Visualizing NBA

Junqi Fu

2023-10-21

Contents

Chapter 1

NBA Data

The data set contains all the players' performance data from season 1996-97 to season 2022-23.

1.1 Data Type

Here's a brief explanation of some important variable in the analysis:

- **player_height**: player's height in the given season.
- **player_weight**: player's weight in the given season.
- **gp**: total games a player has played in the given season.
- **pts**: player's average points per game.
- **reb**: player's average rebound per game.
- **ast**: player's average assist per game.
- **net_rating**: the team's point differential per 100 possessions while a player is on court.
- **oreb_pct**: offensive rebound percentage - an estimate of the percentage of available offensive rebounds a player grabbed.
- **dreb_pct**: defensive rebound percentage - an estimate of the percentage of available defensive rebounds a player grabbed.
- **usg_pct**: usage percentage - an estimate of the percentage of team plays used by a player.
- **ts_pct**: true shooting percentage - a measure of shooting efficiency that takes into account field goals, 3-point field goals, and free throws.
- **ast_pct**: assist percentage - an estimate of the percentage of teammate field goals a player assisted.
- **season**: season - the NBA season for which these stats apply.

1.2 Preliminary Cleaning

1.2.1 Initial Missing Data Imputation

```
sum(is.na(nba))
```

```
## [1] 0
```

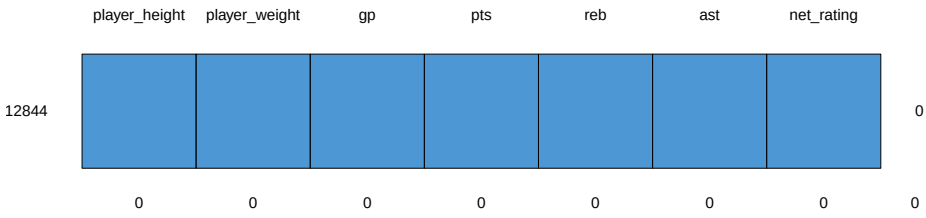
```
md.pattern(nba)
```

```
##  /\     /\
## {  `---'  }
## {  0   0  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```


```
##       X player_name team_abbreviation age player_height player_weight college
## 12844 1           1                 1   1             1             1       1
##       0           0                 0   0             0             0       0
##       country draft_year draft_round draft_number gp pts reb ast net_rating
## 12844       1          1           1            1  1   1   1   1          1
##             0          0           0            0  0   0   0   0          0
##       oreb_pct dreb_pct usg_pct ts_pct ast_pct season
## 12844        1        1       1      1       1      1 0
##              0        0       0      0       0      0 0
```

```
md.pattern(subset(nba,select=c(player_height,player_weight,gp,pts,reb,ast,net_rating)))
```

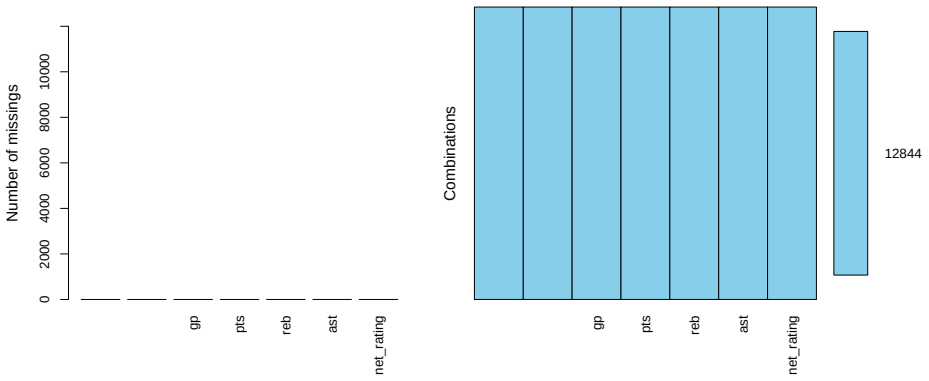
```
##  /\     /\
## {  `---'  }
## {  0   0  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##       player_height player_weight gp pts reb ast net_rating
## 12844             1             1  1   1   1   1          1 0
```

```
##                    0               0   0   0   0   0           0 0
```

```
aggr(subset(nba,select=c(player_height,player_weight,gp,pts,reb,ast,net_rating)),prop=F,numbers=T
```



1.2.2 Data Cleaning

identify outlier exists in the net_rating, with the 300 as the max and -250 as the min.

