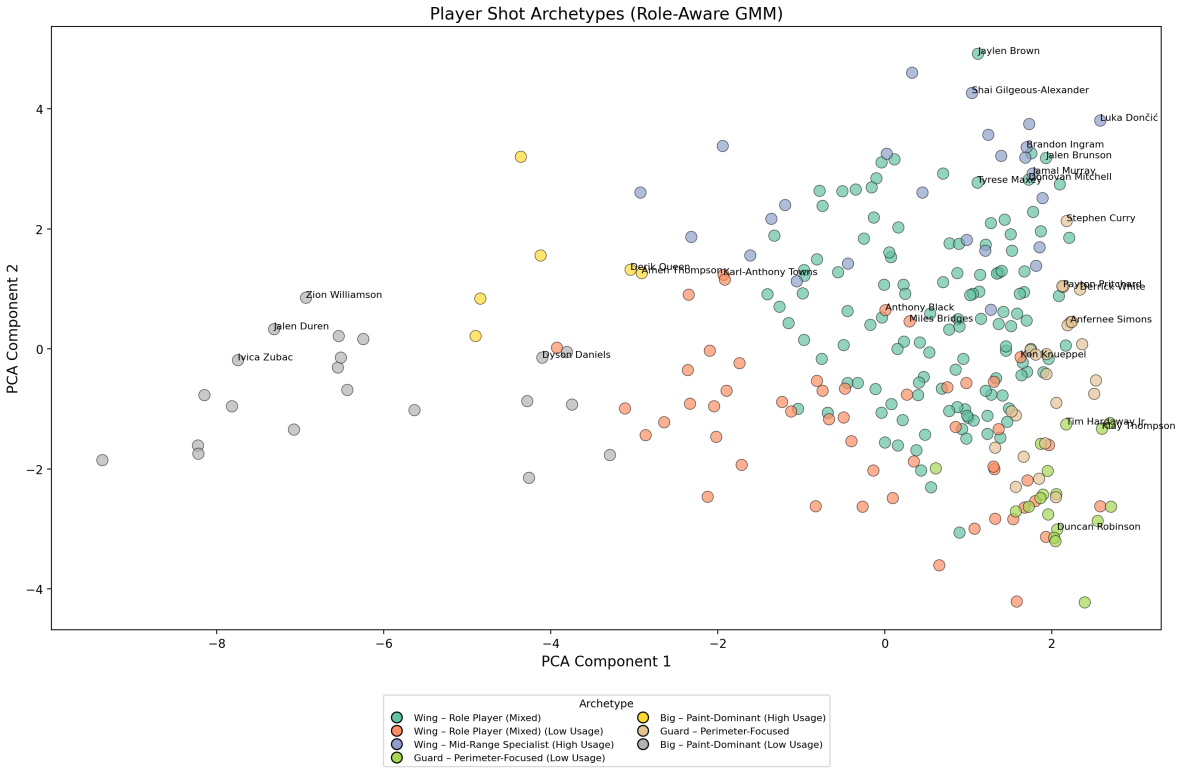


Figure 6: PCA visualization of player shot archetypes (color = archetype).

Data: Player feature matrix (shot diet, distance distribution, difficulty, usage) from 2025-26 shots.
Filters: Players with >=200 attempts; role-aware GMM clustering (k chosen by BIC).
Exclusions: Low-volume players; missing usage when unavailable.
Why: Visualize clustering of shot-profile archetypes (PCA used only for visualization).

