BasketballWorksheetKey

# SCORE Module Worksheet Key – NBA Role Evolution

# Monica Lazaro

# Load packages and dataset

library(readr)

Warning: package 'readr' was built under R version 4.4.3

library(dplyr)

Warning: package 'dplyr' was built under R version 4.4.3

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

library(ggplot2)

Warning: package 'ggplot2' was built under R version 4.4.3

nba\_all <- read\_csv("../Data/nba\_all.csv")

Rows: 2520 Columns: 34

── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (5): Player, Team, Pos, Awards, Player-additional  
dbl (29): Rk, Age, G, GS, MP, FG, FGA, FG%, 3P, 3PA, 3P%, 2P, 2PA, 2P%, eFG%...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Structure of data  
glimpse(nba\_all)

Rows: 2,520  
Columns: 34  
$ Rk <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,…  
$ Player <chr> "Michael Jordan", "Hakeem Olajuwon", "Shaquille O'…  
$ Age <dbl> 32, 33, 23, 32, 30, 22, 23, 32, 25, 30, 33, 22, 24…  
$ Team <chr> "CHI", "HOU", "ORL", "UTA", "SAS", "WSB", "DAL", "…  
$ Pos <chr> "SG", "C", "C", "PF", "C", "PF", "SF", "PF", "C", …  
$ G <dbl> 82, 72, 54, 82, 82, 15, 18, 71, 70, 81, 76, 81, 82…  
$ GS <dbl> 82, 72, 52, 82, 82, 15, 18, 71, 70, 81, 76, 81, 82…  
$ MP <dbl> 37.7, 38.8, 36.0, 38.0, 36.8, 37.2, 37.2, 37.1, 38…  
$ FG <dbl> 11.2, 10.7, 11.0, 9.6, 8.7, 10.0, 8.1, 8.2, 8.0, 7…  
$ FGA <dbl> 22.6, 20.8, 19.1, 18.5, 16.8, 18.4, 21.3, 16.3, 15…  
$ `FG%` <dbl> 0.495, 0.514, 0.573, 0.519, 0.516, 0.543, 0.379, 0…  
$ `3P` <dbl> 1.4, 0.0, 0.0, 0.2, 0.0, 1.0, 1.9, 0.7, 0.1, 2.8, …  
$ `3PA` <dbl> 3.2, 0.2, 0.0, 0.5, 0.1, 2.3, 5.7, 2.5, 0.4, 6.4, …  
$ `3P%` <dbl> 0.427, 0.214, 0.500, 0.400, 0.333, 0.441, 0.343, 0…  
$ `2P` <dbl> 9.8, 10.6, 10.9, 9.4, 8.6, 9.0, 6.1, 7.5, 7.9, 4.8…  
$ `2PA` <dbl> 19.4, 20.6, 19.1, 18.0, 16.7, 16.1, 15.6, 13.9, 14…  
$ `2P%` <dbl> 0.506, 0.517, 0.573, 0.522, 0.517, 0.558, 0.391, 0…  
$ `eFG%` <dbl> 0.525, 0.515, 0.574, 0.524, 0.517, 0.571, 0.424, 0…  
$ FT <dbl> 6.7, 5.5, 4.6, 6.2, 7.6, 2.7, 5.4, 6.2, 7.0, 5.2, …  
$ FTA <dbl> 8.0, 7.6, 9.5, 8.6, 10.0, 4.6, 7.4, 8.0, 10.2, 6.1…  
$ `FT%` <dbl> 0.834, 0.724, 0.487, 0.723, 0.761, 0.594, 0.729, 0…  
$ ORB <dbl> 1.8, 2.4, 3.4, 2.1, 3.9, 2.5, 2.1, 3.4, 3.1, 0.7, …  
$ DRB <dbl> 4.8, 8.4, 7.7, 7.7, 8.3, 5.1, 3.3, 8.1, 7.3, 2.7, …  
$ TRB <dbl> 6.6, 10.9, 11.0, 9.8, 12.2, 7.6, 5.4, 11.6, 10.4, …  
$ AST <dbl> 4.3, 3.6, 2.9, 4.2, 3.0, 5.0, 2.8, 3.7, 2.3, 3.1, …  
$ STL <dbl> 2.2, 1.6, 0.6, 1.7, 1.4, 1.8, 0.8, 1.6, 1.0, 1.5, …  
$ BLK <dbl> 0.5, 2.9, 2.1, 0.7, 3.3, 0.6, 0.2, 0.8, 2.7, 0.2, …  
$ TOV <dbl> 2.4, 3.4, 2.9, 2.4, 2.3, 3.3, 3.1, 3.1, 3.7, 2.7, …  
$ PF <dbl> 2.4, 3.4, 3.6, 3.0, 3.2, 3.4, 2.2, 2.9, 3.5, 2.9, …  
$ PTS <dbl> 30.4, 26.9, 26.6, 25.7, 25.0, 23.7, 23.4, 23.2, 23…  
$ Awards <chr> "MVP-1DPOY-6ASNBA1DEF1", "MVP-4DPOY-5ASNBA2DEF2", …  
$ `Player-additional` <chr> "jordami01", "olajuha01", "onealsh01", "malonka01"…  
$ season <dbl> 1996, 1996, 1996, 1996, 1996, 1996, 1996, 1996, 19…  
$ Era <dbl> 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 19…