```
summary(nba)
```

```
##        X         player_name        team_abbreviation       age
##  Min.   :    0   Length:12844       Length:12844       Min.   :18.00
##  1st Qu.: 3211   Class :character   Class :character   1st Qu.:24.00
##  Median : 6422   Mode  :character   Mode  :character   Median :26.00
##  Mean   : 6422                                         Mean   :27.05
##  3rd Qu.: 9632                                         3rd Qu.:30.00
##  Max.   :12843                                         Max.   :44.00
##  player_height   player_weight      college            country
##  Min.   :160.0   Min.   : 60.33   Length:12844       Length:12844
##  1st Qu.:193.0   1st Qu.: 90.72   Class :character   Class :character
##  Median :200.7   Median : 99.79   Mode  :character   Mode  :character
##  Mean   :200.6   Mean   :100.26
##  3rd Qu.:208.3   3rd Qu.:108.86
##  Max.   :231.1   Max.   :163.29
##   draft_year        draft_round        draft_number            gp
##  Length:12844       Length:12844       Length:12844       Min.   : 1.00
##  Class :character   Class :character   Class :character   1st Qu.:31.00
##  Mode  :character   Mode  :character   Mode  :character   Median :57.00
##                                                           Mean   :51.15
##                                                           3rd Qu.:73.00
##                                                           Max.   :85.00
##       pts              reb              ast           net_rating
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   :-
250.000
```

```
##  1st Qu.: 3.600   1st Qu.: 1.800   1st Qu.: 0.600   1st Qu.:  -
6.400
##  Median : 6.700   Median : 3.000   Median : 1.200   Median :  -
1.300
##  Mean   : 8.213   Mean   : 3.558   Mean   : 1.825   Mean   :  -
2.226
##  3rd Qu.:11.500   3rd Qu.: 4.700   3rd Qu.: 2.400   3rd Qu.:   3.200
##  Max.   :36.100   Max.   :16.300   Max.   :11.700   Max.   : 300.000
##     oreb_pct          dreb_pct         usg_pct          ts_pct
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.02100   1st Qu.:0.0960   1st Qu.:0.1490   1st Qu.:0.4820
##  Median :0.04000   Median :0.1305   Median :0.1810   Median :0.5250
##  Mean   :0.05407   Mean   :0.1406   Mean   :0.1846   Mean   :0.5131
##  3rd Qu.:0.08300   3rd Qu.:0.1790   3rd Qu.:0.2170   3rd Qu.:0.5630
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.5000
##     ast_pct          season
##  Min.   :0.0000   Length:12844
##  1st Qu.:0.0660   Class :character
##  Median :0.1030   Mode  :character
##  Mean   :0.1316
##  3rd Qu.:0.1790
##  Max.   :1.0000
```

```
nba_clean<-(which(nba$net_rating>=300 |nba$net_rating<=-200))
nba<-nba[-nba_clean] #remove the outliers.
```

1.3 Preliminary Descriptive Statistics

- **age**: the range is between a minimum of 18 years and a maximum of 44 years, with the average age being 26 years old.

```
summary(nba$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   24.00   26.00   27.05   30.00   44.00
```

- **height**: the range is between a minimum of 160 cm and a maximum of 231.1 cm, with the average height as 200.7 cm.

```
summary(nba$player_height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   160.0   193.0   200.7   200.6   208.3   231.1
```

- **weight**: the range is between a minimum of 60.33 kg and a maximum of 163.29 kg, with the average as 100 kg.

```
summary(nba$player_weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.33   90.72   99.79  100.26  108.86  163.29
```

- **player**: with a highly competitive threshold, only 2551 players have played in the league since 1996.

```
summary(unique(nba$player_name))
```

```
##    Length     Class      Mode
##      2551 character character
```

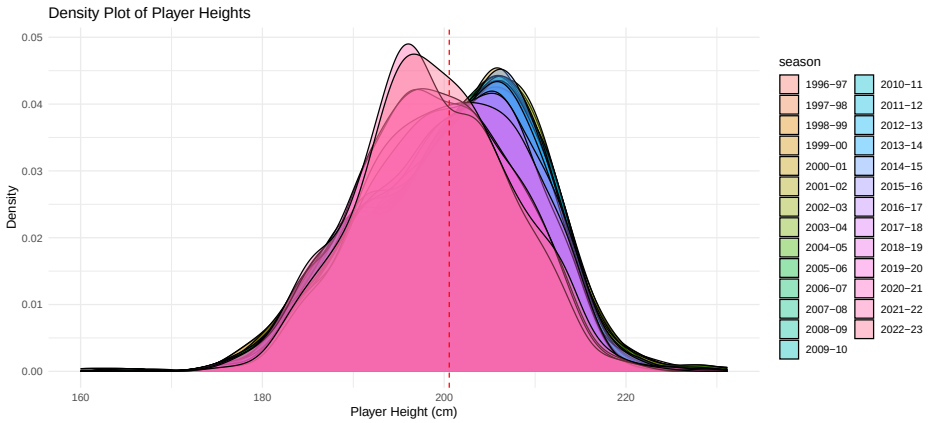
Chapter 2

Hypothesis

2.1 Density Plot for Height and Weight

2.1.1 Density Plot for Height Distribution Over Seasons