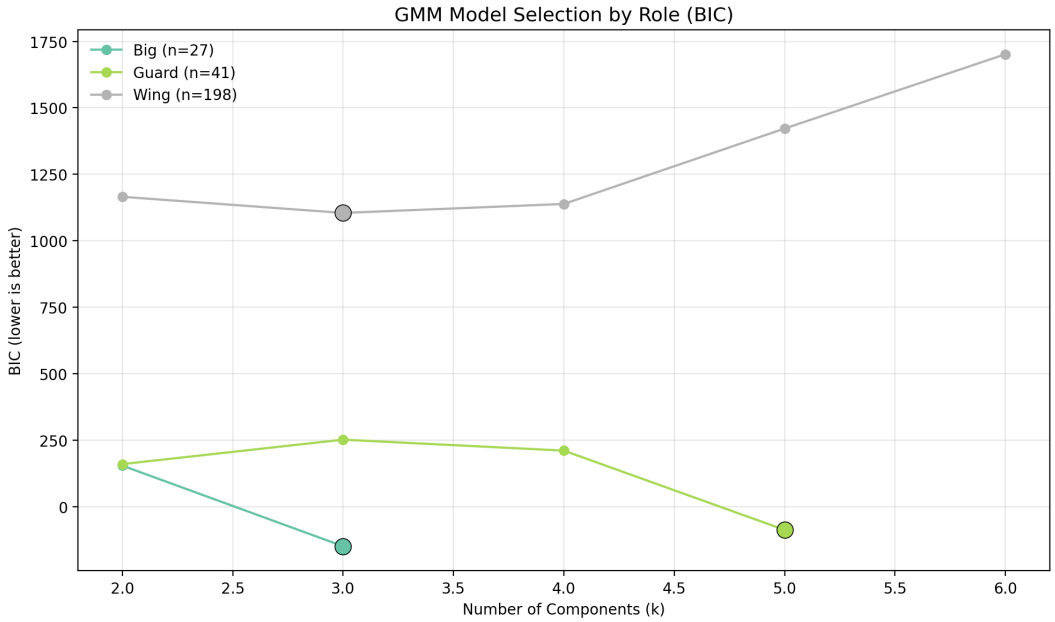


Figure 7: GMM model selection by role using BIC (lower is better). Selected k is highlighted for each role.

Data: Role-split player feature matrix; BIC evaluated for k=2..6.
Filters: Players with >=200 attempts.
Exclusions: Roles with fewer than two players skip model selection.
Why: Show how k was chosen per role and the relative tradeoff between fit and complexity.

