Univariate Data

2020

Distinction between the types of data in statistics

There is a distinction between the types of data in statistics. Univariate data can be of three types:

- **Categorical** - collection of data that is divided into groups or categories
- **Numerical**
  - **Discrete**
  - **Continuous**

Examples:

- Categorical
  - Survey that records whether a person is for or against a proposition $\{Yes, No\}$
  - Hair colors: $\{brunette, brown, red, blonde\}$
  - Eye colors: $\{black, brown, green, blue\}$
  - The blood type of a person: $\{A, B, AB, 0\}$
  - Educational level: $\{high\ school, BSc., MSc, PhD\}$
  - Customer satisfaction: $\{very\ poor, poor, neutral, good, very\ good\}$
  - Traveling motives: $\{business, leisure, family, study, health\}$
  - Computer brands: $\{Dell, HP, Apple, Lenovo, Microsoft, Asus, Acer\}$

    - Car brands: $\{Nissan, Fiat, Ford, Toyota, Volkswagen, Ferrari\}$
    - Weather: $\{Sunny, Cloudy, Stormy, Fog, Rain, Snow\}$
- Numerical discrete
    - Number of students taking this class: $\{..., 15, 16, 17, ...\}$
    - Age of a person: $\{5, 10, 20, 22, 23, 50, 70\}$
    - Number of children: $\{1, 2, 3, ...\}$
    - Height: $\{157, 160, 170, 180, 190\}$
    - Weight: $\{57, 60, 62, 64, 70, 73, 80, 90, 100\}$
- Numerical continuous
    - Age of a person: $\{0.7, 5.5, 10.3, 15.8, 20.1\}$
    - Height: $\{1.57, 1.60, 1.70, 1.80, 1.90\}$
    - Weight: $\{57.3, 60.7, 62.4, 64.2, 70.5, 73.2, 80.5, 90.7, 100.4\}$
    - Body Temperature: $\{36.5, 37, 38.2, 39\}$

Methods for viewing and summarising the data depend on the type of the variable:

- Categorical
    - frequency table
    - probability table
    - bar graph
    - pie chart
- Numeric
    - measuring the center
        - mean
        - trimmed mean
        - median
    - measuring the spread
        - range
        - standard deviation (sd)
        - variance (var)
        - quantile
        - quartile
        - hinge
        - inter quartile range (IQR)

        - median average deviation (MAD)
    - summary
    - fivenum
    - stem-and-leaf charts
    - hist
    - boxplot

Categorical Data

Frequency table

Frequency tables show how many observations we have in the separate groups. In R we can take it with the `table` function.

```
> library(MASS)
> table(survey$Exer)

Freq None Some 
 115   24   98 
> table(survey$Smoke)

Heavy Never Occas Regul 
   11   189    19    17 
```

Using the `survey` data frame `Exec` shows how often the students have been exercising. From the frequency table we see that 115 of the students are frequently exercising, 98 of the students are sometimes exercising and 24 of the students aren't exercising. `Smoke` shows how frequently the students are smoking. From the frequency table we see that 189 of the students aren't smoking, 19 are smoking occasionally, 17 are smoking regularly and 11 are heavily smoking.

Proportion table

Proportion tables show what part of the observations fall in the separate groups. In R for univatriate data we can observe this using the frequency table and dividing on the number of observations.

```
> library(MASS)
> table(survey$Exer) / length(survey$Exer)
```

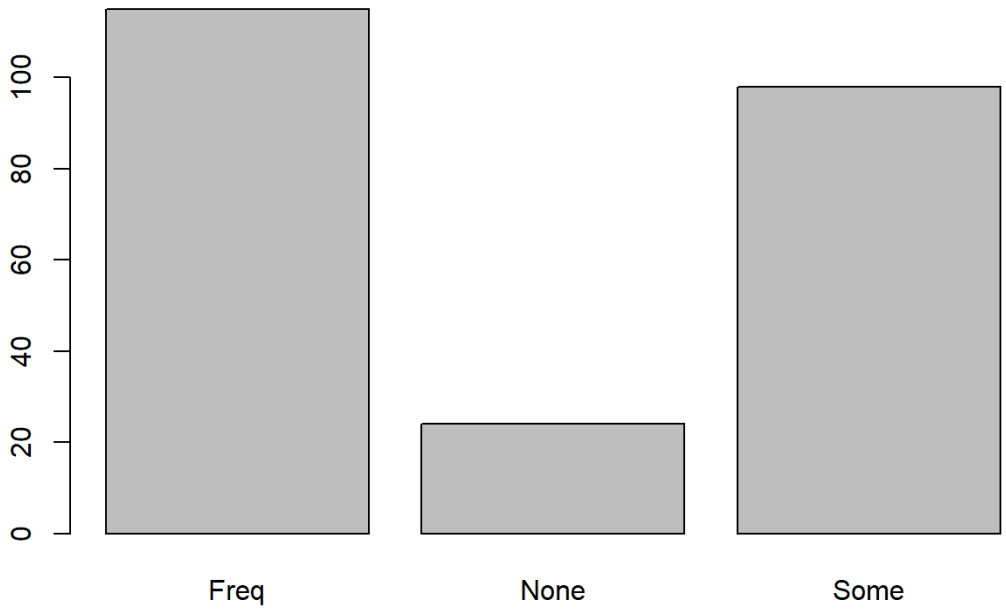
```
     Freq      None      Some 
0.4852321 0.1012658 0.4135021 
> table(survey$Smoke) / length(survey$Smoke)

     Heavy      Never      Occas      Regul 
0.04641350 0.79746835 0.08016878 0.07172996 
```

From the proportion table we see that 0.49 of the students are frequently exercising, 0.41 of the students are sometimes exercising and 0.1 of the students aren't exercising.
From the proportion table we see that 0.8 of the students aren't smoking, 0.08 are smoking occasionally, 0.07 are smoking regularly and 0.05 are heavily smoking.