```
heightplot <- ggplot(nba, aes(x = player_height)) +
  geom_density(aes(fill = season), alpha = 0.4) +
  geom_vline(aes(xintercept = mean(player_height)), linetype = "dashed", color = "red") +
  ggtitle("Density Plot of Player Heights") +
  xlab("Player Height (cm)") +
  ylab("Density") +
  theme_minimal()
print(heightplot)
```

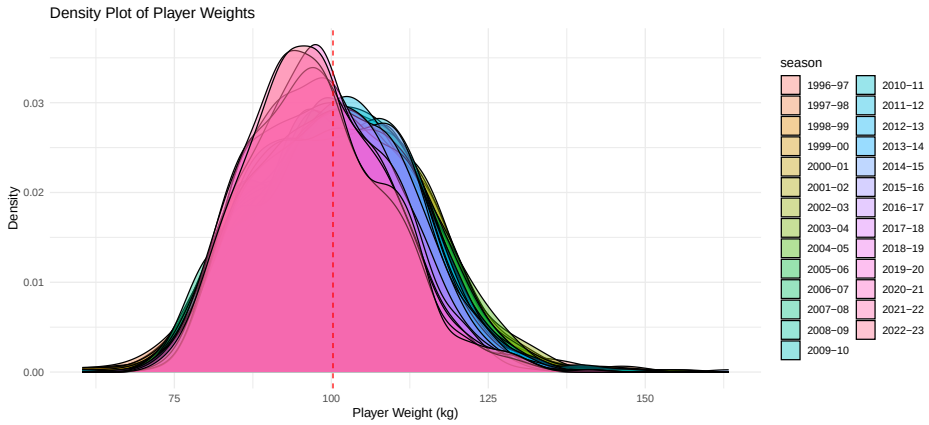


2.1.1.1 Summary

- A more noticeable shift in the mean height of players over seasons. The mean height has decreased steadily.
- A narrower range of heights among players in the recent NBA compared to the past.
- There's a noticeable peak around the 200-210 cm range, suggesting that a significant number of players fall within this height bracket.

2.1.2 Density Plot for Weight Distribution Over Seasons

```
weightplot <- ggplot(nba, aes(x = player_weight)) +
  geom_density(aes(fill = season), alpha = 0.4) +
  geom_vline(aes(xintercept = mean(player_weight)), linetype = "dashed", color = "red")
  ggtitle("Density Plot of Player Weights") +
  xlab("Player Weight (kg)") +
  ylab("Density") +
  theme_minimal()
print(weightplot)
```



2.1.2.1 Summary

- The mean weight seems to have shifted towards the left, suggesting that players have, on average, become lighter over the years.
- The distributions for the recent seasons appear narrower, indicating less variability in player weights now than in earlier seasons.

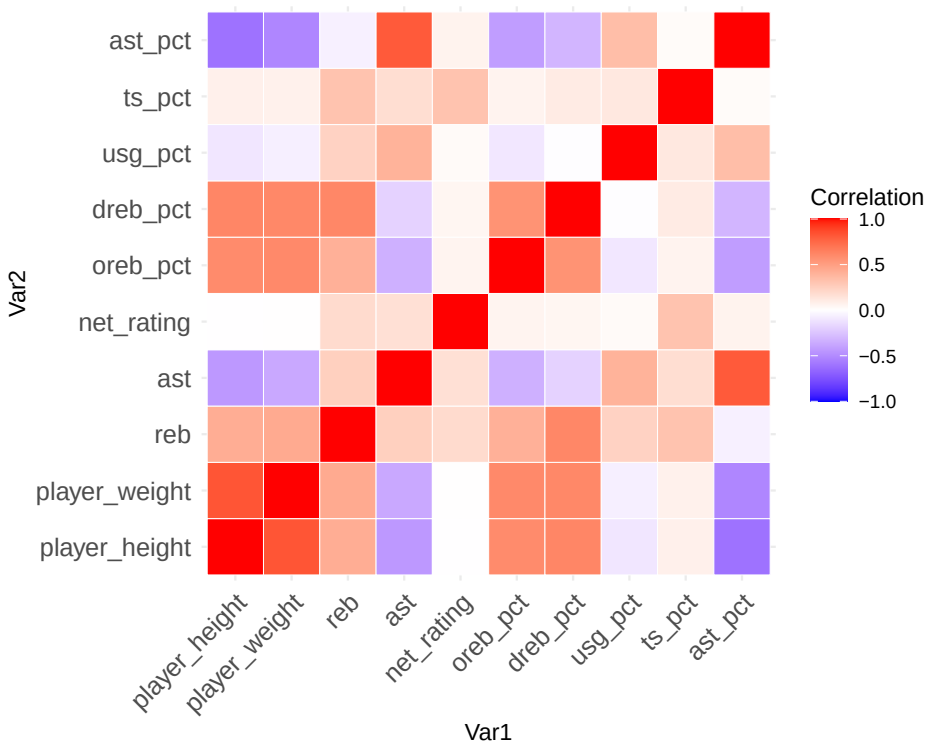
2.1.3 In summary, over the years, NBA players have, on average, become shorter and lighter, with a narrower range of both weights and heights represented in recent seasons.

2.2 Inference & Hypothesis Based On The Density Plot and Initial Summary

- **H1** Versatility & Position-less Basketball: the narrower range of heights and weights indicates that there might be an increasing trend of "position-less" basketball, specifically players are no longer strictly confined to traditional roles based on their physical attributes.
- **H2** Evolution in Playing Style: the traditional center-focused style of play cannot adapt to the pace of the modern NBA.
- **H3** Defensive Switching: With players might having a wider range of skills irrespective of their height or weight, teams can employ more switching on defense. Players are more equipped to defend multiple positions, making it harder for offenses to exploit mismatches.

2.3 Correlation analysis

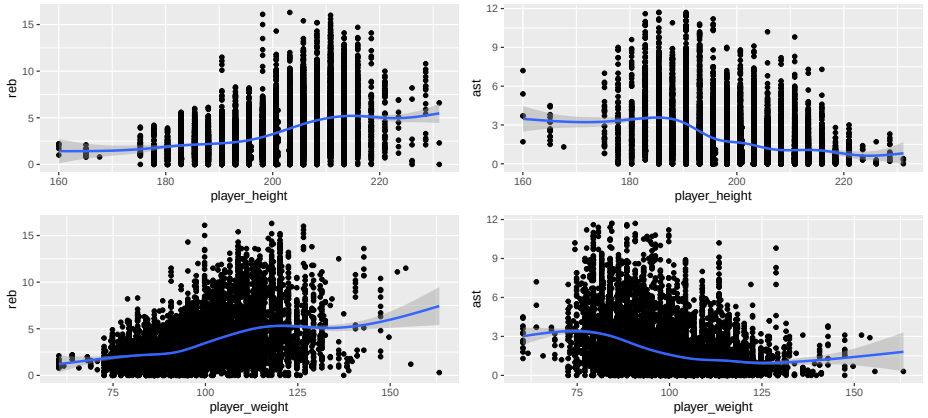
```
ggplot(data = melted_cormat, aes(x=Var1, y=Var2)) +
  geom_tile(aes(fill=value), color='white') +
  scale_fill_gradient2(low='blue', high='red', mid='white', midpoint=0, limit=c(-1,1), space='Lab
  theme_minimal() +
  theme(axis.text.x=element_text(angle=45, vjust=1, size=12, hjust=1),
        axis.text.y=element_text(size=12)) +
  coord_fixed()
```



2.3.1.1 Correlational Matrix.