Shot Density

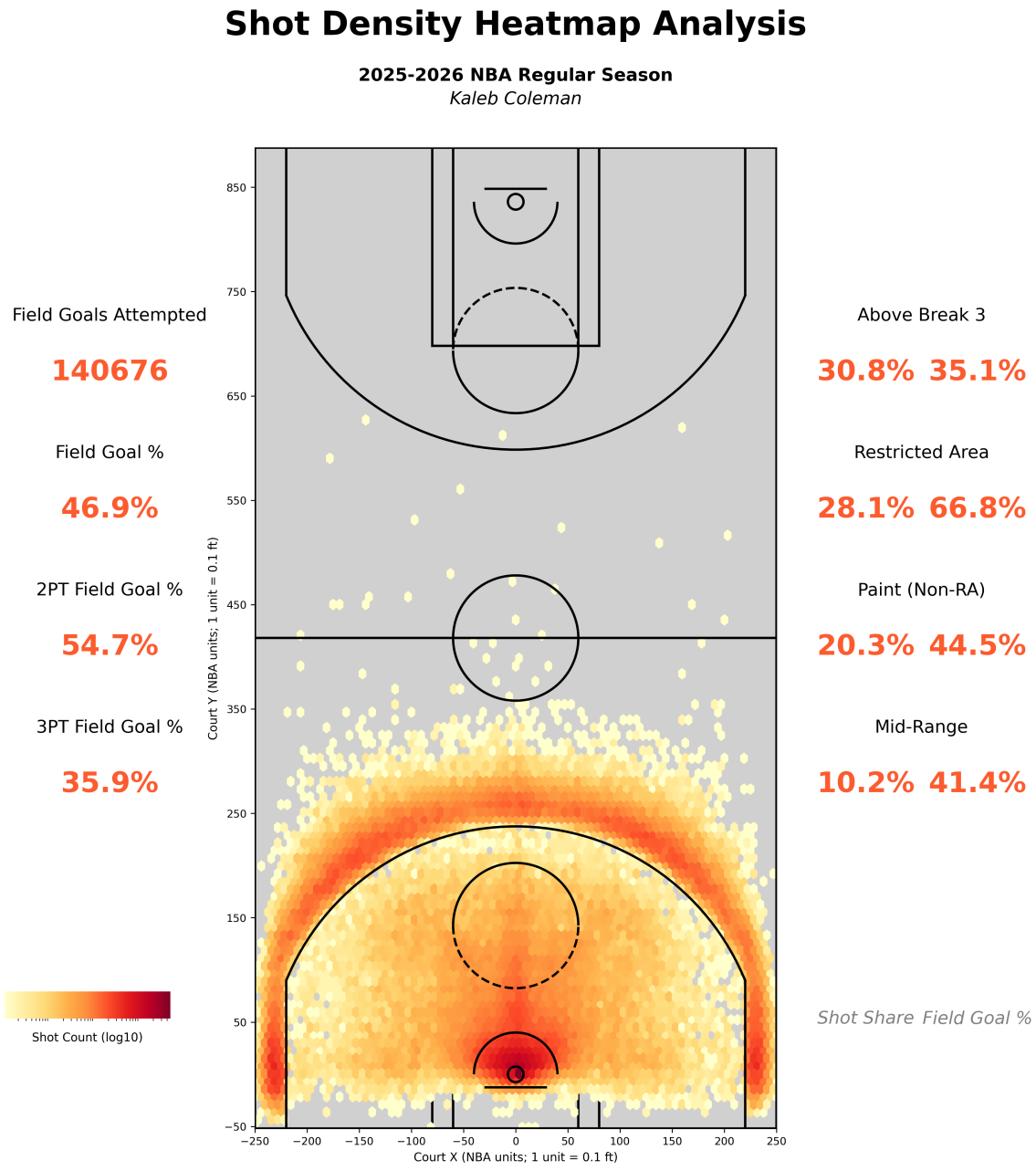


Figure 8: League-wide shot density heatmap for the 2025-26 regular season.

Data: 2025-26 regular-season shots from nba_stats_shots (SQLite).
Filters: SHOT_ATTEMPTED_FLAG=1; full-court hexbin density with log scaling.
Exclusions: None beyond missing rows in the source table.
Why: Show league-wide shot concentration patterns.

GAM Partial Dependence

The GAM highlights nonlinear effects via marginal log-odds contributions. The distance panel focuses on the marginal distance term and its widening uncertainty in the long-distance tail.

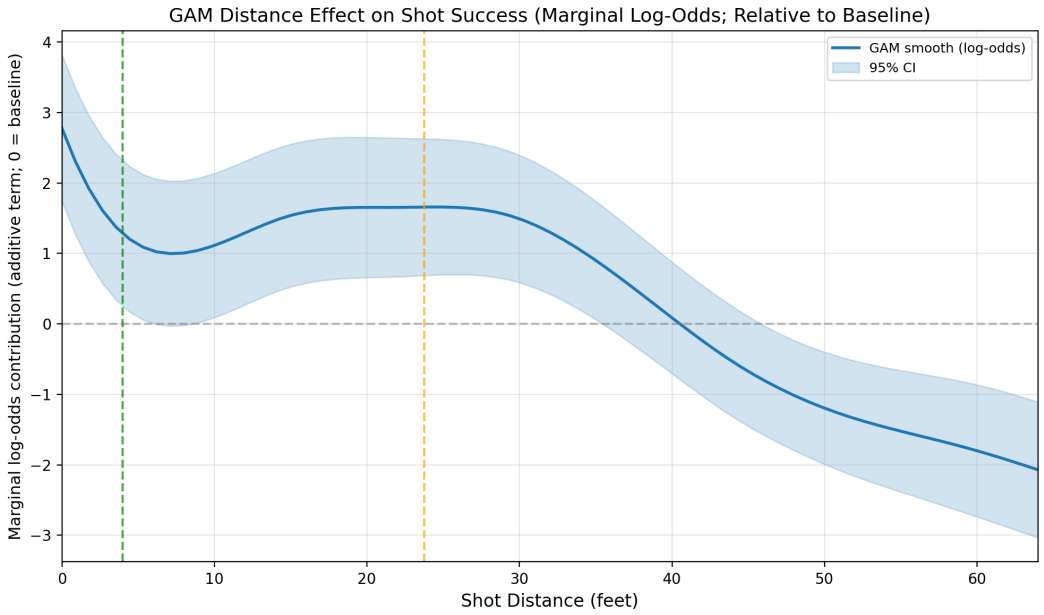


Figure 9: GAM distance partial dependence (marginal log-odds). Confidence intervals widen in the long-distance tail due to sparse samples.

