Basic Skills in Visualing Data and Exporatory Data Analysis Using R: part I - univariate datasets

Ziv Shkedy, Adetayo Kasim et al. (2020)

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
## Warning: package 'mvtnorm' was built under R version 3.6.2
```

Introduction

John W. Tukey said that "Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone - as the first step". This book provides the practical guide to acquire the fundamental skills in Exploratory data analysis (EDA) and visualizing data (VD) using R.

R ?

Only basic knowledge in R is required. We focused on a user level skills rather than on programming skills and we use mainly the lattice and gg2plot2 R packages and the basic graphical functions in R to vizuaized the structure in the data.

Example: the iris data

The iris dataset was used by R.A.Fisher in 1936 to illustrate few methods in multivariate analysis. The data contains information about 4 measurements on 50 flowers from each of 3 species of iris. Sepal length and width, and petal length and width are measured in centimeters. Species are Setosa, Versicolor and Virginica. In this section we focus on the Sepal length of Setosa which is shown bellow.

```r
Sepal.L <- iris$Sepal.Length[iris$Species=="setosa"]
Sepal.L
```

```
##  [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1 5.7
## [20] 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0 5.5 4.9
## [39] 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0
```

Throughout this book, we use both numerical summaries and graphical displays to visualize data structures. Let us focus on a very simple example of the sepal length data. The mean and the median of sepal length are 5.006 and 5 respectively.

```r
c(mean(Sepal.L),median(Sepal.L))
```

```
## [1] 5.006 5.000
```

Why do we use the mean and the median ? What do these values tell us about the distribution of the sepal length ? Furthermore, what can we say about the distribution of the sepal length of Setosa ? How does the distribution look like ? A Histogram is a graphical display that can be used to visualize the distribution of the sepal length. Figure @ref(fig:figchp11) presents a histogram that was produced using a basic plot function in R, the hist() function.

```r
hist(Sepal.L, nclass = 20, col = 0)
```

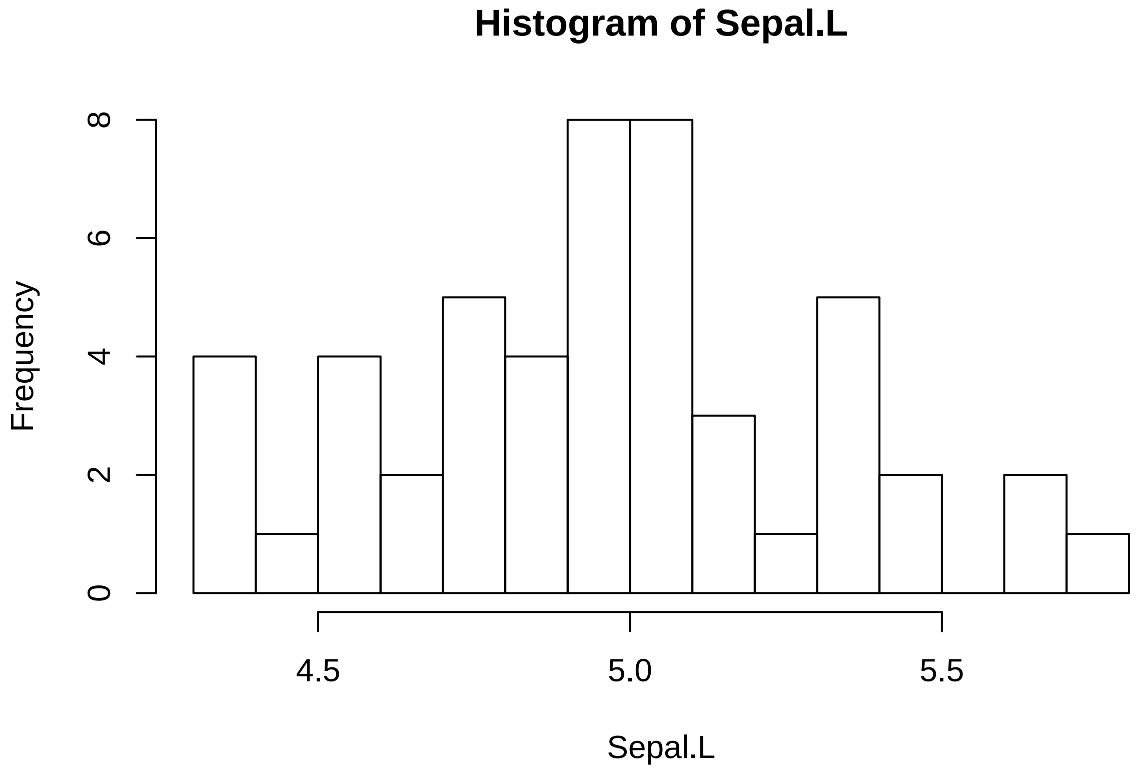


Figure 1: Sepal length (I)

Alternatively, we can use the function histogram() from the R package lattice to produce the histogram in Figure @ref(fig:figchp12). The histogram() function is more advanced than the hist() and it allows to produce histograms for a more complex data structures.

```
histogram(Sepal.L,
                  layout = c(1, 1),
              aspect = 1,
                 xlab = "Sepal length")
```

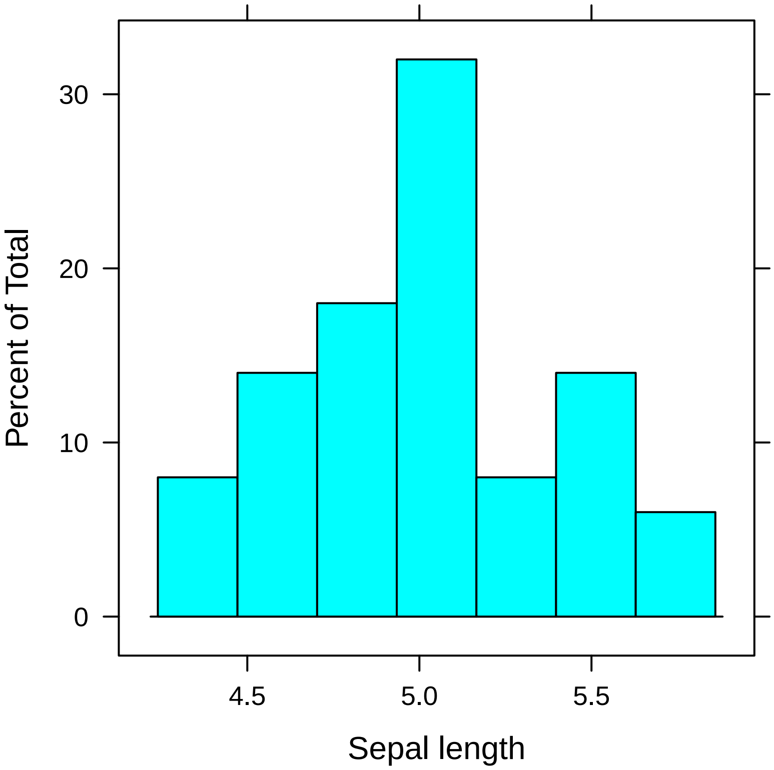


Figure 2: Sepal length (II)

A third option to produce a histogram for the sepal length, Figure @ref(fig:figchp12a), is to use the ggplot2 package and the qplot() function with the argument geom = “histogram”.

```
Iris.R<-data.frame(Sepal.L)
qplot(Sepal.L, data = Iris.R, geom = "histogram", binwidth = 0.1)
```

Book Structure

Part 1: Location, Spread and Shape in univariate data

In the first part of the book we focus on descriptive measures, numerical and graphical, to characterize and visualize the features of a particular univariate distribution. The following three main factors are usually used to specify a paticular distribution:

- Location

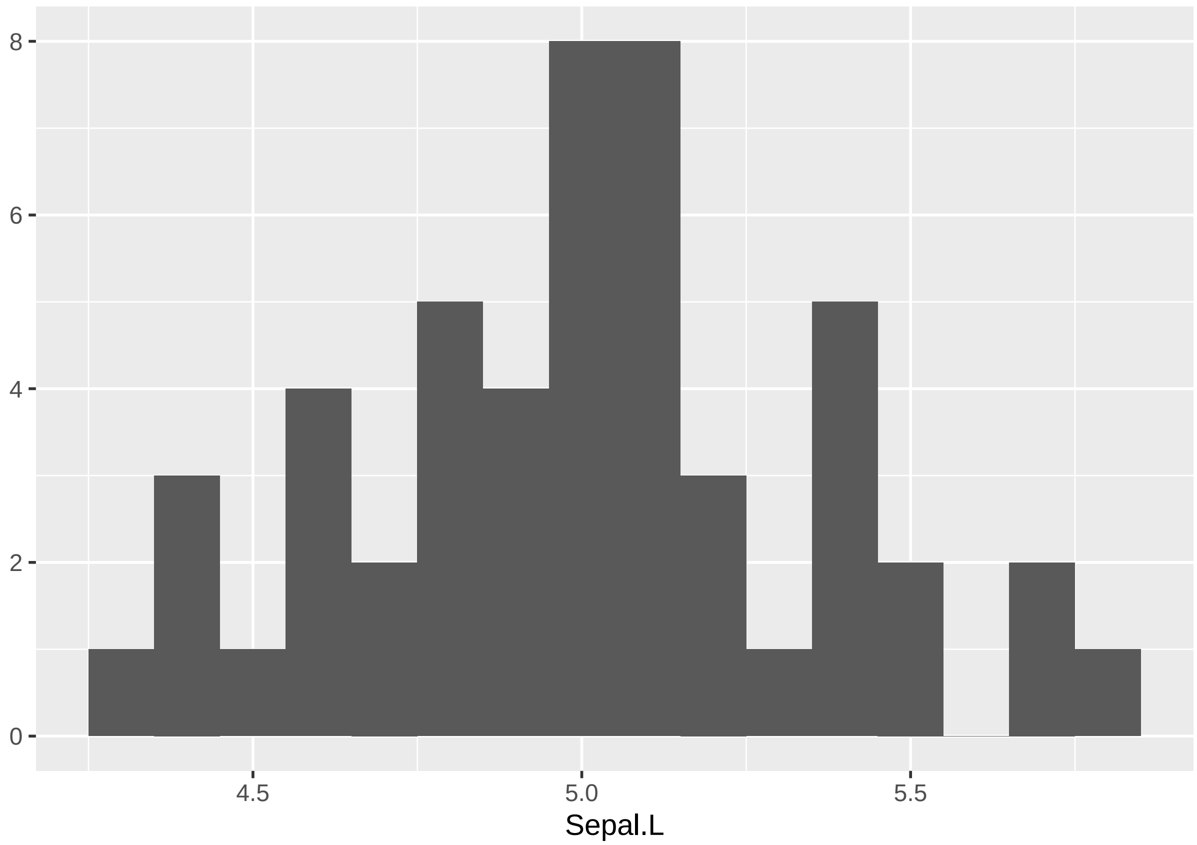


Figure 3: Sepal length (III)

- Spread
- Shape

Each of these control different characteristics of a distribution.

Location

Location is the center of the distribution. Figure @ref(fig:figchp12c) presents distributions with different locations: two normal densities with mean equal to 0 (the black line) and mean equal to 2 (the red line).

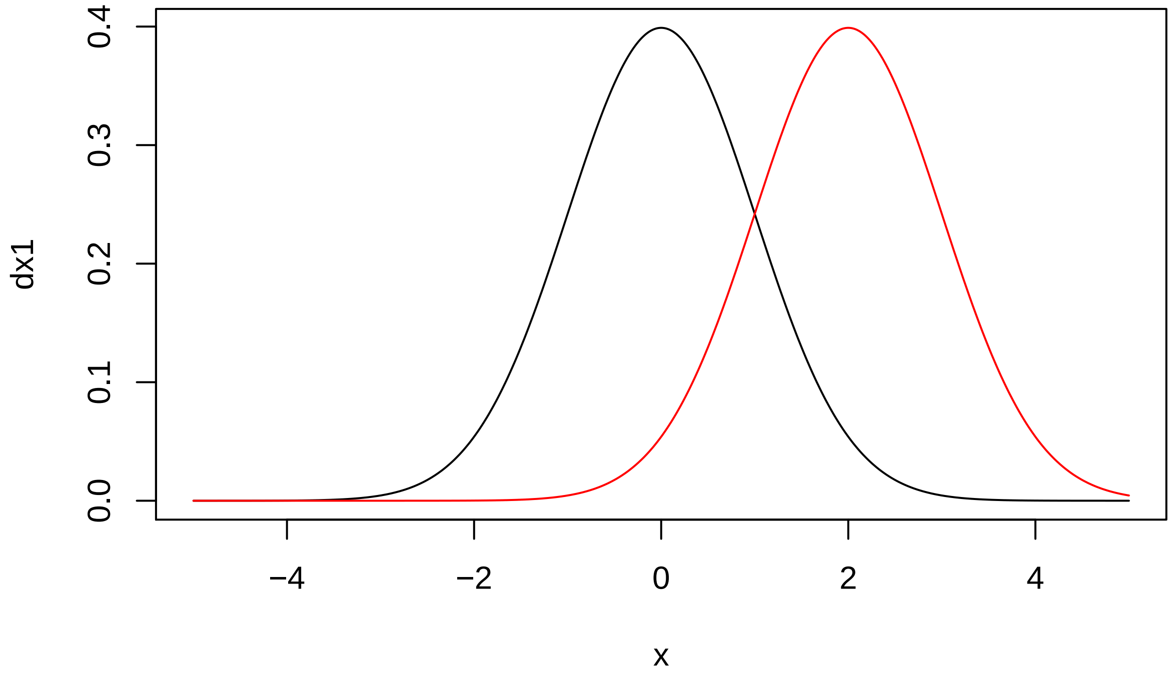


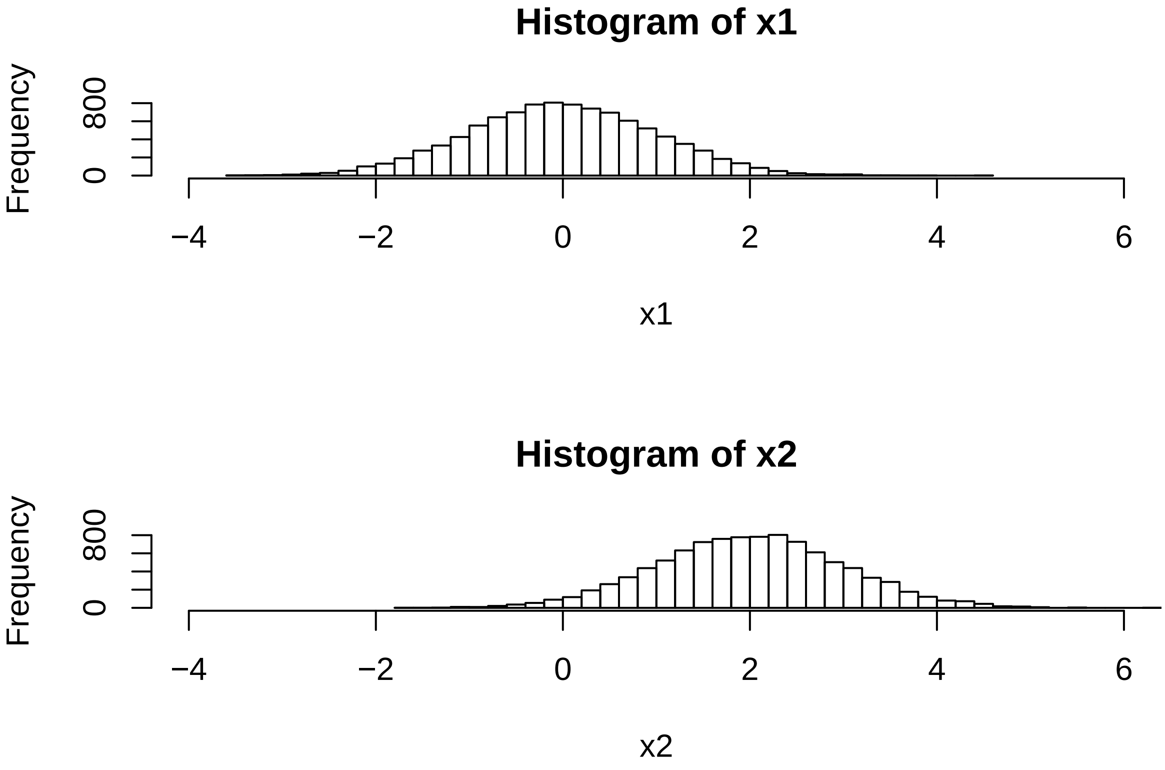
Figure 4: Location

Figure @ref(fig:figchp12d) shows histograms for two *random samples* drawn from normal distribution with the same variance but different mean. Both histograms show that the data are symmetric around the sample mean but the histogram of x_2 is located to the right relative to the histogram of $x1$.

```
x1 <- rnorm(10000, 0, 1)
x2 <- rnorm(10000, 2, 1)
par(mfrow = c(2, 1))
hist(x1, col = 0, nclass = 50, xlim = c(-4, 6))
hist(x2, col = 0, nclass = 50, xlim = c(-4, 6))
```

Spread

Figure @ref(fig:figchp12e) presents two normal densities that have different variability (or spread). The density with the black line has variance 1 and density with the green line has variance 2.



Histogram of x1

Histogram of x2

Figure 5: Samples from normal distribution.

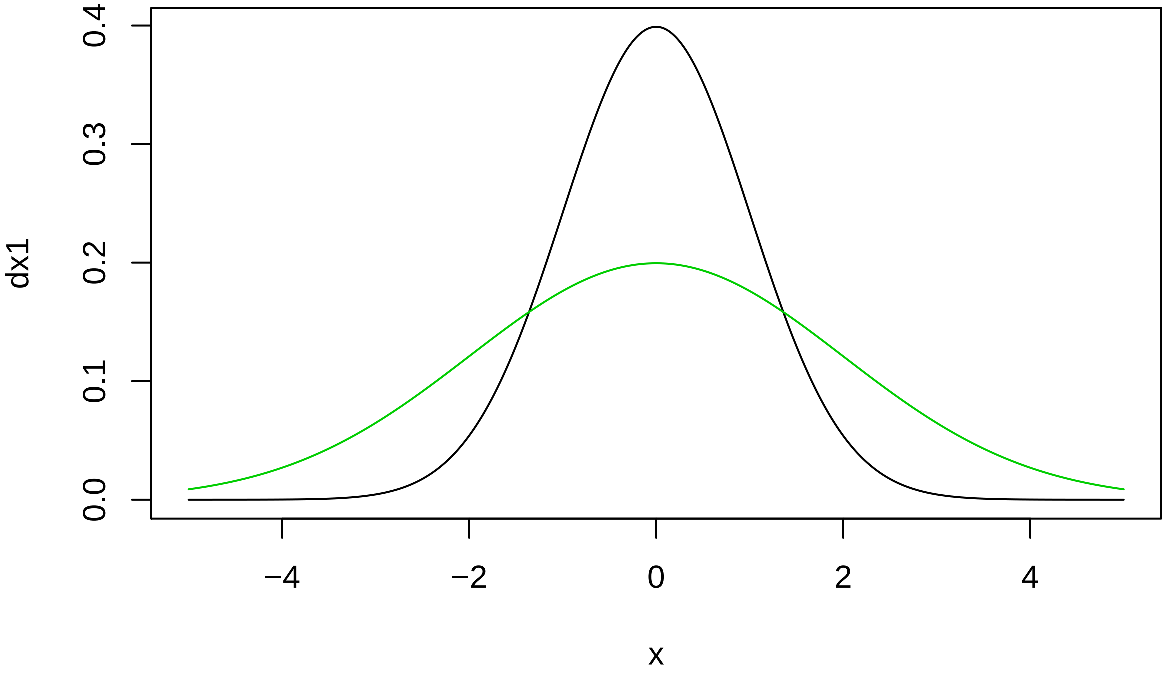


Figure 6: Spread

The two samples in Figure @ref(fig:figchp12f) were drawn from normal distribution with mean equal to 0 but with different variance. The two distributions have the same shape, both histograms are symmetric around 0 as expected. The spread in the histogram of x_2 is much higher than the spread in the histogram of x_1.

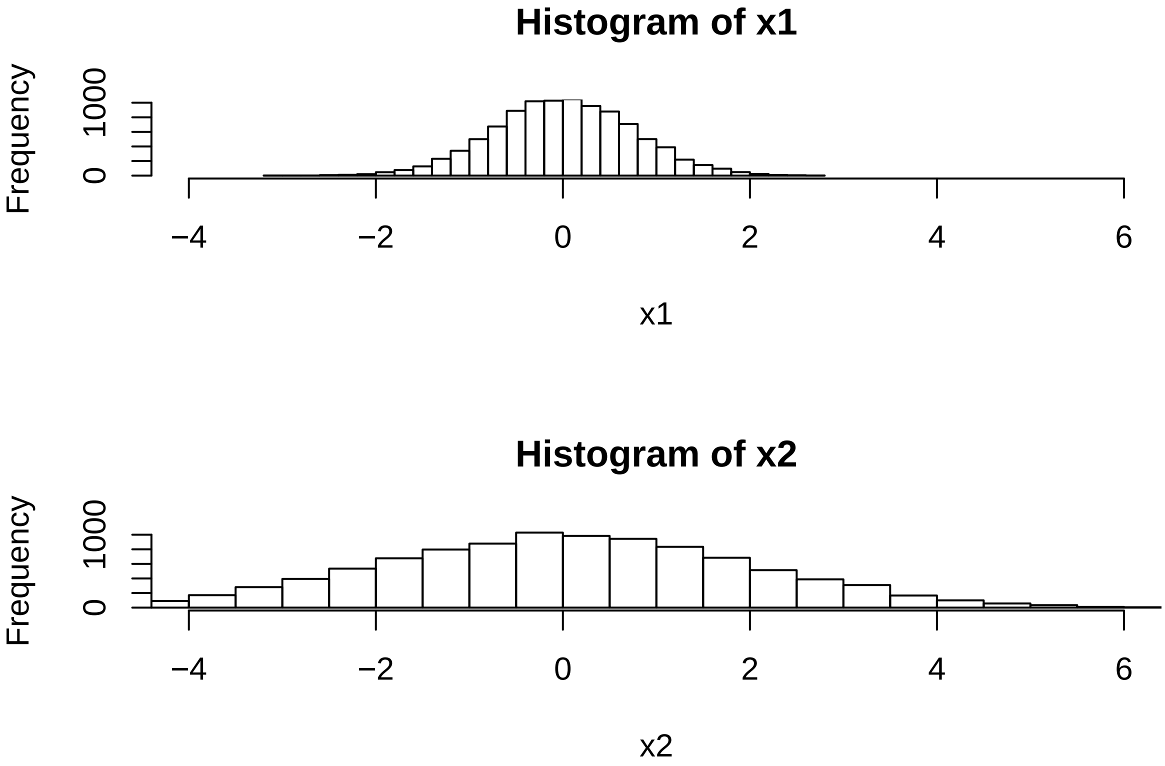


Figure 7: Random samples from normal distribution

Shape

Figure @ref(fig:figchp12i) presents two beta densities having diffent shapes, the black line has shape parameters 2 and 2 whereas the red line has shape parameters 2 and 4.

Figure @ref(fig:figchp12j) shows 4 samples (each with 10000 observations) that were drawn from different distributions.The first two samples were drawn from symmetric distributions. However, the distributions do not have the same shape. The other two samples (x_3 and x_4) were drawn from skewed distributions. The distributions of x_3 is skewed to the left and the distribution of x_4 to the right.

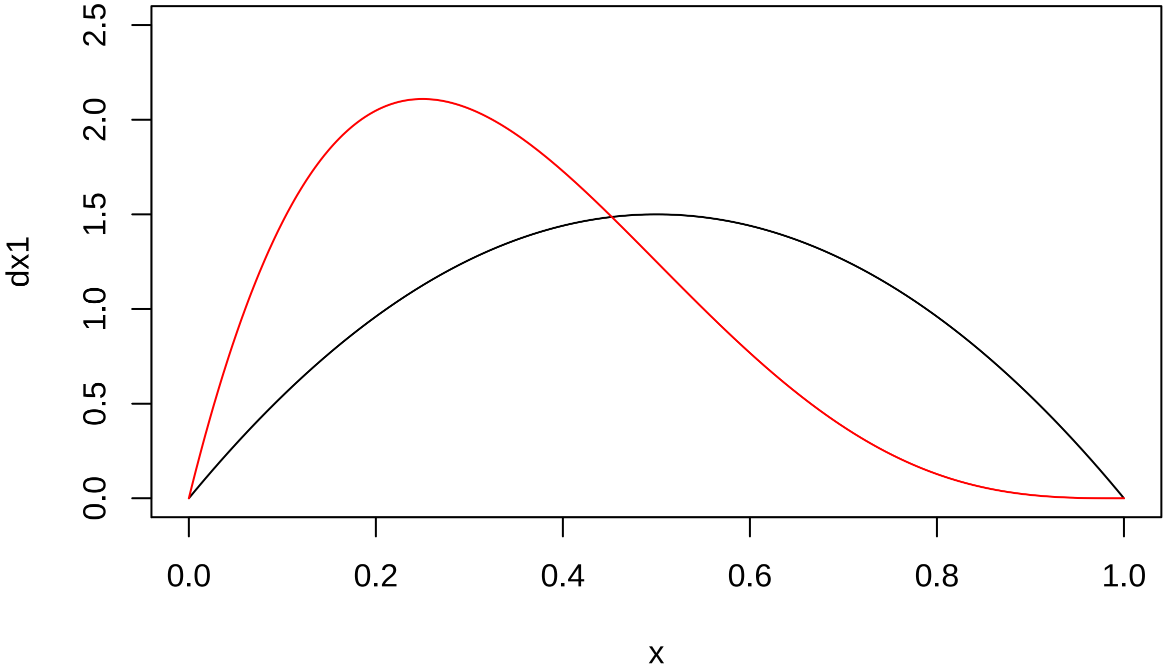


Figure 8: Shape

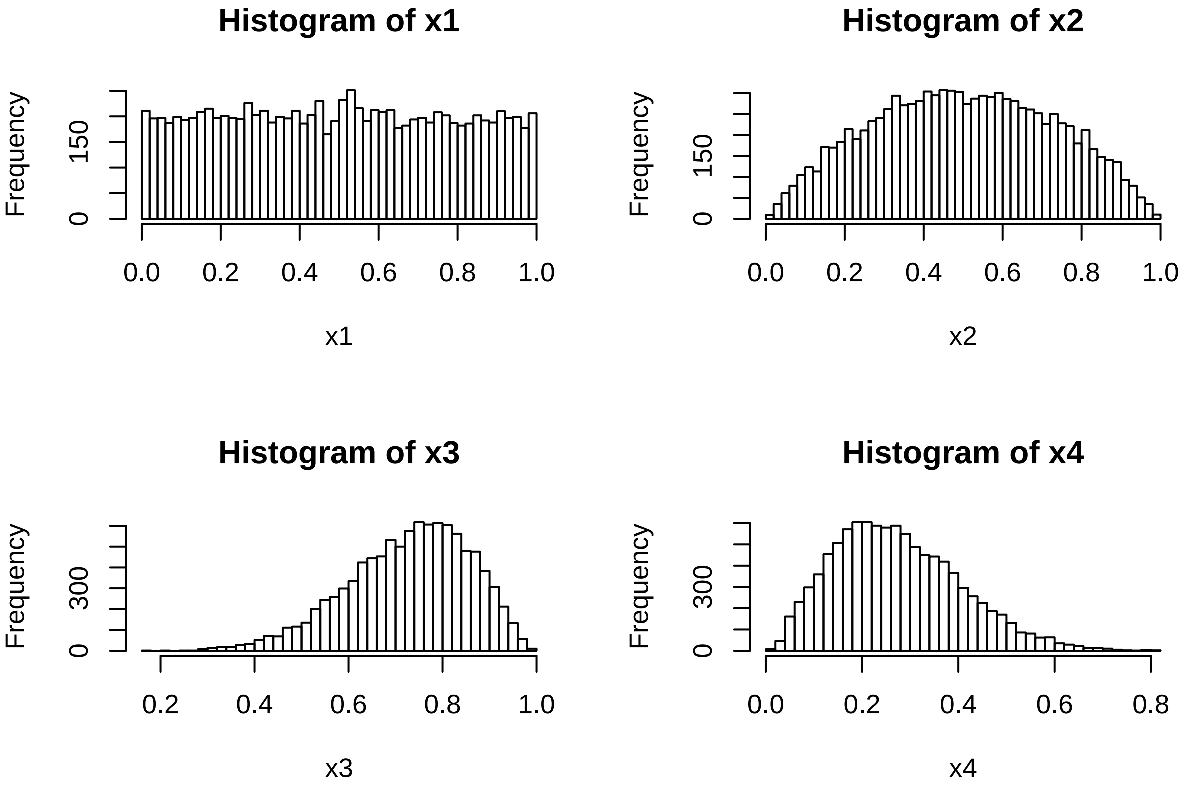


Figure 9: Random Samples from Beta distribution.

Part 2: Correlation and association in multivariate datasets.

Part 3:Clustering.

Part 4: Biclustering

R, again ?

If you did not have a problem to understand the R code above, you will not have any problem to understand the R code that we use to produce all the output for the examples discuss in this book. If you had a difficulty to understand the code for the examples above you need a short training, at a beginner level, in R.

YouTube tutorial: introduction to R

For a short online YouTube introduction, by Stuar51XT, see YTVD1.

R course online

An introductory course for R is available online in the >eR-BioStat website here >eR-BioStat.

R datasets for illustraions

In order to simplify the usage of this book, the data we used for illustrations are R datasets or NBA data. For the later, see next section. We give a short description of each data in the relevant chapters, more details can be found with help(dataset name) or in

- Part 1
  - The singers data: singers.
  - The airquality data: airquality
  - The cars data: cars
  - The Iris data: iris
  - The Old Faithful Geyser Data: oldfaithful
  - The beavers data: beavers
- Part 2

NBA data

For many examples we use NDA data about different teams and different players. Data were download from the basketball reference website NBA. The NBA data consists of performance indicators (on a team and individual levels) and we use the data to illustrate visualization tools to detect and present trend s/structures in the data. Topics of NBA analytics that we cover in this part of the book include:

- Magic Johnson and Larry Bird's rookie season in the NBA.
- The number of 3 points attempts per game made by the Golden State Warriors: 2010-2019.
- Michael Jordan's game score in championship years.
- Kareem Abdul Jabbar and Karl Malone's total number of points.
- Lebron James and Kobe Bryant total number of points.
- Kobe Bryant's performance indicators over the championship years.

Telling the story: skills and tools

Part I

Basic Concepts: Location, Speard and Shape

The 5-number summary and stem-and-leaf

Introduction

In a given dataset we want to know

- Which values seems to be typical (Location) ?
- How much variation there is in the data (Spraed) ?
- How the distribution of the data looks like (Shape) ?

John W. Tukey (1977, page 32) proposed the five number summary - the median, the minimum and maximum and the upper and lower quartiles - as a numerical summary of the data. The extremes (the minimum and maximum) give information about the depth of the data (i.e., spread), the median gives information about the center of the distribution (location) and the lower and upper quartiles give information on both spread (at the center of the distribution of the data) and shape.

Sorting and ranking the sample

Consider a sample of size 5 from $N(0, 2^2)$: x_1, x_2, x_3, x_4, x_5

```
x <- rnorm(5, 0, 2)
x
```

```
## [1]  0.36216435  0.20214062 -0.07142807 -2.99804413 -2.25041687
```

The Ordered sample is the sample sorted from the lowest to the highest value:

$x_{(1)}, x_{(2)}, x_{(3)}, x_{(4)}, x_{(5)}$

```
sort(x)
```

```
## [1] -2.99804413 -2.25041687 -0.07142807  0.20214062  0.36216435
```

Numerical summaries for sample distribution

Median (M)

The median is the center of the sample in terms of counting: $\frac{1}{2}$ of the observations are smaller or equals to the median and $\frac{1}{2}$ are greater than the median.

```
 median(x) #note that the sample size is 5
```

```
## [1] -0.07142807
```

If the sample size is even then

```
y <- c(10, 2, 50, 6, 100, 25)
sort(y)
```

```
## [1]   2   6  10  25  50 100
```

```
median(y) #=(10+25)/2
```

```
## [1] 17.5
```

The extremes

The smallest and the largest values in the data.

```
min(y)
```

```
## [1] 2
```

```
max(y)
```

```
## [1] 100
```

```
range<-max(y)-min(y)
range
```

```
## [1] 98
```

Lower and Upper fourth (the lower and upper quartiles)

The lower fourth (F_L) is the value that $\frac{1}{4}$ of the observations are smaller or equals to and $\frac{3}{4}$ of the observations are greater than F_L. The upper forth (F_U) is the value that $\frac{3}{4}$ of the observations are smaller or equals to and $\frac{1}{4}$ of the observations are greater than F_U.

Summary of a single sample

The 5-number summary

The idea is to summarize the information in the data with 5 numbers: The median, the fourths and the extremes, x_1, F_L, M, F_U, x_n.

In R, The 5-number summary of a sample $x = (x_1, \ldots, x_n)$ can be produced using the function quantile().

Online tutorials

YouTube tutorial: 5 number summary

For a short online YouTube introduction, by Simple Learning Pro, about the five numbers Boxplots, and Outliers with R see YTVD2.

YouTube tutorial: stem-and-leaf plot

For a short online YouTube introduction, by Ed Boone, stem-and-leaf using R see YTVD3.

Example: the old faithful geyser data

The Old faithful geyser dataset, the R object faithful, gives information about the Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA. The duration of the eruption are listed below.