```
grid.arrange(height_reb+geom_smooth(), height_ast+geom_smooth(), weight_reb+geom_smooth
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

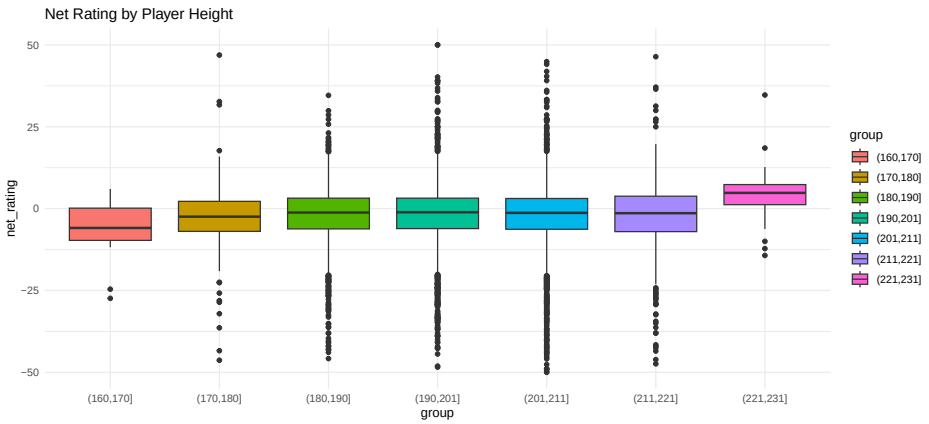


```r
heightdatapart <- data.frame(player_height = nba$player_height, net_rating = nba$net_rating)

heightdatapart <- heightdatapart[(heightdatapart$net_rating >= -50 & heightdatapart$net_rating <=

heightdatapart$group <- cut(heightdatapart$player_height, breaks = 7)

ph <- ggplot(data = heightdatapart, aes(x = group, y = net_rating, fill = group)) +
  geom_boxplot() +
  ggtitle("Net Rating by Player Height") +
  theme_minimal()
print(ph)
```

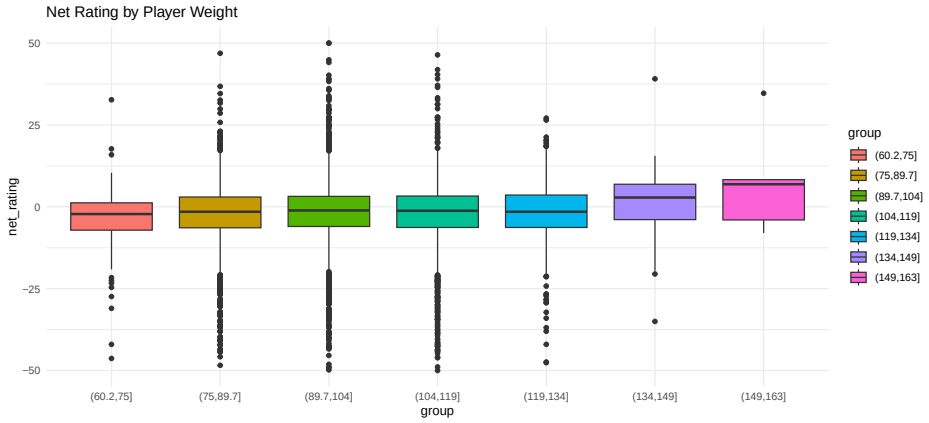



```r
weightdatapart <- data.frame(player_weight = nba$player_weight, net_rating = nba$net_rating)

weightdatapart <- weightdatapart[(weightdatapart$net_rating >= -50 & weightdatapart$net_rating <=

weightdatapart$group <- cut(weightdatapart$player_weight, breaks = 7)

pw <- ggplot(data = weightdatapart, aes(x = group, y = net_rating, fill = group)) +
  geom_boxplot() +
  ggtitle("Net Rating by Player Weight") +
  theme_minimal()
print(pw)
```