Although the distance smooth can appear flat or slightly negative near the rim, it is a marginal log-odds term relative to the model baseline and other spatial effects. The rim advantage is largely captured by the spatial surface and correlated covariates, so the distance smooth reflects only incremental effects. This does not imply rim shots are difficult: league-wide FG% remains highest at the rim. At very long distances, sample sizes are tiny, so confidence intervals widen substantially and should be interpreted cautiously. Data: GAM fit on 2020-21 through 2024-25 regular-season shots (SQLite).
Filters: Model uses shots with valid location/clock/action.
Exclusions: Rows with missing distance or make flag.
Why: Isolate the marginal distance effect on shot success.

\newpage

Player Performance (POE Shot Charts)

Player-level POE shot charts show where individual shot-making exceeds or falls short of model expectations.

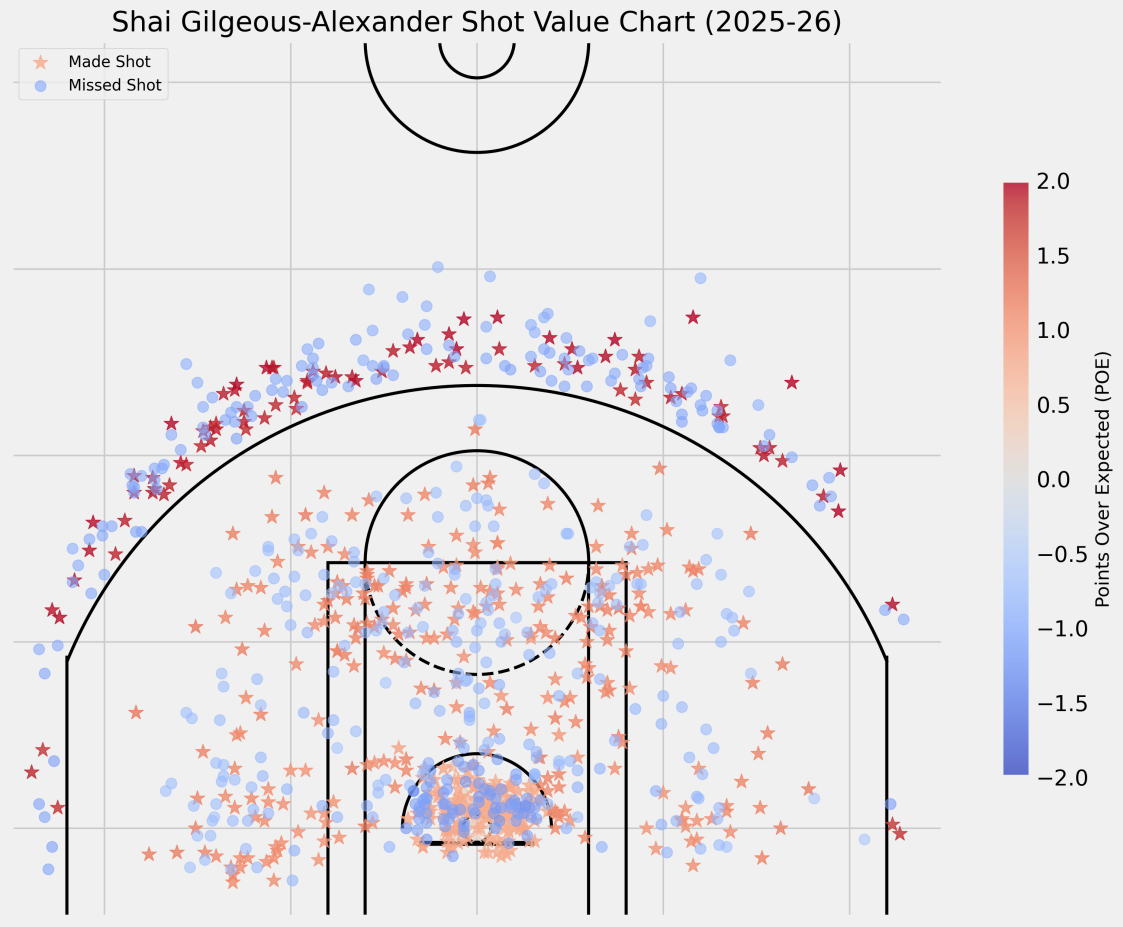


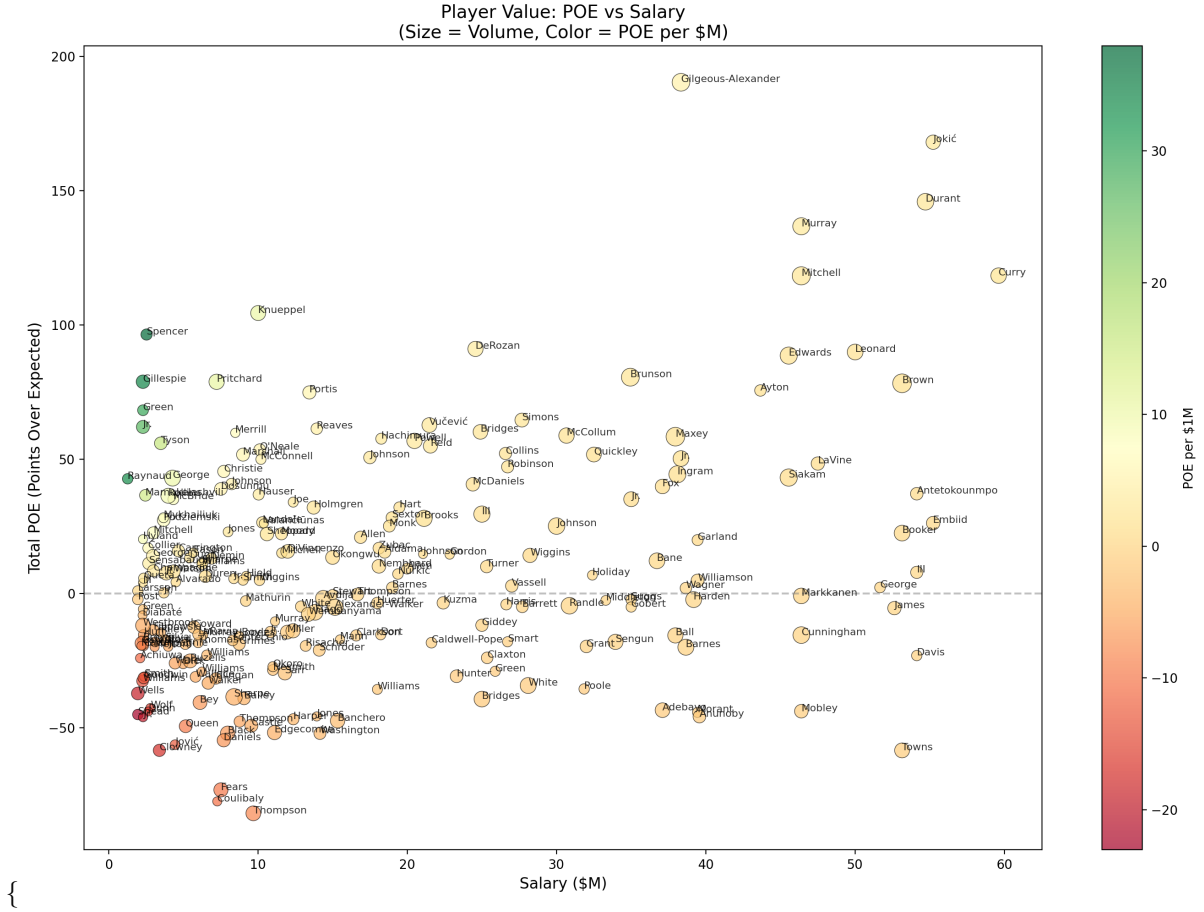
Figure 10: Shai Gilgeous-Alexander POE shot chart (2025-26).