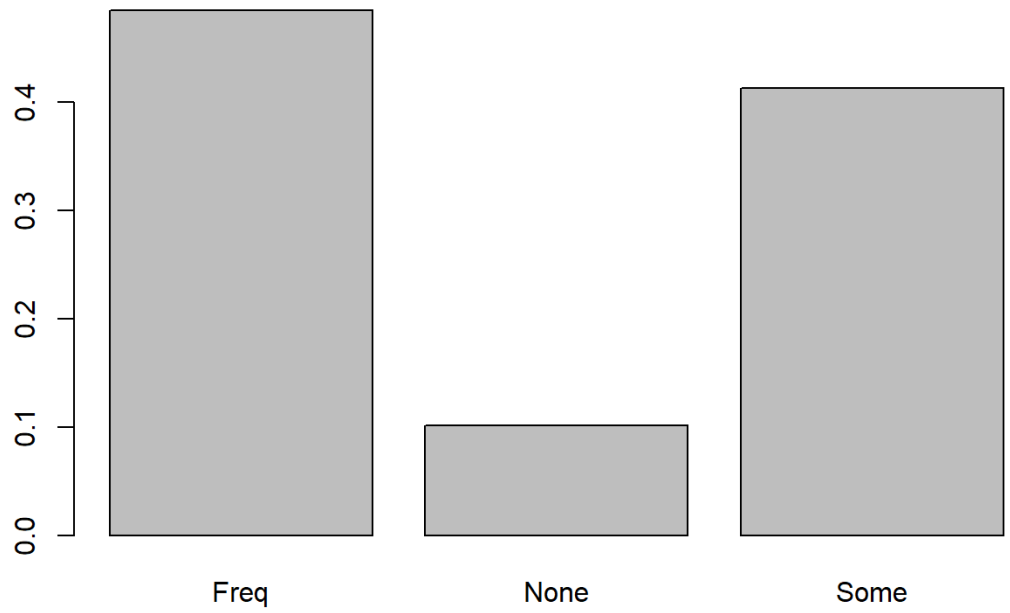
Barplot chart

Bar plot chart draws a bar with a height proportional to the count in the table. We can present the height using the frequency and proportion tables.

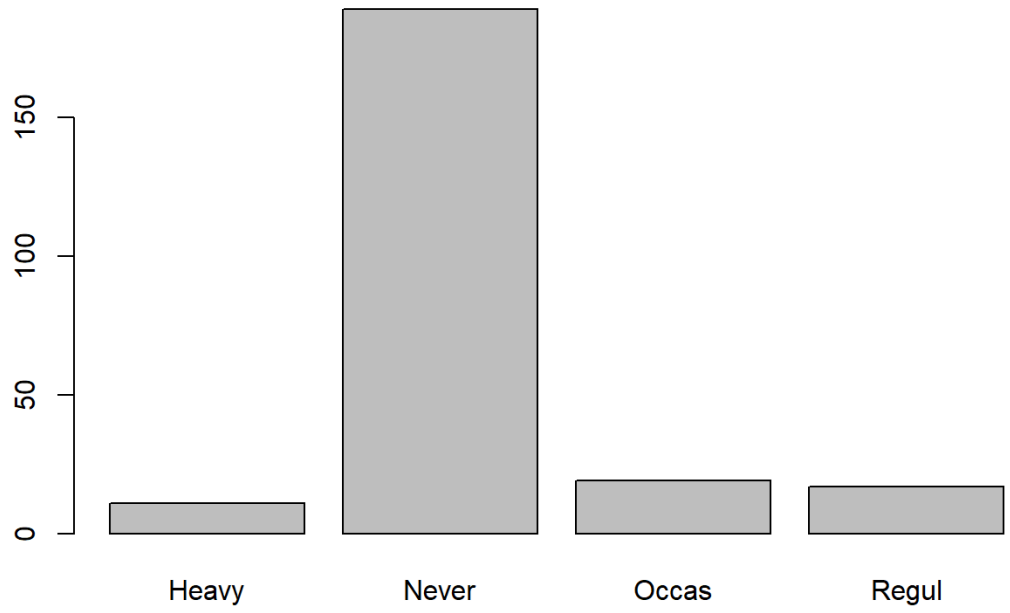
```
> library(MASS)
> barplot(table(survey$Exer))
```



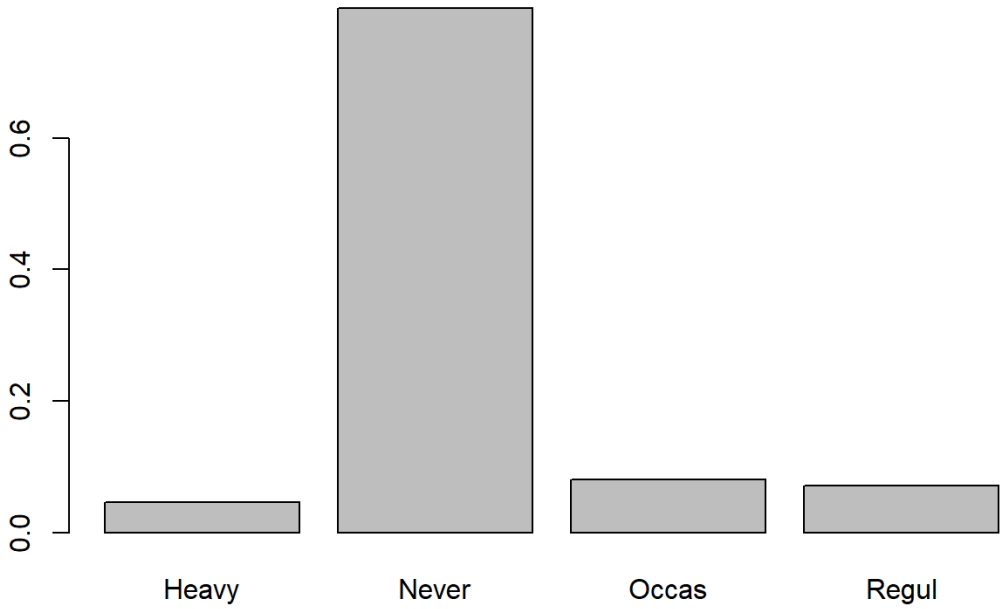
```
> barplot(table(survey$Exer) / length(survey$Exer))
```




```
> barplot(table(survey$Smoke))
```