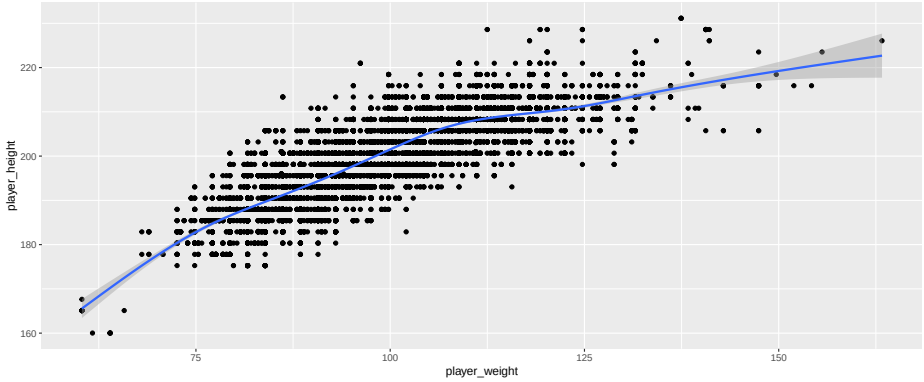

Net Rating by Player Weight
net_rating
50
25
0
−25
−50
(60.2,75]
(75,89.7]
(89.7,104]
(104,119]
(119,134]
(134,149]
(149,163]
group
group
(60.2,75]
(75,89.7]
(89.7,104]
(104,119]
(119,134]
(134,149]
(149,163]

Chapter 3

In-depth Analysis Between Weight and Height

```
p1<-ggplot(nba,aes(x=player_weight,y=player_height))
p1<-p1+geom_point()
p1+geom_smooth()
```

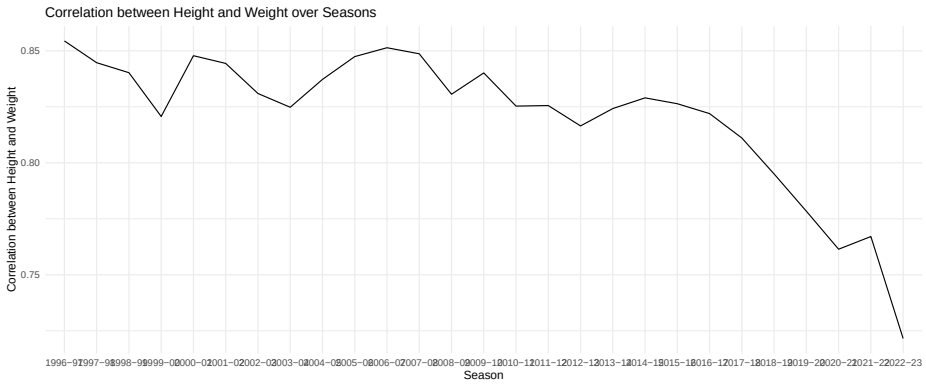
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



3.1 Time-series Visualization

```
p <- ggplot(cor_df, aes(x = season, y = correlation, group = 1)) +  # Set group=1 to connect all
  geom_line() +
  xlab("Season") +
  ylab("Correlation between Height and Weight") +
  ggtitle("Correlation between Height and Weight over Seasons") +
```

```
  theme_minimal()
print(p)
```



3.2 Clustering

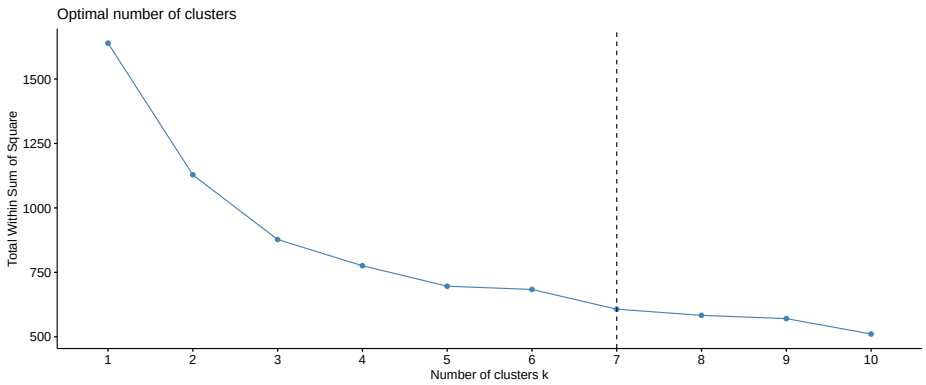
3.2.1 k-means Clustering

```
nba_selected_season <- nba[(which(nba$season == "2022-23")),]
nba_selected_season<-nba_selected_season[nba_selected_season$pts>10,]
nba_selected_season<-nba_selected_season[nba_selected_season$gp>50,]
rownames(nba_selected_season)<-nba_selected_season$player_name

selected_features <- c('player_height', 'pts','player_weight', 'reb', 'ast', 'net_ratin
nba_for_clustering <- nba_selected_season %>% select(all_of(selected_features))

df <- as.data.frame(scale(nba_for_clustering))


fviz_nbclust(df, kmeans, method = "wss") + geom_vline(xintercept = 7, linetype = 2)
```