Data: 2025-26 shots_with_xp for Shai Gilgeous-Alexander.
Filters: Player-specific; POE colored by make/miss.
Exclusions: None beyond missing shot metadata.
Why: Show where shot value exceeds or falls short of expectation.

Player Value (POE vs Salary)

We combine POE with salary estimates to contextualize performance efficiency. This is not a definitive value model (it omits defense and usage context), but it provides a first-order lens on shot-making return per $1M.

\begin{figure}[H]



{

}

\caption{Player value scatter: POE vs Salary with color encoding POE per \$1M.} \end{figure} Data: player_summary.csv (>=250 attempts) merged with 2024-25 salary data.
Filters: Players with salary match; point size by attempts.
Exclusions: Players without salary match; salary year lags by one season.
Why: Contextualize POE output relative to contract cost.

Player Value (FG Residual vs Salary)

POE can be influenced by free throws and usage context. To isolate *pure shot-making*, we also plot FG% residual (Actual FG% - Expected FG%) against salary. This view highlights which players beat expected shot quality on their field-goal attempts regardless of foul-drawing.

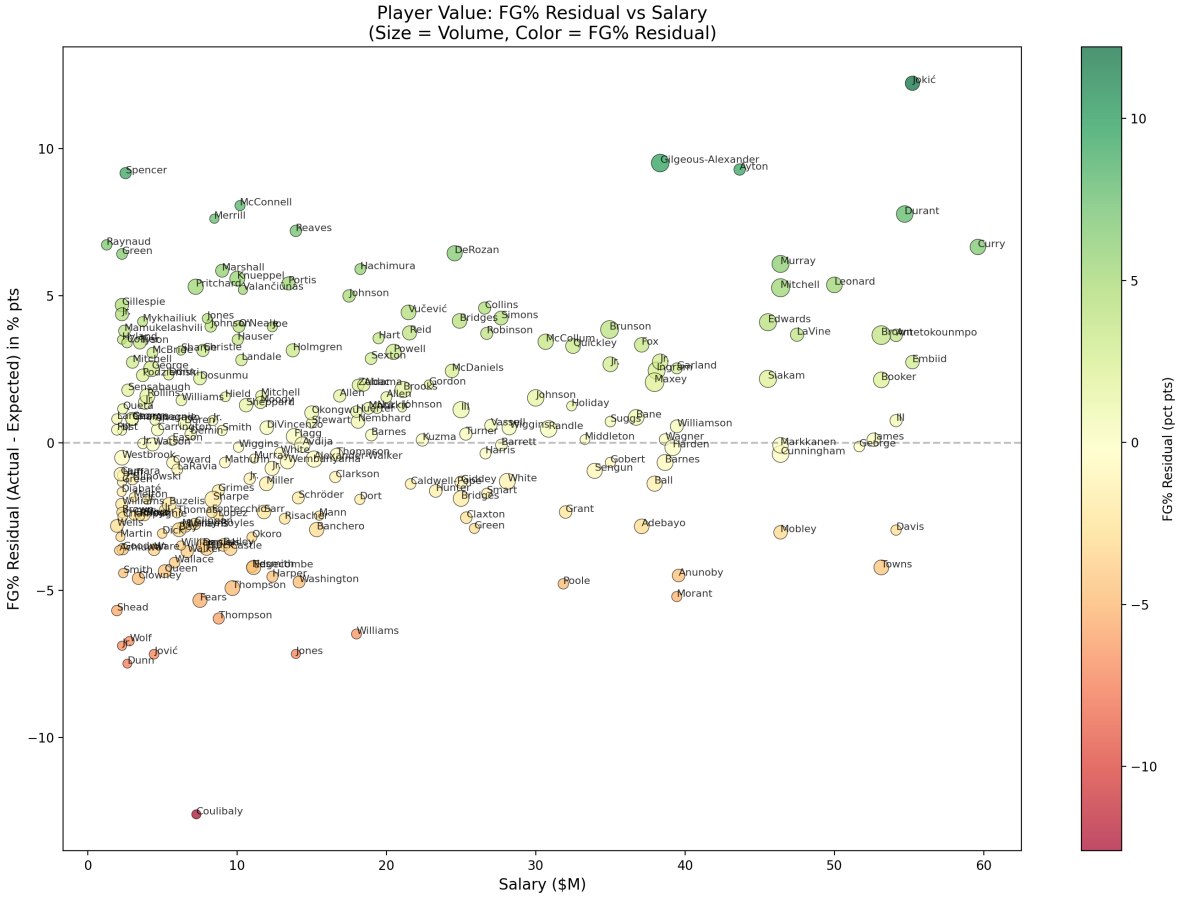


Figure 11: Player value scatter: FG% residual vs Salary. Positive values indicate above-expected shot making.

Data: Same merge as POE vs salary; FG residual = actual FG% - expected.
Filters: Players with salary match; point size by attempts.
Exclusions: Players without salary match; salary year lags by one season.
Why: Isolate pure shot-making efficiency relative to salary.

Discussion

Key Findings

1. **Model Performance**: The logistic regression xFG model achieves 63% accuracy, near the theoretical ceiling without defender tracking data.
2. **Overperformers**: Luke Kennard leads with +12.3% residual, indicating elite shot-making ability beyond what location and context predict.
3. **Shot Selection**: League-wide charts confirm the rim/three prioritization with suppressed mid-range volume.
4. **Shot Difficulty**: Players like Isaiah Joe and AJ Green take the most difficult shots (high SDI), while rim attackers like Gobert have low SDI.
5. **Archetypes**: Distinct *shot-profile* types emerged based on zone mix, SDI, and usage. Labels are spatial-descriptive:
    - **Paint-Dominant**: Rim/paint share above the season `q70` threshold.
    - **Perimeter-Focused**: 3PT share above the season `q70` threshold.
    - **Shot Creator (Mixed)**: Multi-zone profile with SDI above season median.
    - **Role Player (Mixed)**: Multi-zone profile with SDI at/below season median.