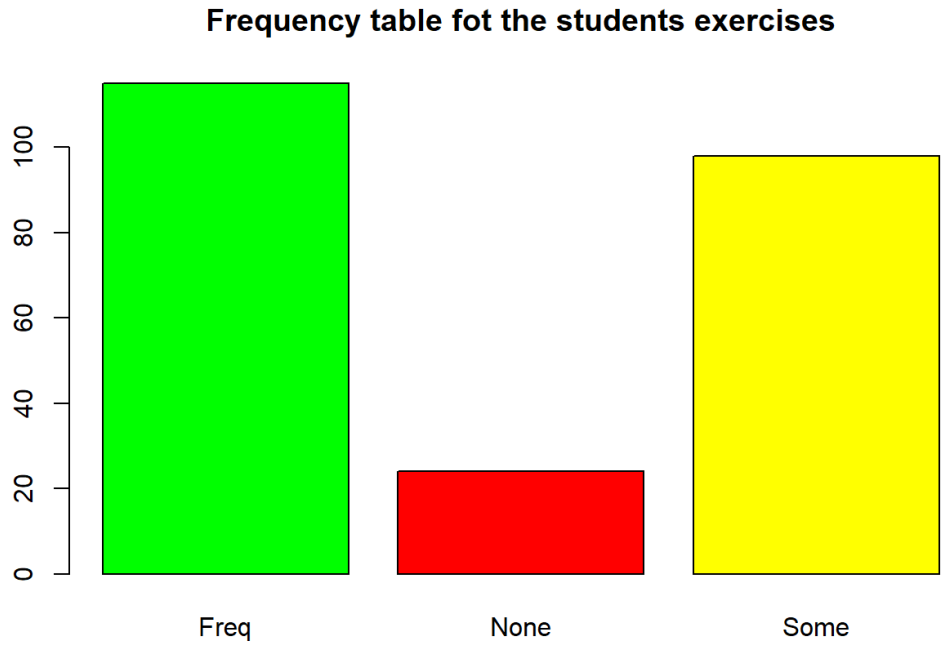
```
> barplot(table(survey$Smoke) / length(survey$Smoke))
```



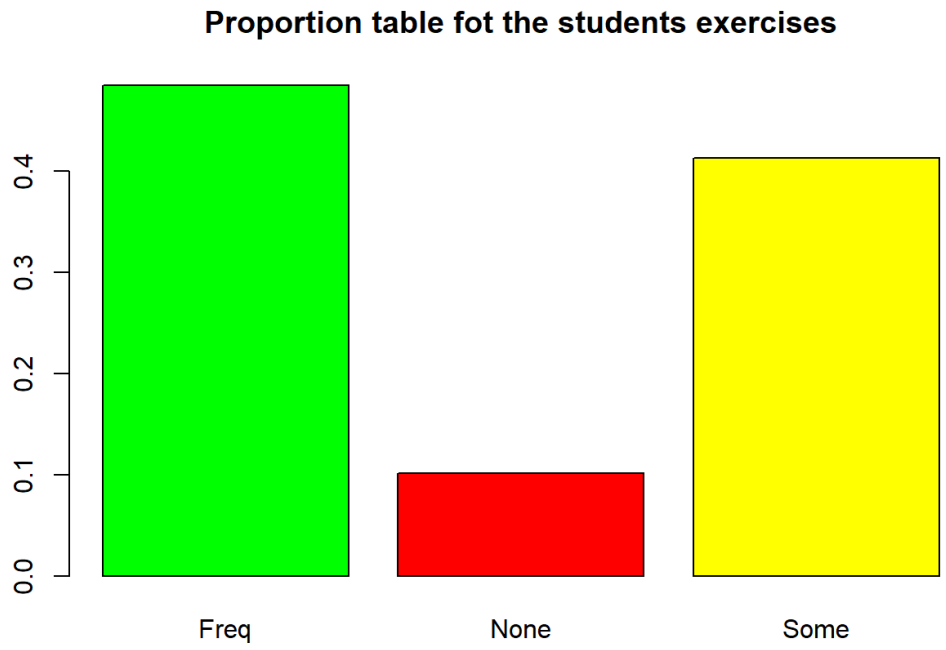
It is important to notice that we are not using the `barplot` with the raw data.

We can add headers and colors to the graphics

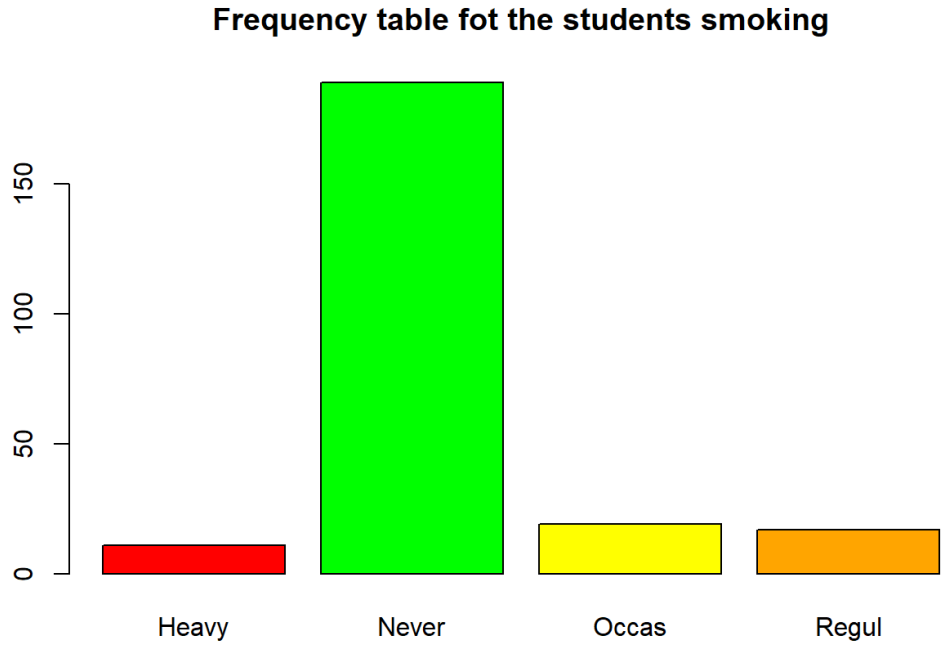
```
> library(MASS)
> barplot(table(survey$Exer), main = "Frequency table for
the students exercises", col = c("Green", "Red",
"Yellow"))
```