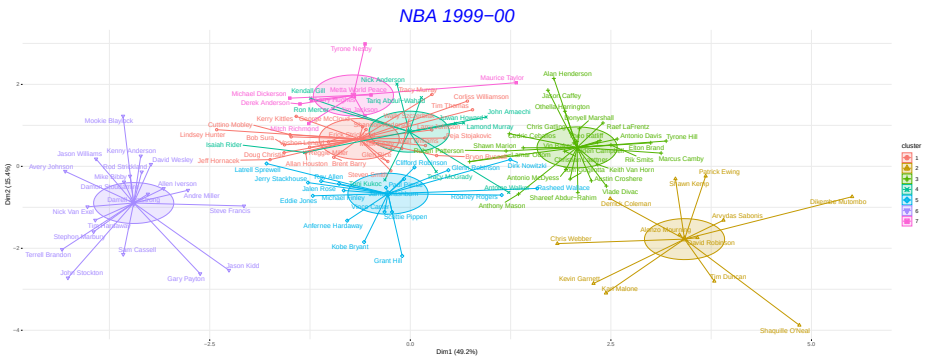
```
set.seed(123)
km_result <- kmeans(df, centers = 7)
```

```
clustering_23<-fviz_cluster(km_result, data = df,
             ellipse.type = "euclid",
             ellipse.level=0.5,
             ellipse.ratio=0.8,
             star.plot = TRUE,
             repel = TRUE,
             main="NBA 2022-2023",
             ggtheme = theme_minimal())

clustering_23 <- clustering_23 +
  theme(
    plot.title = element_text(
      size = 30,
      face = "italic",
      color = "blue",
      hjust = 0.5,
      vjust = 1,
      angle = 0,
      lineheight = 1.2
    )
  )
```

For season 1999-00

```
print(clustering_00)
```

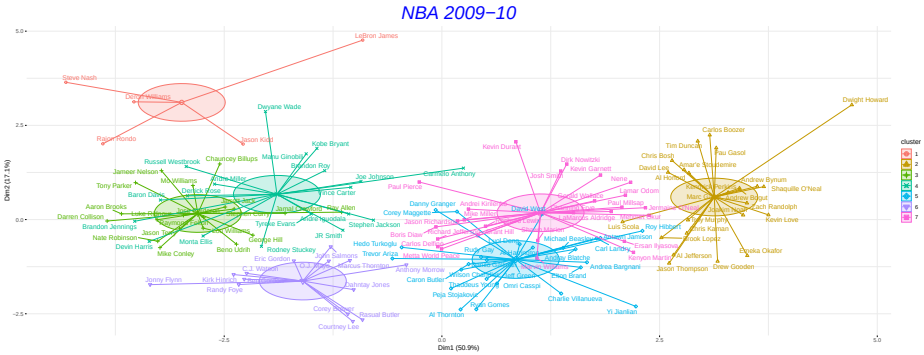



```
kable(mean_values_00)
```

cluster	player_height	player_weight	pts	reb	ast
1	198.9667	96.38830	12.92917	3.879167	2.558333
2	212.3017	119.63489	19.84167	10.483333	2.650000
3	207.4008	108.30381	13.78462	7.665385	1.826923
4	202.0455	104.65605	14.86364	5.009091	2.709091
5	201.9300	99.89104	19.59444	5.505556	3.811111
6	186.1820	83.95988	15.68500	3.680000	7.250000
7	197.8025	98.59956	15.82500	4.275000	2.475000

For season 2009-10

```
print(clustering_10)
```

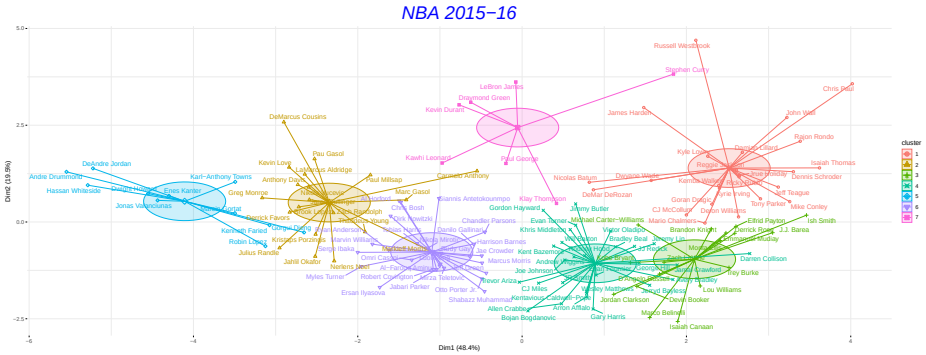



```
kable(mean_values_10)
```

cluster	player_height	player_weight	pts	reb	ast
1	192.5320	92.16989	17.78000	4.920000	9.800000
2	210.0792	117.59373	16.15000	9.754167	1.991667
3	187.4253	84.94107	14.20526	2.705263	4.673684
4	195.1789	93.79805	19.55789	4.378947	5.252632
5	206.3262	106.50689	15.16923	5.715385	1.834615
6	193.9471	91.36639	13.10714	2.964286	2.650000
7	205.5446	107.67576	14.13462	6.519231	2.338462