6. **Value Context**: POE per $1M adds a salary lens for identifying efficient shot makers relative to contract cost.
7. **Shot-Making Value**: FG% residual vs salary isolates pure shot-making value from free-throw effects, surfacing players who convert above expectation even without heavy foul drawing.
8. **Nonlinear Effects**: GAM partial dependence plots confirm strong non-linear penalties for shot distance, reinforcing the value of rim attempts and closer midrange looks.

Implications

- **Player Evaluation**: Residual analysis identifies players who consistently beat expectations, valuable for player development and trade evaluation.
- **Shot Selection**: SDI helps distinguish players who take difficult shots by choice vs. those forced into them.
- **Team Building**: Archetype clustering can inform roster construction and lineup optimization.
- **Contract Efficiency**: Salary overlays provide a first-order signal for value-based decision making.
- **Shot-Making ROI**: FG% residual vs salary separates shot-making efficiency from free throws, helping identify underpaid pure shooters.
- **Coaching Emphasis**: GAM patterns quantify the marginal penalty of longer-distance attempts, supporting shot selection priorities.

Limitations

This study uses league-wide averages and does not account for defensive pressure, which is the primary driver of shot difficulty. The xFG model is limited by the absence of tracking data. Salary data is scraped from a public source and may include reporting lags or contract nuances (current workflow uses 2024-25 salary data). FG residuals focus only on field-goal attempts and therefore do not capture value from free throws or playmaking.

The 5-season rolling window training ensures true out-of-sample evaluation, but assumes that shooting patterns from 2020-21 onward remain stable. Rule changes, court distance modifications, or significant style shifts could introduce distribution drift.

Conclusion

This paper presented an xFG model and three advanced analytics products for NBA shot analysis. The residual analysis, SDI, and player clustering provide actionable insights for player evaluation and team strategy. Future work should incorporate tracking data for defender distance and extend the analysis to team-level and lineup-level aggregations.

References

Basketball-Reference Player Contracts. n.d. Online. https://www.basketball-reference.com/contracts/players.html.

ESPN NBA API. n.d. Online. https://site.api.espn.com/apis/site/v2/sports/basketball/nba/scoreboard.

NBA Stats API. n.d. Online. https://stats.nba.com/stats/.