# Warm-Up Answers:

# 1. Columns include: Player, Pos, Age, Team, PTS, AST, TRB, 3P, 3PA, FG%, etc.

# 2. Era = a label based on season: 1990s, 2000s, 2010s, 2020s

# 3. Total rows:

nrow(nba\_all)

[1] 2520

length(unique(nba\_all$Player))

[1] 1699

# Summarize Key Stats

nba\_all |>   
 group\_by(Era) |>   
 summarise(  
 avg\_pts = mean(as.numeric(PTS), na.rm = TRUE),  
 avg\_ast = mean(as.numeric(AST), na.rm = TRUE),  
 avg\_trb = mean(as.numeric(TRB), na.rm = TRUE),  
 avg\_3p = mean(as.numeric(`3P`), na.rm = TRUE),  
 avg\_3pa = mean(as.numeric(`3PA`), na.rm = TRUE),  
 avg\_fg\_pct = mean(as.numeric(`FG%`), na.rm = TRUE)  
 )

# A tibble: 4 × 7  
 Era avg\_pts avg\_ast avg\_trb avg\_3p avg\_3pa avg\_fg\_pct  
 <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 1990 7.98 1.86 3.43 0.453 1.27 0.444  
2 2000 7.78 1.74 3.43 0.463 1.32 0.427  
3 2010 7.80 1.81 3.35 0.632 1.87 0.427  
4 2020 8.02 1.93 3.24 0.909 2.58 0.448

# Observations:

# - 3P highest in 2020s

# - Assists increased over time

# - Suggests more team passing and shooting roles

# — Part 2: Visualizations —

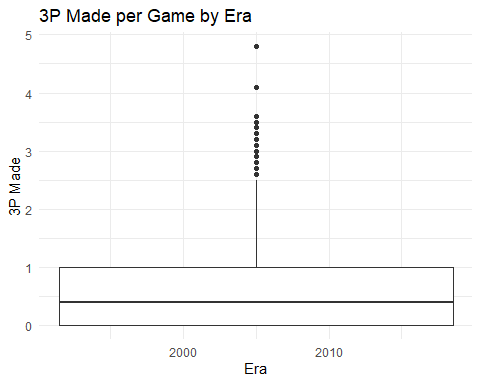
# Boxplot: 3P

ggplot(nba\_all, aes(x = Era, y = as.numeric(`3P`), fill = Era)) +  
 geom\_boxplot() +  
 labs(title = "3P Made per Game by Era", y = "3P Made", x = "Era") +  
 theme\_minimal()

Warning: Continuous x aesthetic  
ℹ did you forget `aes(group = ...)`?

Warning: Removed 4 rows containing non-finite outside the scale range  
(`stat\_boxplot()`).

Warning: The following aesthetics were dropped during statistical transformation: fill.  
ℹ This can happen when ggplot fails to infer the correct grouping structure in  
 the data.  
ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
 variable into a factor?