For season 2015-16

```
print(clustering_16)
```

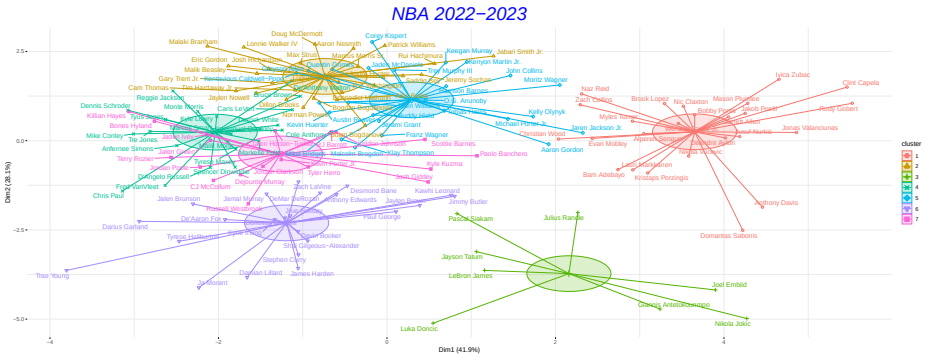



```
kable(mean_values_16)
```

cluster	player_height	player_weight	pts	reb	ast
1	189.7592	88.50714	17.87917	4.016667	6.491667
2	209.5500	114.50930	16.73000	8.585000	2.370000
3	191.5459	87.30312	13.50000	3.147059	3.817647
4	196.7593	94.16894	14.13929	3.557143	2.671429
5	211.0509	114.71754	13.36364	10.263636	1.145455
6	206.3262	104.76231	13.67692	5.723077	1.788461
7	201.0229	102.05820	23.42857	6.871429	4.957143

For season 2022-23

```
print(clustering_23)
```

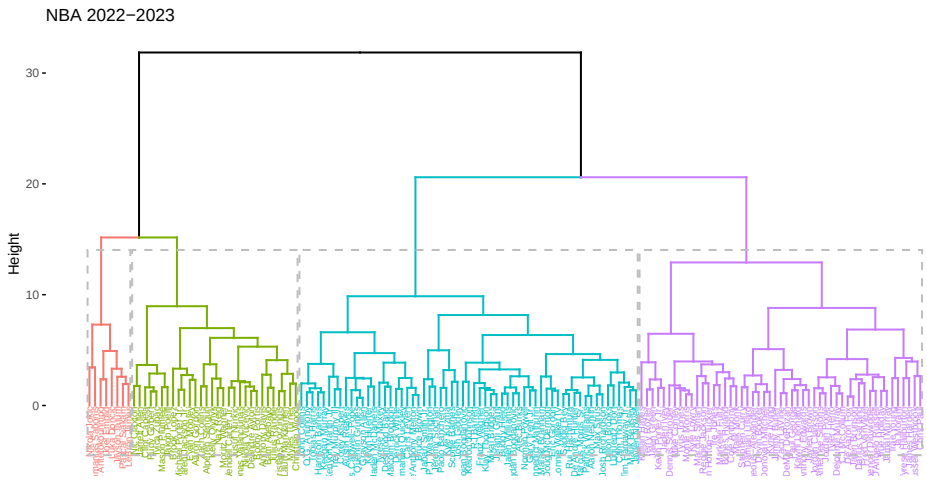



```
kable(mean_values_23)
```

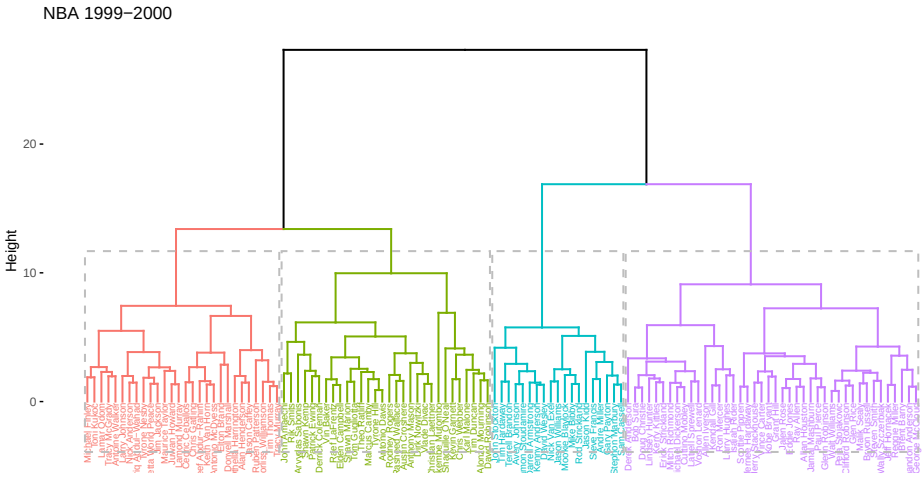
cluster	player_height	player_weight	pts	reb	ast
1	211.3280	113.25285	15.90000	9.196000	2.352000
2	197.4615	97.10229	13.09259	3.555556	1.855556
3	206.6925	112.09392	28.67500	9.662500	6.125000
4	189.2300	88.18241	13.73182	3.427273	5.009091
5	201.2462	98.91795	14.81154	4.538462	2.276923
6	193.2609	92.27639	24.90000	4.917391	6.239130
7	195.5800	92.41340	18.14737	4.826316	4.700000