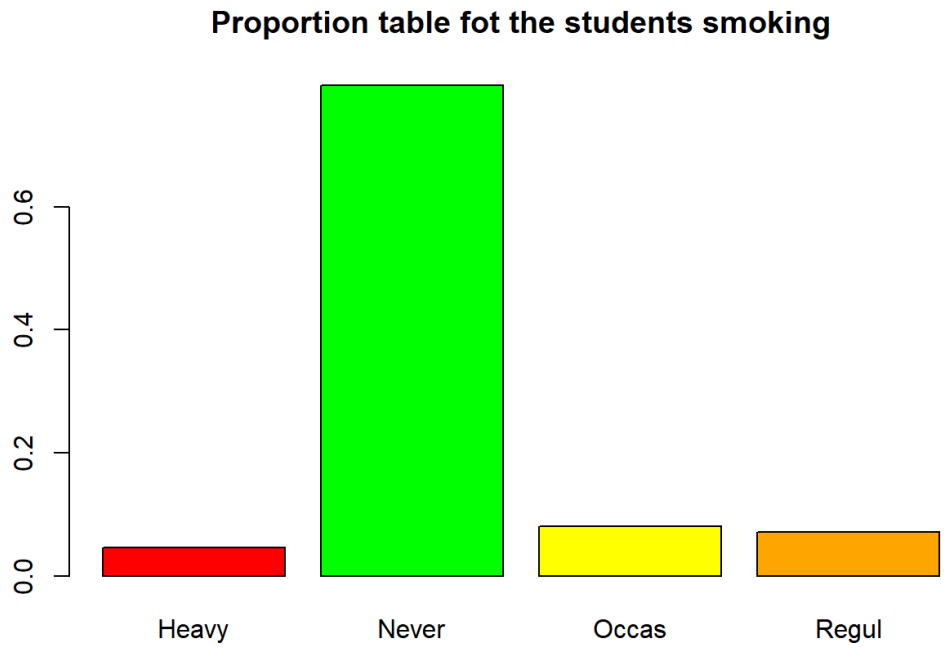
```
> barplot(table(survey$Exer) / length(survey$Exer), main
= "Proportion table fot the students exercises", col =
c("Green", "Red", "Yellow"))
```




```
> barplot(table(survey$Smoke), main = "Frequency table
fot the students smoking", col = c("Red", "Green",
"Yellow", "Orange"))
```



```
> barplot(table(survey$Smoke) / length(survey$Smoke),
main = "Proportion table fot the students smoking", col =
c("Red", "Green", "Yellow", "Orange"))
```

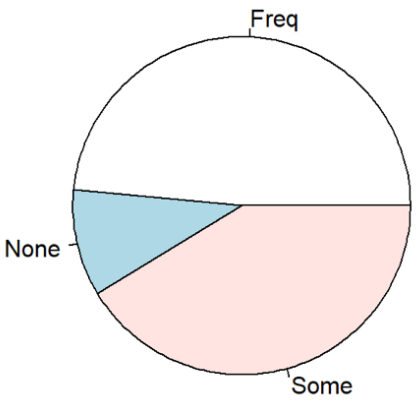


There is a lot of other options that you can see to this graphs, you can see them from the help.

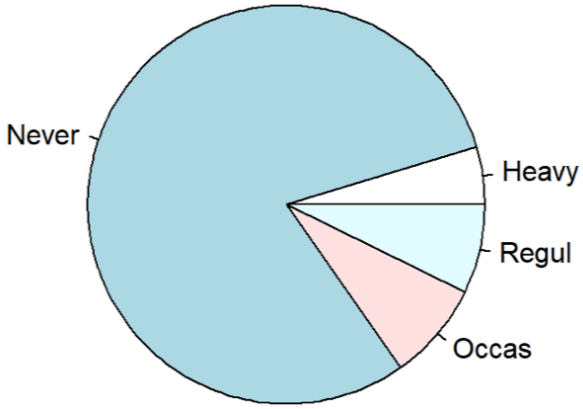
Pie chart

Pie chart draws a circle in which every category is presented as part of this circle.

```
> pie(table(survey$Exer))
```