```
faithful$eruptions
```

```
##   [1] 3.600 1.800 3.333 2.283 4.533 2.883 4.700 3.600 1.950 4.350 1.833 3.917
##  [13] 4.200 1.750 4.700 2.167 1.750 4.800 1.600 4.250 1.800 1.750 3.450 3.067
##  [25] 4.533 3.600 1.967 4.083 3.850 4.433 4.300 4.467 3.367 4.033 3.833 2.017
##  [37] 1.867 4.833 1.833 4.783 4.350 1.883 4.567 1.750 4.533 3.317 3.833 2.100
##  [49] 4.633 2.000 4.800 4.716 1.833 4.833 1.733 4.883 3.717 1.667 4.567 4.317
##  [61] 2.233 4.500 1.750 4.800 1.817 4.400 4.167 4.700 2.067 4.700 4.033 1.967
##  [73] 4.500 4.000 1.983 5.067 2.017 4.567 3.883 3.600 4.133 4.333 4.100 2.633
##  [85] 4.067 4.933 3.950 4.517 2.167 4.000 2.200 4.333 1.867 4.817 1.833 4.300
##  [97] 4.667 3.750 1.867 4.900 2.483 4.367 2.100 4.500 4.050 1.867 4.700 1.783
## [109] 4.850 3.683 4.733 2.300 4.900 4.417 1.700 4.633 2.317 4.600 1.817 4.417
## [121] 2.617 4.067 4.250 1.967 4.600 3.767 1.917 4.500 2.267 4.650 1.867 4.167
## [133] 2.800 4.333 1.833 4.383 1.883 4.933 2.033 3.733 4.233 2.233 4.533 4.817
## [145] 4.333 1.983 4.633 2.017 5.100 1.800 5.033 4.000 2.400 4.600 3.567 4.000
## [157] 4.500 4.083 1.800 3.967 2.200 4.150 2.000 3.833 3.500 4.583 2.367 5.000
## [169] 1.933 4.617 1.917 2.083 4.583 3.333 4.167 4.333 4.500 2.417 4.000 4.167
## [181] 1.883 4.583 4.250 3.767 2.033 4.433 4.083 1.833 4.417 2.183 4.800 1.833
## [193] 4.800 4.100 3.966 4.233 3.500 4.366 2.250 4.667 2.100 4.350 4.133 1.867
## [205] 4.600 1.783 4.367 3.850 1.933 4.500 2.383 4.700 1.867 3.833 3.417 4.233
## [217] 2.400 4.800 2.000 4.150 1.867 4.267 1.750 4.483 4.000 4.117 4.083 4.267
## [229] 3.917 4.550 4.083 2.417 4.183 2.217 4.450 1.883 1.850 4.283 3.950 2.333
## [241] 4.150 2.350 4.933 2.900 4.583 3.833 2.083 4.367 2.133 4.350 2.200 4.450
## [253] 3.567 4.500 4.150 3.817 3.917 4.450 2.000 4.283 4.767 4.533 1.850 4.250
## [265] 1.983 2.250 4.750 4.117 2.150 4.417 1.817 4.467
```

The 5 number summary for the duration of the eruption is

```r
quantile(faithful$eruptions)
```

```
##      0%     25%     50%     75%    100% 
## 1.60000 2.16275 4.00000 4.45425 5.10000
```

Indicating that in 50% the duration is more than 4.00 (location) and that in 50% of the times the furation is between 2.16 to 4.45 (spread). The range of the data is between 1.6 to 5.1 (spread).

Stem-and-leaf display

Stem-and-leaf enables to organize the data graphically in a way that directs our attention to the main pattern in the data. This is a basic and simple but very effective tool to visualize the pattern in the data.

Constructing stem-and-leaf:

- data value $\Longrightarrow$ split $\Longrightarrow$ stem and leaf
- 10.5 $\Longrightarrow$ 10|5 $\Longrightarrow$ 10 and 5

We use the R function stem() to construct a stem-and-leaf. For the Old faithful geyser data we use

```r
stem(faithful$eruptions)
```

```
## 
##   The decimal point is 1 digit(s) to the left of the |
## 
##   16 | 070355555588
##   18 | 000022233333335577777777888822335777888
##   20 | 00002223378800035778
##   22 | 0002335578023578
##   24 | 00228
##   26 | 23
##   28 | 080
##   30 | 7
##   32 | 2337
##   34 | 250077
##   36 | 0000823577
##   38 | 2333335582225577
##   40 | 0000003357788888002233555577778
##   42 | 03335555778800233333555577778
##   44 | 02222335557780000000023333357778888
##   46 | 0000233357700000023578
##   48 | 00000022335800333
##   50 | 0370
```

The stem-and-leaf plot of the Old faithful geyser data reveals a clear bimodel for the duration ofeuroption.

The 5-number summary and samples comparison

The five number summery can be used for a comparisons between two samples. The relatioship between the quantiles of the two samples can give information about the difference between the two samples in terms of location and spread.

The beaver data

The beaver dataset (the R objects beaver1 and beaver2), pblished by Reynolds (1994), describes a small part of a study of the long-term temperature dynamics of beaver Castor canadensis in north-central Wisconsin. Body temperature was measured by telemetry every 10 minutes for four females.For illusrtraion we use data obtaioned for two beavers and focused on the body temperature (the variable temp) in a specific day (346). A part of the data for one beaver is listed below.

```
head(beaver1[beaver1$day==346& beaver1$activ==0,])
```

```
##   day time  temp activ
## 1 346  840 36.33     0
## 2 346  850 36.34     0
## 3 346  900 36.35     0
## 4 346  910 36.42     0
## 5 346  920 36.55     0
## 6 346  930 36.69     0
```

A comparison between the two beavers: stem-and-leaf

The stem-and-leaf plot and 5 number summry are shown below

```
temp1<-beaver1[beaver1$day==346& beaver1$activ==0,]$temp
stem(temp1)
```

```
##
##   The decimal point is 1 digit(s) to the left of the |
##
##   363 | 345
##   364 | 2
##   365 | 04559
##   366 | 224577999
##   367 | 145567789
##   368 | 01123455556777788999999
##   369 | 111222334445567788999
##   370 | 0001259
##   371 | 08
##   372 | 00013
```

```
quantile(temp1)
```

```
##      0%     25%     50%     75%    100% 
## 36.3300 36.7525 36.8800 36.9575 37.2300
```

For the second beaver we have

```
temp2<- beaver2[beaver2$day==307 & beaver2$activ==0,]$temp
stem(temp2)
```

```
## 
##   The decimal point is 1 digit(s) to the left of the |
## 
##   364 | 8
##   366 | 3
##   368 | 90035577899
##   370 | 0114472223445577
##   372 | 34488
##   374 | 411
##   376 | 4
```

```
quantile(temp2)
```

```
##      0%     25%     50%     75%    100% 
## 36.5800 36.9725 37.0950 37.1700 37.6400
```

A comparison between the two beavers: the 5 number summary

We can see that the 5 number summary values obtained for the second beaver are higher than those obtained for the first beaver, indicating that the temperature distribution of the second beaver located to the right compare to the temperature distribution of the first beaver. This can be seen in Figure @ref(fig:figchp12k) that compares the 5 number summaries of the two beavers and shows that the quantiles obtained for the second beaver are above the 45^o line.

```
plot(quantile(temp1),quantile(temp2))
abline(0,1)
```

Example: Magic Johnson and Larry Bird's rookie season in the NBA

Earvin "Magic" Johnson (https://en.wikipedia.org/wiki/Magic_Johnson) and Larry Bird (https://en.wikipedia.org/wiki/Larry_Bird) are two NBA legends who entered the NBA league in the season of 1979/1980. The Battles between the Lakers (with Magic Johnson) and the Celtics (with Larry Bird) in their time is considered by many as one of the most intense rivalry in NBA history. In this section we compare two performance indicators, the number of field goals attempts(FGA) and the number of assist (AST) per game between Johnson and Bird in their rookie season (79/80). The R objects that contains that data are x1 (Larry Bird) and x2 (Magic Johnson).

```
LB <- read.csv("C:/projects/NBA/LBMJ/LB_1979_RS.csv",header = TRUE,sep =",")
MJ <- read.csv("C:/projects/NBA/LBMJ/MJ_1979_RS.csv",header = TRUE,sep =",")
```

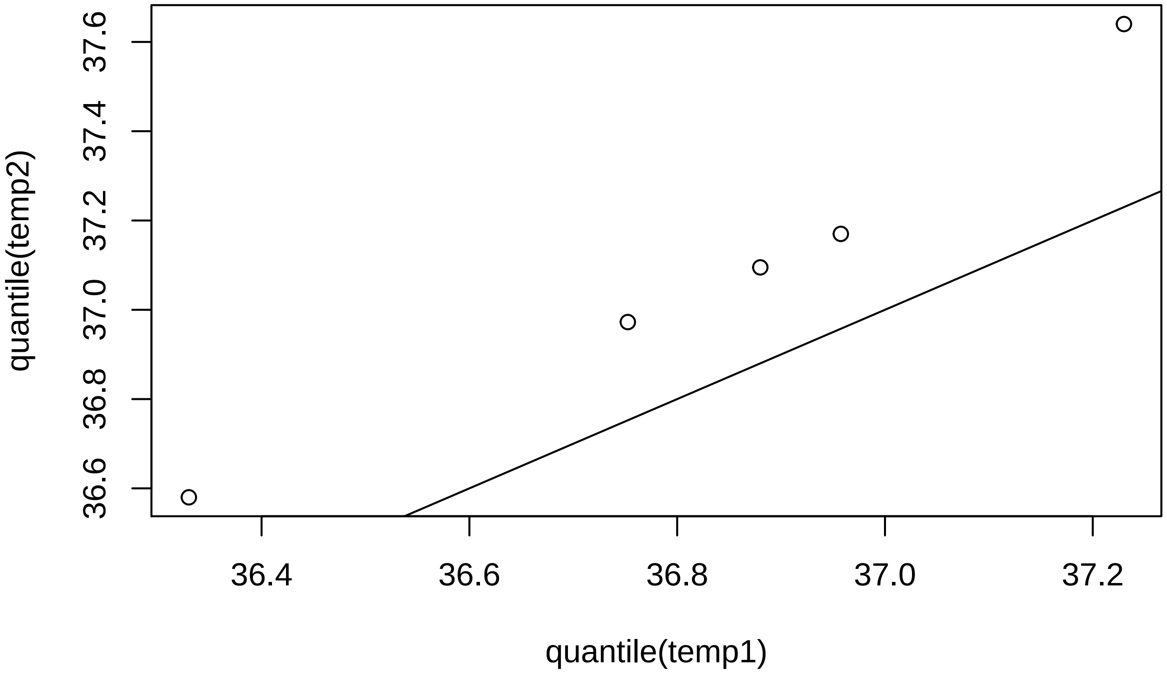


Figure 10: the 5 number summary of two beavers.

```
k<-12 #FGA
#k<-23 #AST
x1<-na.omit(LB[,k])
x2<-na.omit(MJ[,k])
```

Field goal attempts (FGA) per game

Larry Bird's mean FGA (per game), 17.6, is higher than Magic Johnson's mean, 12.34, indicating that Larry Bird took more shorts than Magic Johnson,

```
xx<-data.frame(c("MEJ","LB"),c(mean(x2),mean(x1)))
names(xx)<-c("Player","Average")
xx
```

```
##   Player  Average
## 1    MEJ 12.32468
## 2     LB 17.63235
```

The stem-and-leaf diagram below shows that Bird's mode of FGA is between 15-19 compare to Johnson's mode FGA which is between 10-12.

```
stem(x1)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   0 | 69
##   1 | 0012222244444444
##   1 | 5555555556666777777778888899999
##   2 | 00111112244
##   2 | 557799
##   3 | 02
```

```
q.lb<-quantile(x1)
q.lb
```

```
##    0%   25%   50%   75%  100%
##  6.00 14.00 17.00 20.25 32.00
```

It also shows that Bird's FGA per game is more variable compare to Johnson's FGA per game. Note that Bird's range is 26 (32-6) higher than Magic's range of 20.

```
stem(x2)
```

```
##
##   The decimal point is at the |
##
##    2 | 00
##    4 | 0
##    6 | 00000
##    8 | 0000000
##   10 | 000000000000000000
##   12 | 0000000000000000000
##   14 | 00000000
##   16 | 0000000
##   18 | 00000000
##   20 | 0
##   22 | 0
```

```
q.mej<-quantile(x2)
q.mej
```

```
##   0%  25%  50%  75% 100% 
##    2   10   12   15   22
```

The same pattern is visualized in Figure @ref(fig:figchp12l) that shows that Bird's 5 number summary values are higher than Johnson's Bird's 5 number summary values.

```
plot(q.mej,q.lb,xlab="Magic Johnson",ylab="Larry Bird")
abline(0,1)
```

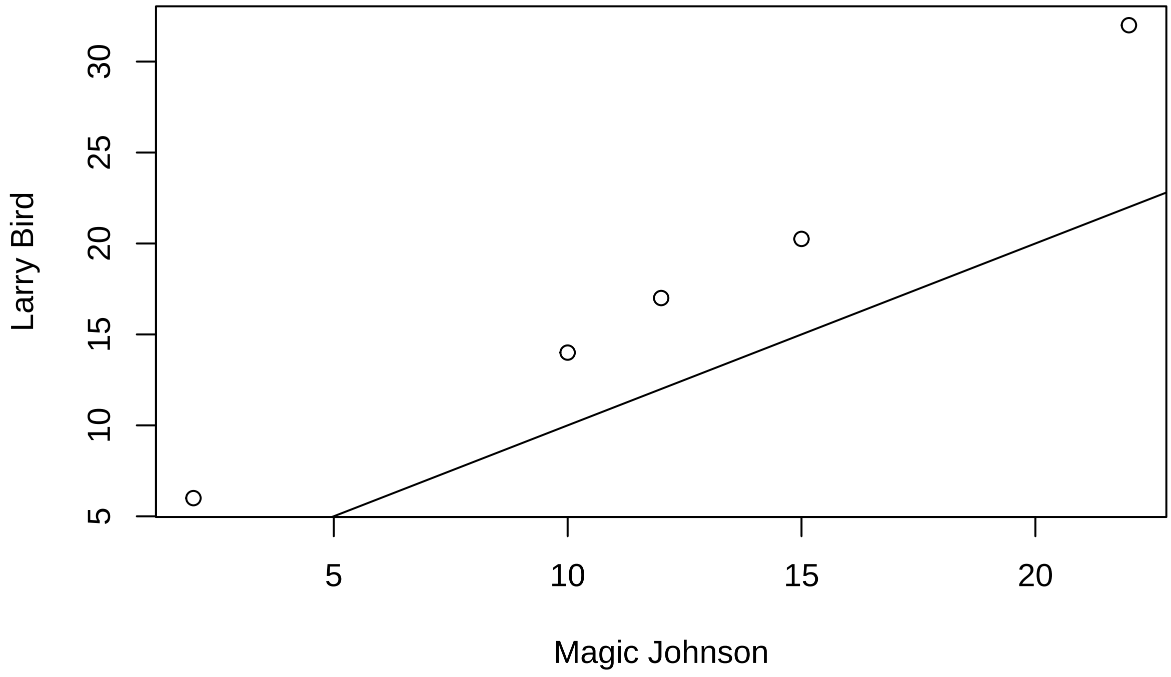


Figure 11: FGA: the 5 number summary of Magic Johnson and Larry Bird.

Number of asists (AST) per game

In his rookie season Magic Johnson passed 7.31 assists per game much higher than Larry Bird's 4.63 assists per game.

```
k<-23 #AST
x1<-na.omit(LB[,k])
x2<-na.omit(MJ[,k])
xx<-data.frame(c("MEJ","LB"),c(mean(x2),mean(x1)))
names(xx)<-c("Player","Average")
xx
```

```
##   Player  Average
## 1    MEJ 7.311688
## 2     LB 4.637681
```

The stem-and-leaf diagram shows that Magic Johnson's AST distribution is located to the right and has higher variability compare to Larry Bird's distribution.

```
stem(x1)
```

```
## 
##   The decimal point is at the |
## 
##    1 | 00
##    2 | 00000
##    3 | 00000000000000
##    4 | 000000000000000
##    5 | 00000000000000
##    6 | 000000
##    7 | 0000000
##    8 | 0000
##    9 | 0
##   10 | 0
```

```
stem(x2)
```

```
## 
##   The decimal point is at the |
## 
##    2 | 000000
##    4 | 0000000000000
##    6 | 0000000000000000000000000
##    8 | 00000000000000000
##   10 | 0000000000
##   12 | 000000
```

This is also indicated by Magic Johnson's median AST per game (7) compared to Larry Bird's median AST per game (4).

```
q.lb<-quantile(x1)
q.lb
```

```
##   0%  25%  50%  75% 100% 
##    1    3    4    6   10
```

```
q.mej<-quantile(x2)
q.mej
```

```
##   0%  25%  50%  75% 100% 
##    2    6    7    9   13
```

Note how Magic Johnson's 5 number summary, shown in Figure @ref(fig:figchp12m) is laying (almost parallel) below the 45^o lines.

```
plot(q.mej,q.lb,xlab="Magic Johnson",ylab="Larry Bird")
abline(0,1)
```

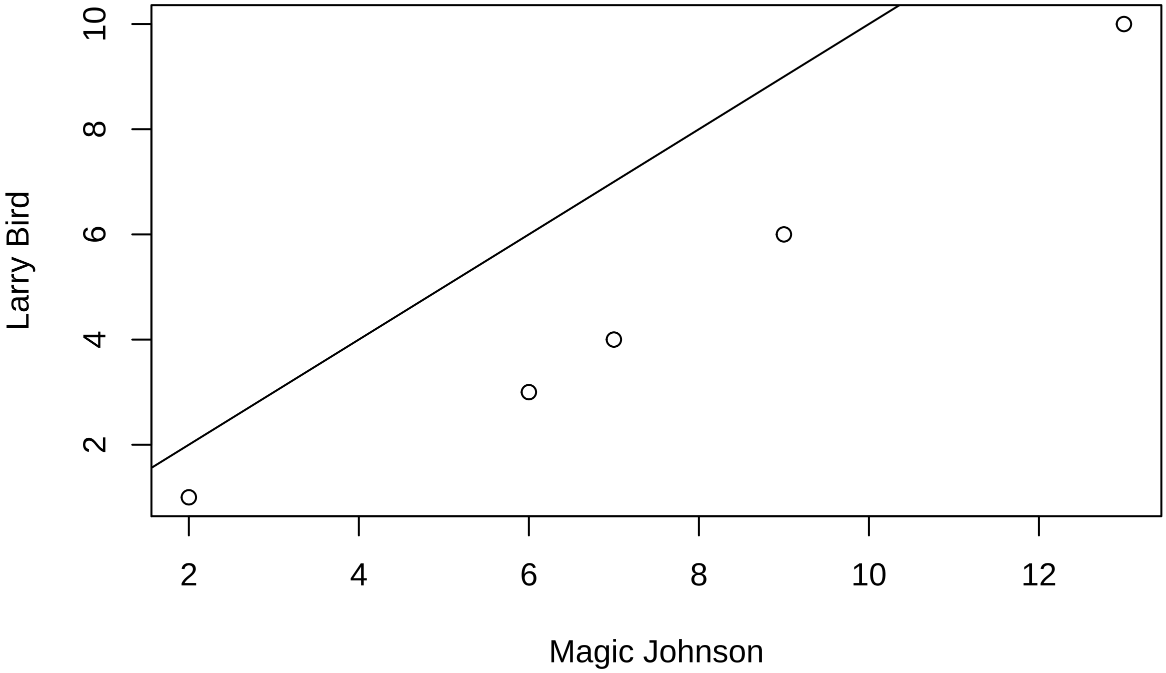


Figure 12: AST: the 5 number summary of Magic Johnson and Larry Bird.

Numerical summeries for location

Location and location parameter

The location of a distribution is the center of the distribution. It is the place where the data are concentrated. When several distributions have the same shape and width but are shifted relative to each other, then the location parameter is a measure of this shift. For example, consider a normal distribution with variance $\sigma^2 = 1$ and mean μ, the density function is given by

$$f(x, \mu, 1) = \frac{1}{\sqrt{2\pi}} e^{-} \frac{(x-\mu)^2}{2}.$$

Figure @ref(fig:figchp31a) shows three density functions of the normal distribution with $\mu = -2, 0$ and 2. The three distributions are shifted relative to each other and the value of μ determines the shift.

```r
x<-seq(from=-6,to=6,length=100)
dx1<-dnorm(x,-2,1)
dx2<-dnorm(x,0,1)
dx3<-dnorm(x,2,1)
plot(x,dx1,type="l",xlab="x",ylab="f(x)")
lines(x,dx2,col=2)
lines(x,dx3,col=3)
```

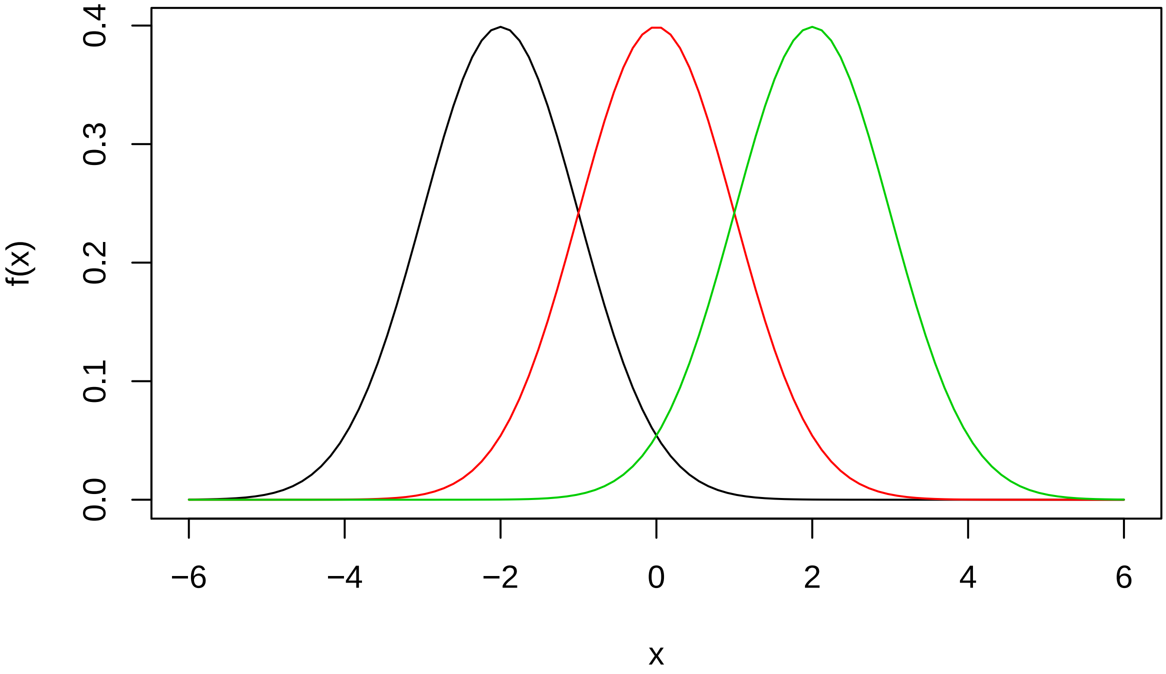


Figure 13: Location.

Location estimators

The sample mean, median and the trimmed mean are all estimators that can be used to estimate the location parameter. They belong to the class of $L-estimators$ which implies that these estimators are a linear function of the observations.

Order statistics

Let $X_1, X_2......X_n$ be a sample of size n. The sorted values $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$,from smallest to largest , are called the order statistic. For example, consider the sample 35, 80, 105, 96 and 35. The sorted values are 35, 35, 80, 96,105 imply that, $X_{(1)} = 35$, $X_{(3)} = 80$ and the maximum $X_{(5)} = 105$.

L-estimators

Let $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$ be the order statistics of a sample of size n, and let $a_1, a_2 \ldots, a_n$ be weights such that $0 \leq a_i \leq 1$ and $\sum_{i=1}^{n} a_i = 1$. An L-estimator, T_L, with weights $a_1, a_2 \ldots, a_n$ can be defined by

$$T_L = \sum_{i=1}^{n} a_i X_{(i)}.$$

Examples of L-estimators

YouTube tutorial: calculating Mean, Median, Range, Minimum and Maximum using R studio

For a short online YouTube tutorial, by BIO-RESEARCH, about the mean, median etc, using R see YTVD4.

The sample mean

Let $a_i = \frac{1}{n}$, $i = 1, 2, \ldots, n$, then

$$T = \sum_{i=1}^{n} a_i X_{(i)} = \sum_{i=1}^{n} \frac{1}{n} X_{(i)} = \bar{X}.$$

In out example the sample mean is $\bar{X} = \frac{1}{5}(35 + 35 + 80 + 96 + 105) = 70.2$.

```
x<-c(35,35,80,96,105)
mean(x)
```

```
## [1] 70.2
```

The sample median

The median is the value such that 50% of the data are less than or equal to. When n is odd, the median is equal to $X_{(\frac{n-1}{2}+1)}$. For example, the median of $35, 80, 105, 96, 35$ is 80, which is also $X_{(3)}$.

```
sort(x)
```

```
## [1]  35  35  80  96 105
```

```
median(x)
```

```
## [1] 80
```

If we define the weights

$$a_i = \begin{cases} 1 & \text{if } i = \frac{n-1}{2} + 1 \\ 0 & \text{otherwise} \end{cases}$$

then it follows that

$$T = \sum_{i=1}^{n} a_i X_{(i)} = 1 \times X_{(\frac{n-1}{2}+1)} = Median.$$

When n is even then

$$a_i = \begin{cases} 0.5 & \text{if } i = \frac{n}{2}, \frac{n}{2} + 1, \\ 0 & \text{otherwize.} \end{cases}$$

The median is $\frac{1}{2}X_{(\frac{n}{2})} + \frac{1}{2}X_{(\frac{n}{2}+1)}$.

The trimmed mean

The trimmed mean is the mean of the sample obtained after trimming a certain proportion of the observations at the upper and lower tails. For example, consider a sample of size 10:

$$25, 27, 39, 57, 57, 63, 69, 75, 76, 94$$

The 10%-trimmed mean , $T(0.2)$, is the average of the 8 observations

$$27, 39, 57, 57, 63, 69, 75, 76$$

which remain after trimming 10% of the observations at each tail, i.e the largest and the smallest values of the sample (25 and 94). The 10%-trimmed mean is therefore obtained by discarding 20% of the data (hence the notaion $T(0.2)$). In this example $T(0.2) = 57.875$. In R, we use the function mean() with the argument trim=. For our example,

```r
x<-c(25,27,39,57,57,63,69,75,76,94)
mean(x,trim=0.1)
```

```
## [1] 57.875
```

A 20%-trimmed mean, $T(0.4)$, is the average of the data after trimming 20% of the data at each side, i.e.,

$$T(0.4) = \frac{1}{6}\sum_{i=3}^{8} x_{(i)} = \frac{1}{6}(39 + 57 + 57 + 63 + 69 + 75) = 60$$

which can be calculate in R by

```r
mean(x,trim=0.2)
```

```
## [1] 60
```

The α%-trimmed mean belongs to the L-estimators class. If we define

$$a_i = \begin{cases} \frac{1}{n-2\alpha n} & \text{if} \quad n\alpha + 1 \le i \le n - n\alpha, \\ 0 & \text{otherwise.} \end{cases}$$

For $n = 10$ and $\alpha = 0.2$ (40% of the observations are trimmed), we have: $n - 2\alpha n = 10 - 4 = 6$, $n\alpha + 1 = 3$ and $n - n\alpha = 8$ and

$$T(2\alpha) = \sum_{i=1}^{n} a_i X_{(i)} = \frac{1}{n - 2\alpha n}\sum_{i=n\alpha+1}^{n-n\alpha} X_{(i)} = \frac{1}{6}\sum_{i=3}^{8} X_{(i)}$$

Example: the airquality data

The airquality dataset is a data frame in R contints information about daily air quality measurements in New York, May to September 1973. We focus on the daily mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island (the variable Ozone). Data are shwon below:

```
print(airquality$Ozone)
```

```
##   [1]  41  36  12  18  NA  28  23  19   8  NA   7  16  11  14  18  14  34   6
##  [19]  30  11   1  11   4  32  NA  NA  NA  23  45 115  37  NA  NA  NA  NA  NA
##  [37]  NA  29  NA  71  39  NA  NA  23  NA  NA  21  37  20  12  13  NA  NA  NA
##  [55]  NA  NA  NA  NA  NA  NA  NA 135  49  32  NA  64  40  77  97  97  85  NA
##  [73]  10  27  NA   7  48  35  61  79  63  16  NA  NA  80 108  20  52  82  50
##  [91]  64  59  39   9  16  78  35  66 122  89 110  NA  NA  44  28  65  NA  22
## [109]  59  23  31  44  21   9  NA  45 168  73  NA  76 118  84  85  96  78  73
## [127]  91  47  32  20  23  21  24  44  21  28   9  13  46  18  13  24  16  13
## [145]  23  36   7  14  30  NA  14  18  20
```

Figure @ref(fig:figchp32a) shows the histogram for the ozone level.

```
hist(airquality$Ozone,main="Ozone level")
```

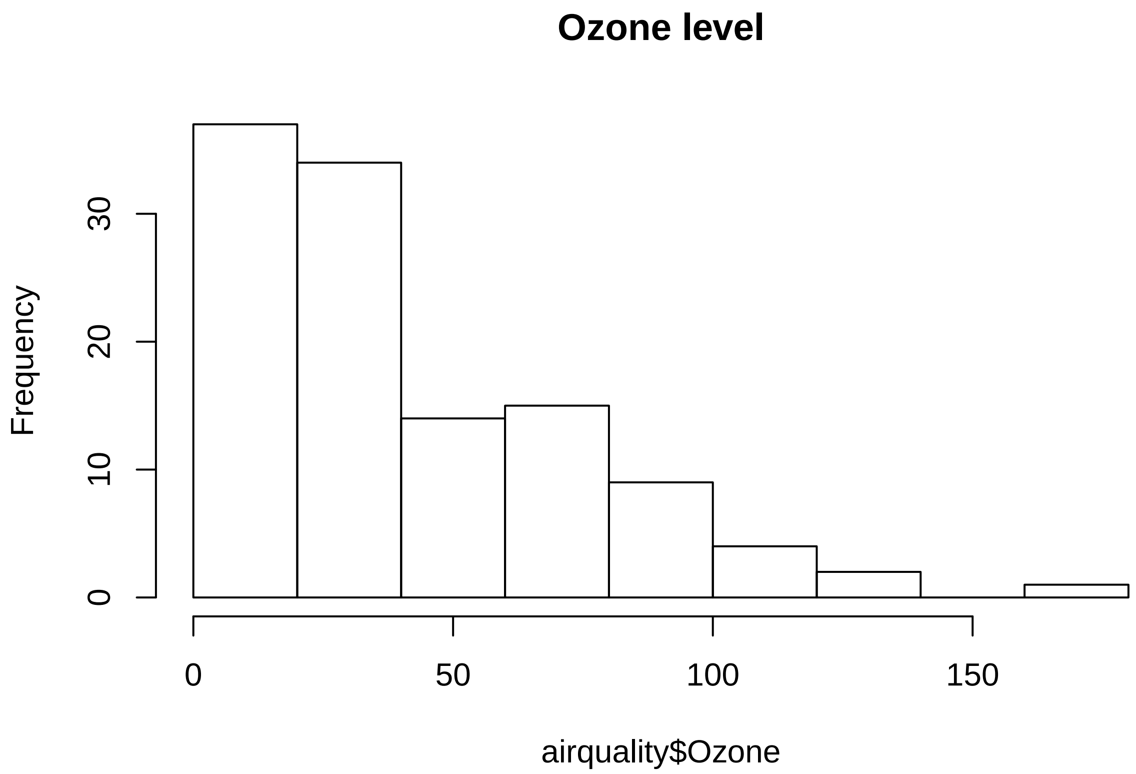


Figure 14: Histogram for the ozone level.

An equivalent information can be seen in the stem-and-leaf diagram.

```
stem(airquality$Ozone)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##    0 | 1467778999
##    1 | 01112233334444666688889
##    2 | 0000111123333334478889
##    3 | 001222455667799
##    4 | 01444556789
##    5 | 0299
##    6 | 134456
##    7 | 13367889
##    8 | 024559
##    9 | 1677
##   10 | 8
##   11 | 058
##   12 | 2
##   13 | 5
##   14 |
##   15 |
##   16 | 8
```

The distributiion is skwed to the right. This means that most of the days have a low ozone level while a (relarivly) small proportion of the days have a high ozone level. The ozone level for these days are located in the right tell of the distribution.The mean ozone level is equal 42.12.

```
Ozone1<-na.omit(airquality$Ozone)
 mean(Ozone1)
```

```
## [1] 42.12931
```

The sample mean infeluences from the higher ozone leves at the right tail of the distribution and therefore is higher than the median (31.5) which is the average of the two centeral values of the sample.

```
median(Ozone1)
```

```
## [1] 31.5
```

Note that the trimmed mean ($\alpha = 10\%$) is equal to 43.8 and its value is between the mean and the median.

```
mean(Ozone1, trim = 0.2)
```

```
## [1] 34.8
```

The number of 3 points attempt per game (3PA) made by the Golden State Warriors: 2010-2019

The Golden State Worrier (GSW) is one of the dominant teams in the NBA during the second decade of the 21 century. The team won 3 championships (2014/2015,2017/2018,2018/2019) and 5 Conference titles. The data frame GSW1 contains information about the three points attempts (per game) made by the GSW players during the regular season's games between 28/10/2009 and 10/04/2019. The team played 739 games during this period and made 7084 attempts for three points.

```
load("C:\\projects\\NBA\\GSW\\GSW1.RData")
head(xxx)
```

```
##   A3P YEAR
## 1   1 2010
## 2   3 2010
## 3   2 2010
## 4   1 2010
## 5   2 2010
## 6   2 2010
```

```r
length(xxx$A3P)
```

```
## [1] 739
```

```r
sum(xxx$A3P)
```

```
## [1] 7084
```

Our aim in the section is to visualized the pattern in the number of 3 point attempts over the years. The total number of 3 points attempts increased from 380 in 2010 to 1339 in 2019.

```r
tapply(xxx$A3P,xxx$YEAR,sum)
```

```
## 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
##  380  342  121  606  615  646  886  789 1360 1339
```

This trend can be detected when we use the mean and median of the number of 3 points attempts (3PA) per game as a summery statistics over the years. Notice how the median number of 3PA increases from 4 in 2010 to 17 in 2019.

```r
tapply(xxx$A3P,xxx$YEAR,mean)
```

```
##      2010      2011      2012      2013      2014      2015      2016      2017
##  4.750000  4.621622  4.653846  7.670886  7.884615  8.075000 11.215190  9.987342
##      2018      2019
## 16.585366 16.329268
```

```r
tapply(xxx$A3P,xxx$YEAR,median)
```

```
## 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
##  4.0  4.5  4.0  8.0  7.5  8.0 11.0 10.0 16.0 17.0
```

Both mean and median summarized the distribution of 3PA using one number, the location estimator. We can visualize the shift in 3PA using the stem-and-leaf plot discussed in Chapter 2. For example, the right shift in the 3PA's distribution can be clearly seen when we compare the stem-and-leaf plot of 2010 and 2018 (the last championship's season)

```r
stem(xxx$A3P[xxx$YEAR=="2010"])
```

```
## 
##   The decimal point is at the |
## 
##    0 | 00000
##    2 | 00000000000000000000000
##    4 | 000000000000000000000000
##    6 | 0000000000000
##    8 | 0000000000
##   10 | 00000
```

```r
stem(xxx$A3P[xxx$YEAR=="2018"])
```

```
## 
##   The decimal point is at the |
## 
##    6 | 000
##    8 | 000000
##   10 | 00000000
##   12 | 00000000
##   14 | 0000000000000
##   16 | 0000000
##   18 | 0000000000
##   20 | 000000
##   22 | 000000000000000
##   24 | 0000
##   26 | 0
##   28 |
##   30 | 0
```

The shift can be detected when we use the 5 number summary to compare the 3PA distribution between the two years.