# Answers:

# - 2020s = highest median

# - Variability grew by era

# - Matches numeric summary

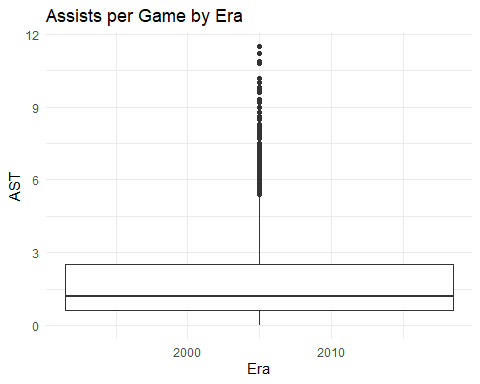
# Boxplot: AST

ggplot(nba\_all, aes(x = Era, y = as.numeric(AST), fill = Era)) +  
 geom\_boxplot() +  
 labs(title = "Assists per Game by Era", y = "AST", x = "Era") +  
 theme\_minimal()

Warning: Continuous x aesthetic  
ℹ did you forget `aes(group = ...)`?

Warning: Removed 4 rows containing non-finite outside the scale range  
(`stat\_boxplot()`).

Warning: The following aesthetics were dropped during statistical transformation: fill.  
ℹ This can happen when ggplot fails to infer the correct grouping structure in  
 the data.  
ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
 variable into a factor?

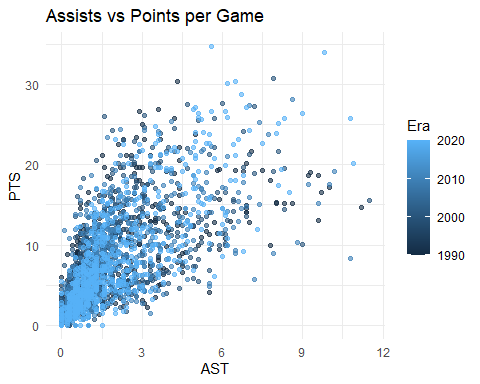


# Answer: Assists have risen steadily — more ball movement

# Scatterplot: AST vs PTS

ggplot(nba\_all, aes(x = as.numeric(AST), y = as.numeric(PTS), color = Era)) +  
 geom\_point(alpha = 0.6) +  
 labs(title = "Assists vs Points per Game", x = "AST", y = "PTS") +  
 theme\_minimal()

Warning: Removed 4 rows containing missing values or values outside the scale range  
(`geom\_point()`).



# Answer: Strong correlation — more assists usually = more scoring

# — Part 3: Regression Model —

model <- lm(PTS ~ AST + TRB + `3P`, data = nba\_all)  
summary(model)

Call:  
lm(formula = PTS ~ AST + TRB + `3P`, data = nba\_all)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-16.1241 -1.3918 -0.1122 1.0055 13.1701   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -0.48980 0.10115 -4.843 1.36e-06 \*\*\*  
AST 1.08721 0.03574 30.418 < 2e-16 \*\*\*  
TRB 1.30174 0.02304 56.509 < 2e-16 \*\*\*  
`3P` 3.18089 0.08554 37.186 < 2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 2.651 on 2512 degrees of freedom  
 (4 observations deleted due to missingness)  
Multiple R-squared: 0.7997, Adjusted R-squared: 0.7994   
F-statistic: 3343 on 3 and 2512 DF, p-value: < 2.2e-16

# Interpretation:

# - All predictors likely significant

# - Positive coefficients → all contribute to PTS

# - R² around 0.6–0.7

# - Doesn’t capture defense or spacing roles

# — Part 4: Clustering —

nba\_cluster <- nba\_all |>   
 mutate(  
 PTS = as.numeric(PTS),  
 AST = as.numeric(AST),  
 TRB = as.numeric(TRB),  
 `3P` = as.numeric(`3P`)  
 ) |>   
 filter(!is.na(PTS), !is.na(AST), !is.na(TRB), !is.na(`3P`))  
  
set.seed(123)  
k\_result <- kmeans(scale(nba\_cluster |> select(PTS, AST, TRB, `3P`)), centers = 4, nstart = 20)  
  
nba\_cluster$Cluster <- as.factor(k\_result$cluster)  
  
table(nba\_cluster$Cluster)