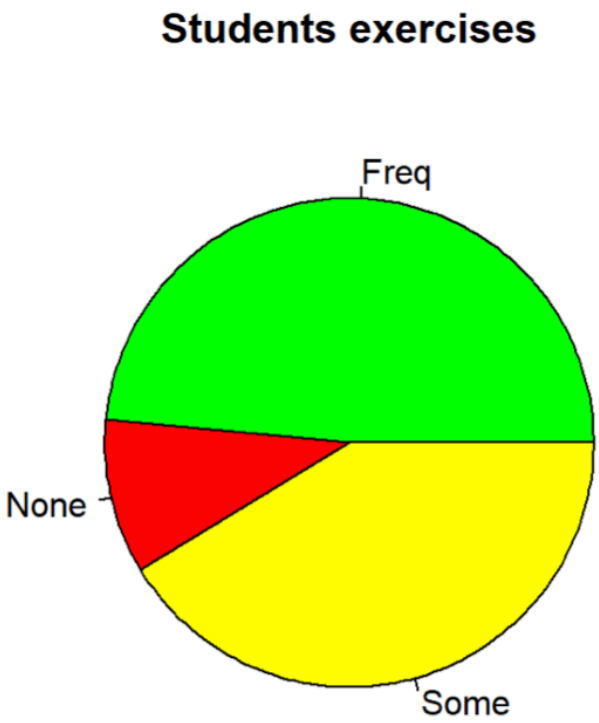


```
> pie(table(survey$Smoke))
```

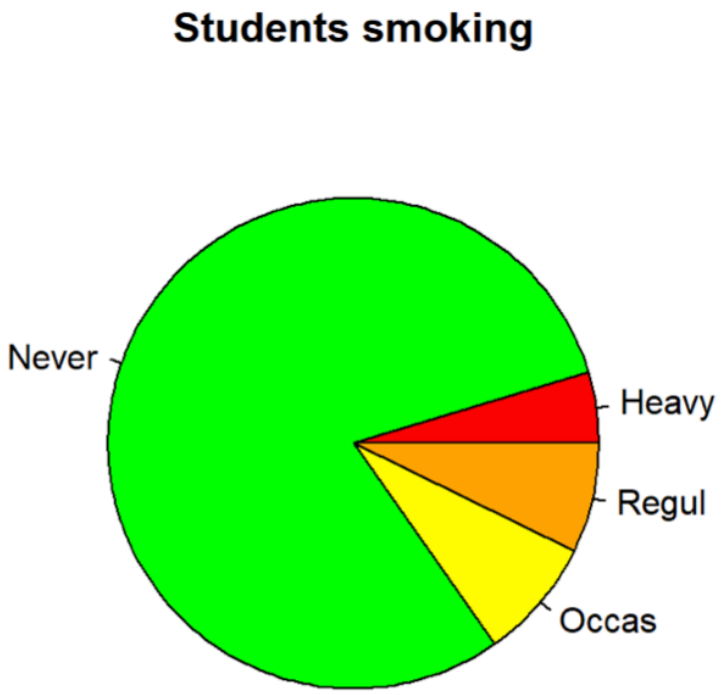


It is important to notice that we are not using the `pie` with the raw data. We can add headers and colors to the graphics

```
> pie(table(survey$Exer), main = "Students exercises",
col = c("Green", "Red", "Yellow"))
```



```
> pie(table(survey$Smoke), main = "Students smoking", col
= c("Red", "Green", "Yellow", "Orange"))
```



There is a lot of other options that you can see to this graphs, you can see them from the help.

Numerical data

To describe a distribution we often want to know where is it centered and what is the spread. These are typically measured with mean and variance(or standard deviation). We have already reviewed some of the functions in the Statistics tab, but lets see some more details and examples.

```
> min(survey$Age)
[1] 16.75
> max(survey$Age)
[1] 73
> range(survey$Age)
[1] 16.75 73.00
```

We see that the Age of the students in the survey are between 16.75 and 73 years old.

```
> mean(survey$Age)
[1] 20.37451
> median(survey$Age)
[1] 18.583
> quantile(survey$Age, 0.25)
   25%
17.667
> quantile(survey$Age, 0.75)
   75%
20.167
> summary(survey$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  16.75   17.67   18.58   20.37   20.17   73.00
> fivenum(survey$Age)
[1] 16.750 17.667 18.583 20.167 73.000
```

We see that the average age of the students in the survey is 20.37. We see that 25% of the student are under age of 17.67, 50% of the students are under age of 18.58 and 75% of the students are under age of 20.17.

We can also see the trimmed mean

```
> mean(survey$Age, trim=10/237)
[1] 19.37985
```

Standard deviation of the age of the students taking place in the survey is 6.47

```
> sd(survey$Age)
[1] 6.474335
> var(survey$Age)
[1] 41.91701
```

The standard deviation and the variance are also sensitive to outliers. Resistant measure of spread include the IQR and the MAD.

```
> IQR(survey$Age)
[1] 2.5
> mad(survey$Age)
[1] 1.605656
```

Stem and leaf chart

When the data set is really small we can use the stem and leaf diagram to see the shape of the distribution and the values. The number on the left of the bar is the stem and the number on the right is the digit. We must put them together to find the observation.

```
> stem(survey$Age)

  The decimal point is 1 digit(s) to the right of the |

  1 |
77777777777777777777777777777777777788888888888888888888888888+80
  2 |
00000000000000000000000011111111111111122223333334444445677999
  3 | 1133
  3 | 667
  4 | 0244
  4 |
  5 |
  5 |
  6 |
  6 |
  7 | 03
> stem(survey$Age, scale = 5)

  The decimal point is at the |

  16 | 8999
  17 |
00111111122222222223333333334444444445555555555555566667777778888888
  18 |
00000111222222233333333333444445555555566666777778888889999999999
  19 | 00001122222333333334457778889999
  20 | 0001122223333457888 9
  21 | 0011222334569
  22 | 389
  23 | 013456678
```