```
tpa10<-quantile(xxx$A3P[xxx$YEAR=="2010"])
tpa10
```

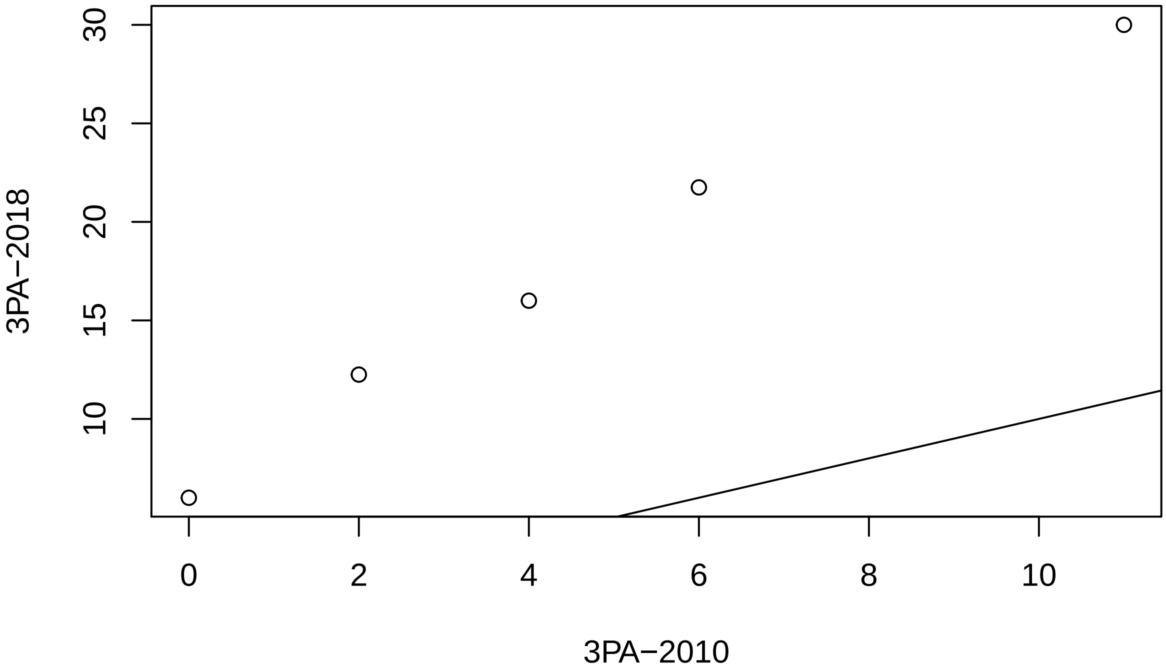
```
##   0%  25%  50%  75% 100%
##    0    2    4    6   11
```

```
tpa18<-quantile(xxx$A3P[xxx$YEAR=="2018"])
tpa18
```

```
##    0%   25%   50%   75%  100%
##  6.00 12.25 16.00 21.75 30.00
```

Figure @ref(fig:figchp33a) presents the 5 number summary of 3PA for 2010 and 2018 and shows the shift in the distribution between the two years.Note that the median increased from 4 to 16 for 2010 and 2018, respectively.

```
plot(tpa10,tpa18,xlab="3PA-2010",ylab="3PA-2018")
abline(0,1)
```


3PA−2018
30
25
20
15
10
0
2
4
6
8
10
3PA−2010

Graphical display for location

Introduction

In Chapter 2 and 3 we discuss numerical summaries for location and spread. This chapter is focused on graphical displays for location that are used to visualize the distribution of the data and use the numerical summaries that were previously discussed to represent the location of the distribution. We discuss the following graphical displays: dotplot, boxplot, histogram. For illustrations, we use both the lattice and the ggplot2 packages.

Online tutorials

YouTube tutorial: A dotplot using the lattice package

For a short online YouTube tutorial, by ramstatvid, about dotplot using the `lattice` package see YTVD5.

YouTube tutorial: A dotplot using the gg2plot package

For a short online YouTube tutorial, by ramstatvid, about dotplot using the `gg2plot` package see YTVD6

The singer dataset

The singer datset (the R object singer) is a data frame giving the heights of singers in the New York Choral Society. The variables are named height (inches) and voice.part which is the voice group of the singer (altos, sporanos, tenors and bass) Note that each voice group is subdivied into two groups, high voice and low voice (for example Bass1 and Bass2 and the lower and higher Bass voices, respectivly). Graphical despalys in the section were produced using the R packages lattice and ggplot2. The first 6 observatioans are given below.

```
head(singer)
```

```
##   height voice.part
## 1     64  Soprano 1
## 2     62  Soprano 1
## 3     66  Soprano 1
## 4     65  Soprano 1
## 5     60  Soprano 1
## 6     61  Soprano 1
```

The display in Figure @ref(fig:figchp41a) is a strip plot which plots the data of each voice group in a different strip. Figure @ref(fig:figchp41a) shows that there is a clear main pattern in the data: it is easy to distinguish between women (Sopranos and Altos) and men (Tenors and Basses). Women are clearly shorter than men. Other patterns will be discussed later.

To produce Figure figchp41a, we use the function dotplot() from the lattice R package. This function fas the general form of dotplot(factor~continuous variable). For the singer example we use

```
dotplot(singer$voice.part~singer$height,
        aspect=1,
        xlab="Mean Height (inches)")
```

An equivalent strip plot, show in Figure @ref(fig:figchp41b), can be produced using the ggplot() package. The function geom jitter implies that observatios with the same values will be ploted side by side and will not overlap so sample size would be seen as well in the plot.

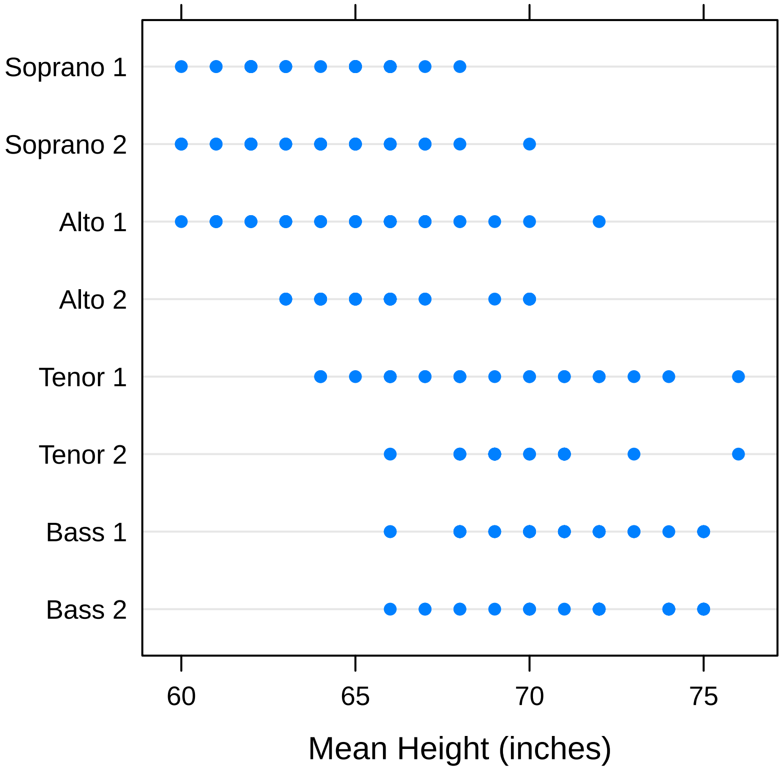


Figure 15: The singer dataset: dotplot by voice group using the lattice package.

```
ggplot(singer, aes(voice.part,height)) + geom_jitter(position = position_jitter(width = .05))
```

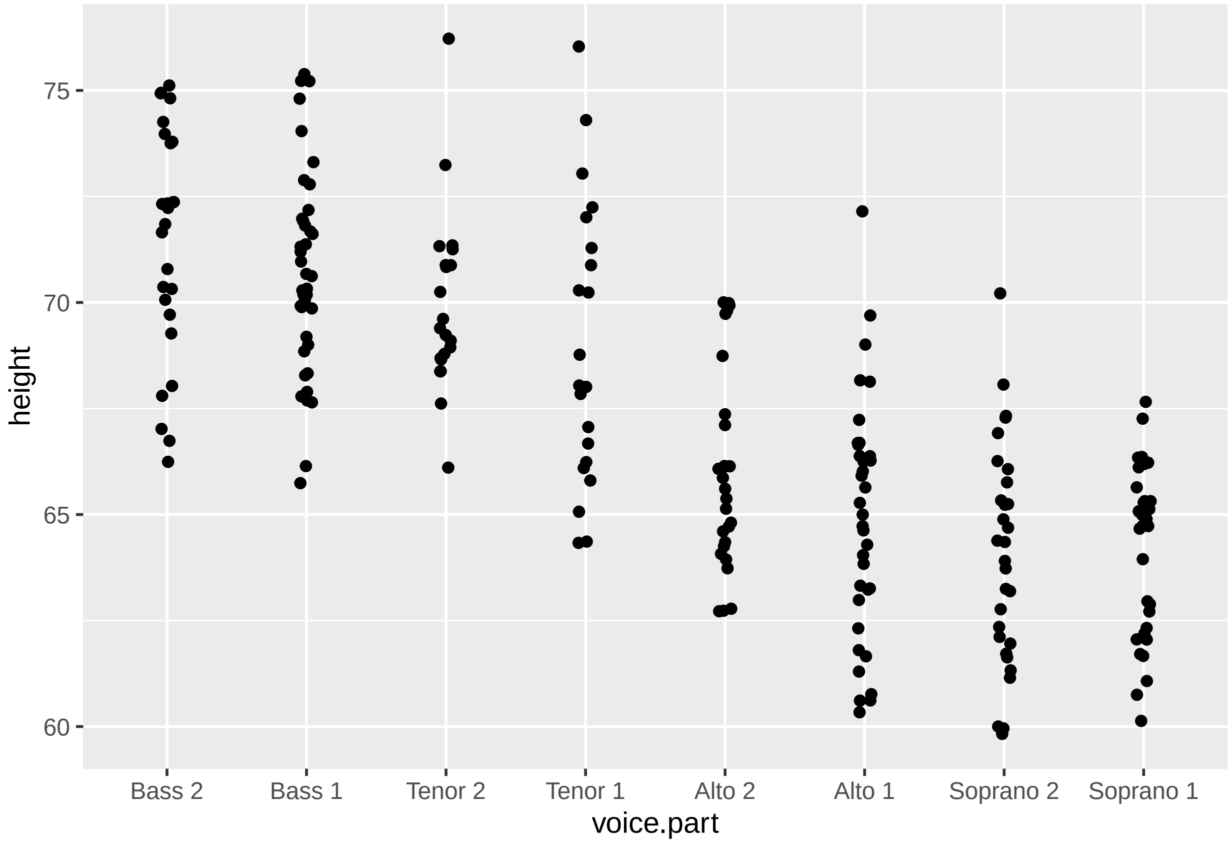


Figure 16: Dotplot using the ggplot2 package.

In order to get a better insight of other patterns, we can summarize the distribution of each group with the sample mean. In R this can be done using the function tapply(). A general call of the function has the form

<tt>tapply(numerical vector, factor, statistics)</tt>

To calculate mean of height by the voice group in the singer dataset we use

```
attach(singer)
tapply(singer$height,singer$voice.part,mean)
```

```
##    Bass 2    Bass 1   Tenor 2   Tenor 1    Alto 2    Alto 1 Soprano 2 Soprano 1
##  71.38462  70.71795  69.90476  68.90476  66.03704  64.88571  63.96667  64.25000
```

The group means points on the pattern that was already detected: on average men are taller than women. In addition, we can see that within each gender group, singers with lower voice are taller than singers with higher voice. For example, the average of the two bass groups (71.38 and 70.71 for Bass 1 and Bass 2 respectively) are higher than the average in the tenor groups (69.90 and 68.90 for Tenor 1 and Tenor 2 respectively). Among women, the sopranos are shorter, on average, than the altos. Within each voice group (all except the sopranos), the singers with lower voices (the second voice group Bass 2, Tenor 2 and Alto 2) are taller than the singers with the higher voices (the first group Bass 1, Tenor 1 and Alto 1). For example the mean of the Bass 2 group (71.38) is higher than the mean of the Bass 1 group (70.71), the same holding for the altos and the tenors. In order to vizualizse both data and the location sumamry, the mean in our example, we can use the function summary() with the option fun.y=mean which will include the mean for each voice group in the plot. Figure @ref(fig:figchp41c) shows the same information as Figure @ref(fig:figchp41b) with the addtion of the mean for each voice group.

```
ggplot(singer, aes(voice.part,height)) +
geom_point() +
stat_summary(geom = "point", fun.y = "mean", colour = "red", size = 4)
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

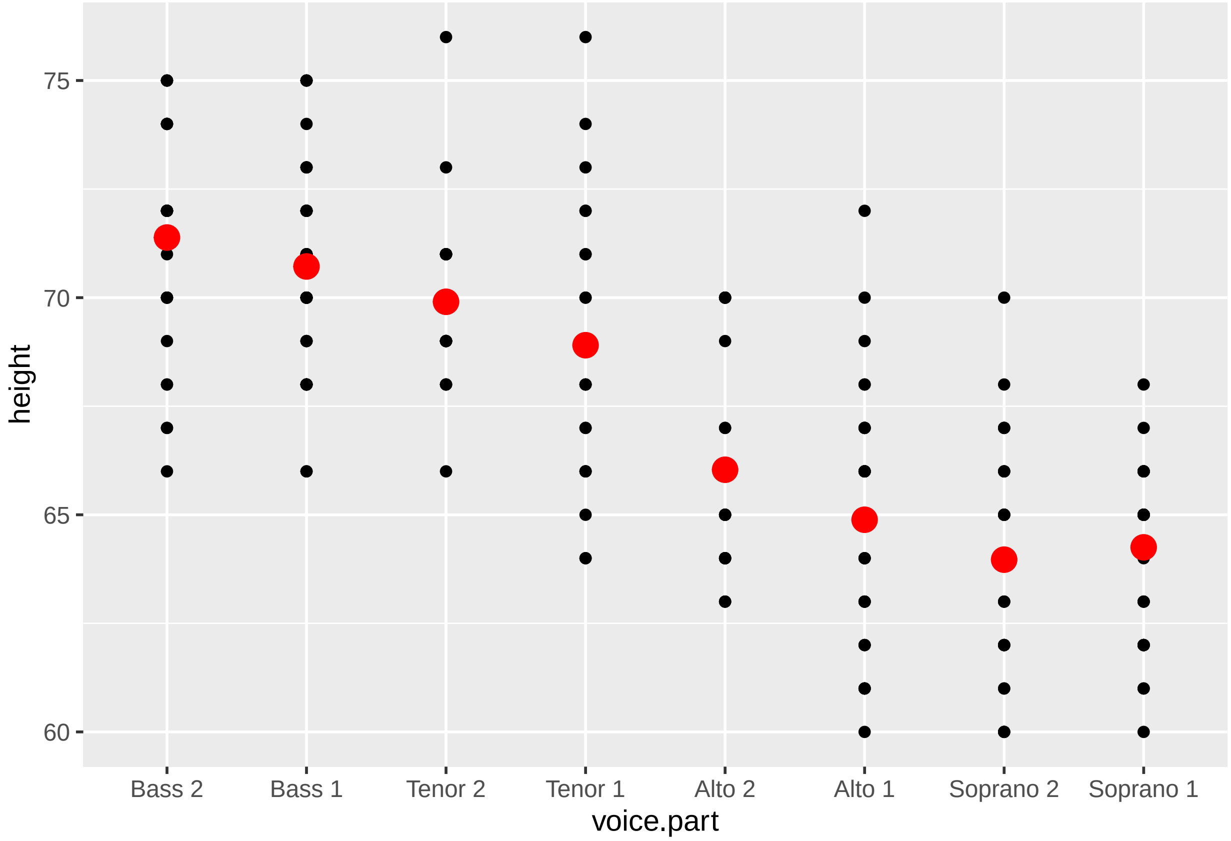


Figure 17: Dotplot using the ggplot2 package - data and mean at each voice group.

Another display of the location of each distribution is the box plot in Figure @ref(fig:figchp41d). In this figure, the location of each group is summarized by the median (the dot inside the box). Other aspects of this plot will be discussed in later chapters). Note that the function bwplot() is a part of the lattice package.

```
bwplot(as.factor(singer$voice.part)~ singer$height,
        data=singer,
        aspect=1,
        xlab="Height (inches)")
```

The multiway histogram shown in Figure @ref(fig:figchp41e) present the distributions of heights across the voice groups. Note how the distribution of height is shiffted from left to right across the voice levels.

```
histogram(~ singer$height | singer$voice.part,
        data=singer,
          layout = c(2, 4),
          aspect = 0.5,
       xlab = "height")
```

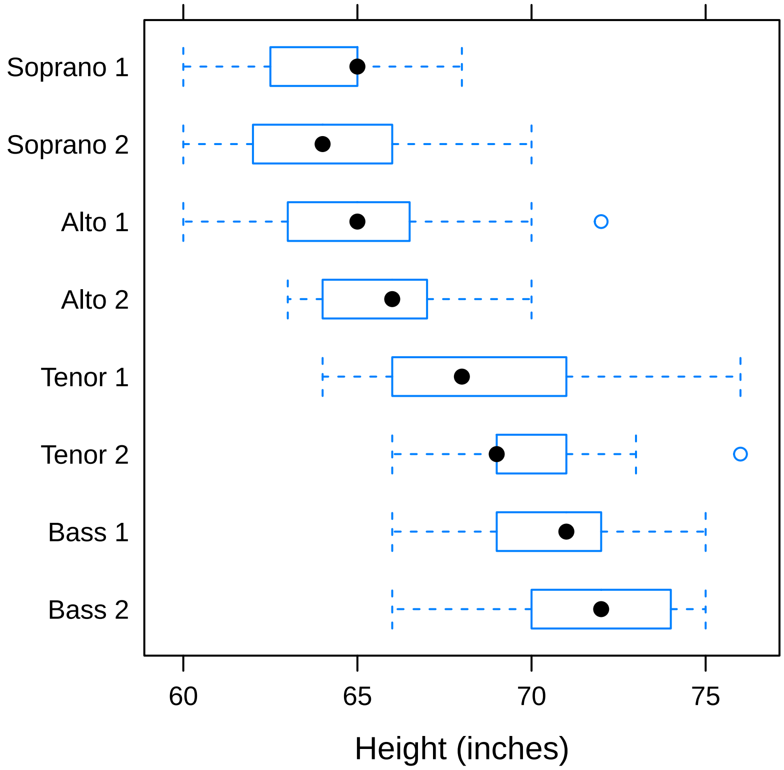


Figure 18: Side by side boxplots by voice group.

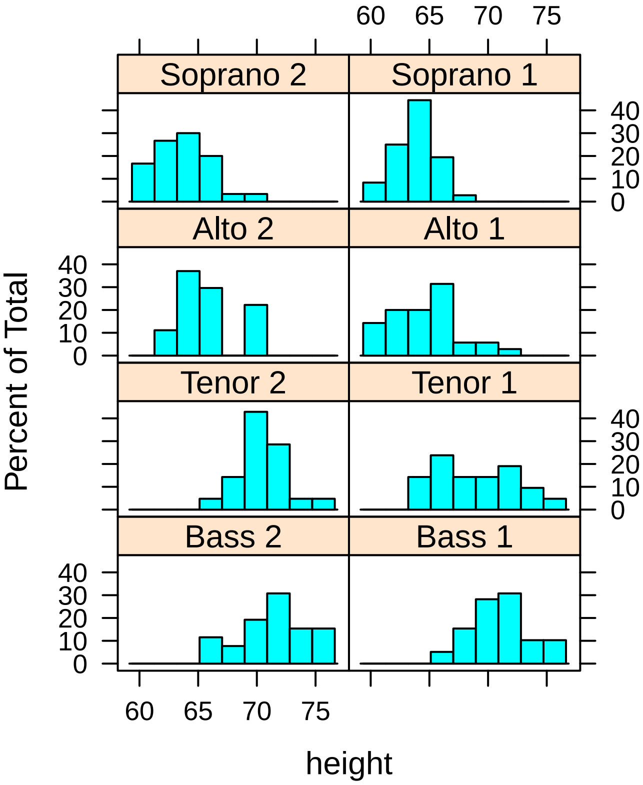


Figure 19: Histogram by voice group using the lattice package.

An equivalent multiway histogram can be produce with the ggplot2 package using addtional two layers in the basic plot. The first layer spacifys the plot type histogram() while the second layer indicates the factor for the plot partions, the function wrap(factor). Figure @ref(fig:figchp41f) shows the histogram produced using the ggplot2 package.

```
ggplot(singer, aes(height,fill = voice.part)) +
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

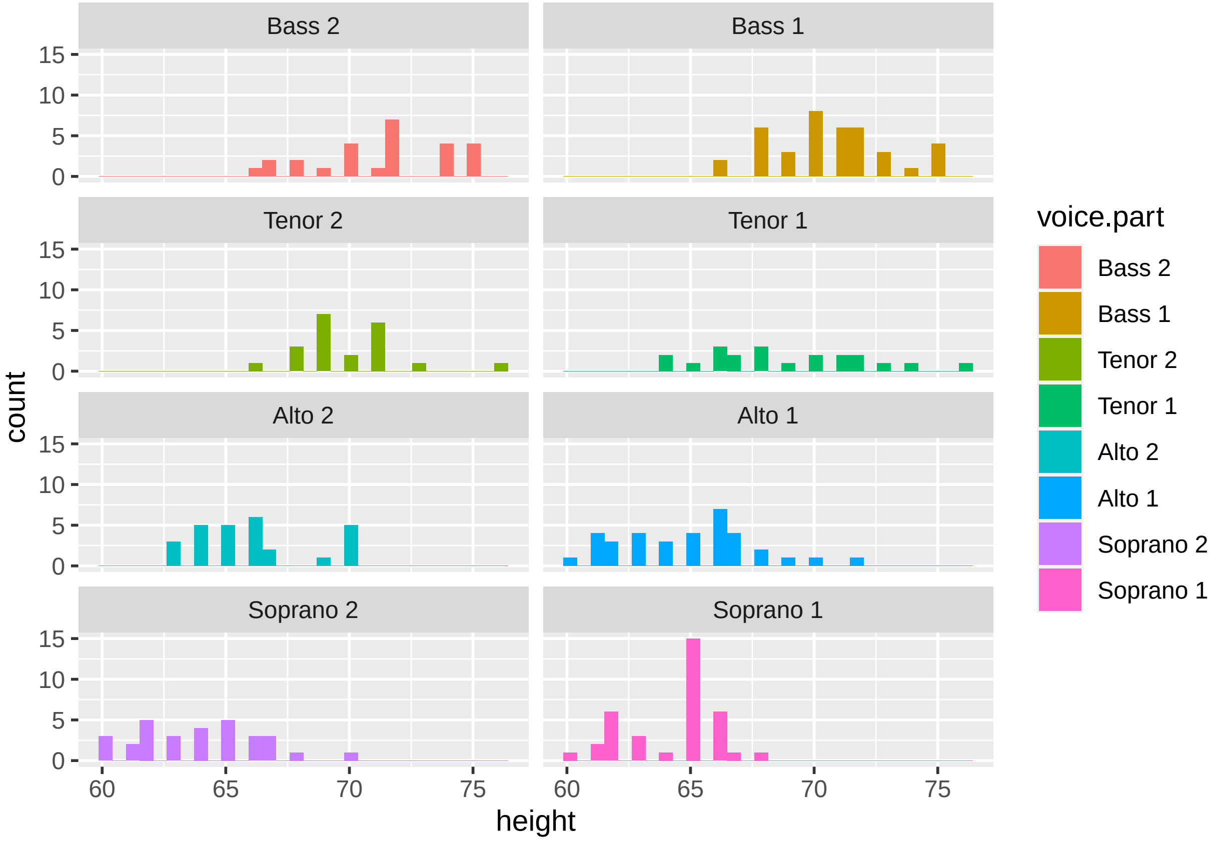


Figure 20: Histogram by voice group using the ggplot2 package.

Online tutorial:

YouTube tutorial: Histogram using the lattice package

For a short online YouTube tutorial, by ramstatvid, about histogram using the `gg2plot` package see YTVD7

Web tutorial: Histograms using the ggplot2 package

The the R Graph Gallery provide code for advance histogram visualization using the `ggplot` package. For examples of histograms see WMVD1

Michael Jordan's game score in championship years

Michael Jordan is arguably the greatest basketball player of all time (https://en.wikipedia.org/wiki/Michael_Jordan). During his career he won 6 NBA championships, in the seasons of 90/91, 91/92 92/93 and 95/96, 96/97, 97/98. He retired after the third championship (the first tripeat) and made first comeback to win the other three (the second tripeat). He retired again form the game after the 97/98 season and made a comeback in the season 2001/2002 and 02/03. In this section we focus on Michael Jordan's game score, in the regular season and playoff, in his 6 championship seasons. The game score (GmSc) is an individual performance indicator that will be discussed in length in Chapter XX.

```
load("C:\\projects\\NBA\\MJKB\\MJ2.RData")
load("C:\\projects\\NBA\\MJKB\\MJ3.RData")
```

The mean GmSc in the regular season games is given below and shows a decreasing pattern from the first three championship to the second three-peat.

```
tapply(xx21$x.rs,xx21$y.rs,mean)
```

```
##     1991     1992     1993     1996     1997     1998
## 26.21341 24.43875 25.51538 23.43537 22.13171 20.01951
```

The same pattern is observed for the GmSc in the playoff games.

```
tapply(xx21.po$x.po,xx21.po$y.po,mean)
```

```
##     1991     1992     1993     1996     1997     1998
## 27.42941 25.07727 26.17368 21.93333 22.22105 22.18571
```

We can visualize the distribution of the GmSc by year using the bwplot() function of the package lattice. Note that the R object y.rs is the year and the object x.rs is the GmSc. Figure @ref(fig:figchp41g1) shows the boxplot for the GmSc by year in the regular season while Figure @ref(fig:figchp41g2) shows the boxplot for the GmSc by year in the playoff.

```
  bwplot(xx21$y.rs~xx21$x.rs,
              data = xx21,
              layout = c(1, 1),
              aspect = 1,
              xlab = " Michael Jordan game score per year (regular season)")
```

```
   bwplot(xx21.po$y.po~xx21.po$x.po,
              data = xx21.po,
              layout = c(1, 1),
              aspect = 1,
              xlab = " Michael Jordan game score per year (playoff)")
```

A dotplot of MJ game score with both mean and median as numerical summaries for location, shown in Figure @ref(fig:figchp41h1), can be produced using the function stat_summary with the options fun.y = "mean" and fun.y = "median", respectively. The smooth line is added using the function smooth(aes(group=1)).

```
ggplot(xx21, aes(xx21$y.rs,xx21$x.rs)) +
geom_point() +
geom_smooth(aes(group=1))+
stat_summary(geom = "point", fun.y = "mean", colour = "red", size = 4)+
stat_summary(geom = "point", fun.y = "median", colour = "blue", size = 4)+
xlab("Year")+
ylab("GmSc")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

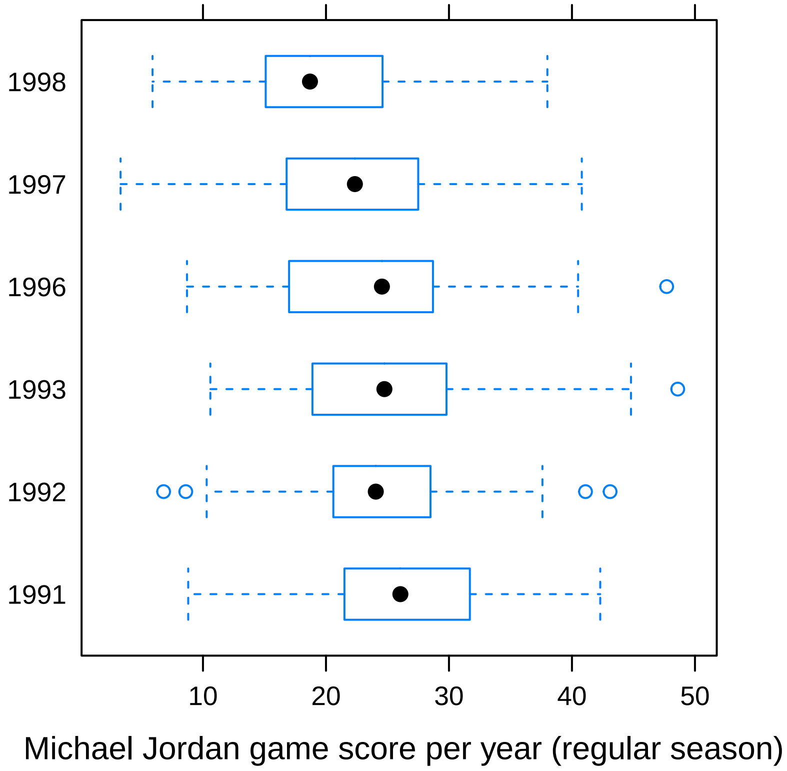


Figure 21: Boxplot for MJ game score in championchip years.

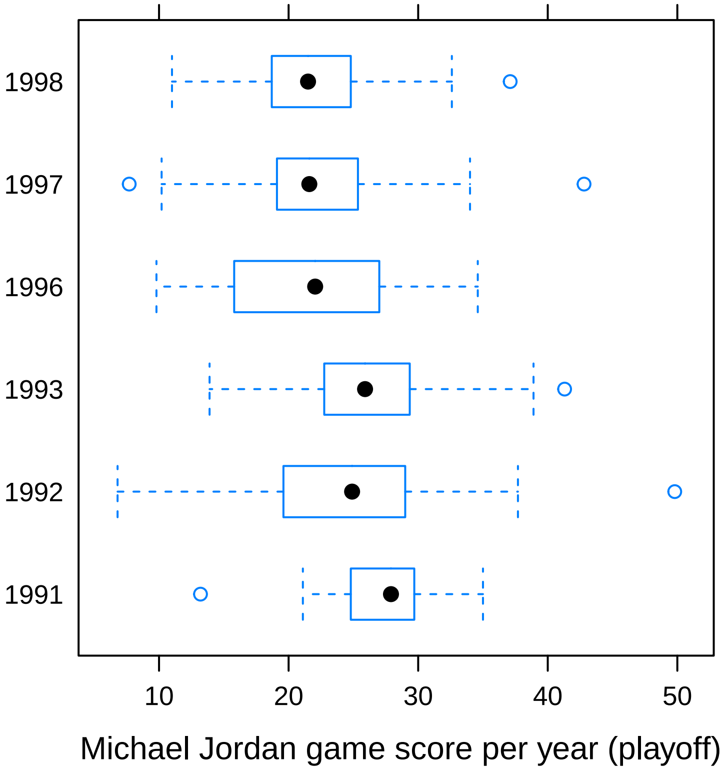


Figure 22: Boxplot for MJ game score in championchip years.

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
## Warning: Use of `xx21$y.rs` is discouraged. Use `y.rs` instead.
## Warning: Use of `xx21$x.rs` is discouraged. Use `x.rs` instead.
## Warning: Use of `xx21$y.rs` is discouraged. Use `y.rs` instead.
## Warning: Use of `xx21$x.rs` is discouraged. Use `x.rs` instead.
## Warning: Use of `xx21$y.rs` is discouraged. Use `y.rs` instead.
## Warning: Use of `xx21$x.rs` is discouraged. Use `x.rs` instead.
## Warning: Use of `xx21$y.rs` is discouraged. Use `y.rs` instead.
## Warning: Use of `xx21$x.rs` is discouraged. Use `x.rs` instead.
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

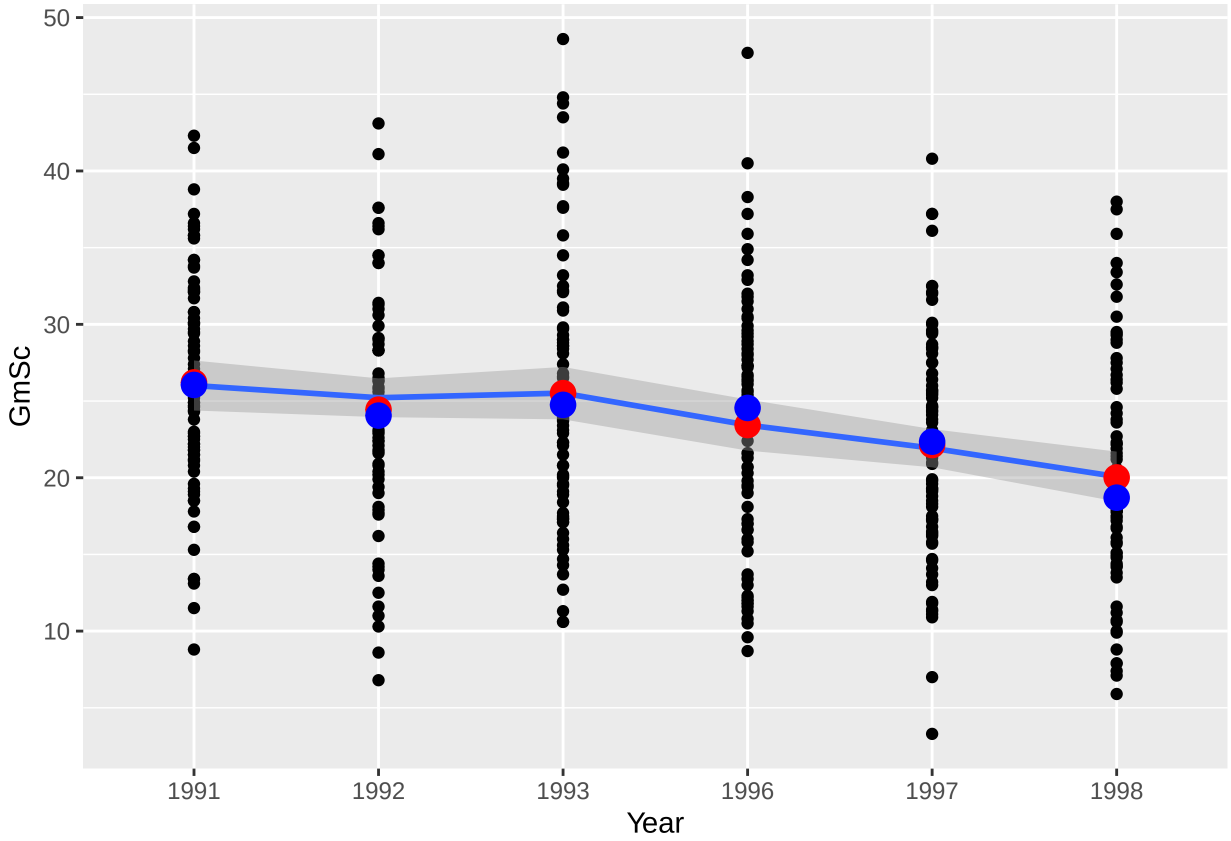


Figure 23: Dotplot for MJ game score in championchip years (regular season).