1 2 3 4   
 291 369 1265 591

# Answers:

# - Cluster 1 = scorers

# - Cluster 2 = passers

# - Cluster 3 = bigs/rebounders

# - Cluster 4 = bench/low usage

# — Part 5: Shiny App —

# Setup

library(shiny)  
  
nba\_data <- read\_csv("../Data/nba\_all.csv")|>   
 mutate(  
 PTS = as.numeric(PTS),  
 AST = as.numeric(AST),  
 TRB = as.numeric(TRB),  
 `3P` = as.numeric(`3P`)  
 ) |>   
 filter(!is.na(PTS), !is.na(AST), !is.na(TRB), !is.na(`3P`))

Rows: 2520 Columns: 34  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (5): Player, Team, Pos, Awards, Player-additional  
dbl (29): Rk, Age, G, GS, MP, FG, FGA, FG%, 3P, 3PA, 3P%, 2P, 2PA, 2P%, eFG%...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

set.seed(123)  
k\_result <- kmeans(scale(nba\_data |> select(PTS, AST, TRB, `3P`)), centers = 4, nstart = 20)  
nba\_data$Cluster <- as.factor(k\_result$cluster)

# UI

library(shiny)  
  
nba\_data <- read\_csv("../Data/nba\_all.csv")|>   
 mutate(  
 PTS = as.numeric(PTS),  
 AST = as.numeric(AST),  
 TRB = as.numeric(TRB),  
 `3P` = as.numeric(`3P`)  
 ) |>   
 filter(!is.na(PTS), !is.na(AST), !is.na(TRB), !is.na(`3P`))

Rows: 2520 Columns: 34  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (5): Player, Team, Pos, Awards, Player-additional  
dbl (29): Rk, Age, G, GS, MP, FG, FGA, FG%, 3P, 3PA, 3P%, 2P, 2PA, 2P%, eFG%...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

set.seed(123)  
k\_result <- kmeans(scale(nba\_data |> select(PTS, AST, TRB, `3P`)), centers = 4, nstart = 20)  
nba\_data$Cluster <- as.factor(k\_result$cluster)

# Server

server <- function(input, output) {  
 filtered\_data <- reactive({  
 data <- nba\_data |> filter(Era == input$era)  
 if (!is.null(input$pos)) data <- data |> filter(Pos %in% input$pos)  
 if (input$search != "") data <- data |> filter(grepl(input$search, Player, ignore.case = TRUE))  
 data  
 })  
  
 output$scatterPlot <- renderPlot({  
 ggplot(filtered\_data(), aes(x = AST, y = PTS, color = Cluster)) +  
 geom\_point(alpha = 0.6) +  
 labs(title = "AST vs. PTS by Cluster", x = "Assists", y = "Points")  
 })  
  
 output$boxPlot <- renderPlot({  
 ggplot(filtered\_data(), aes(x = Cluster, y = `3P`, fill = Cluster)) +  
 geom\_boxplot() +  
 labs(title = "3P by Cluster", y = "3P Made", x = "Cluster")  
 })  
  
 output$playerTable <- renderTable({  
 filtered\_data() |> select(Player, Pos, Era, Cluster, PTS, AST, TRB, `3P`)  
 })  
}  
  
# UI and server code commented out just for the sake of renering- include this in your code without it commented out!-... shinyApp(ui = ui, server = server)

# Answers:

# - Interactive filtering helps explore clusters and roles

# - Most recent eras dominate scoring clusters

# — Part 6: Wrap-Up Answer —

# Example summary:

# Modern players are more perimeter-oriented, with more 3-point attempts and assists.

# Regression shows assists and 3P are top predictors of scoring.

# Clusters reveal distinct role types: scorers, facilitators, rebounders.

# Visuals and app show how spacing and guard play have evolved over 3 decades.

# Combining these tools gives us a full picture of NBA role evolution.

# Bonus: Try other cluster counts

set.seed(123)  
k\_result\_alt <- kmeans(scale(nba\_data |> select(PTS, AST, TRB, `3P`)), centers = 3, nstart = 20)  
table(k\_result\_alt$cluster)

1 2 3   
 389 597 1530