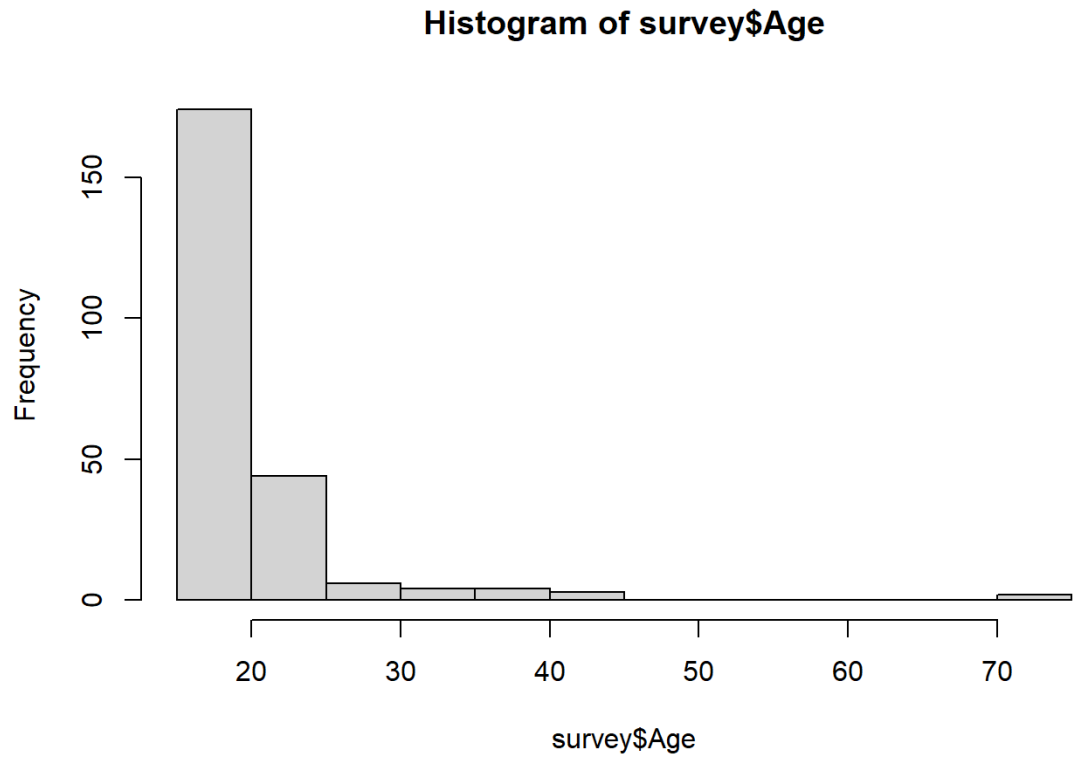
```
24 | 27
25 | 5
26 | 5
27 | 3
28 | 56
29 | 1
30 | 78
31 |
32 | 78
33 |
34 |
35 | 58
36 | 6
37 |
38 |
39 | 8
40 |
41 | 6
42 |
43 | 8
44 | 3
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
```

```
  66 |
  67 |
  68 |
  69 |
  70 | 4
  71 |
  72 |
  73 | 0
```

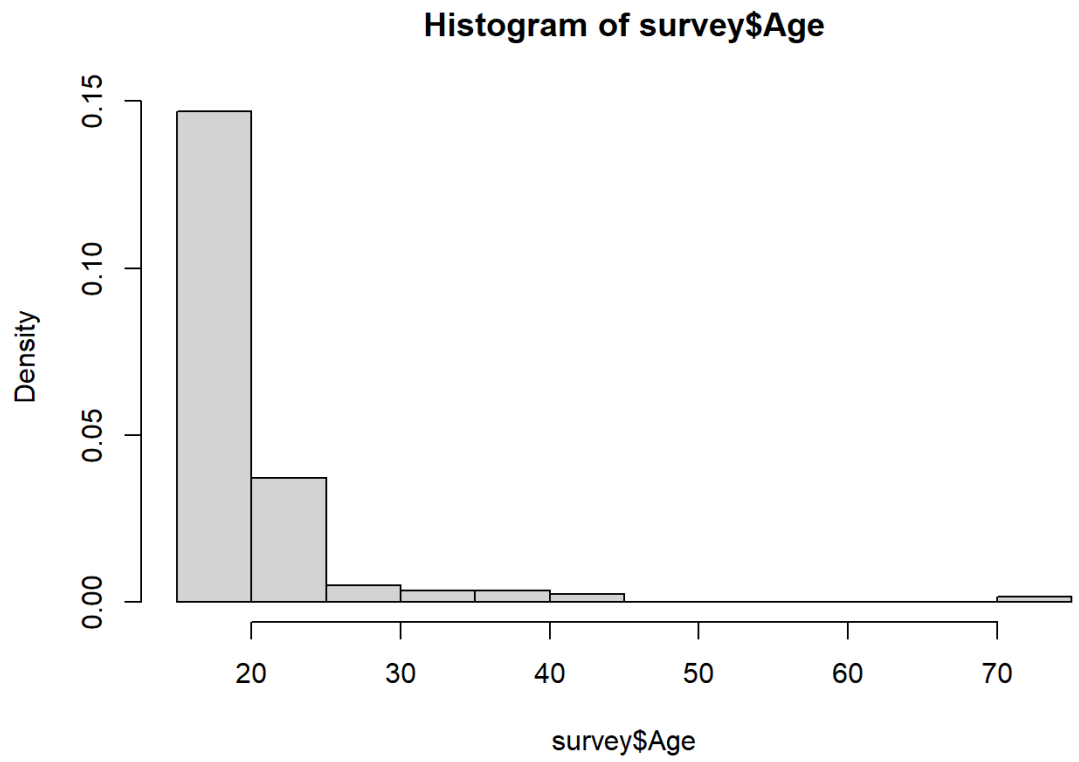
Histogram

If there is more data. The most common graphic is the histogram `hist`. It defines a sequence of breaks and then counts the number of observations in the bins formed by the breaks. Plotting these with a bar similar to the bar chart, but the bars are touching. The height can be the frequencies, or the proportions.

```
> hist(survey$Age)
```