Figure @ref(fig:figchp41h2) presents the dotplot for the playoff that was produced in a similar way.

```
ggplot(xx21.po, aes(xx21.po$y.po,xx21.po$x.po)) +
geom_point() +
geom_smooth(aes(group=1))+
stat_summary(geom = "point", fun.y = "mean", colour = "red", size = 4)+
stat_summary(geom = "point", fun.y = "median", colour = "blue", size = 4)+
xlab("Year")+
ylab("GmSc")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: `fun.y` is deprecated. Use `fun` instead.
## Warning: Use of `xx21.po$y.po` is discouraged. Use `y.po` instead.
## Warning: Use of `xx21.po$x.po` is discouraged. Use `x.po` instead.
## Warning: Use of `xx21.po$y.po` is discouraged. Use `y.po` instead.
## Warning: Use of `xx21.po$x.po` is discouraged. Use `x.po` instead.
## Warning: Use of `xx21.po$y.po` is discouraged. Use `y.po` instead.
## Warning: Use of `xx21.po$x.po` is discouraged. Use `x.po` instead.
## Warning: Use of `xx21.po$y.po` is discouraged. Use `y.po` instead.
## Warning: Use of `xx21.po$x.po` is discouraged. Use `x.po` instead.
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

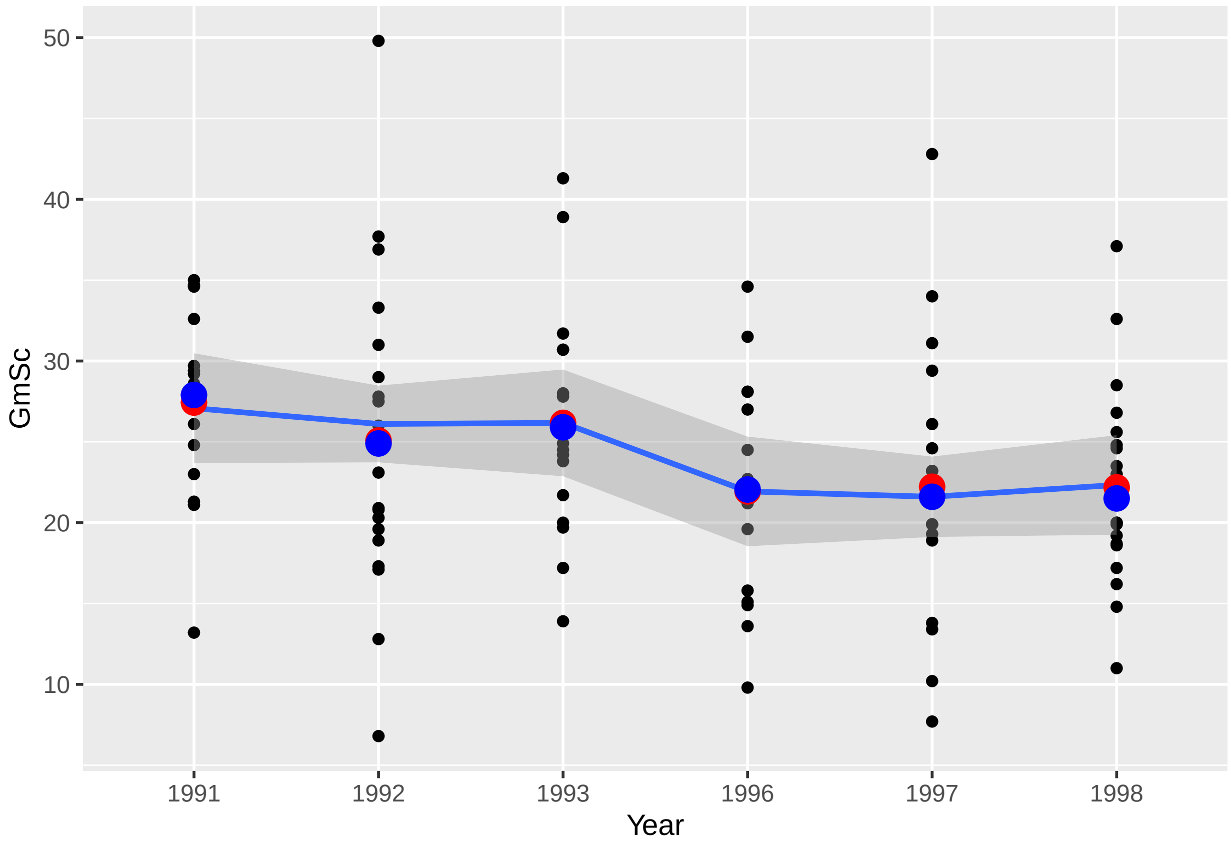


Figure 24: Dotplot for MJ game score in championchip years (playoff).

```
#grid.arrange(p1, p2, nrow = 1)
```

The number of 3 points attempt made by the Golden State Warriors: 2010-2019

In Chapter 3 we saw that the median number of 3 points attempts (3PA) per game made by GSW players increased from 4 in 2010 to 17 in 2019. In total, GSW had 739 games during this period in which the team attempt 7084 three points shots.

```
load("C:\\projects\\NBA\\GSW\\GSW1.RData")
head(xxx)
```

```
##   A3P YEAR
## 1   1 2010
## 2   3 2010
## 3   2 2010
## 4   1 2010
## 5   2 2010
## 6   2 2010
```

```
length(xxx$A3P)
```

```
## [1] 739
```

```
sum(xxx$A3P)
```

```
## [1] 7084
```

The mean and variance of 3PA per game by year is shown below

```
mx<-tapply(xxx$A3P,xxx$YEAR,mean)
vx<-tapply(xxx$A3P,xxx$YEAR,var)
data.frame(mx,vx)
```

```
##             mx        vx
## 2010  4.750000  7.050633
## 2011  4.621622  3.745280
## 2012  4.653846  6.315385
## 2013  7.670886  7.787731
## 2014  7.884615 10.389111
## 2015  8.075000  7.158861
## 2016 11.215190  9.812074
## 2017  9.987342  6.833171
## 2018 16.585366 28.961759
## 2019 16.329268 32.075429
```

The boxplot presented in Figure @ref(fig:figchp41i), was produced using the bwplot() function of the lattice package and reveals clearly the increasing pattern over time. A general call of the function have the form bwplot(continuous variable~factor}. In our example we use

```
   bwplot(xxx$A3P~xxx$YEAR,
                data = xxx,
                layout = c(1, 1),
                aspect = 1,
                xlab = "3PA by year")
```

A dotplot for the 3PA over time in which the sample mean is used as numerical summary for location is shown in Figure @ref(fig:figchp41j). Note that the mean was added to the plot using the function stat_summary.

```
ggplot(xxx, aes(xxx$YEAR,xxx$A3P)) +
geom_point() +
stat_summary(geom = "point", fun.y = "mean", colour = "red", size = 4)
```

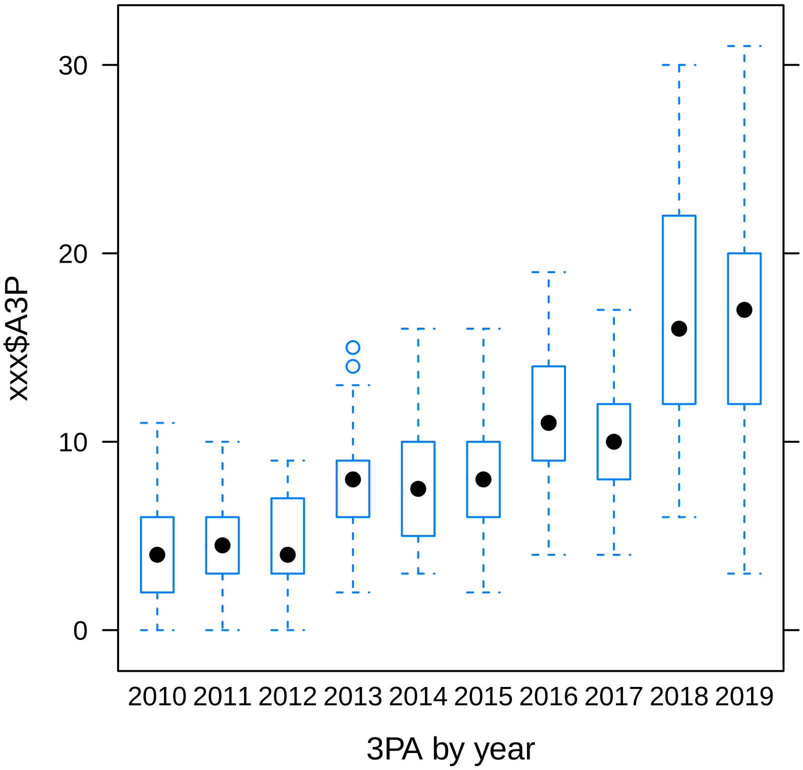


Figure 25: Boxplot over time (the lattice package).

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Use of `xxx$YEAR` is discouraged. Use `YEAR` instead.

## Warning: Use of `xxx$A3P` is discouraged. Use `A3P` instead.

## Warning: Use of `xxx$YEAR` is discouraged. Use `YEAR` instead.

## Warning: Use of `xxx$A3P` is discouraged. Use `A3P` instead.
```

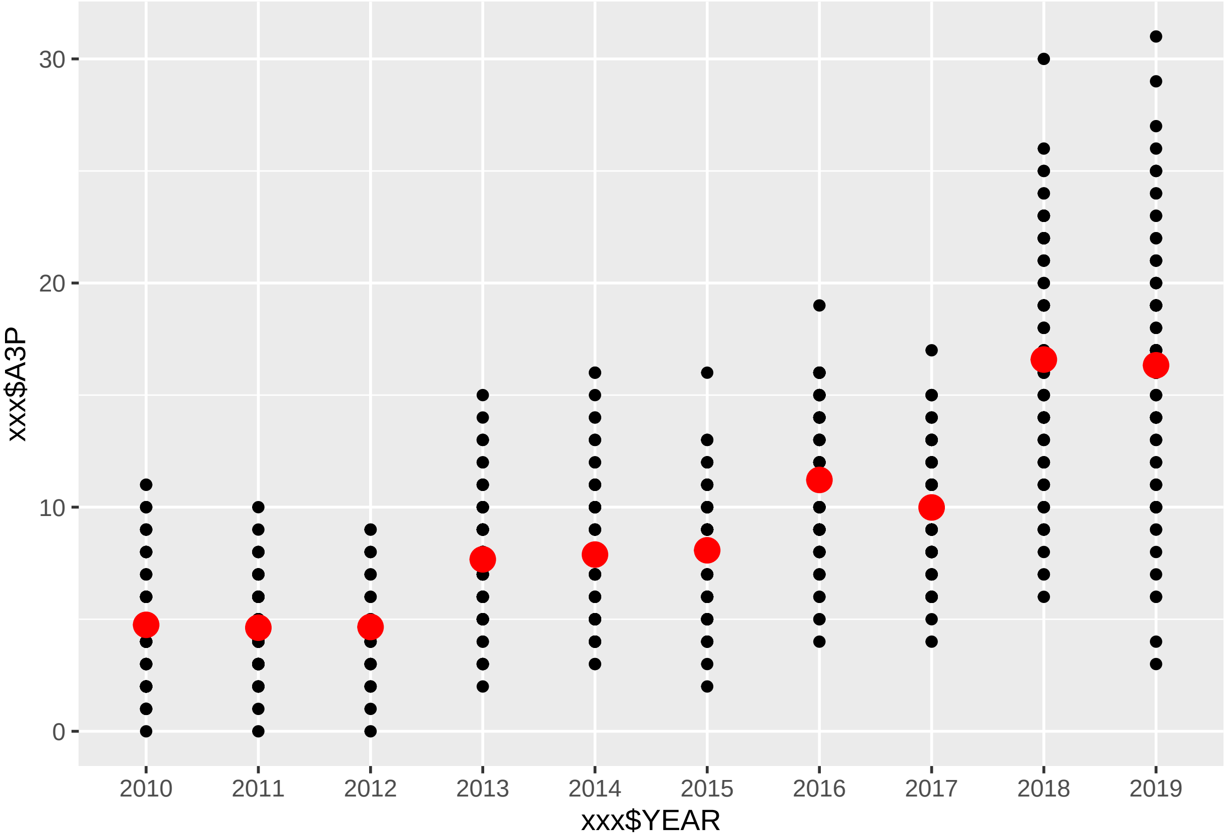


Figure 26: Dotplot over time (the gg2plot package).

Both figures reveal that the variability in the number of 3 point attempts increase with the time (as expected, since the number of 3PA is a count variable for which the variability depends on the mean). This can also be seen in Figure @ref(fig:figchp41k) below.

```
plot(mx,vx,xlab="mean 3PA per year",ylab="variance 3PA per year")
```

```
cor(mx,vx,method = c("spearman"))
```

```
## [1] 0.7939394
```

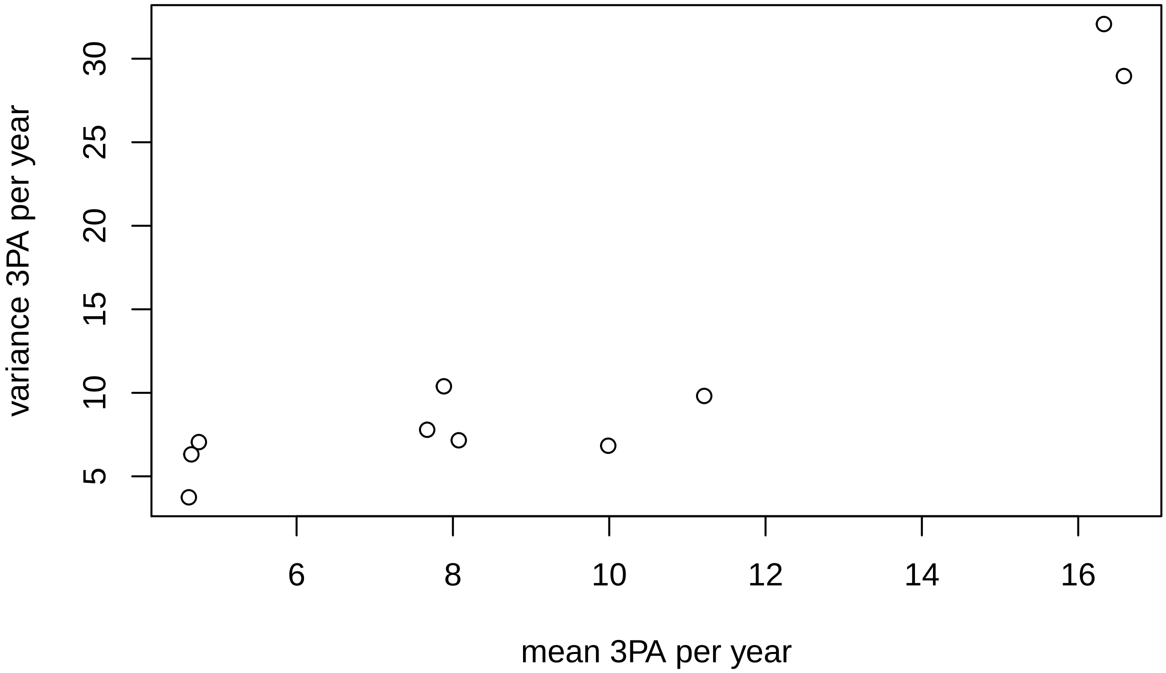


Figure 27: 3PA: mean versus variability over time.

Spread

Main concepts

Up till now we summarized the distribution of the data with location estimators, in this chapter we focus on the spread. We want to measure how close the data are to each other and how concentrate the data around the center of the distribution.We focus on both numerical summaries and graphical displays for spread, we discuss the sample variance, the fourth-spared and the MAD as measures for the sample spread and the boxplot and violin plot as graphical displays for spread.

Consider the following hypothetical samples:

$$(-1, 0, 1) \quad \text{and} \quad (-50, 0, 50).$$

Both samples are symmetric around 0, the location estimators for both sample are the same. The data in the first sample range from -1 to 1, in the second sample the data range from -50 to 50. The variablity in the second sample is higher. In this schapter we focus on estimators to measure the variablity of the sample.

Spread estimators

Sample variance

The most simple measure for spread is the sample variance given by:

$$S_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2.$$

Fourth-spared

A more robust estimator for the spread of the distribution is the fourth-spread (the interquartile range) given by

$$\text{fourth spread} = \text{upper fourth} - \text{lower fourth} = Q_3 - Q_1.$$

The fourth-spread is the difference between the 75% and the 25% quantiles of the data. It is the range of 50% of the data in the center of the distribution and therefore it is more robust estimator than the variance since it does not affect from outliers at the tails as the sample variance. Consider a sample of 8 observations:

$$24, \underbrace{35, 39, 50, 60, 60}, 75, 80$$

the fourth-spread is 25 (60-35) and the sample variance $S_x^2 = 382.9$. Now, suppose that we change the sample to

$$24, \underbrace{35, 39, 50, 60, 60}, 75, 800$$

the fourth-spread remains the same but the sample variance now is equal to 70762.98.

MAD - Median Absolute Deviation

Another robust measure of spread is the MAD, which is the Median of the Absolute Deviation form the median, given by

$$MAD = Median|X_i - M|.$$

where M is the sample median, $M = Median(X_1, X_2 \dots, X_j, \dots, X_n)$. In our example the median is 55, when we subtract the median we have $(-31, -20, -16, -5, 5, 5, 20, 25)$. The MAD is the median of the absolute values, in this example the MAD is 18. In R we have

```
xi <- c(24,35,39,50,60,60,75,80)
median(xi)
```

```
## [1] 55
```

The absolute deviation form the median

```
xi-median(xi)
```

```
## [1] -31 -20 -16  -5   5   5  20  25
```

```
abs(xi-median(xi))
```

```
## [1] 31 20 16  5  5  5 20 25
```

with median equal to 18, which is the MAD.

```
sort(abs(xi-median(xi)))
```

```
## [1]  5  5  5 16 20 20 25 31
```

```
median(abs(xi-median(xi)))           # the MAD
```

```
## [1] 18
```

When we change the last value to 800 we calculate the MAD using the R function mad()

```
xi[8] <- 800                        # change the maximum value to 800
xi
```

```
## [1]  24  35  39  50  60  60  75 800
```

```
mad(xi,constant=1)                  # calculate the MAD
```

```
## [1] 18
```

Boxplot: A graphical display for spread and location

Boxplot is a graphical display which shows the location, the spread and the shape of the distribution. The location is summarized by the median, the spread is summarized by the fourth-spread which is simply the length of the box in the boxplot (the shape will be discussed in Chapter~7). Figure @ref(fig:figchp51) shows histograms and boxplots for random samples of 1000 observations from $N(0, 1)$ and $N(0, 3^2)$.

```
x1<-rnorm(1000,0,1)
par(mfrow=c(2,2))
hist(x1,main="random sample from N(0,1)",xlim=c(-10,10))
boxplot(x1,ylim=c(-10,10))
x2<-rnorm(1000,0,3)
hist(x2,main="random sample from N(0,9)",xlim=c(-10,10))
boxplot(x2,ylim=c(-10,10))
```

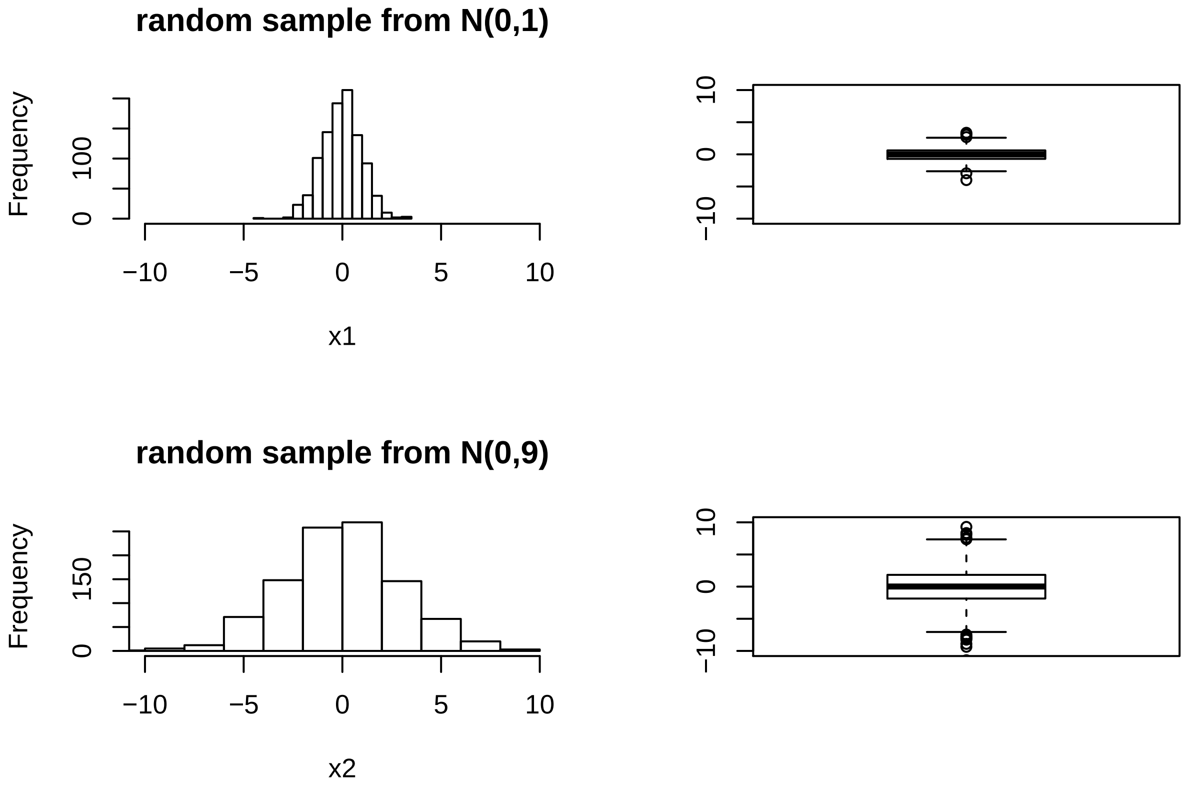


Figure 28: Samples from normal distributions.

The upper and lower adjacent values in the boxplot are given by

$$\text{Upper adjacent value} = Min\left\{max(X), Q_3 + 1.5(Q_3 - Q_1)\right\},$$

$$\text{Lower adjacent value} = max\left\{min(X), Q_1 - 1.5(Q_3 - Q_1)\right\}.$$

The upper and lower adjacent values are used to identify s and extrime values. Observations higher than the upper adjacent value or smaller than the lower adjacent value are considered to be outliers. The histogram and boxplot for the airquality data, discussed in Chapter 4 and shown in Figure @ref(fig:figchp52) reveal a skewed distribution with few outliers at the upper tail (histogram). In the boxplot these outliers can be identified above the upper adjacent value.

```
par(mfrow=c(1,2))
airquality1<-na.omit(airquality)
hist(airquality1$Ozone)
boxplot(airquality1$Ozone)
```

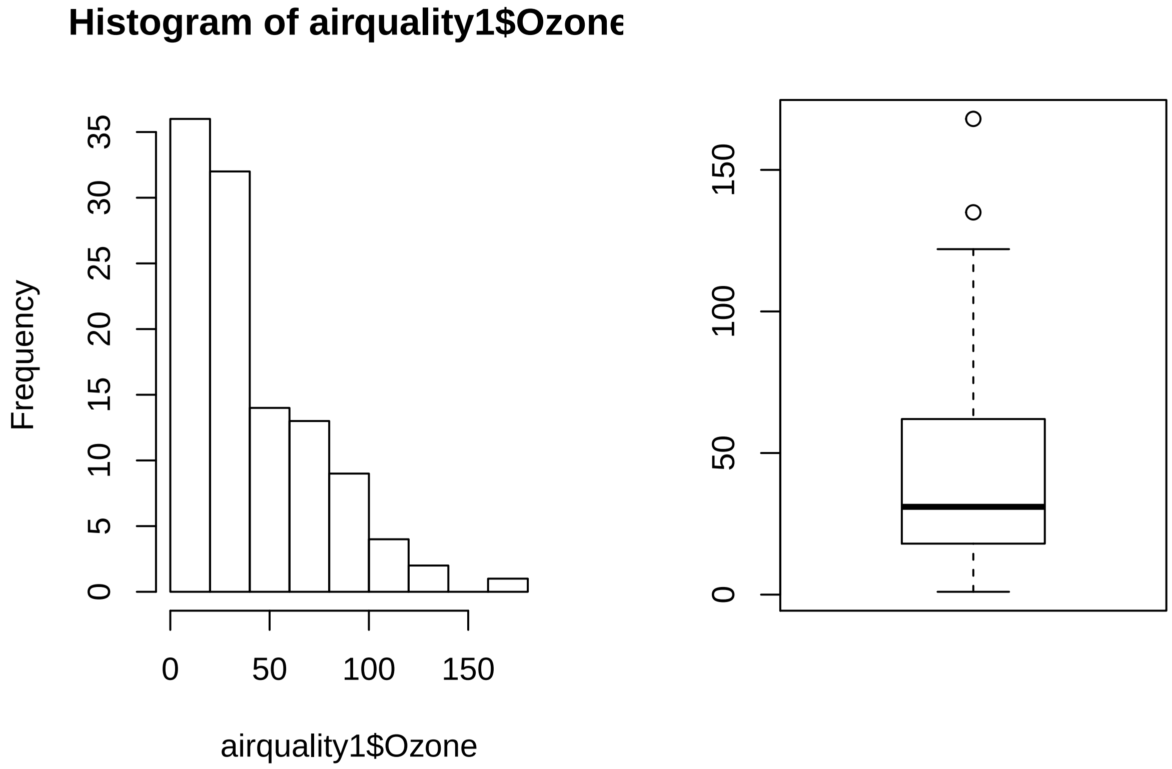


Figure 29: Histogram and boxplot for the airquality data.

Graphical displays for spread and location for the singers dataset

Online tutorials

YouTube tutorial: Boxplot in R

For a short online YouTube tutorials:

- by Data Science Tutorials, about boxplot using the ggplot2 package see YTVD8.
- by LawrenceStats, about boxplot using the ggplot2 package see YTVD9

Web tutorial: Advanced boxplots in R

Example for advanced boxplots in R using the ggplot2 package and code to produce the plots can be found in the R Graph Gallery website here WAVD2.

Boxplot for the singers data

A boxplot for the singers' height by voice group that was produced using the function geom boxplot() is shown in Figure @ref(fig:figchp53)

```
ggplot(singer, aes(voice.part,height)) + geom_boxplot()
```

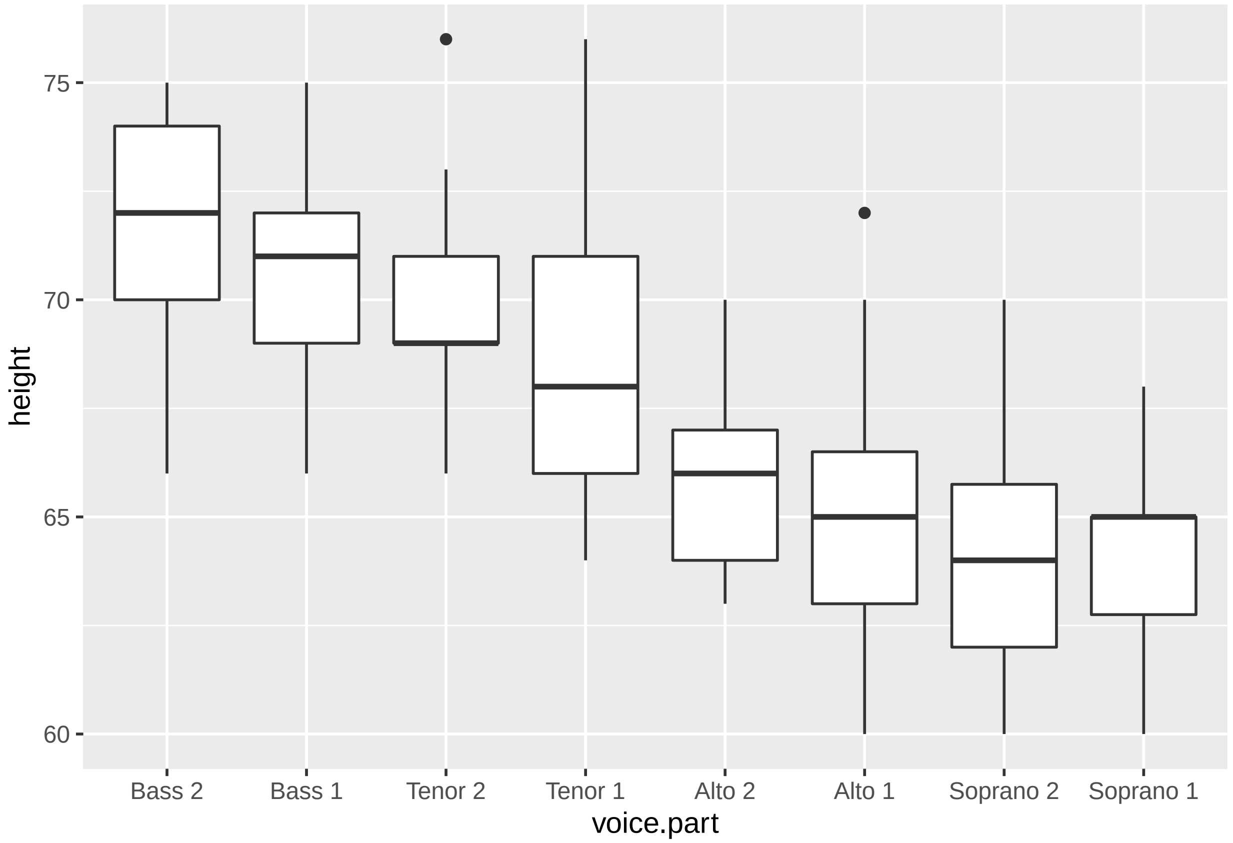


Figure 30: Boxplot for the singers data (1).

Figure @ref(fig:figchp54) shows the same boxplot in which colors (by group) can be added to the boxplot using the argument fill=voice.part. Note that the object voice.part is a factor.

```
ggplot(singer, aes(voice.part,height,fill=voice.part)) + geom_boxplot()
```

Figure @ref(fig:figchp55) shows the boxplot presented Figure @ref(fig:figchp53) in which the data were be added to the boxplot using the argument geom = c("boxplot", "jitter").

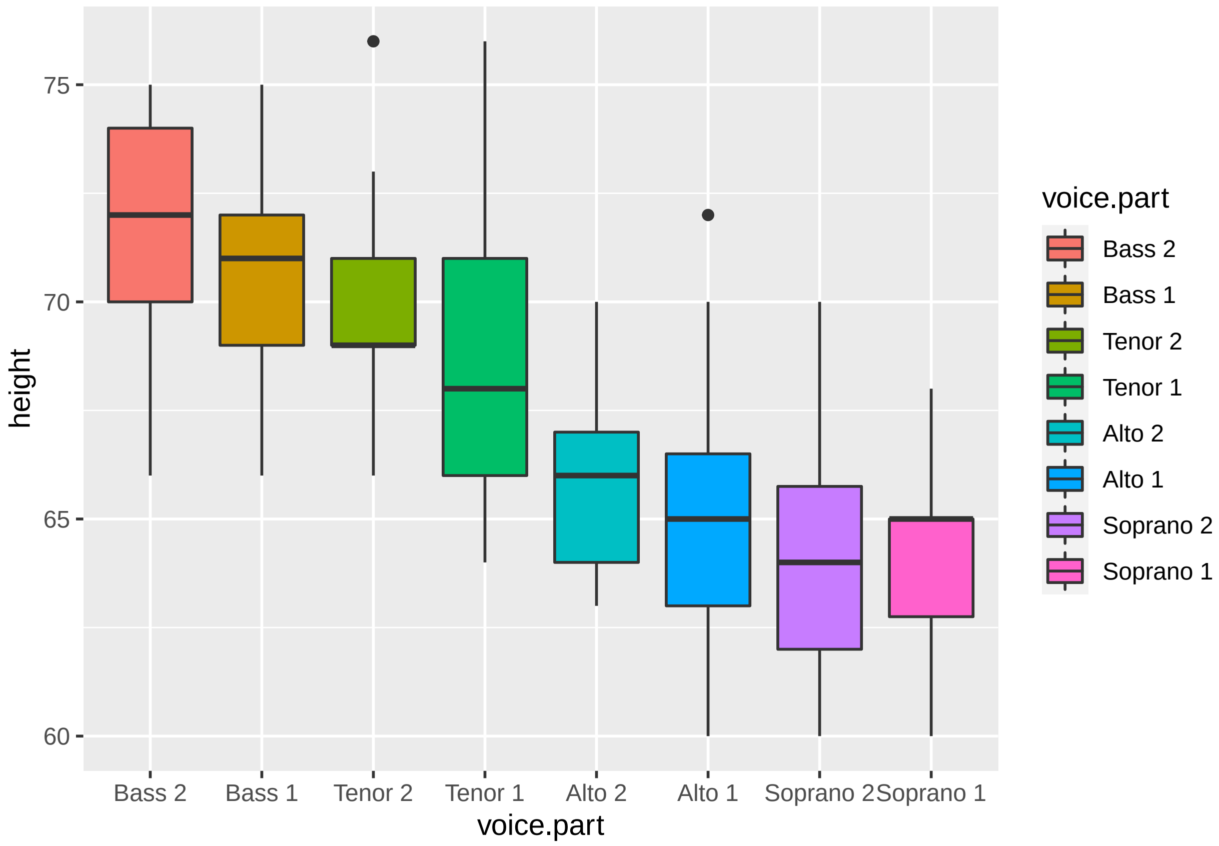


Figure 31: Boxplot for the singers data (2).

```
qplot(voice.part, height, data = singer, geom = c("boxplot", "jitter"))
```

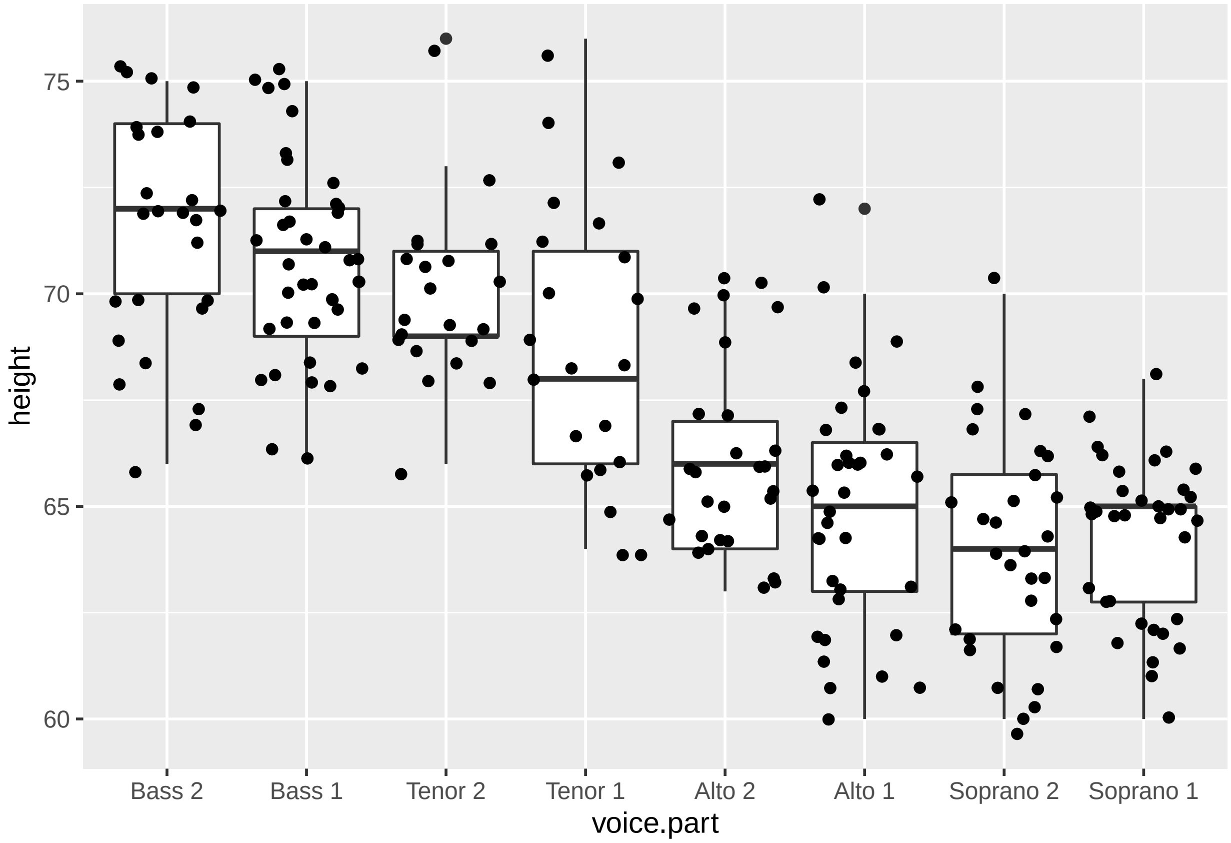


Figure 32: Boxplot for the singers data (3).