3.2.2 Hierarchical Clustering

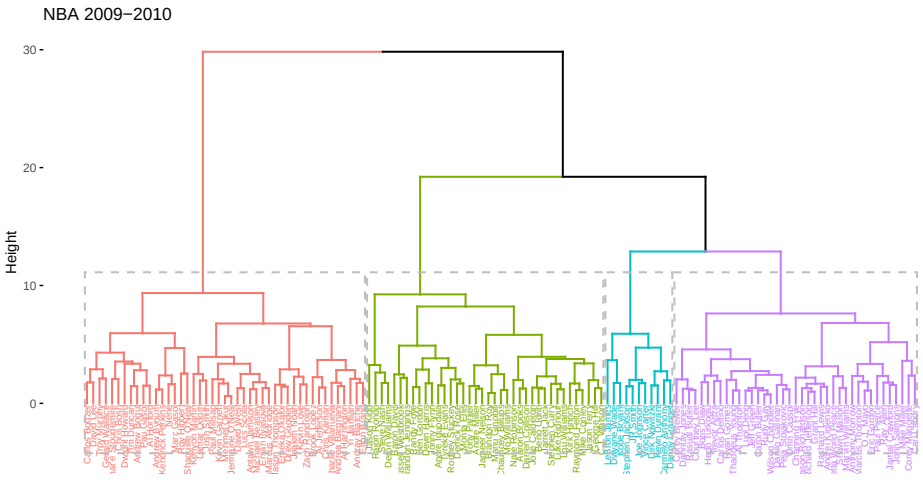
```
result_23 <- dist(df, method = "euclidean")
result_hc <- hclust(d = result_23, method = "ward.D2")
fviz_dend(result_hc, k = 4,
          cex = 0.5,
          color_labels_by_k = TRUE,
          main="NBA 2022-2023",
          rect = TRUE
)
```



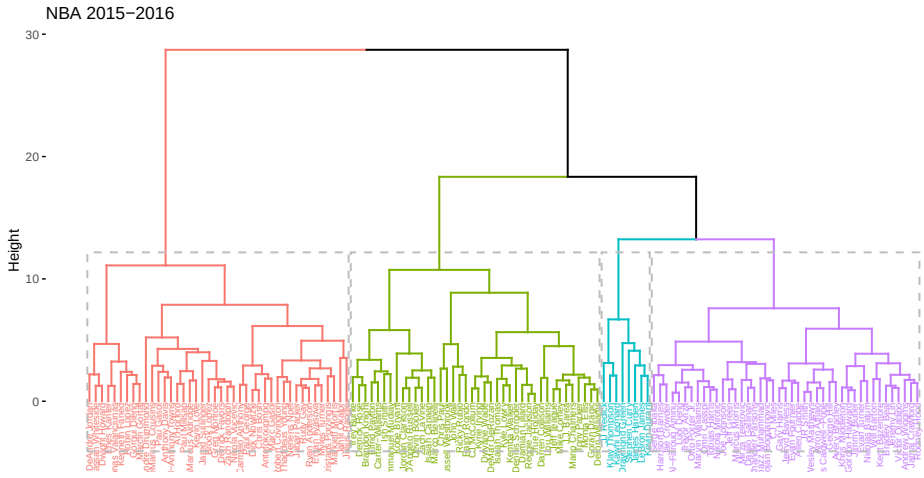

```
fviz_dend(result_hc_00, k = 4,
          cex = 0.5,
          color_labels_by_k = TRUE,
          main="NBA 1999-2000",
          rect = TRUE
)
```




```
fviz_dend(result_hc_10, k = 4,
          cex = 0.5,
          color_labels_by_k = TRUE,
          main="NBA 2009-2010",
          rect = TRUE
)
```




```
fviz_dend(result_hc_16, k = 4,
          cex = 0.5,
          color_labels_by_k = TRUE,
          main="NBA 2015-2016",
          rect = TRUE
)
```


NBA 2015−2016
Height
30
20
10
0

Chapter 4

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 5

Footnotes and citations

5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2023] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Chapter 6

Blocks

6.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{6.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (6.1).

6.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 6.1.

Theorem 6.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

6.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 7

Sharing your book

7.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

7.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

7.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book- all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2023. URL https://github.com/rstudio/bookdown. R package version 0.35.