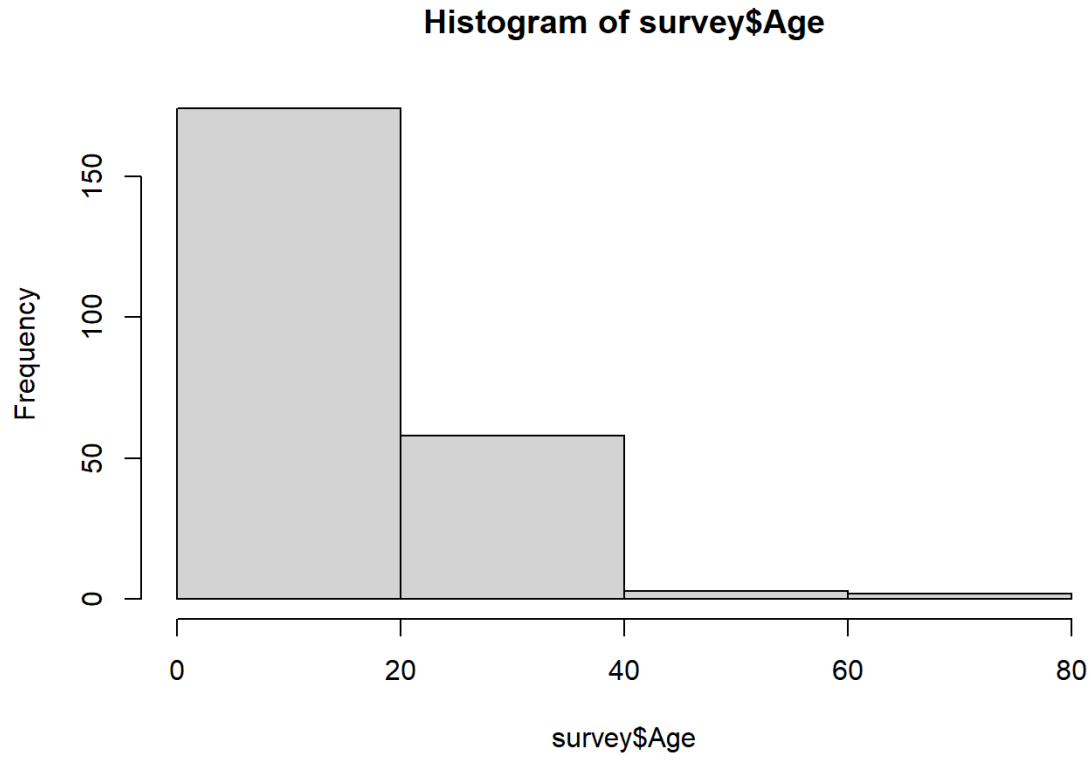


```
> hist(survey$Age, probability = TRUE)
```

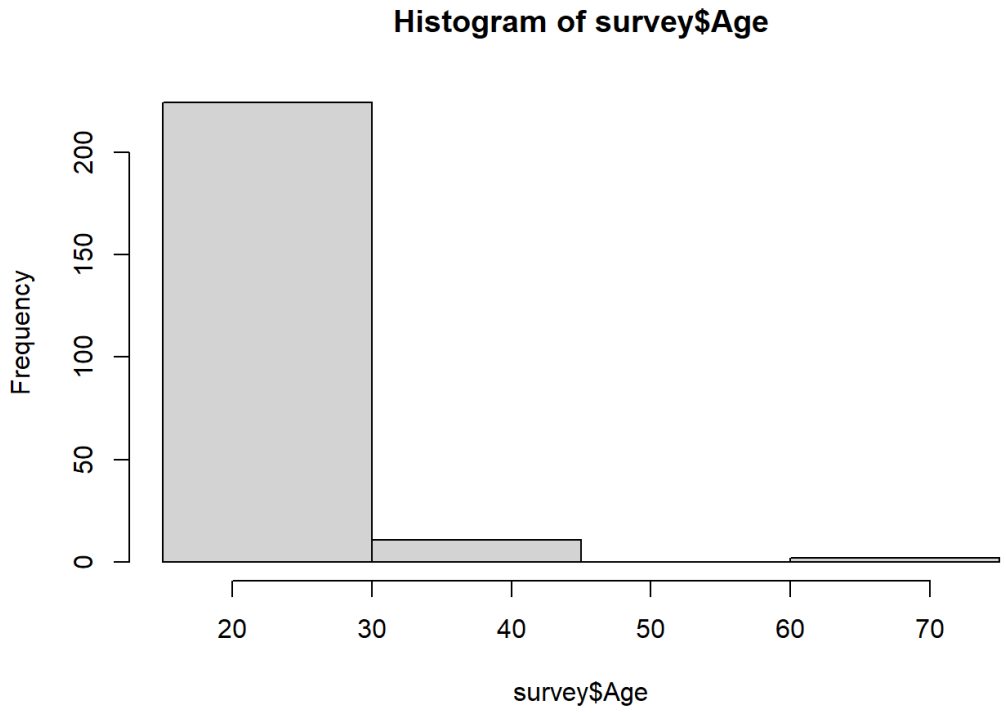


We can specify our own number of breaks or break points.

```
> hist(survey$Age, breaks = 4)
```



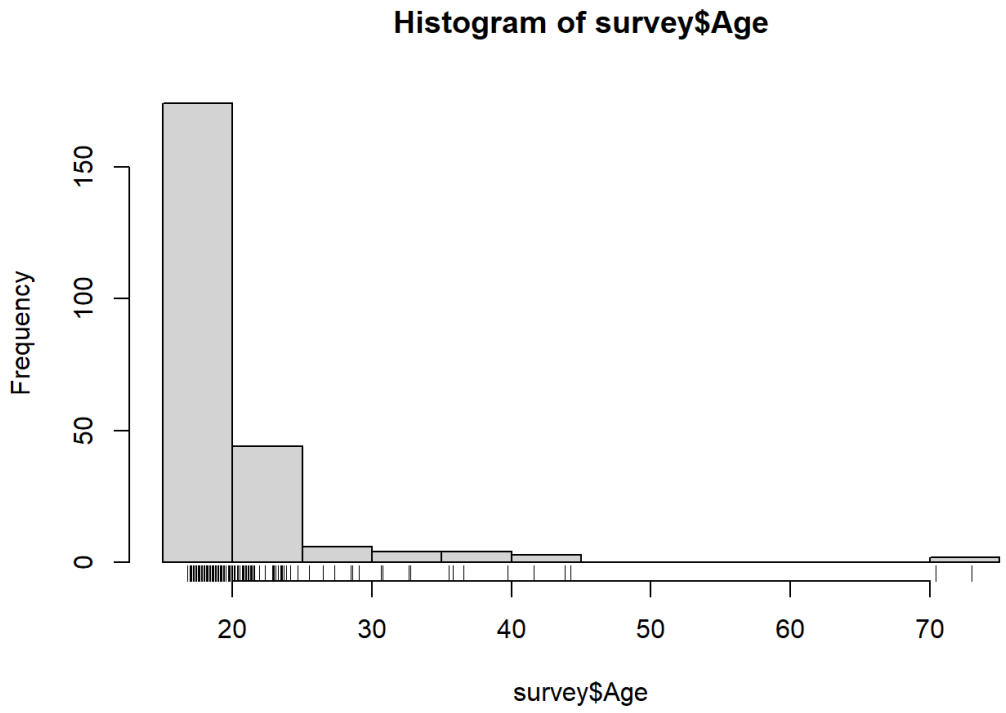
```
> hist(survey$Age, breaks = c(15, 30, 45, 60, 75))
```



It is wort mentioning that the median divides the histogram into two equal area pieces, the median would be the point where the histogram would balance, and the IQR captures exactly the middle half of the data.