When the argument geom violin() is used instead of geom boxplot() the boxplot in Figure @ref(fig:figchp53) become a violin plot shown in Figure @ref(fig:figchp56).

```
ggplot(singer, aes(voice.part,height)) + geom_violin()
```

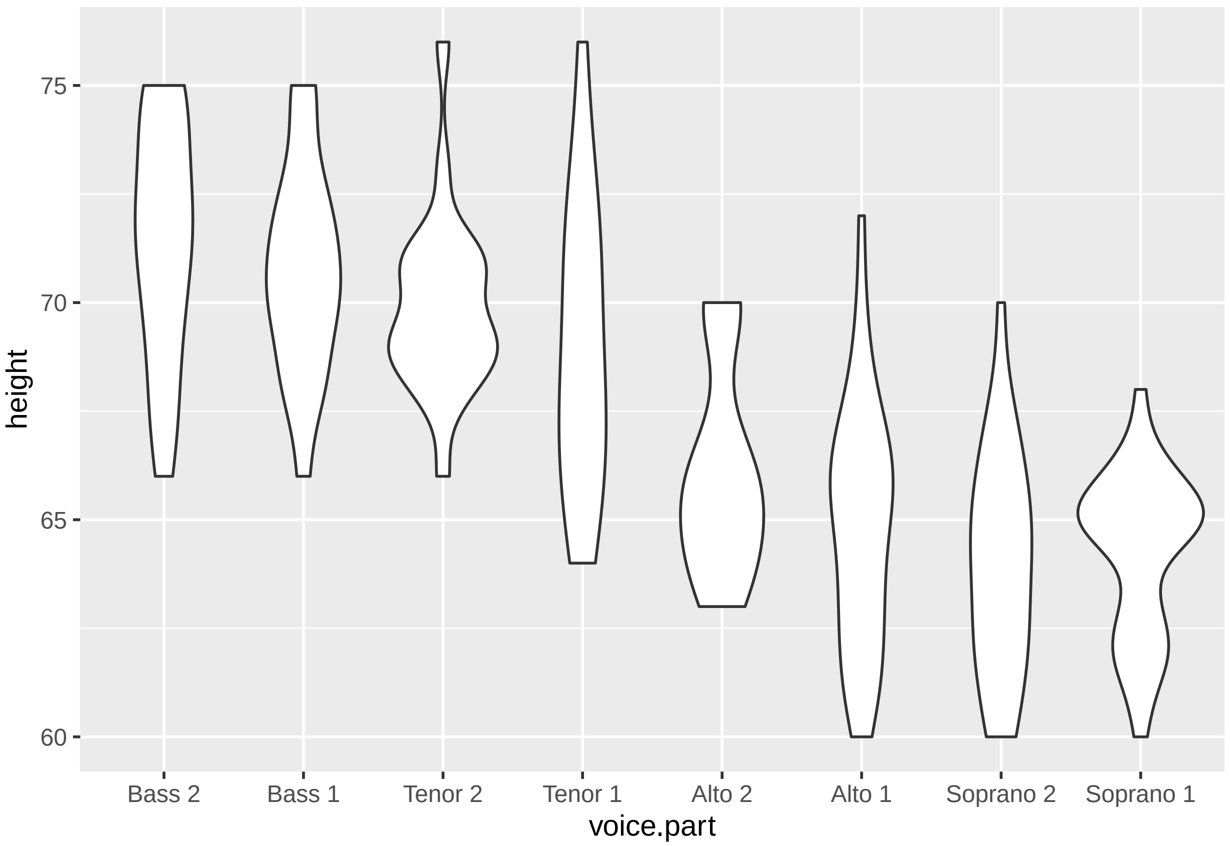


Figure 33: Violin plot for the singers data.

Kareem Abdul Jabbar and Karl Malone's total number of points

Only four NBA players scored more than 33000 points during their career: Kareem Abdul-Jabbar (KAJ, 38387 points in 1560 games), Karl Malone (KM, 36928, 1476), Kobe Bryant KB, (33643,1346) and LeBron James (LJ, 33313,1228). In the section we compare the distribution of the total number of points between Kareem Abdul-Jabbar and Karl Malone. The data are given in the xx1 dataframe.

```
head(xx1)
```

```
##    PTS PLAYER
## 1 1203     KM
## 2 1779     KM
## 3 2268     KM
## 4 2326     KM
## 5 2540     KM
## 6 2382     KM
```

The boxplot and the violin plot, presented in Figure @ref(fig:figchp58) and @ref(fig:figchp59) both indicate that the variability of the total number of points for KAJ and KM is comparable, i.e., the length of the box in Figure @ref(fig:figchp58). The MAD is equal to 326.17 and 363.97 for KM and KAJ, respectively.

```
bwplot(PLAYER~PTS,
                data = xx1,
                layout = c(1, 1),
                aspect = 1,
                xlab = "Yearly number of points")
```

The violin plot, shown in Figure @ref(fig:figchp59) is produced using the function geom violin().

```
ggplot(xx1, aes(PLAYER,PTS,fill=PLAYER)) +
            geom_violin() +
            geom_jitter(color="black", size=0.75, alpha=0.5)
```

The MAD, equal to 326.17 and 363.97 for KM and KAJ, respectively.

```
tapply(xx1$PTS,xx1$PLAYER,mad)
```

```
##       KM      KAJ
## 326.1720 363.9783
```

```
#tapply(xx$PTS,xx$PLAYER,mad)
```

The MAD and the interquartile range, 475 and 491.25 for KM and KAJ, respectively indicate that the variability of KM is slightly lower then the variability of KAJ.

```
Qrange.KM<-diff(quantile(xx1$PTS[xx1$PLAYER=="KM"],probs=c(0.25,0.75)))
Qrange.KAJ<-diff(quantile(xx1$PTS[xx1$PLAYER=="KAJ"],probs=c(0.25,0.75)))
Qrange.KM
```

```
## 75%
## 475
```

```
Qrange.KAJ
```

```
##    75%
## 491.25
```

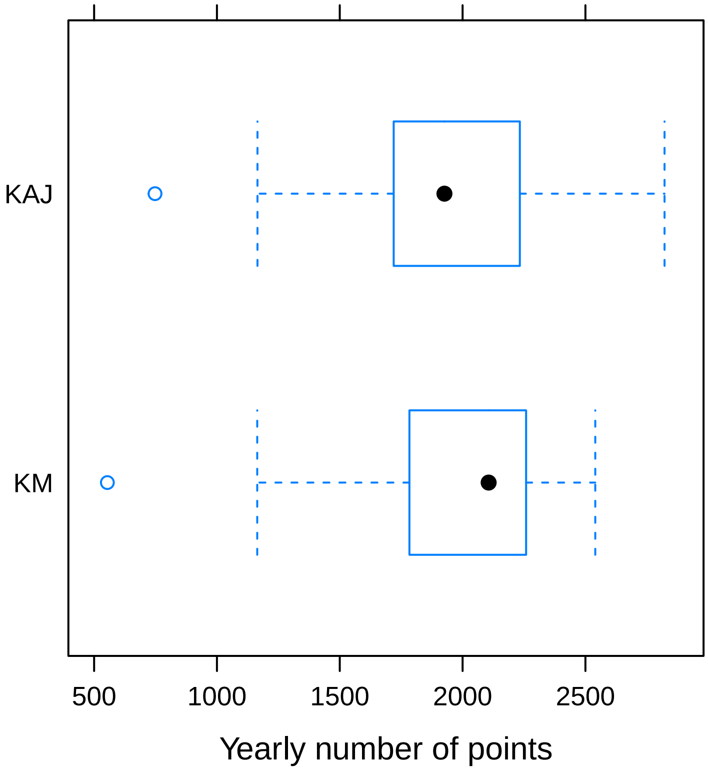


Figure 34: Number of points for Kareem Abdul-Jabbar and Karl Malone (the lattice package).

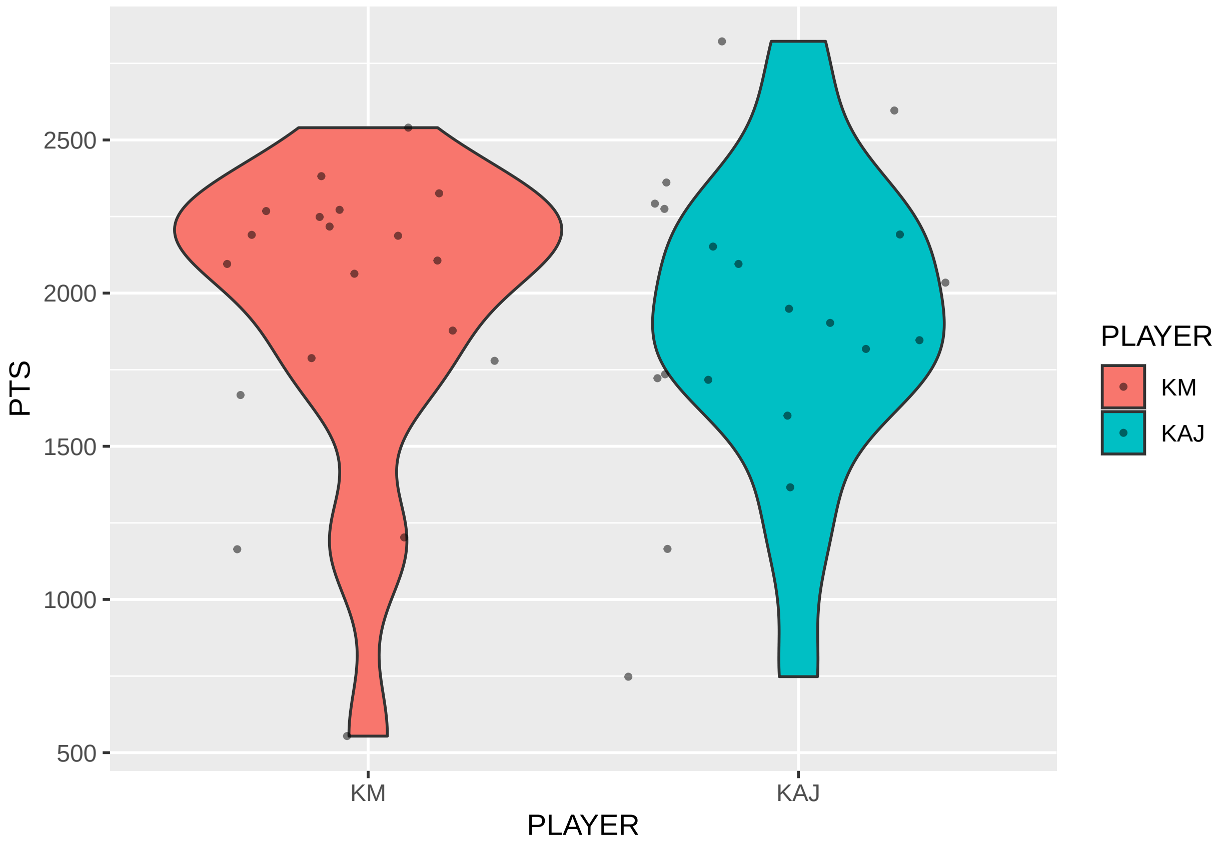


Figure 35: Number of points for Kareem Abdul-Jabbar and Karl Malone (the ggplot2 package).

The number of 3 points attempt made by the Golden State Warriors: 2010-2019

The Golden state worriers data consists of two variable, the season (YEAR) and the number of three points attest per game (A3P).

```
load("C:\\projects\\NBA\\GSW\\GSW1.RData")
head(xxx)
```

```
##   A3P YEAR
## 1   1 2010
## 2   3 2010
## 3   2 2010
## 4   1 2010
## 5   2 2010
## 6   2 2010
```

```
#length(xxx$A3P)
#sum(xxx$A3P)
```

The boxplot for the number of 3PA per game for GSW over time is shown in Figure @ref(fig:figchp510). Note how the variability increases as the median 3PA per game increases over time.

```
qplot(xxx$YEAR, xxx$A3P, data = xxx, geom = c("boxplot", "jitter"))
```

```
## Warning: Use of `xxx$YEAR` is discouraged. Use `YEAR` instead.

## Warning: Use of `xxx$A3P` is discouraged. Use `A3P` instead.

## Warning: Use of `xxx$YEAR` is discouraged. Use `YEAR` instead.

## Warning: Use of `xxx$A3P` is discouraged. Use `A3P` instead.
```

The same pattern using a violin plot in Figure~@ref(fig:figchp512).

```
ggplot(xxx, aes(YEAR,A3P,fill=YEAR)) +
            geom_violin() +
            geom_jitter(color="black", size=0.75, alpha=0.5)
```

We can add a smmothed trend over time, as shown in Figure~@ref(fig:figchp513), using the function geom_smooth.

```
ggplot(xxx, aes(YEAR,A3P,fill=YEAR)) +
      geom_violin()+
      geom_jitter(color="black", size=0.75, alpha=0.5)+
      geom_smooth(aes(group=1))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

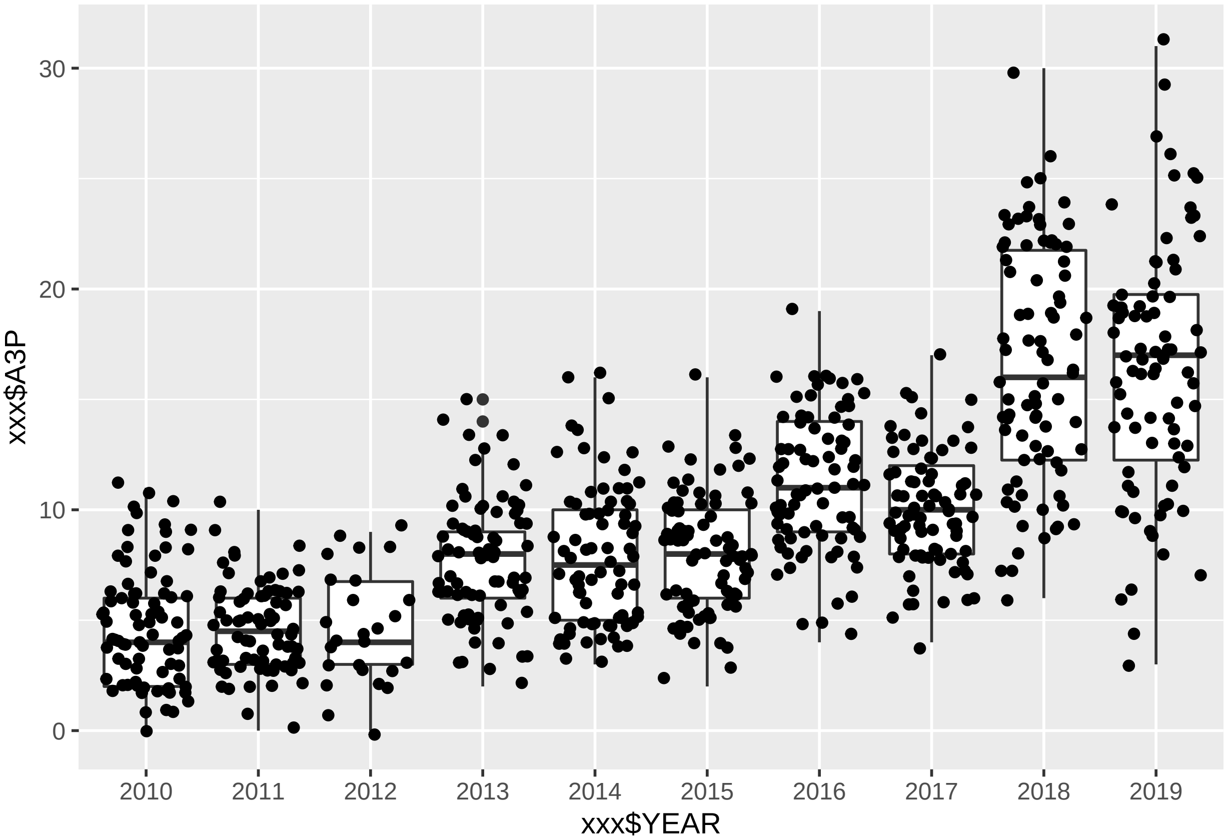


Figure 36: GSW: 3PA per game over time (1).

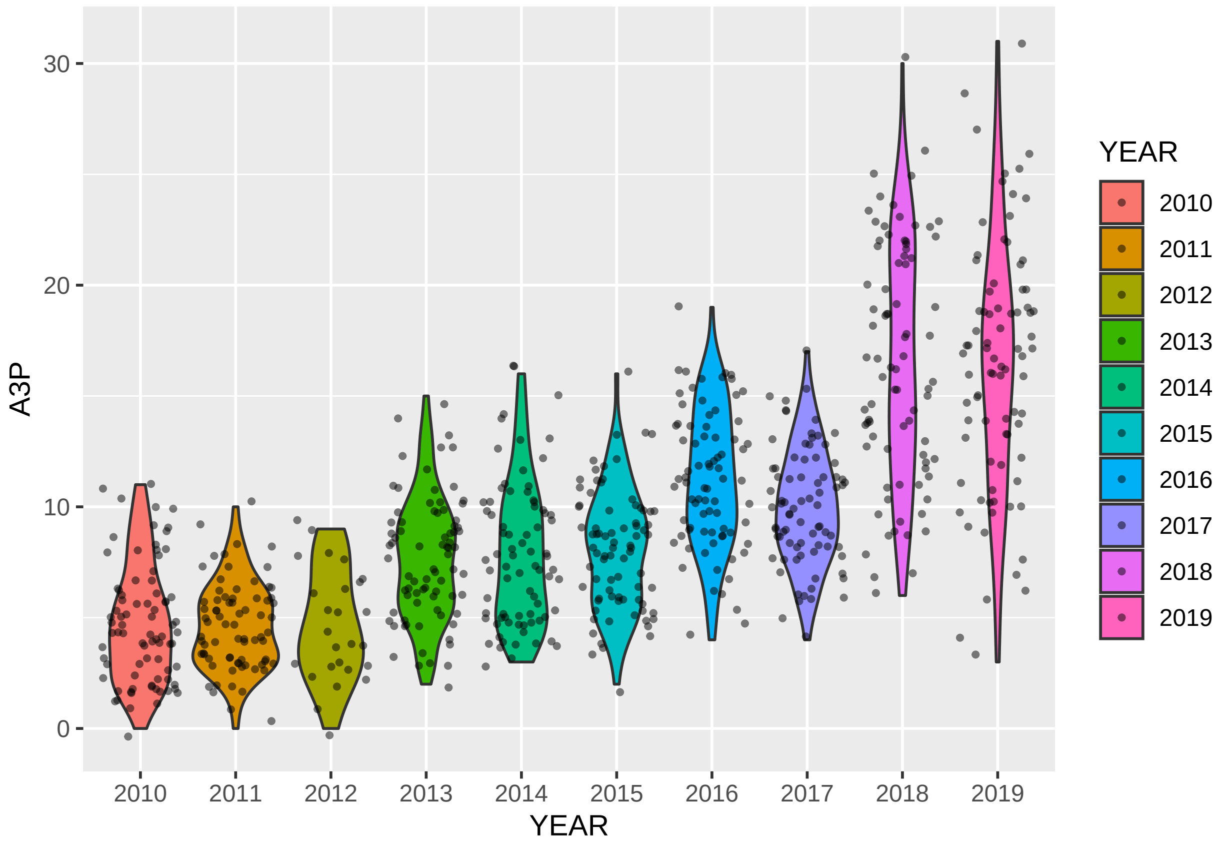


Figure 37: GSW: 3PA per game over time (2).

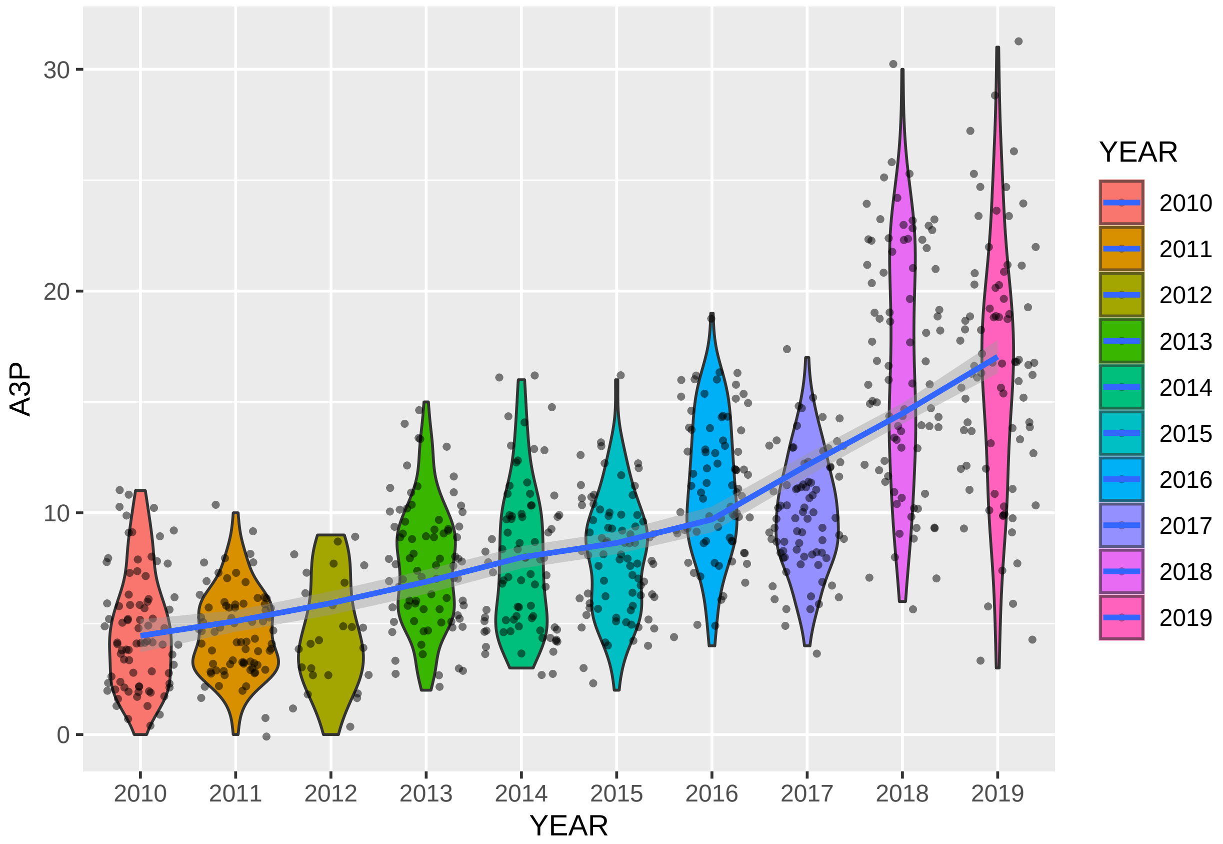


Figure 38: GSW: 3PA per game over time (3).

The quantile plot

Definitions and examples

Quantile

Let $X_{(1)}, X_{(2)} \ldots, X_{(n)}$ be the sorted sample (i.e., the order statistics). The f quantile of the sample is order statistic $X_{(i)}$ that a proportion f of the observations is less than or equal to. For example, the median is the 0.5 quantile since 50% of the observations are less than or equal to the median. The lower fourth is the value that 25% of the observations are less than or equal to. Hence, the lower forth is the 0.25 quantile of the sample.

f-value

The f_i value is the proportion of observations in the sample that are less than or equal to the $i'th$ order statistic, $X_{(i)}$. Hence,

$$f_i = \frac{i}{n}.$$

So $X_{(i)}$ is the f_i quantile of the sample.

Quantile plot

A quantile plot is a figure in which $X_{(i)}$ is plotted versus f_i. Let us focus on the second Bass voice group in the singer dataset for which Figure~@ref(fig:figchp514) presents the quantile plots. Note that the median (72) is the value that cross the Horizontal line of $f = 0.5$.

```
x<-sort(singer$height[singer$voice.part=="Bass 2"])
n<-length(x)
f.value<-c(1:n)/n
cbind(x,f.value)
```

```
##        x    f.value
##  [1,] 66 0.03846154
##  [2,] 67 0.07692308
##  [3,] 67 0.11538462
##  [4,] 68 0.15384615
##  [5,] 68 0.19230769
##  [6,] 69 0.23076923
##  [7,] 70 0.26923077
##  [8,] 70 0.30769231
##  [9,] 70 0.34615385
## [10,] 70 0.38461538
## [11,] 71 0.42307692
## [12,] 72 0.46153846
## [13,] 72 0.50000000
## [14,] 72 0.53846154
## [15,] 72 0.57692308
## [16,] 72 0.61538462
## [17,] 72 0.65384615
## [18,] 72 0.69230769
## [19,] 74 0.73076923
## [20,] 74 0.76923077
## [21,] 74 0.80769231
## [22,] 74 0.84615385
## [23,] 75 0.88461538
## [24,] 75 0.92307692
## [25,] 75 0.96153846
## [26,] 75 1.00000000
```

```
plot(x,f.value,type="s")
abline(0.5,0,col=2)
```

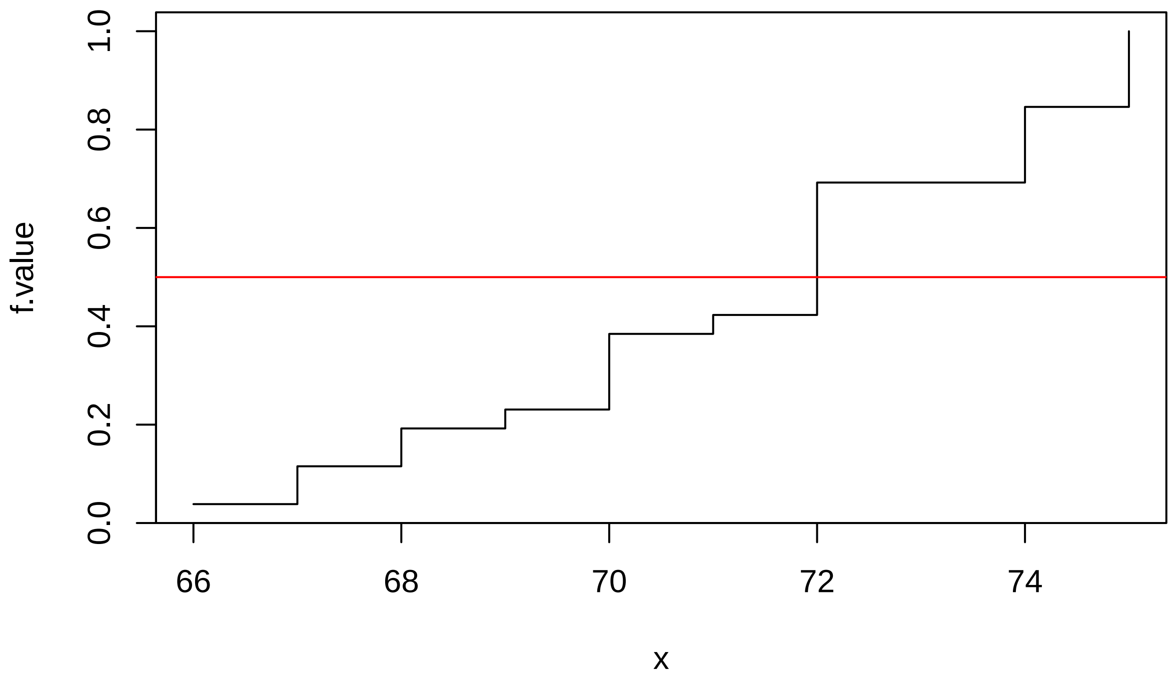


Figure 39: Quantile plot for the singer data for the Bass2 voice group.

```
median(x)
```

```
## [1] 72
```

To produce the quantile plot for the second Bass group, shown in Figure~@ref(fig:figchp515), we can use the function qqmath() of the R package lattice with the argument distribution = qunif in the following way

```
qqmath(~ height[voice.part=="Bass 2"] | voice.part[voice.part=="Bass 2"] , aspect = "xy",
       data = singer,layout=c(1,1),distribution = qunif,
       prepanel = prepanel.qqmathline,
       panel = function(x, ...) {
       panel.qqmathline(x, ...)
        panel.qqmath(x, ...)
       })
```

The argument distribution = qunif implies that quantile for a uniform distribution will be calculated, i.e, $f_i = i/n$. Figure~@ref(fig:figchp516) shows the quantile plot for all voice groups.

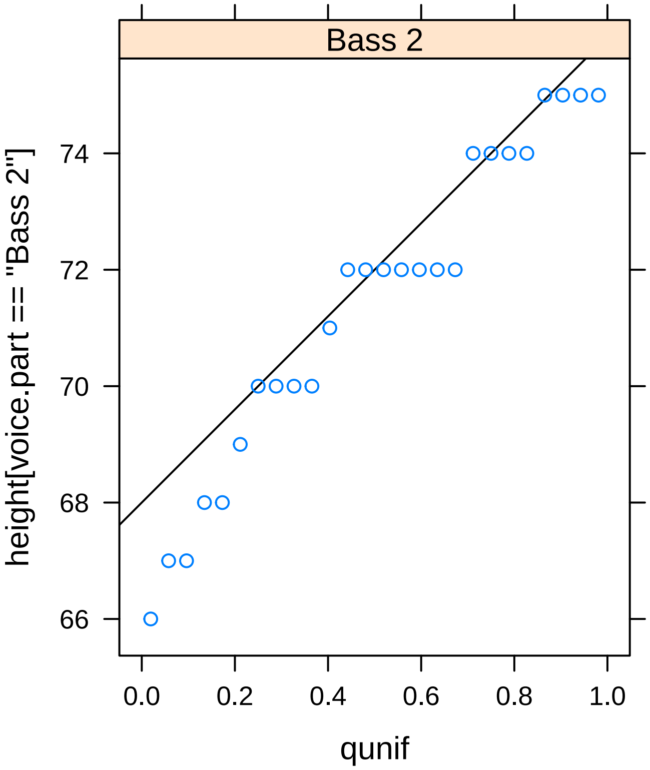


Figure 40: Quantile plot for the second bass group.

```
qqmath(~ height | voice.part, aspect = 0.25, data = singer,layout=c(2,4),
       distribution = qunif,
       prepanel = prepanel.qqmathline,
       panel = function(x, ...) {
       panel.qqmathline(x, ...)
       panel.qqmath(x, ...)
       })
```

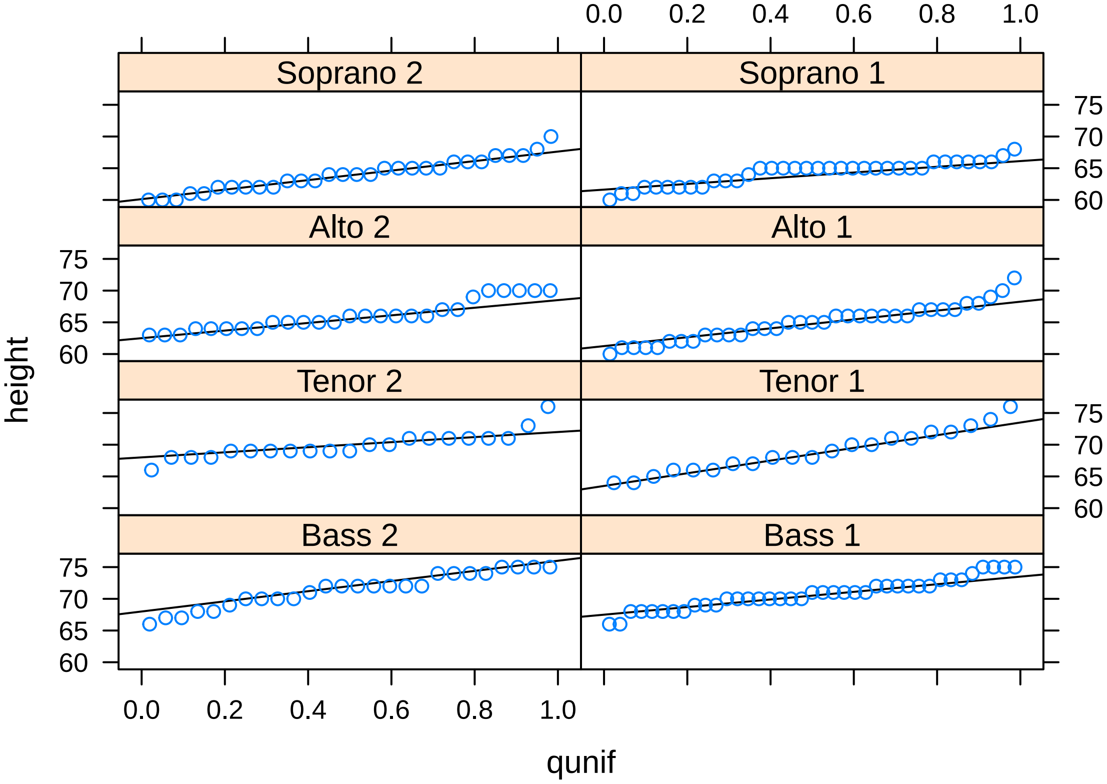


Figure 41: Quantile plot for all voice groups.

Shape: density plots and histograms

Density and density estimate

So far we used histogram to visualize the shape of the distribution of the observations in the sample. In this chapter we discuss density estimates as a method to estimate and visualized the distribution in the population. Figure~@ref(fig:figchp71)a shows a density function of $N(0,1)$ that represents the distribution of a random variable in the population. Suppose that $x_1,\ldots,x_n$ is a random sample of size n from the population. The histogram presented in Figure~@ref(fig:figchp71)b can be used to visualize the shape of the distribution. It is an estimate for the density in the population. A second approach to estimate the distribution of the population is to use a smooth version of the histogram, i.e., a density estimate. The density estimate for our example is shown in Figure~@ref(fig:figchp71)c and Figure~@ref(fig:figchp71)d.

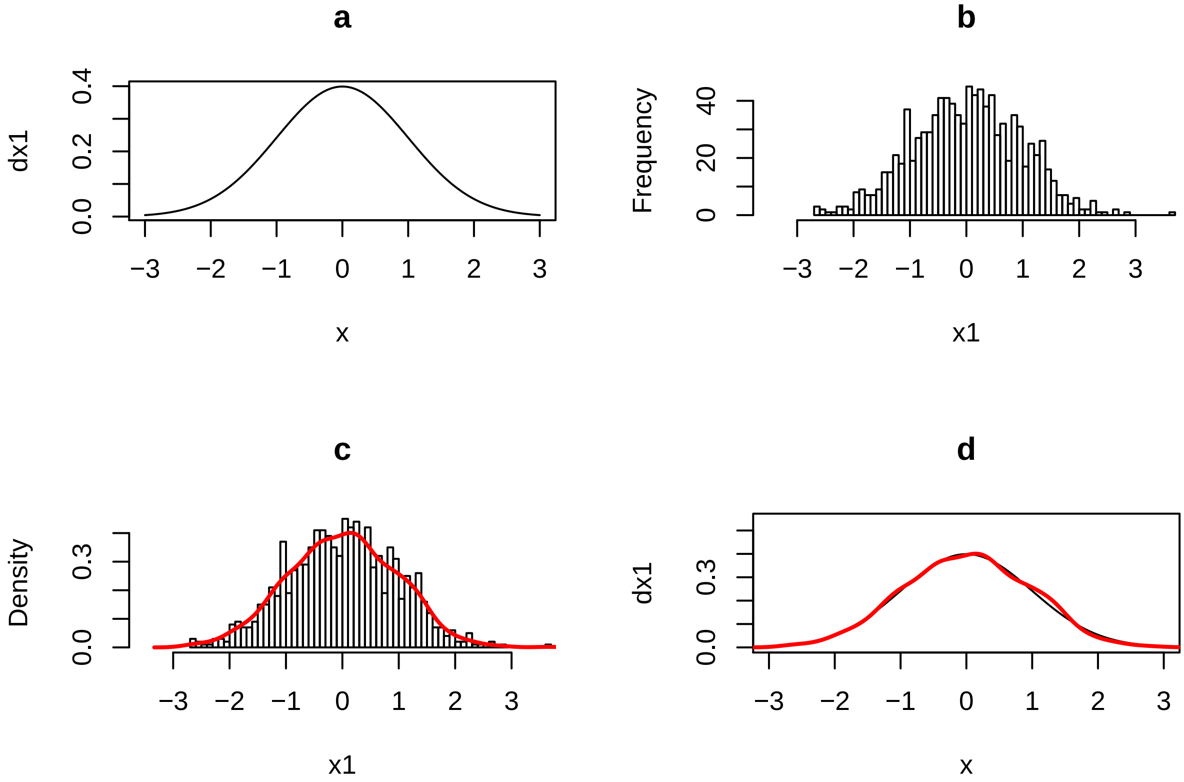


Figure 42: Density and density estimates. Panel a: density of N(0,1). Panel b: histogram for a random sample of 1000 observations from N(0,1). Panel c: density estimae. Panel d: density and density estimate.

Online tutorials

YouTube tutorial: Creating density plots and enhancing it with the ggplot2 package

A short online YouTube tutorial by LawrenceStats, about density plot using the ggplot2 package see YTVD10.

Web tutorial: the ridgeline chart

A Web tutorial about the ridgeline chart using the ggplot2 and ggridges package is given in the the R Graph Gallery website WAVD4.

The old faithful data

As we saw in Chapter~1, the distribution of the eruptions time for the old faithful data is a bi-model with modes at 1.8 and 4.4.

```
stem(faithful$eruptions)
```

```
##
##   The decimal point is 1 digit(s) to the left of the |
##
##   16 | 070355555588
##   18 | 000022233333335577777777888822335777888
##   20 | 00002223378800035778
##   22 | 0002335578023578
##   24 | 00228
##   26 | 23
##   28 | 080
##   30 | 7
##   32 | 2337
##   34 | 250077
##   36 | 0000823577
##   38 | 2333335582225577
##   40 | 0000003357788888002233555577778
##   42 | 03335555778800233333555577778
##   44 | 02222335557780000000023333357778888
##   46 | 0000233357700000023578
##   48 | 00000022335800333
##   50 | 0370
```

We use this data to illustrate the failure of the boxplot to capture this feature of the data as can be seen in Figure~@ref(fig:figchp72).

```
qplot(rep(1,length(eruptions)),eruptions, data=faithful, geom = c("boxplot"))
```

Note that, as shown in Figure~@ref(fig:figchp72a) if we add the data to the boxplot, using the option geom = c("boxplot", "jitter") there is no problem to identify the two parts of the eruptions time distribution.

```
qplot(rep(1,length(eruptions)),eruptions, data=faithful, geom = c("boxplot", "jitter"))
```

As shown in Figure~@ref(fig:figchp73), the histogram is able to capture the shape of the distribution.

```
qplot(eruptions, data=faithful, geom="histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

A density estimate, presented in Figure~@ref(fig:figchp74), was produced using the opion geom="density") provides a smooth estimate of the distribuon.

```
qplot(eruptions, data=faithful, geom="density")
```

Note that, as shown in Figure~@ref(fig:figchp75), we can plot both histogram and density in the same plot as well.

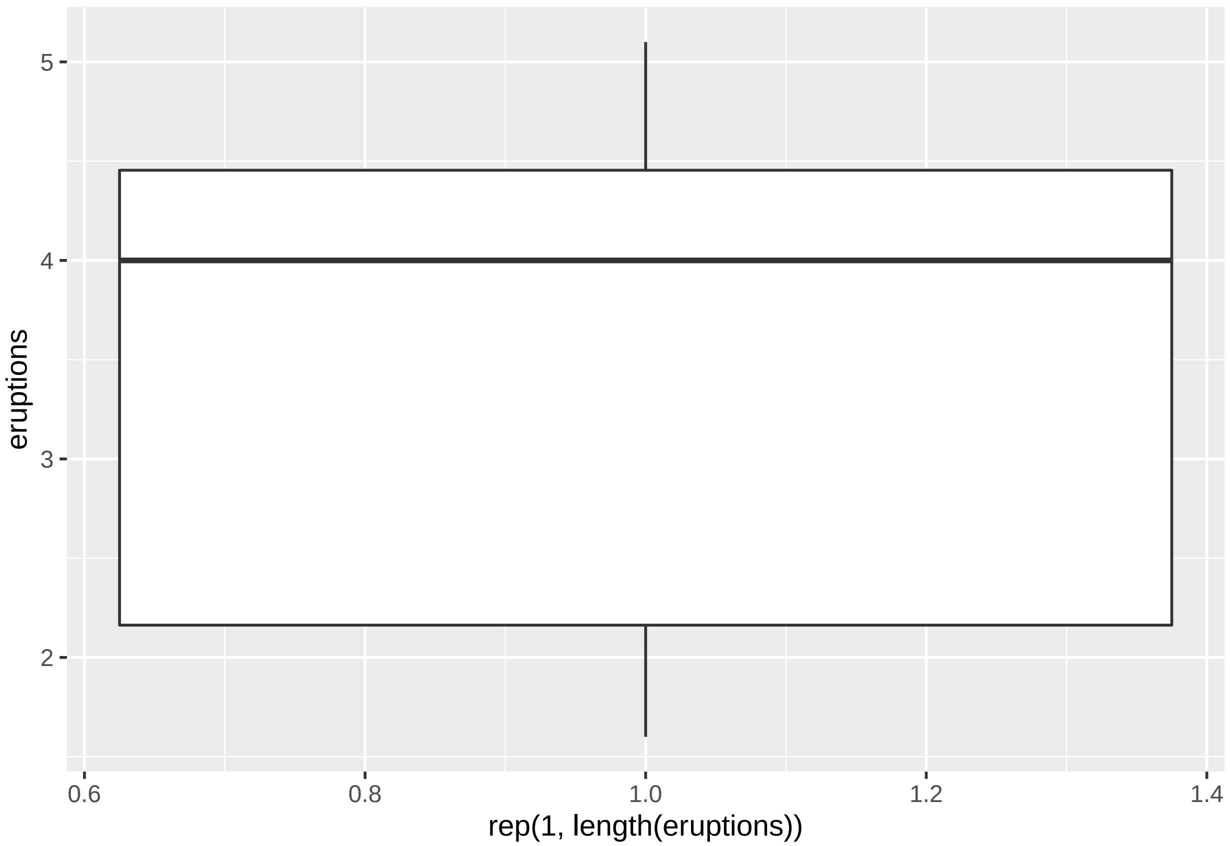


Figure 43: Boxplot for the old faithful data.

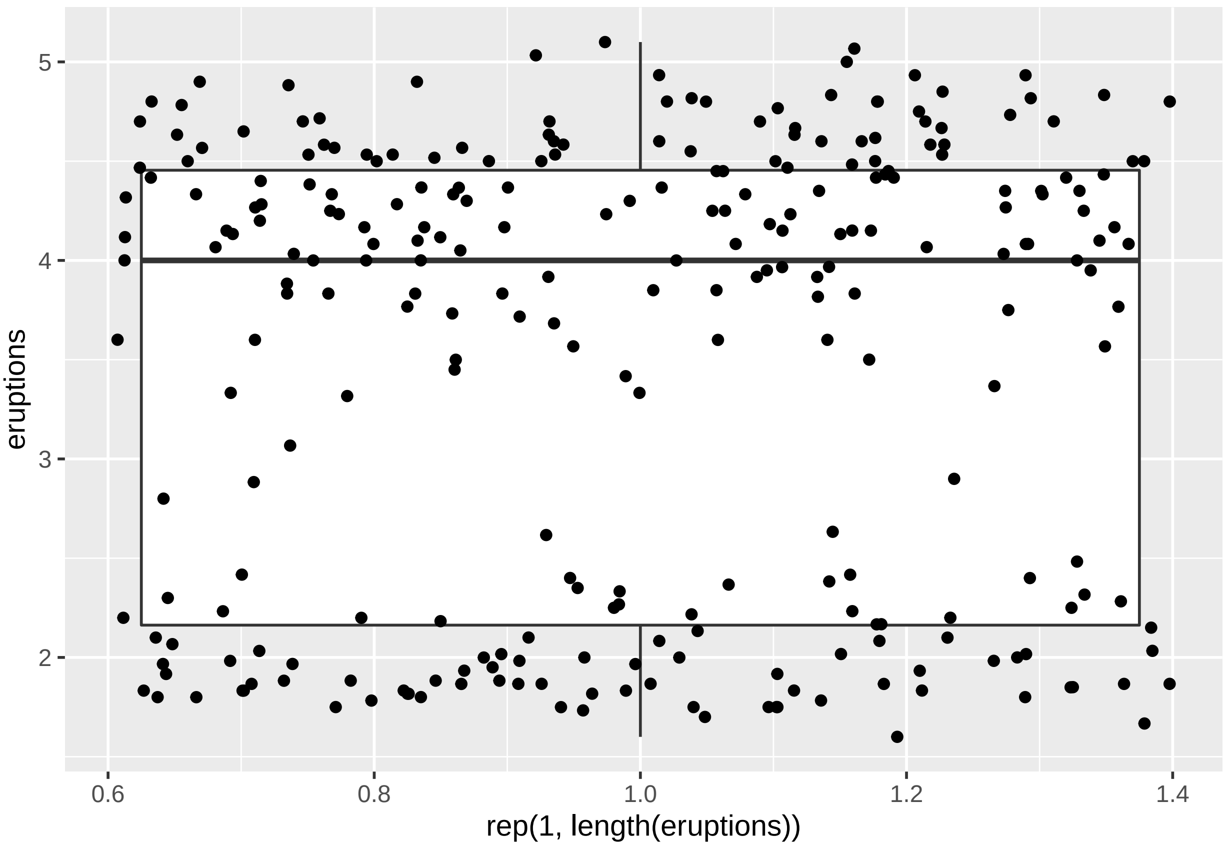


Figure 44: Boxplot for the old faithful data with data points.

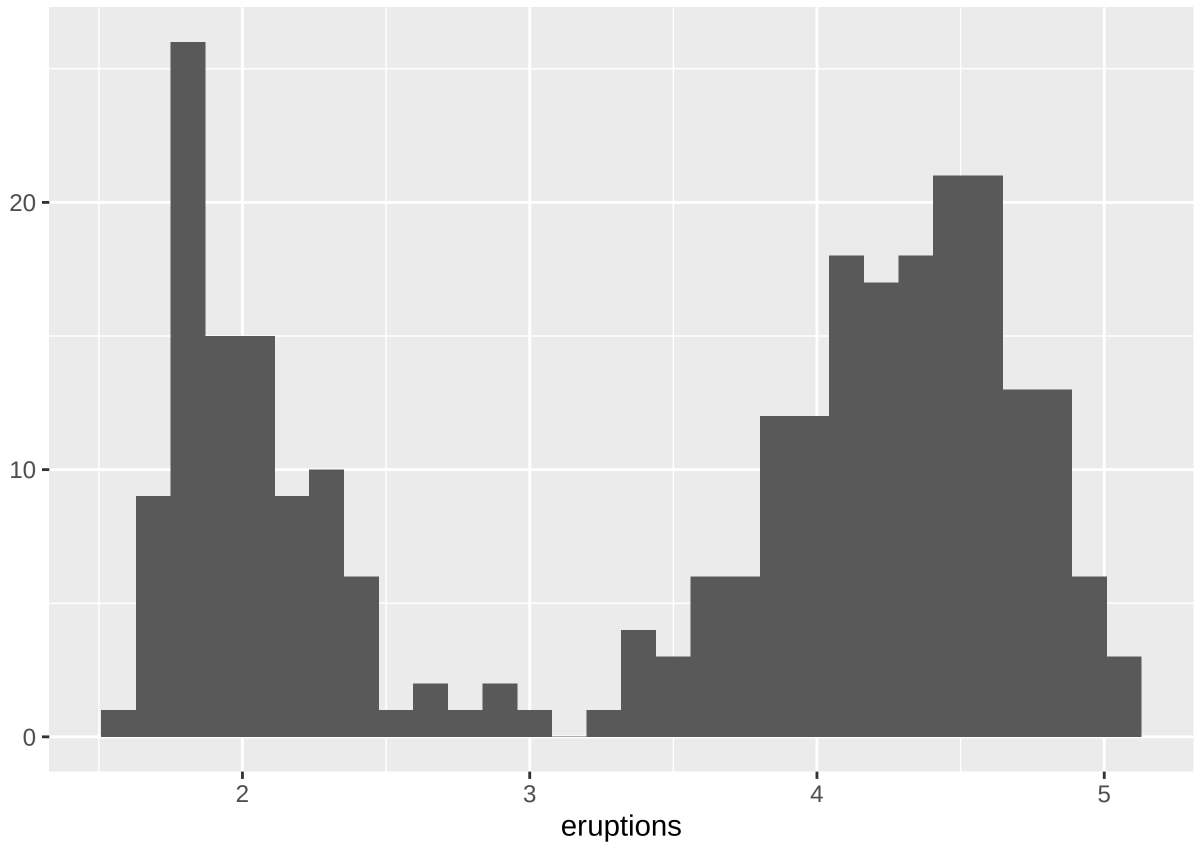


Figure 45: Histogram for the old faithful data.

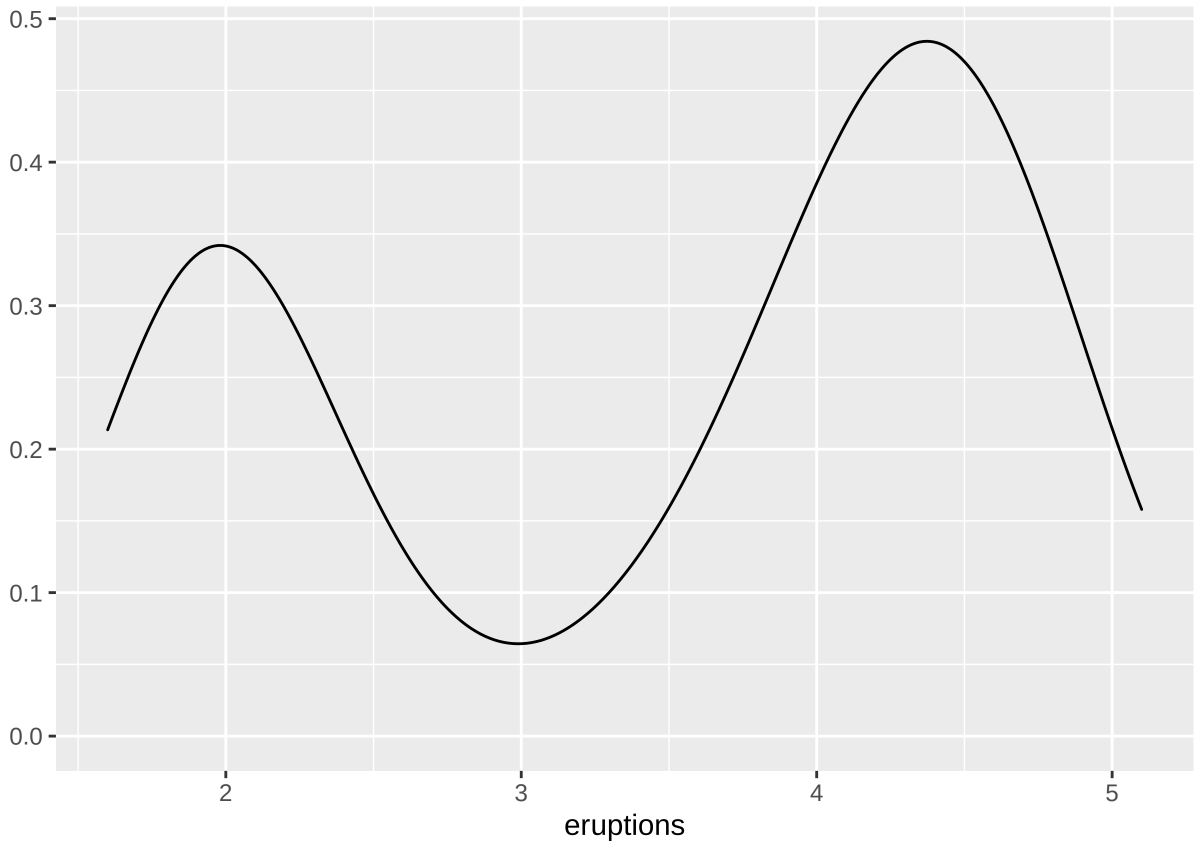


Figure 46: Density plot for the old faithful data (the ggplot2 package).

```
hist(faithful$eruptions,nclass=15,probability = TRUE)
dx<-density(faithful$eruptions)
lines(dx$x,dx$y,lwd=2,col=2)
```

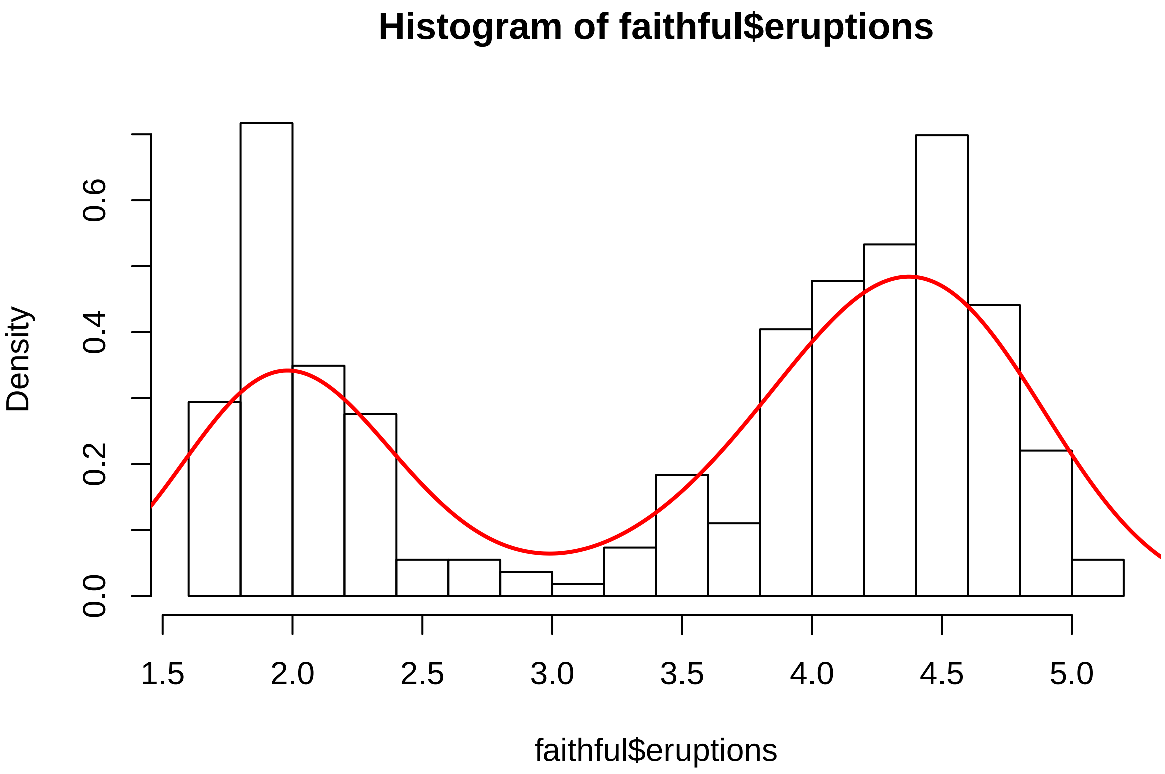


Figure 47: Histogram and a density plot for the old faithful data.

The singer data

In this section we use density plots to visualize the shit of the distribution of the singers' height across the voice part groups that was discussed in previous chapters. The histograms in Figure~@ref(fig:figchp76) ravel the two shifts (1) within each voice group, the heights of the first group are shifted to the right compare to the second group (for example, Alto 1 compared with Alto 2 etc.) and (2) the sift across the voice groups.

```
ggplot(singer, aes(height,fill = voice.part)) +
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

In Both density plots, presented in Figure~@ref(fig:figchp77) and Figure~@ref(fig:figchp78), the difference between the sopranos and altos (women singers, the densities in the left) and the tenors and basses (men singers, densities in the right) is clearly seen but the difference within each group is more difficult to detect.

```
qplot(height, data=singer, geom="density", xlim = c(50,80),
fill = voice.part, alpha = I(0.2))
```

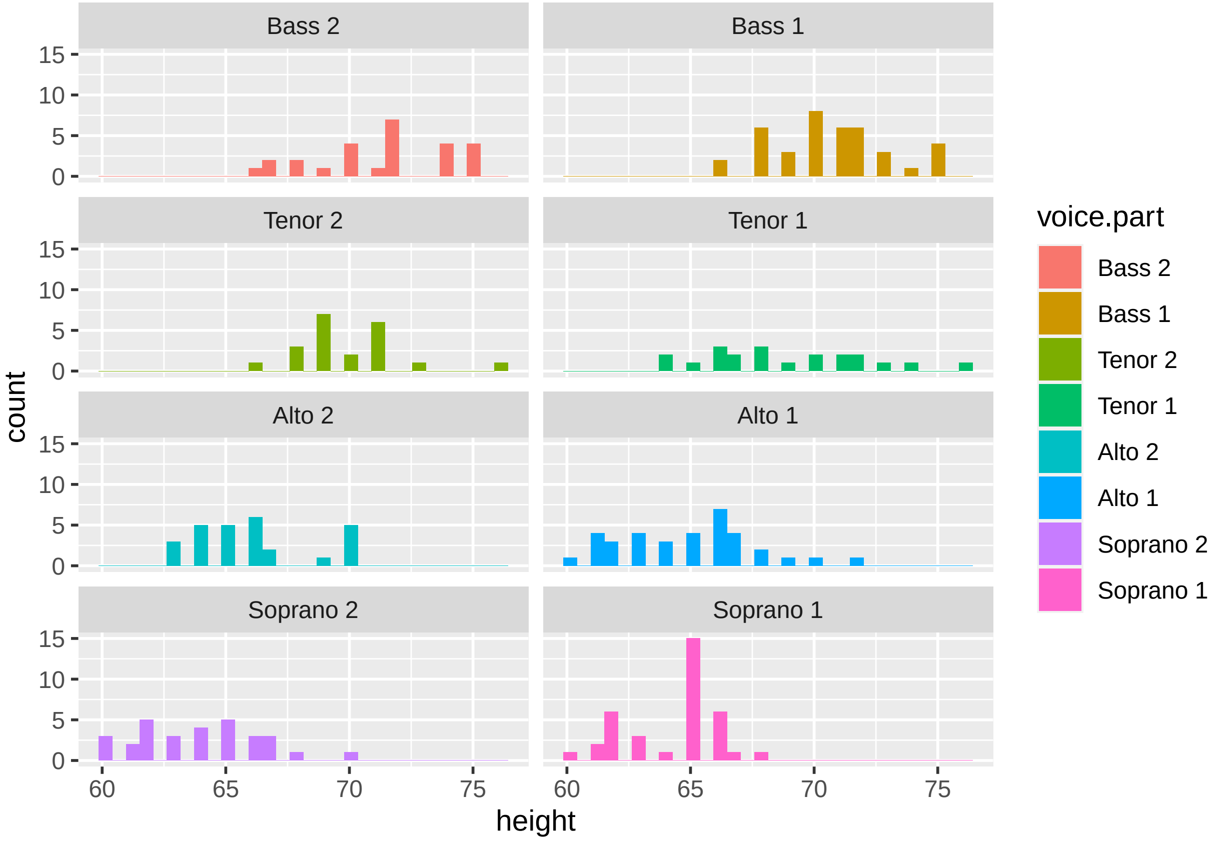


Figure 48: Histogram of the singers' height by voice group.

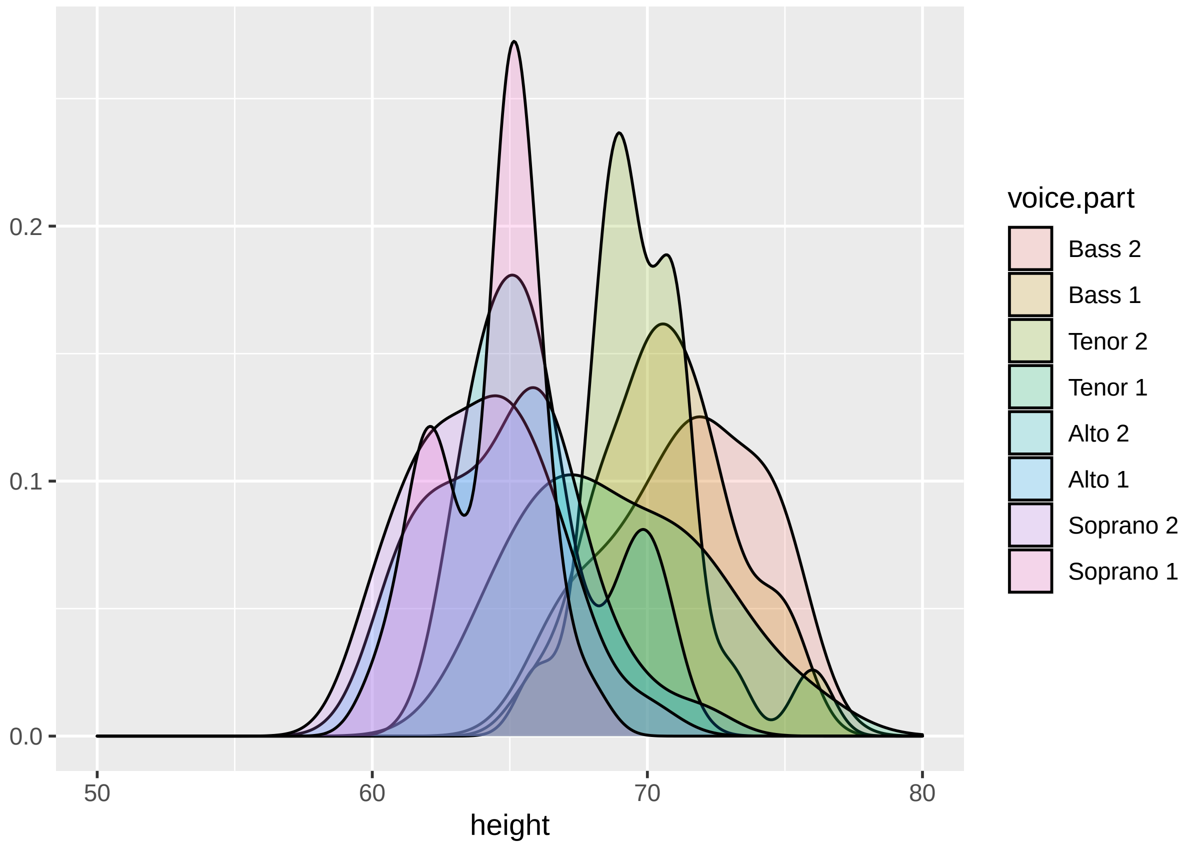


Figure 49: Density plot for the singers dataset (I).

```r
qplot(height, data = singer, geom = "density", colour = voice.part)
```

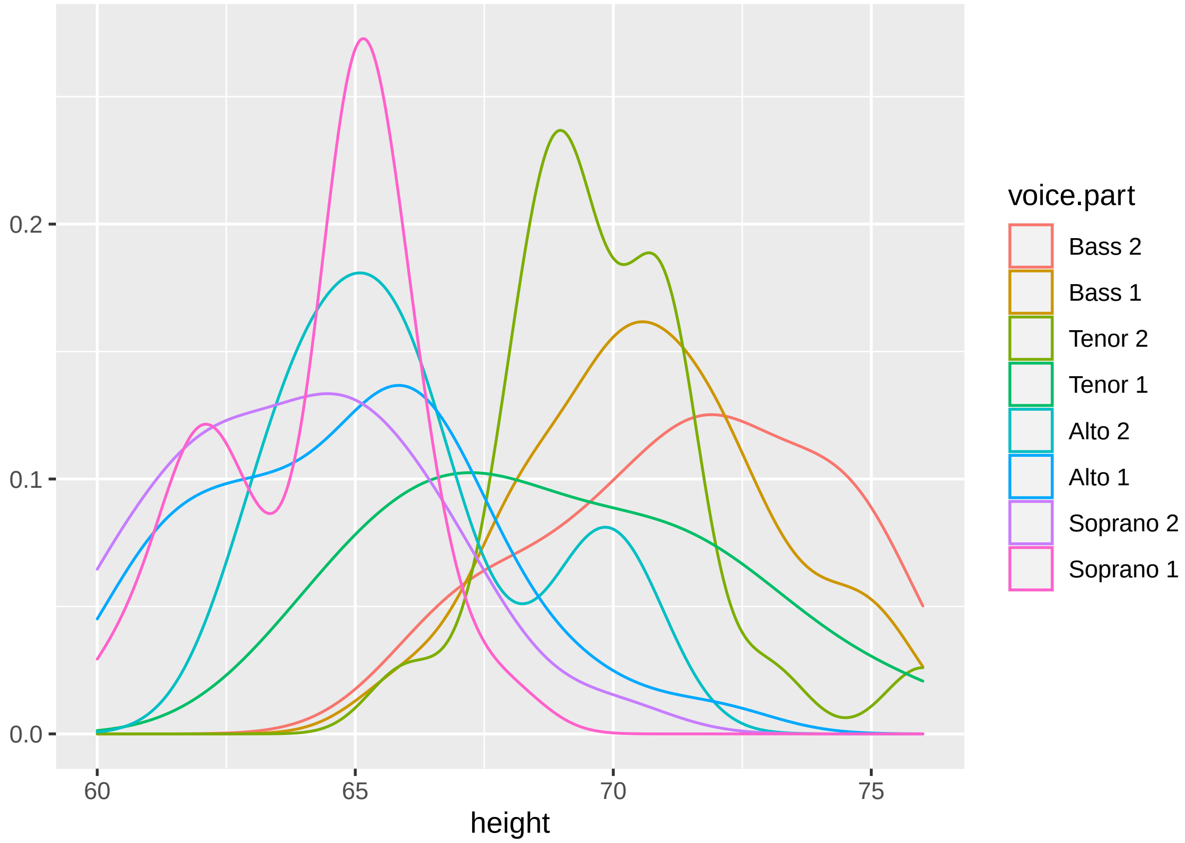


Figure 50: Density plot for the singers dataset (II).

In contrast, the ridgeline charts presented in Figure~@ref(fig:figchp79) visualizes the difference between the groups and within each voice group. Note that the R package ggridges should be instaled to produce such a plot.

```r
library(ggridges)
```

```
## Warning: package 'ggridges' was built under R version 3.6.3
```

```r
ggplot(singer, aes(x=height,y=voice.part,fill = voice.part)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```

```
## Picking joint bandwidth of 1.09
```

Kobe Bryant's performance indicators over the championship years

Kobe Bryant won 5 NBA championships, all with the Lakers, in 99/00, 00/01, 01/02, 08/09 and 09/10. In this section we focus on Bryant's field goal attempts (FGA) per game during the regular season and the playoff in the first and the last championship seasons (1999/2000 and 2008/2009). The densities plot, for FGA, for the regular seasons and playoff in 99/00 and 08/09 are shown in Figure~@ref(fig:figchp710) and Figure~@ref(fig:figchp711) and reveal two different patterns.

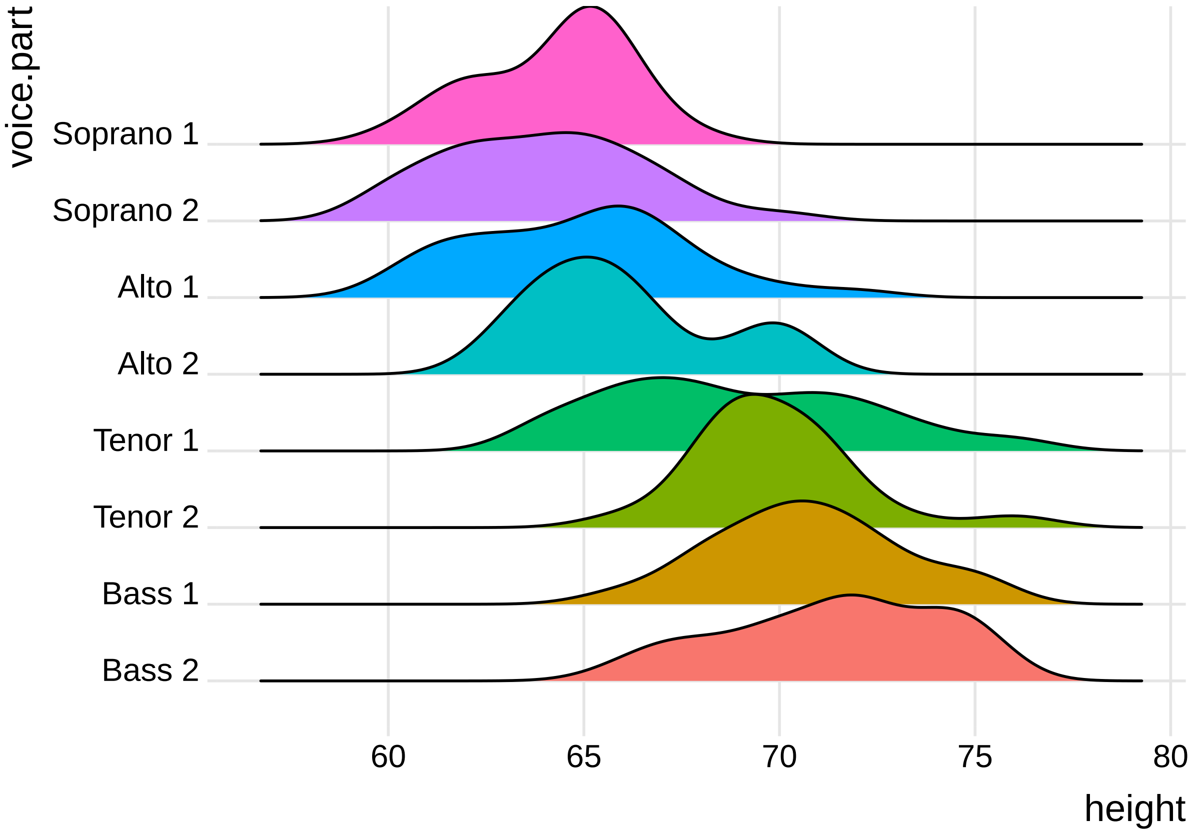


Figure 51: Ridgeline charts for the singers dataset.

```r
qplot(VAR, data=xxx1, geom="density",
fill = TYPE, alpha = I(0.2))+
xlab("FGA,2009")
```

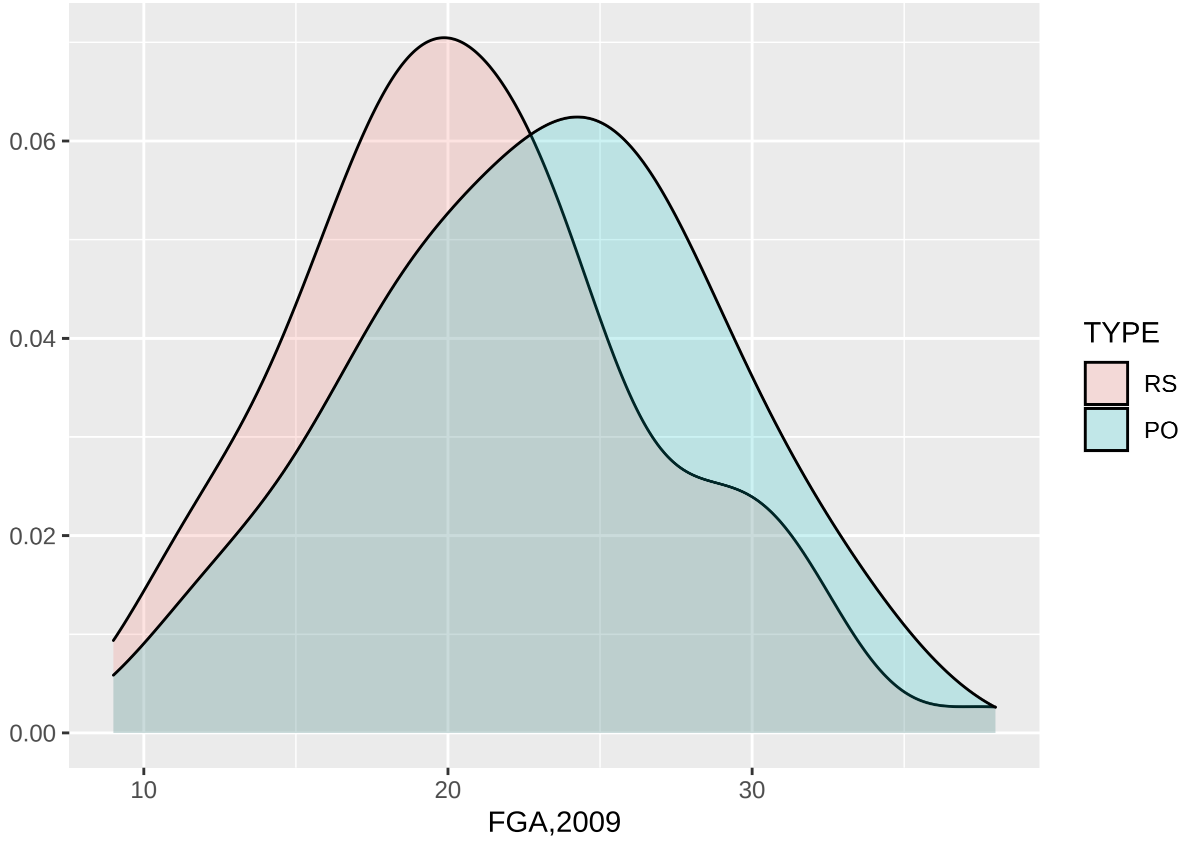


Figure 52: Density plot for FGA for regular season (RS) and playoff (PO) in 2009.

In 08/09, there is a shift in the distribution of the FGA; the distribution in playoff games is located to the right compared to the regular season, indicating a higher FGA per game in the playoff compared to the regular season. In 99/00, the location of the distribution in the regular season and playoff games seems to be similar while the variability is higher in playoff games.

```r
qplot(VAR, data=xxx2, geom="density",
fill = TYPE, alpha = I(0.2))+
xlab("FGA, 2000")
```

This pattern is more clear when ridgeline charts, shown in Figure~@ref(fig:figchp712) and Figure~@ref(fig:figchp713) are used to visualize the distributions in the seasons of 08/09 and 99/00

```r
library(ggridges)
ggplot(xxx1, aes(x=VAR,y=TYPE,fill = TYPE)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```

```
## Picking joint bandwidth of 2.32
```

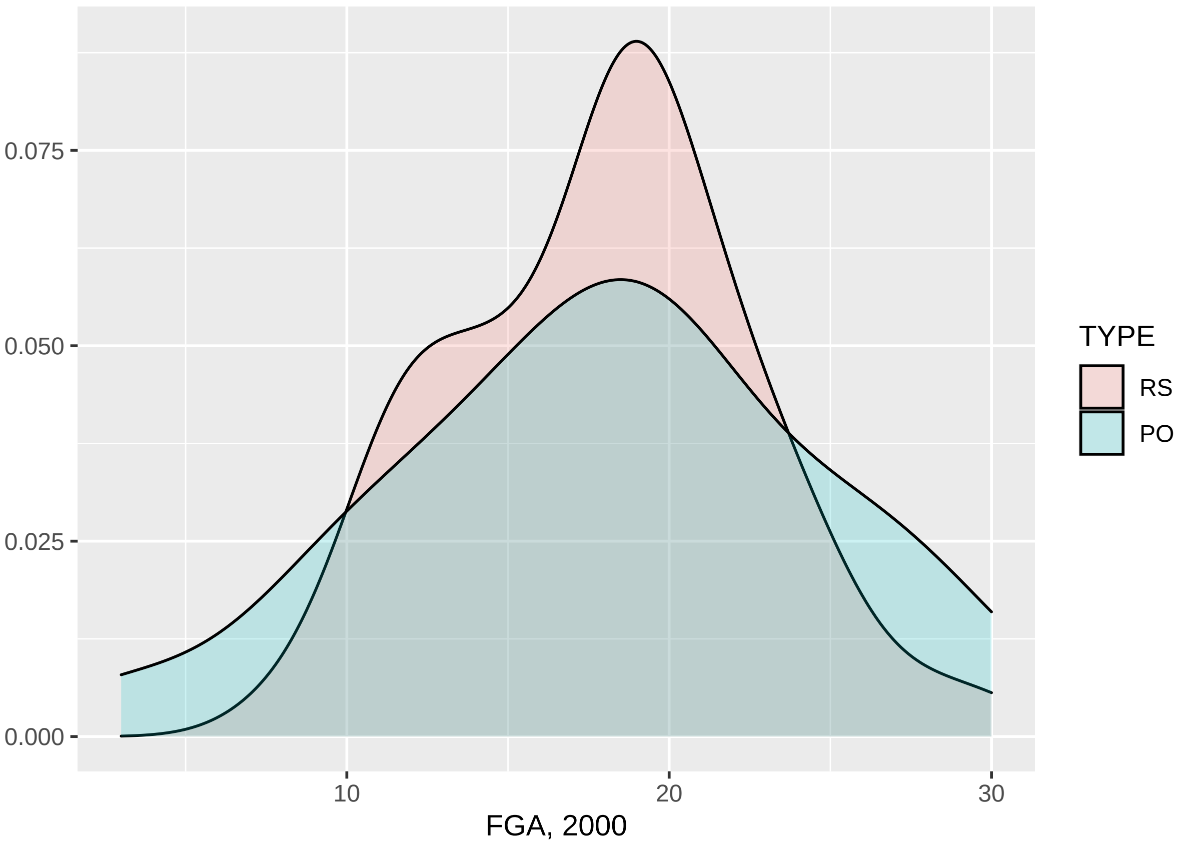


Figure 53: Density plot for FGA for regular season (RS) and playoff (PO) in 2000.

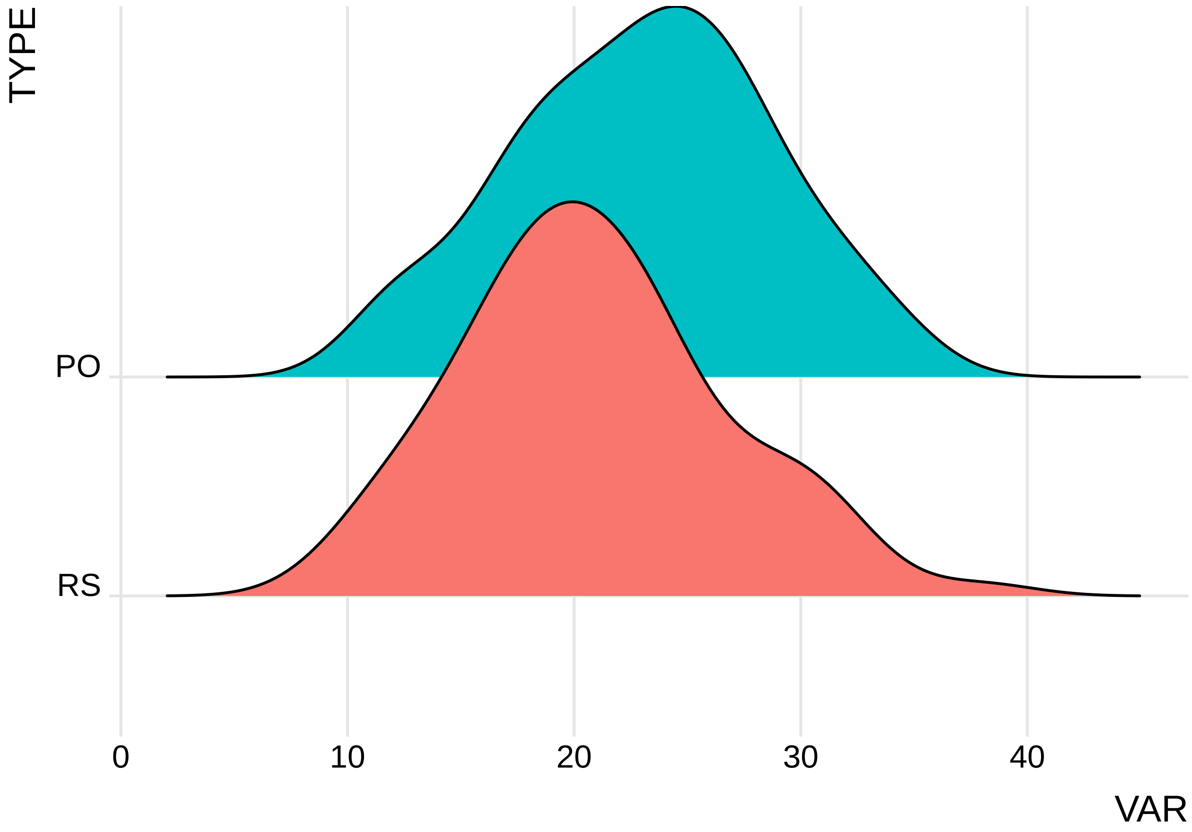


Figure 54: FGA: 2009

```
ggplot(xxx2, aes(x=VAR,y=TYPE,fill = TYPE)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```

```
## Picking joint bandwidth of 2.32
```

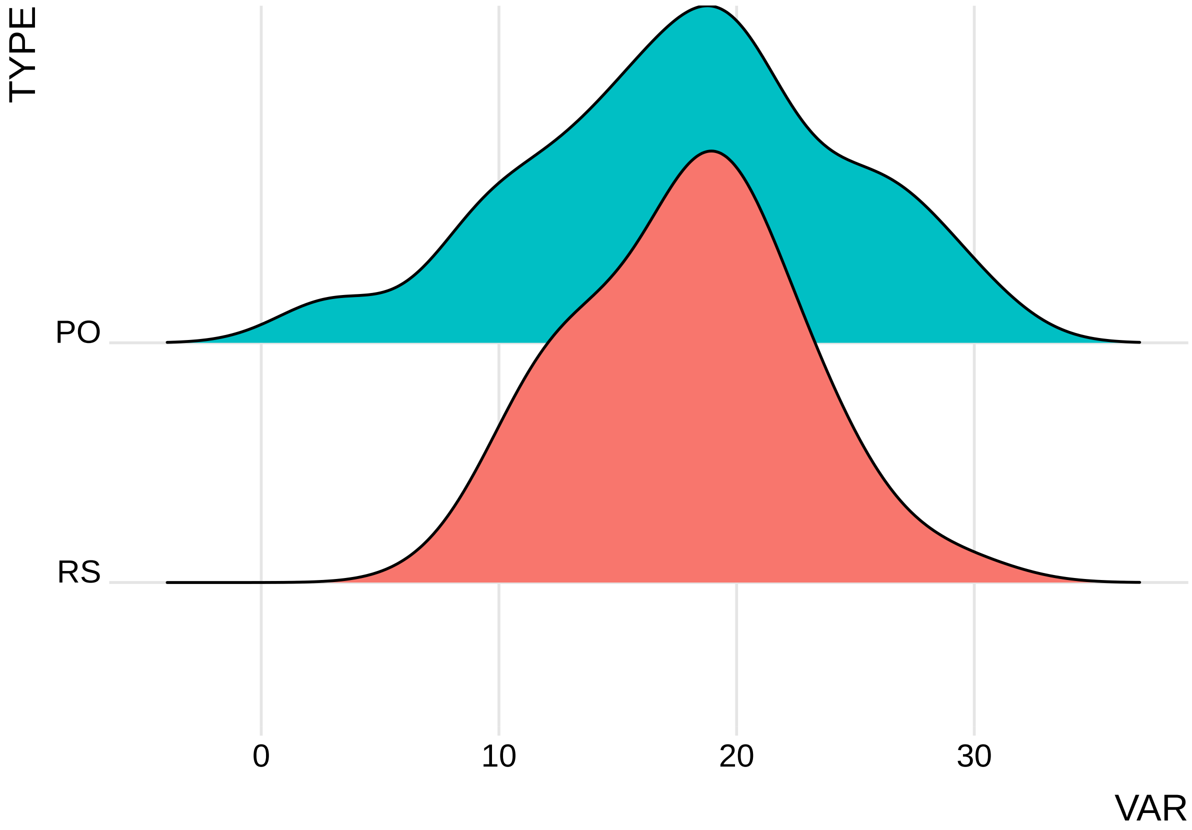


Figure 55: FGA: 2000

This pattern can be seen clearly in Figure~@ref(fig:figchp714) and Figure~@ref(fig:figchp715) that present the side-by-side boxplots. Note how the box is shifted in 08/09 while in 99/00 the length of the box is higher for the playoff games compared to the regular season games.

```
qplot(TYPE, VAR, data = xxx1, geom = c("boxplot", "jitter"))+
xlab("2009")
```

```
qplot(TYPE, VAR, data = xxx2, geom = c("boxplot", "jitter"))+
xlab("2000")
```

Quantiles plot for the distributions in the regular season and playoff games for 2000 and 2009 are shown in Figure~@ref(fig:figchp716) and Figure~@ref(fig:figchp717).

```
qqmath(~ VAR | TYPE,
        distribution=qunif,
        data=xxx1,
           layout=c(1,2),
             prepanel = prepanel.qqmathline,
```

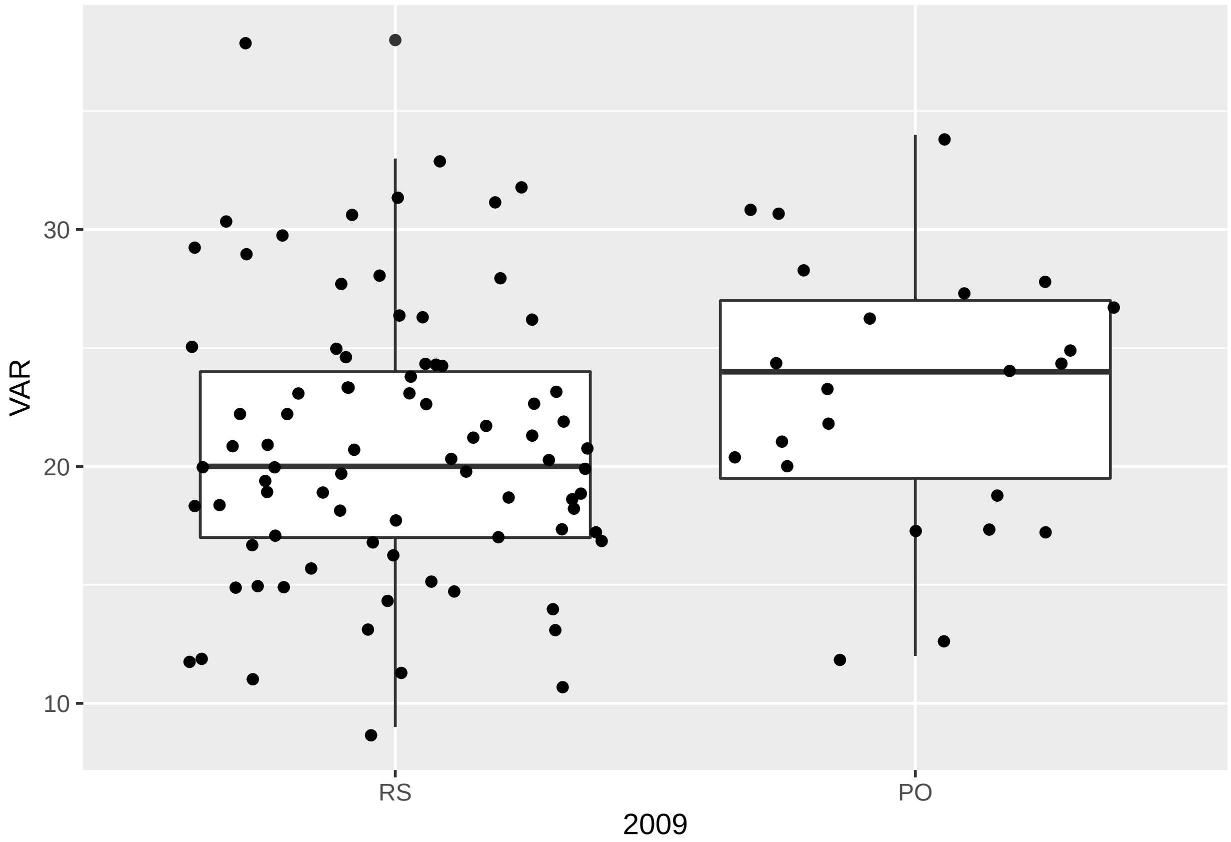


Figure 56: Boxplot for FGA for regular season and playoff in 2009.

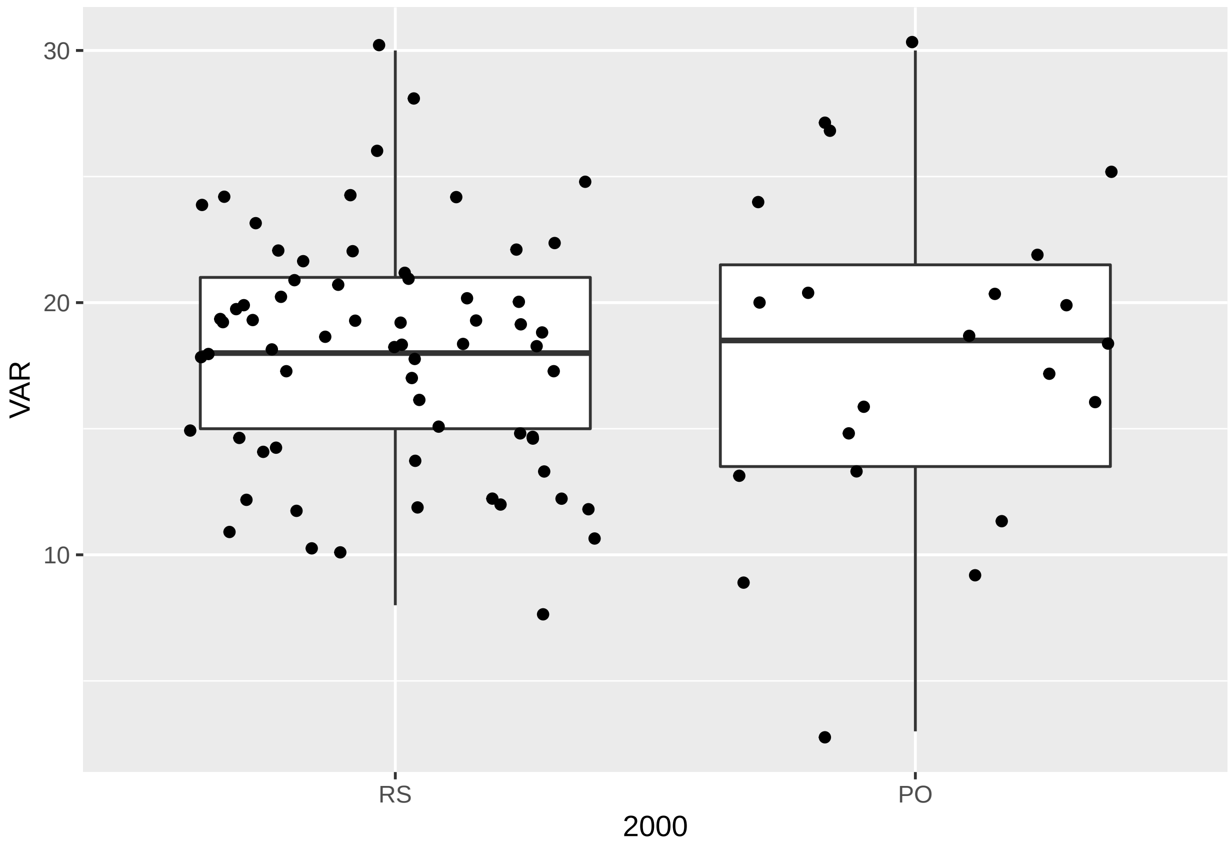


Figure 57: Boxplot for FGA for regular season and playoff in 2000.

```
            panel = function(x, ...) {
            panel.grid()
            panel.qqmathline(x, ...)
            panel.qqmath(x, ...)
            },
        aspect=0.5,
        xlab = "f-value, 2009",
        ylab="FGA")
```

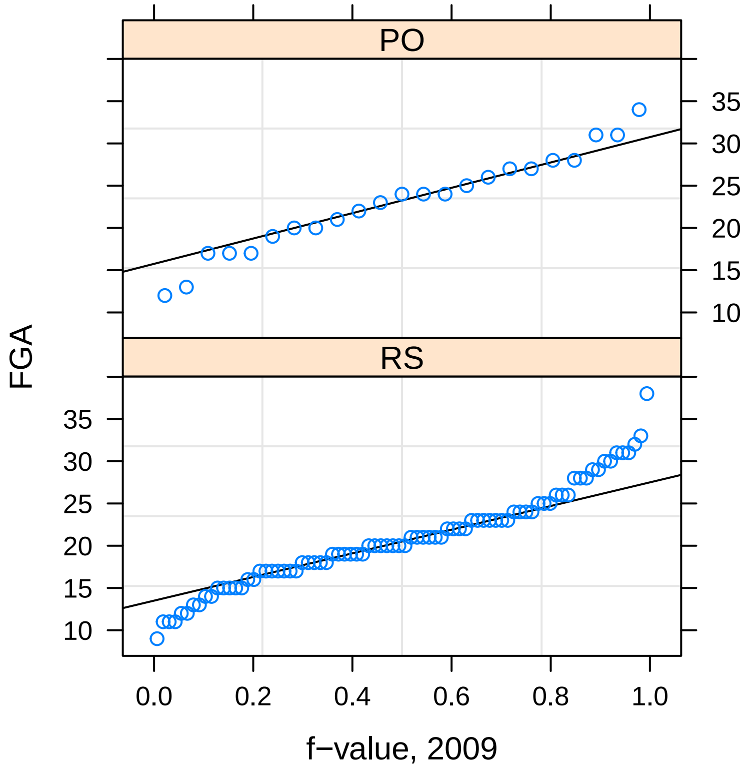


Figure 58: quentile plot for FGA for regular season and playoff in 2009.

```
qqmath(~ VAR | TYPE,
        distribution=qunif,
        data=xxx2,
           layout=c(1,2),
             prepanel = prepanel.qqmathline,
            panel = function(x, ...) {
            panel.grid()
            panel.qqmathline(x, ...)
            panel.qqmath(x, ...)
            },
        aspect=0.5,
        xlab = "f-value, 2000",
        ylab="FGA")
```

qqplot for FGA in the regular season and playoff games for 2000 and 2009 are shown in Fig-

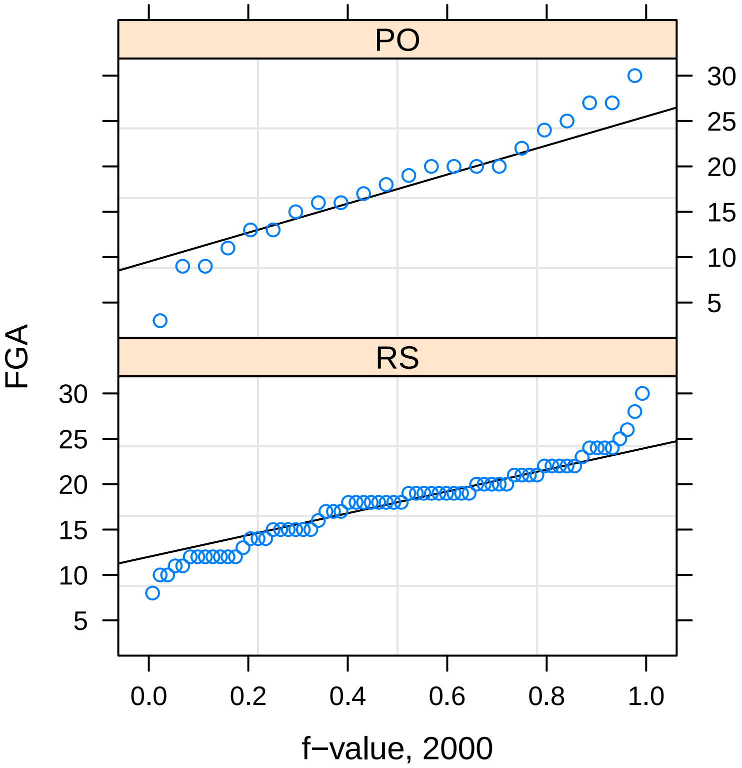


Figure 59: quentile plot for FGA for regular season and playoff in 2000.

ure~@ref(fig:figchp718) and Figure~@ref(fig:figchp719).

```
qq(xxx1$TYPE ~ xxx1$VAR,data=xxx1,
        aspect=1,
        sub = list("2009",cex=.8),
        xlab = "FGA: regular season",
        ylab = "FGA: playoff")
```

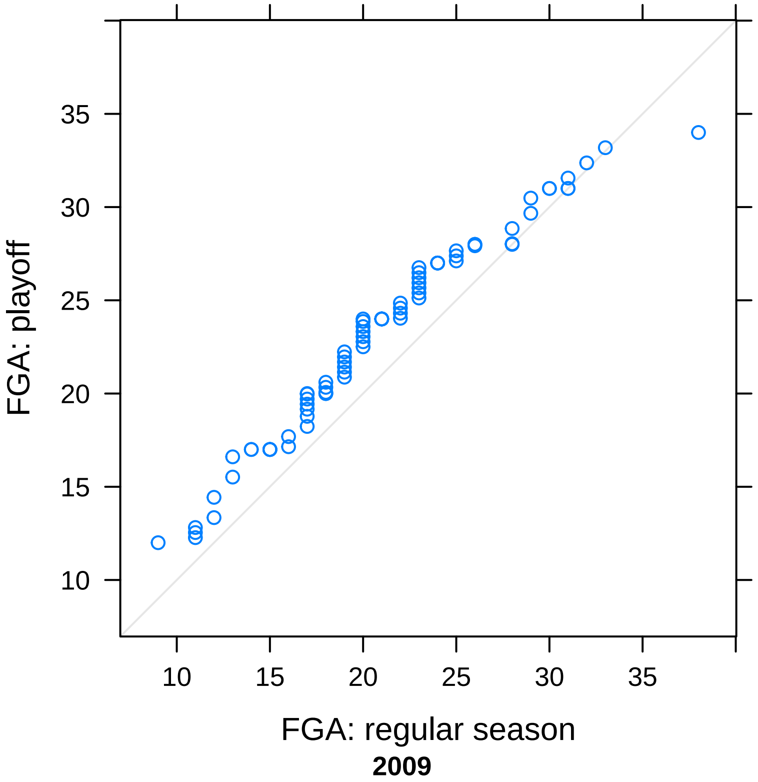


Figure 60: qqplot for FGA for regular season and playoff in 2009.

```
qq(xxx2$TYPE ~ xxx2$VAR,data=xxx2,
        aspect=1,
        sub = list("2000",cex=.8),
        xlab = "FGA: regsular season",
        ylab = "FGA: playoff")
```

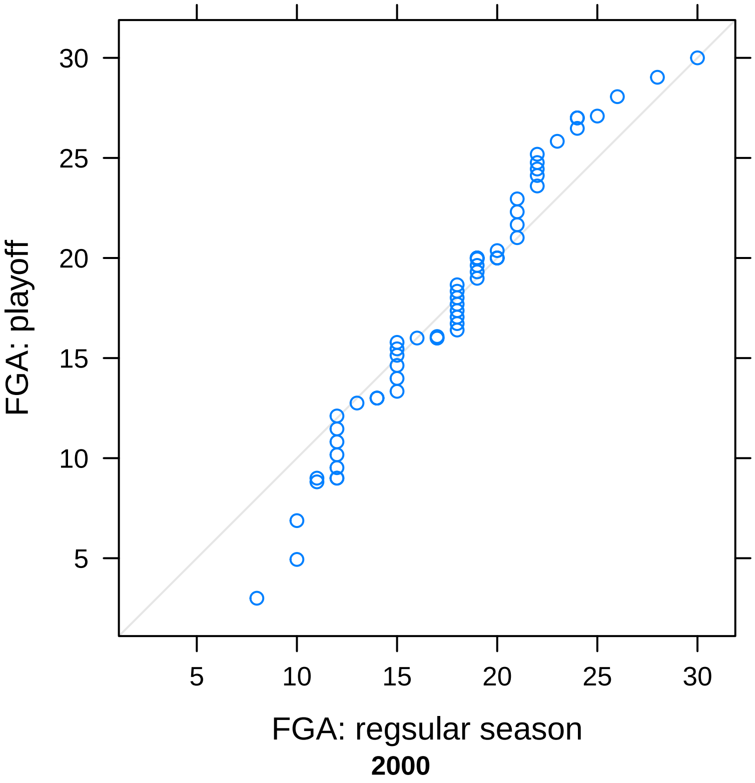


Figure 61: qqplot for FGA for regular season and playoff in 2000.

Normal probability plot

Introduction

A normal probability plot is a plot in which the quantile of the samples are plotted versus the corresponding quantiles of a standard normal distribution $N(0, 1)$. In this chapter we discuss the normal probability plot as a graphical tools to vizualise the shape of a distribution. In addtion, histograms and boxplots, that were discuused in the privious chapters, will be used as well. Using histograms and boxplots we were able to investigate the shape of the distribution focusing on the follwing issues:

- How nearly symmetric the distribution of the data is.
- Whether the distribution of the data is single-peaked, or whether it is multi-peaked.
- Whether it is skewed.
- Whether a few values are far away from the rest.
- Whether there are concentrations of data.
- Whether there are gaps in the data.

Quantile of $N(\mu, \sigma^2)$,

Definition and a simple example

A qq normal plot is a graphical disply to investigate how now nearly is the sample to a normal distribution. Let $q_{\mu,\sigma}(f)$ be a quantile of $N(\mu, \sigma^2)$, it can be expressed as

$$q_{\mu,\sigma}(f) = \mu + \sigma q_{0,1}(f).$$

For example, the \$2.5 % \$ quantile of the standard normal distribution is -1.96.

```
qnorm(0.025,0,1)
```

```
## [1] -1.959964
```

For $N(2, 5^2)$ we have

$$q_{2,5}(2.5\%) = 2 + 5 \times -1.96 = -7.8$$

```
qnorm(0.025,2,5)
```

```
## [1] -7.79982
```

YouTube tutorial: QQ-plots in RStudio

For a short online YouTube tutorials by UTSSC, about normal probability plot using R studio see YTVD12.

Examples

In this section we focus on few examples and use normal probability plots to access the shape of the distributions.

Sample fron N(0,1)

We use the R function qqnorm() to produce the normal probabilty plot shown in Figure~@ref(fig:figchp81). For this example, since data were sampled from $N(0,1)$, we expect that all points in the normal probability plot will lay on the on the 45^o line.

```
x <- rnorm(1000, 0, 1)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```

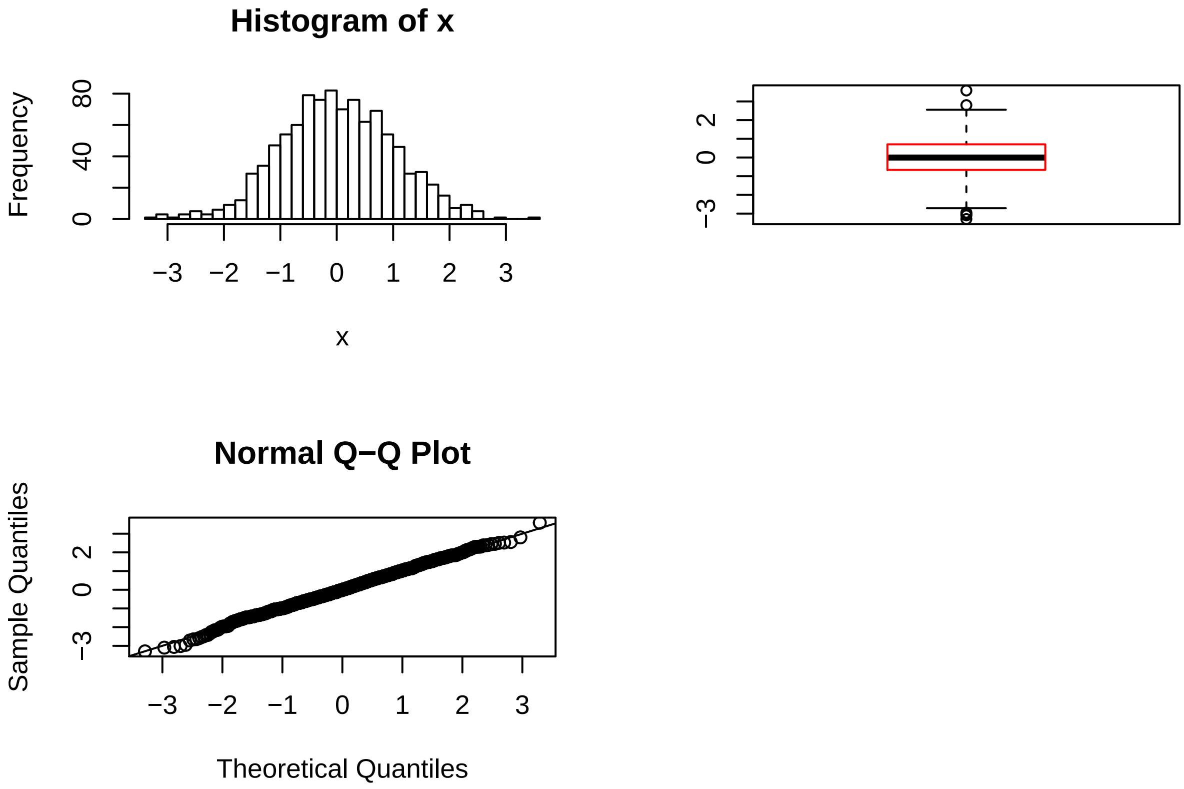


Figure 62: Random sample from N(0,1).

Sample fron N(2,1)

The sample was drawn from $N(2,1)$, i.e., it represents a shift model with the same variabiliy comapre to $N(0,1)$. In this case, presented in Figure~@ref(fig:figchp82), we expect that all points in the normal probability plot will lay above and parallel to the 45^o lines.

```
x <- rnorm(1000, 2, 1)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
```

```
qqnorm(x)
abline(0, 1)
```

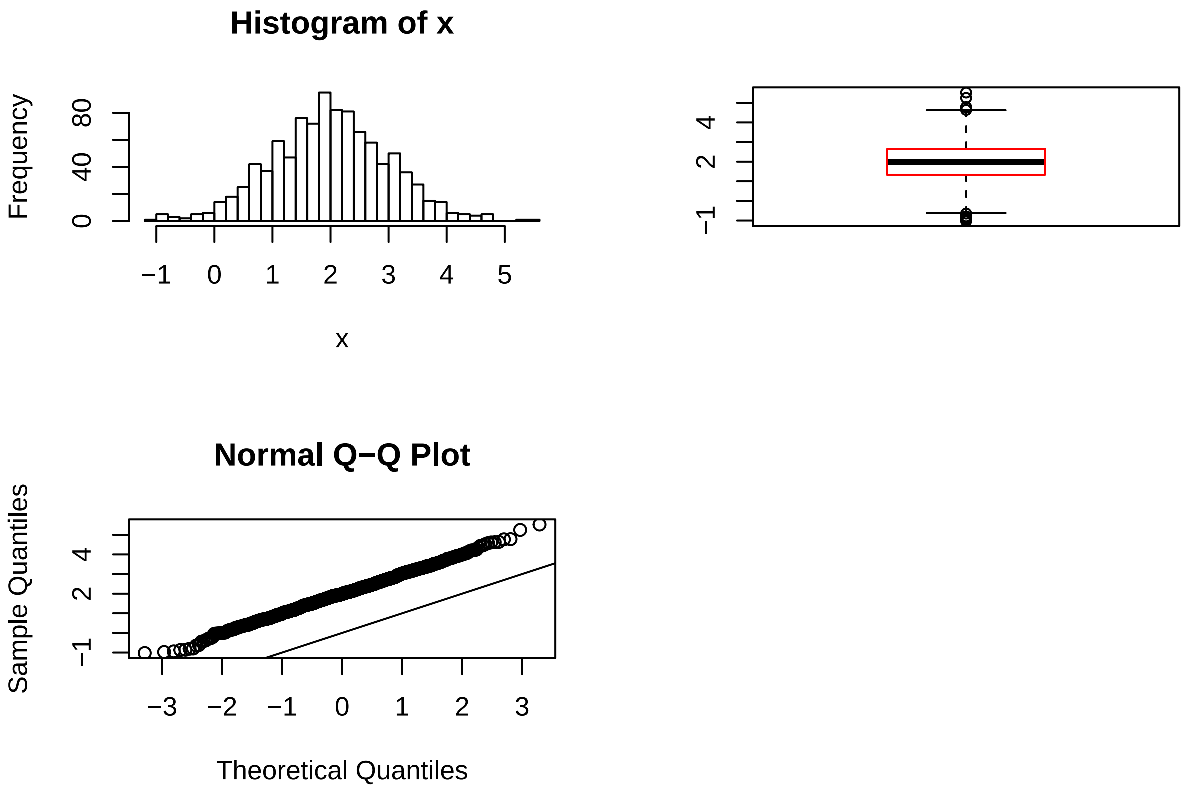


Figure 63: Random sample from N(2,1).

Sample fron N(0,2)

Figure~@ref(fig:figchp83) presents and an example in which the sample was drawn from $N(0, 2^2)$ which implies that the mean is the same as $N(0, 1)$ but the variability is higher. We expect the points in the normal probability plot to form a straight line with higher slope than 1.

```
x <- rnorm(1000, 0, 2)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```

Sample fron $t_{(3)}$

A $t_{(3)}$ has the same mean as $N(0, 1)$ but longer tails as shown in Figure~@ref(fig:figchp84). Note that the two distribution and centered around zero.

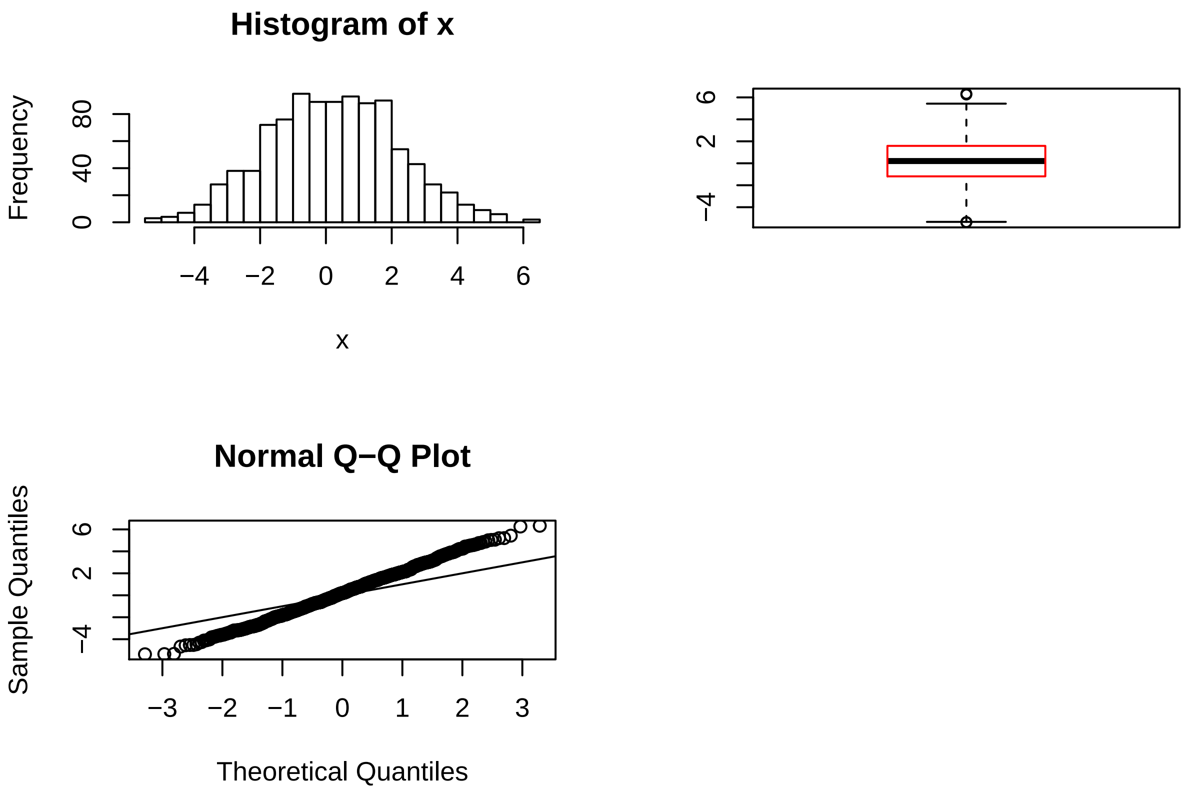


Figure 64: Random sample from N(0,2).

```
par(mfrow = c(1, 1))
qx <- seq(from = -7, to = 7, length = 1000)
xn <- dnorm(qx, mean = 0, sd = 1)
xt <- dt(qx, 3)
plot(qx, xn, xlim = c(-7, 7), type = "l")
lines(qx, xt, col=2)
```

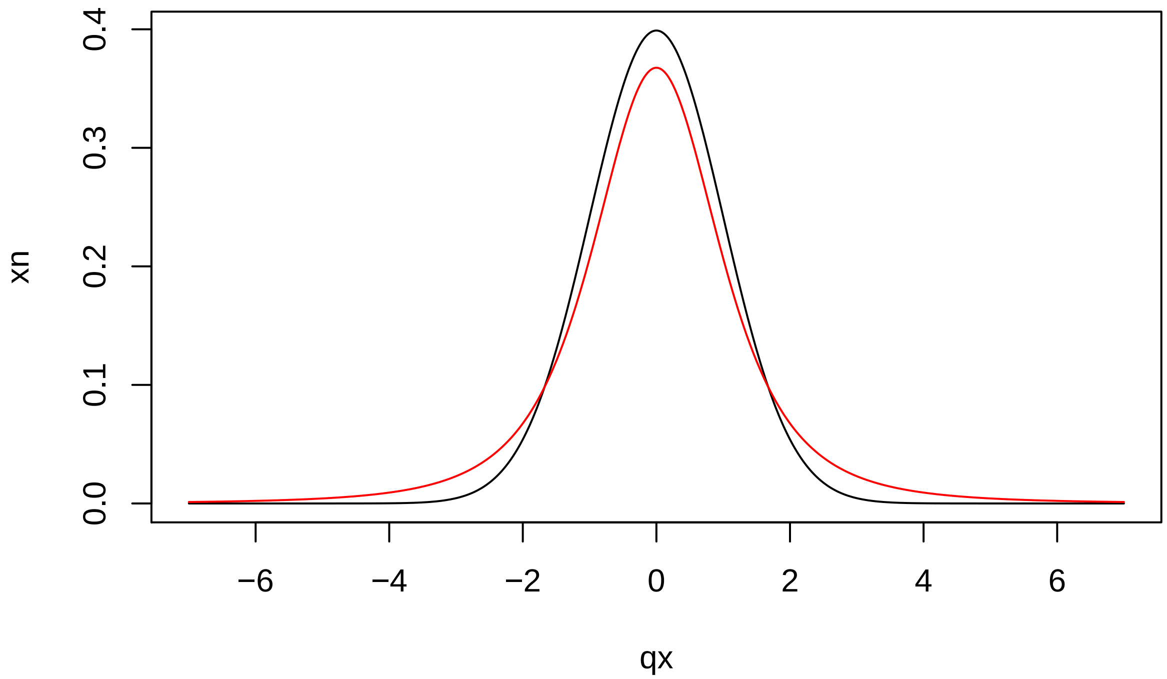


Figure 65: Density plot for N(0,1) and t(3).

This implies that in the normal probability plot we expect that the points will lay on the 45^o line in the center but with more extreme values as shown in Figure~@ref(fig:figchp85).

```
x <- rt(1000, 3)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```

Sample fron $U(-3, 3)$

As can be seen in the histogram presented in Figure~@ref(fig:figchp86), the data of this example is uniformly distributed across the minimum and maximum values. Therefore, we expect the points in the normal probability plot to cross the 45^o lines and to lay relatively far from the line.

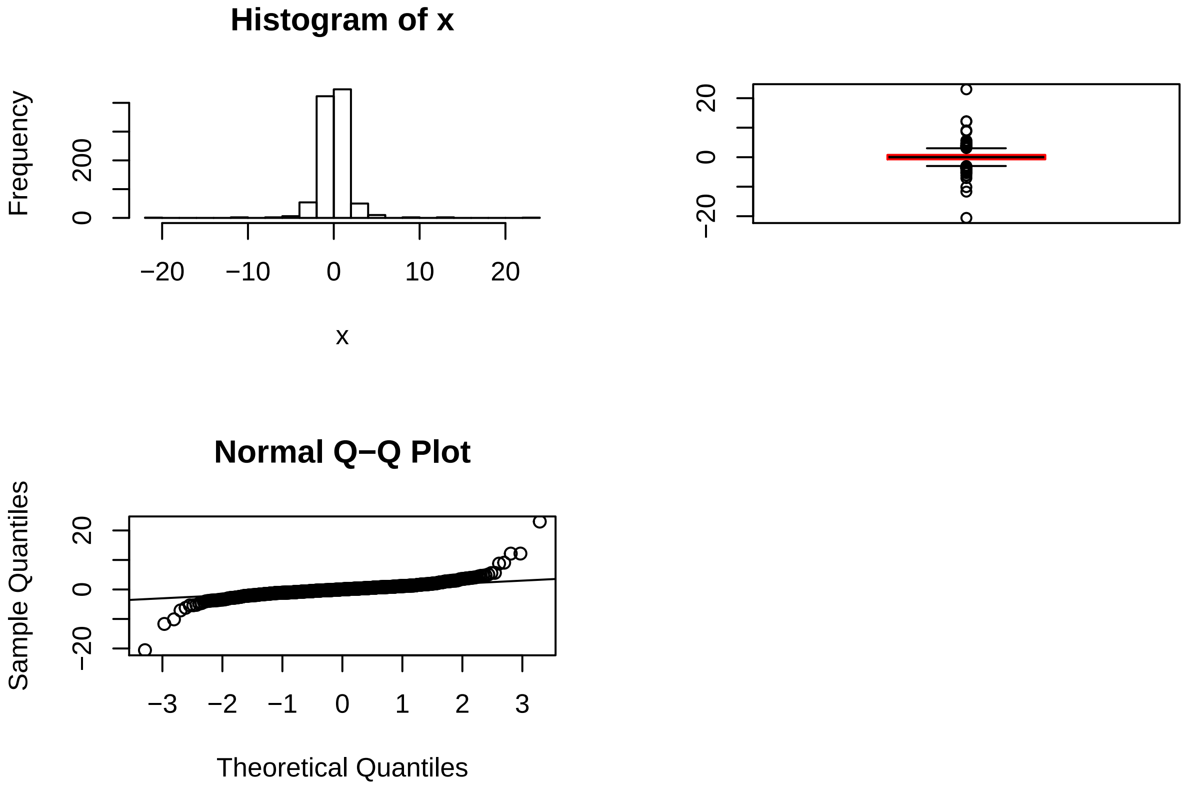


Figure 66: Random sample from t(3).

```
x <- runif(1000, -3, 3)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```

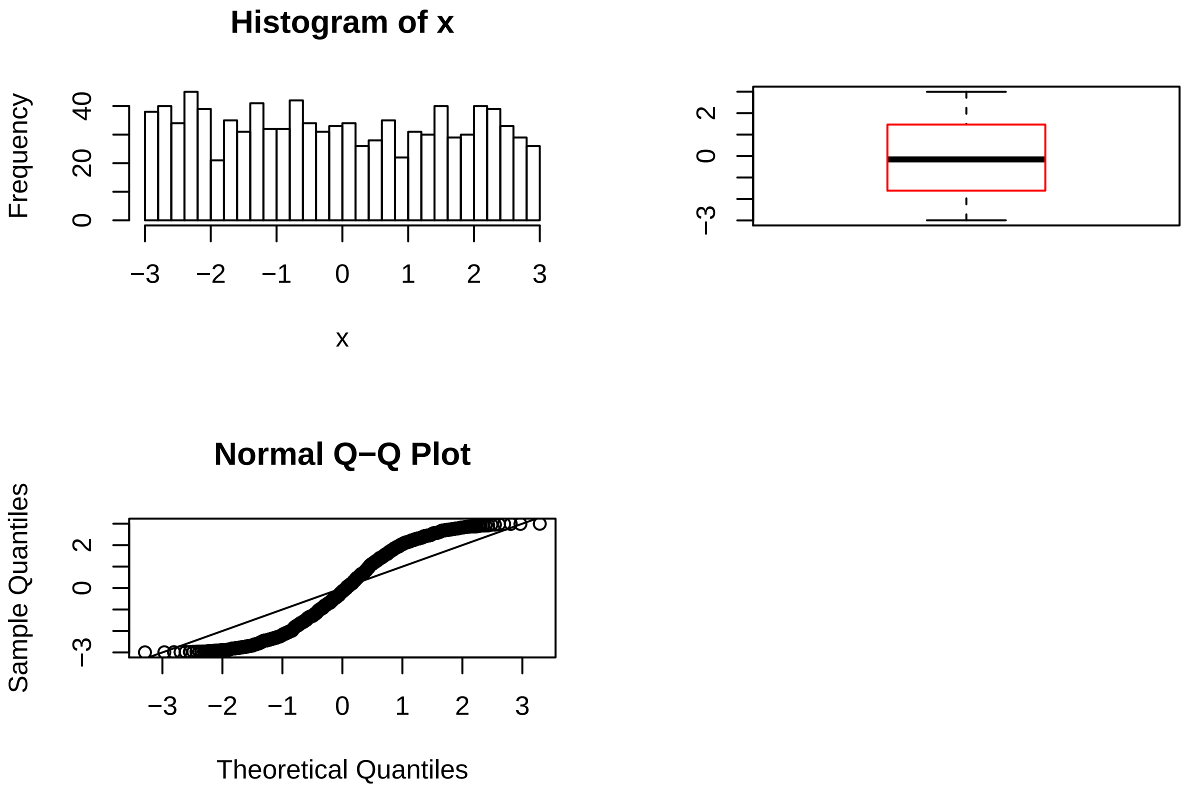


Figure 67: Random sample from U(-3,3).

Kobe Bryant’s performance indicators over the championship years

In this section we focus on the distribution of the average field goal success (FG%) and game score (GmSc) in Kobe Bryant’s five championship years and in particular we investigate if the distribution of these variables can be approximated by a normal distribution.

FG%

Figure~@ref(fig:figchp87) shows the multi-way qqnormal plot for the FG% reveals that the distribution of the variable exhibits a departure from a normal distribution approximation. Note that the qqmath() function that was used to produce Figure~@ref(fig:figchp87) is a part of the lattice package.

```
qqmath(~ FG. | year.i,
        distribution = qnorm,
        data=xx21.b,
           layout=c(3,2),
             prepanel = prepanel.qqmathline,
            panel = function(x, ...) {
            panel.grid()
            panel.qqmathline(x, ...)
            panel.qqmath(x, ...)
            },
        aspect=1,
        xlab = "f-value",
        ylab="field goal success")
```

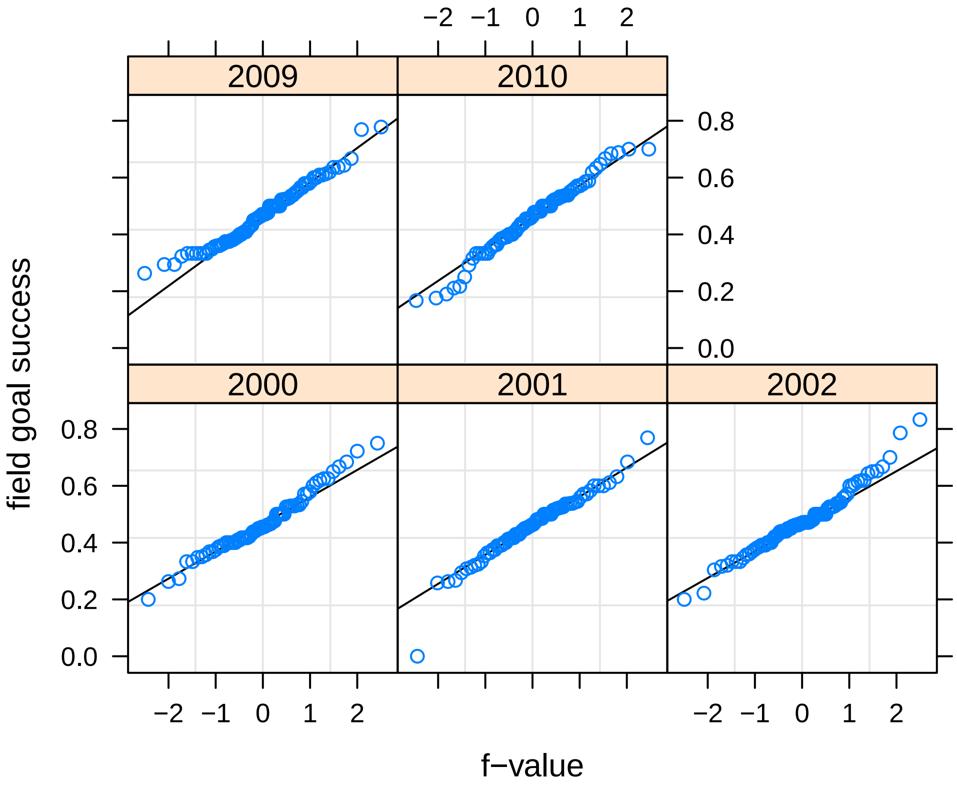


Figure 68: normal probability plot for the distribution of FG%.

For example, the bi-model distribution in 2009 or the heavy tailed distribution in 2009 and 2010 that can be seen in Figure~@ref(fig:figchp88).

```
qplot(FG., data=xx21.b, geom="density",
fill = year.i, alpha = I(0.2))+
xlab("%FG")
```

These patterns can be seen more clearly when a ridgeplot, shown in Figure~@ref(fig:figchp89), is used to visualize the data.

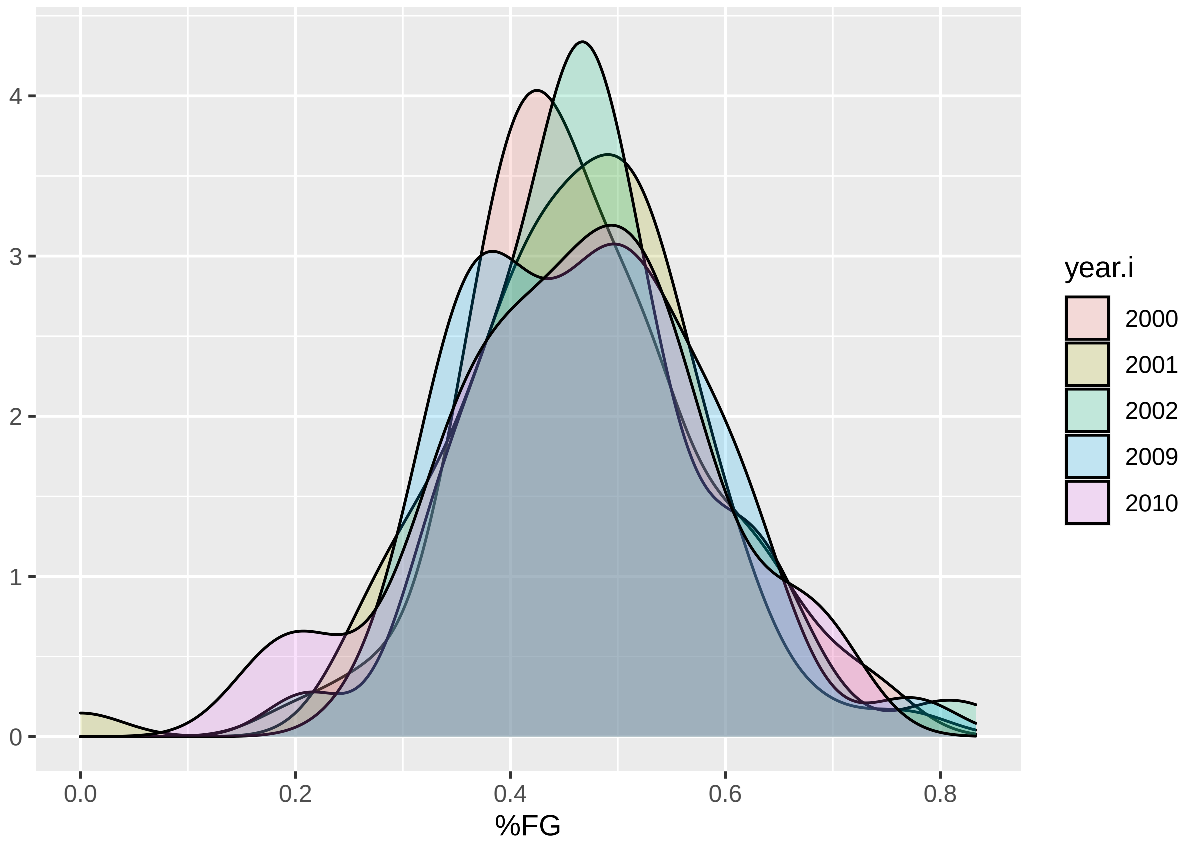


Figure 69: Density plot for the distribution of FG%.

```
library(ggridges)
ggplot(xx21.b, aes(x=FG.,y=year.i,fill = year.i)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```

```
## Picking joint bandwidth of 0.0393
```

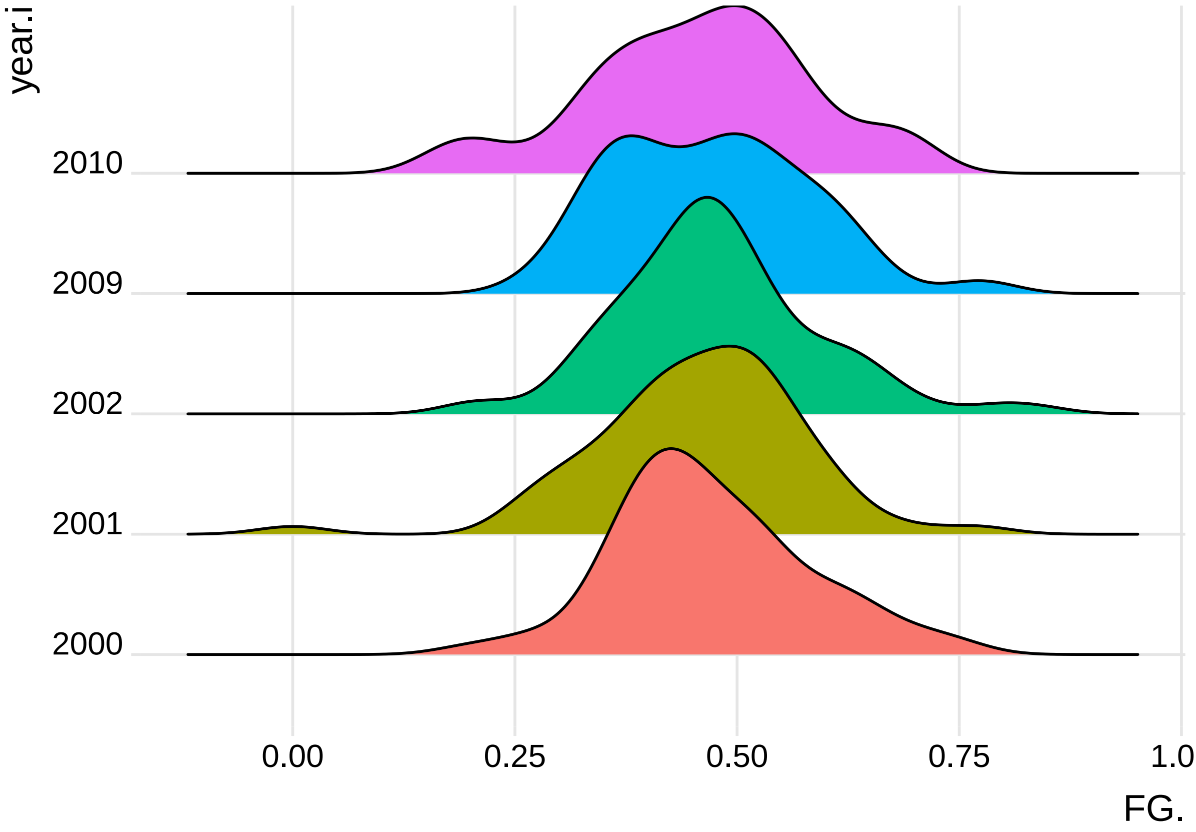


Figure 70: Ridge plot for FG%: 2009

Game score (GmSc)

The multi-way qqnormal plot for game score is presented in Figure~@ref(fig:figchp810) reveals that the distributions are close to normal distributions.

```
qqmath(~ GmSc | year.i,
        distribution = qnorm,
        data=xx21.b,
           layout=c(3,2),
             prepanel = prepanel.qqmathline,
            panel = function(x, ...) {
            panel.grid()
            panel.qqmathline(x, ...)
            panel.qqmath(x, ...)
            },
        aspect=1,
        xlab = "f-value",
        ylab="GmSc")
```

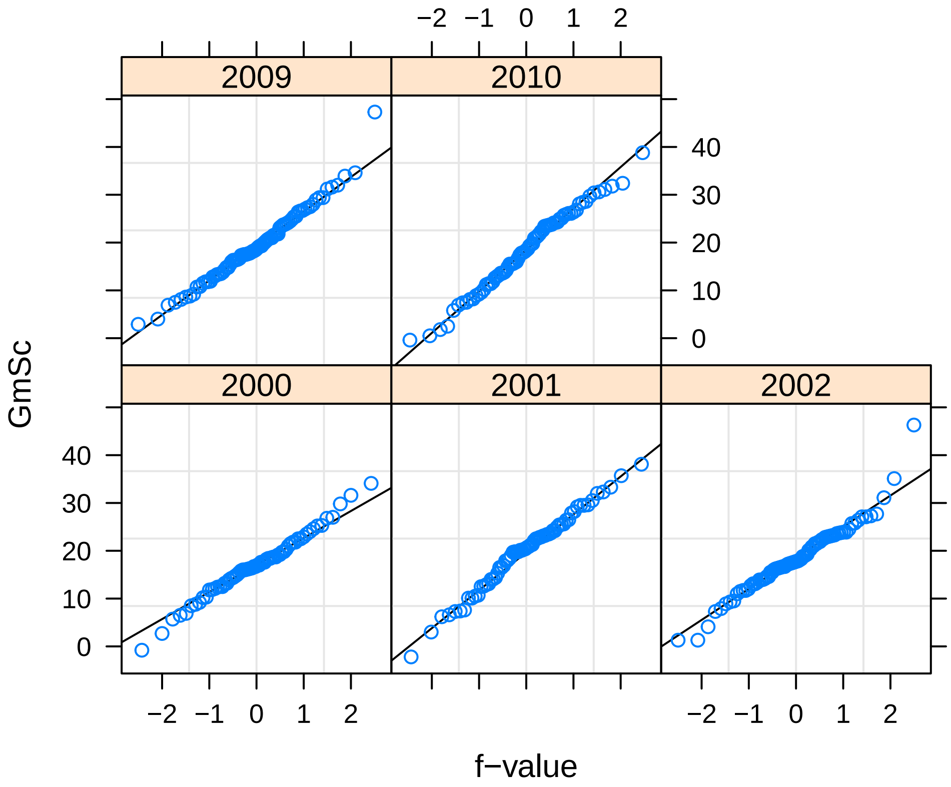


Figure 71: normal probability plot for the distribution of GmSc

Multiway histogram is shown in Figure~@ref(fig:figchp811).

```
ggplot(xx21.b, aes(GmSc,fill = year.i)) +
geom_histogram() +
facet_wrap(~year.i,ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

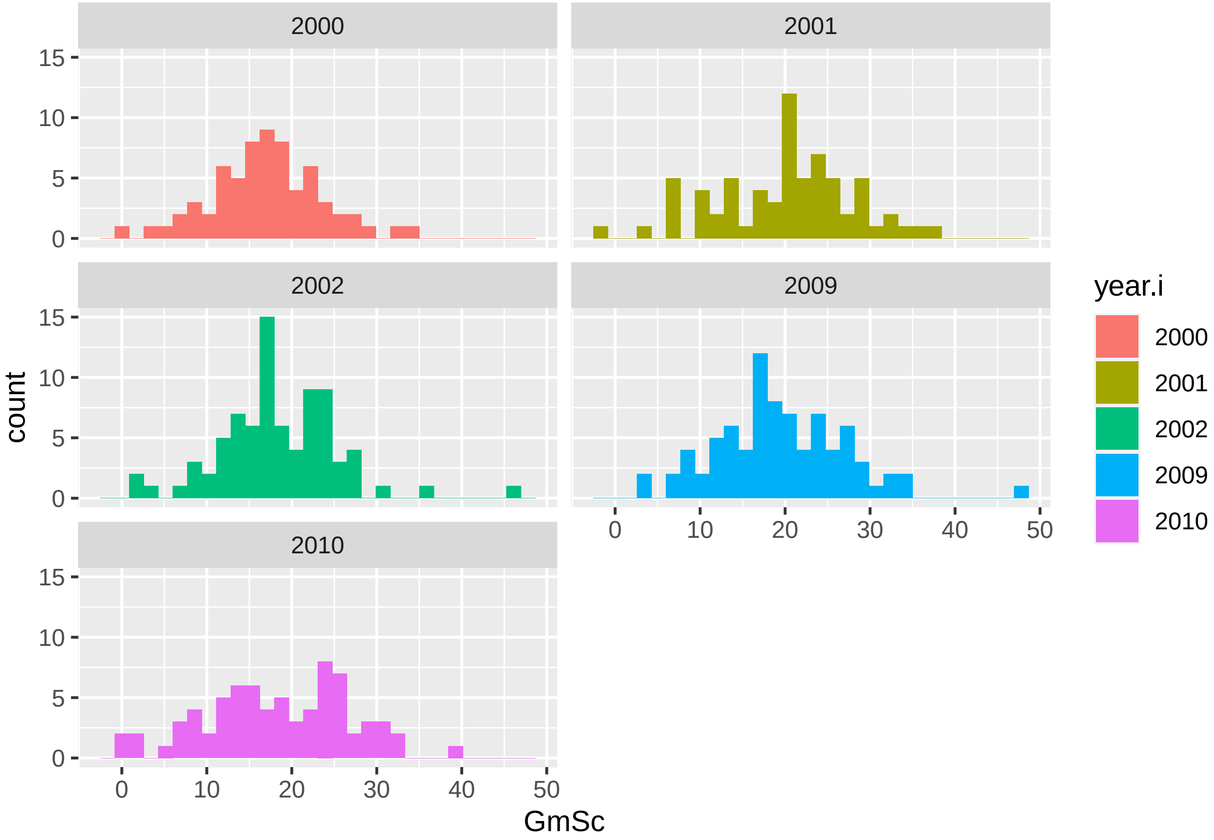


Figure 72: Histogram for the distribution of GmSc