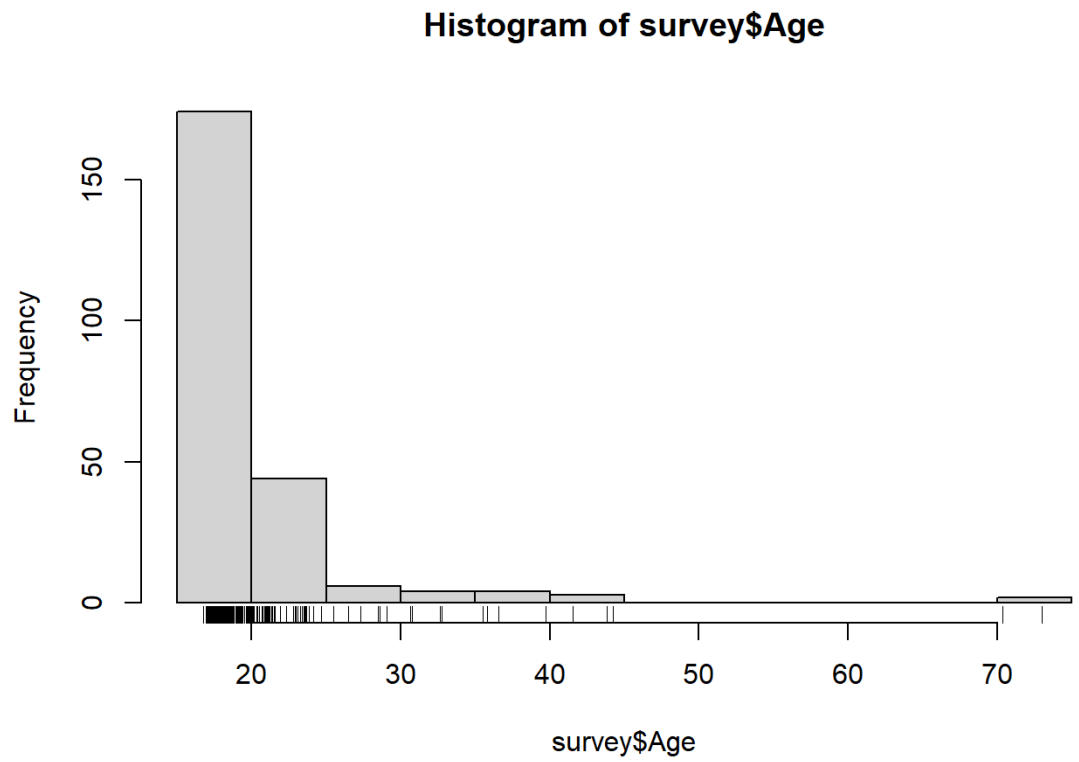
We can use `rug` function to represent the data to the plot

```
> hist(survey$Age)
> rug(survey$Age)
```



and we can use `jitter` to add small amount of noise to the numeric vector, so we can see the overlapping dots.

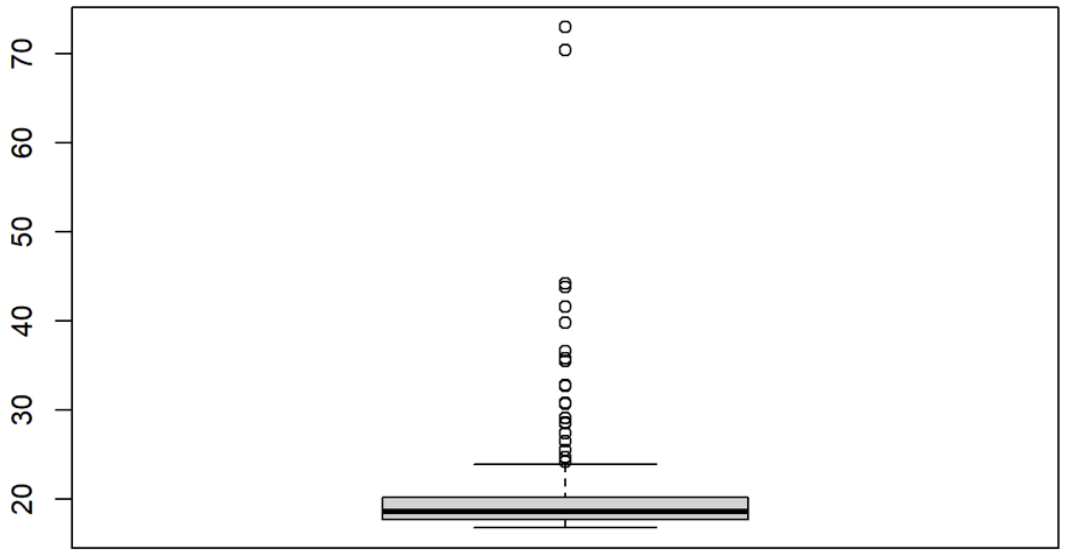
```
> head(survey$Age)
[1] 18.250 17.583 16.917 20.333 23.667 21.000
> head(jitter(survey$Age))
[1] 18.24628 17.57149 16.90796 20.33290 23.66721 21.01301
> hist(survey$Age)
> rug(jitter(survey$Age))
```



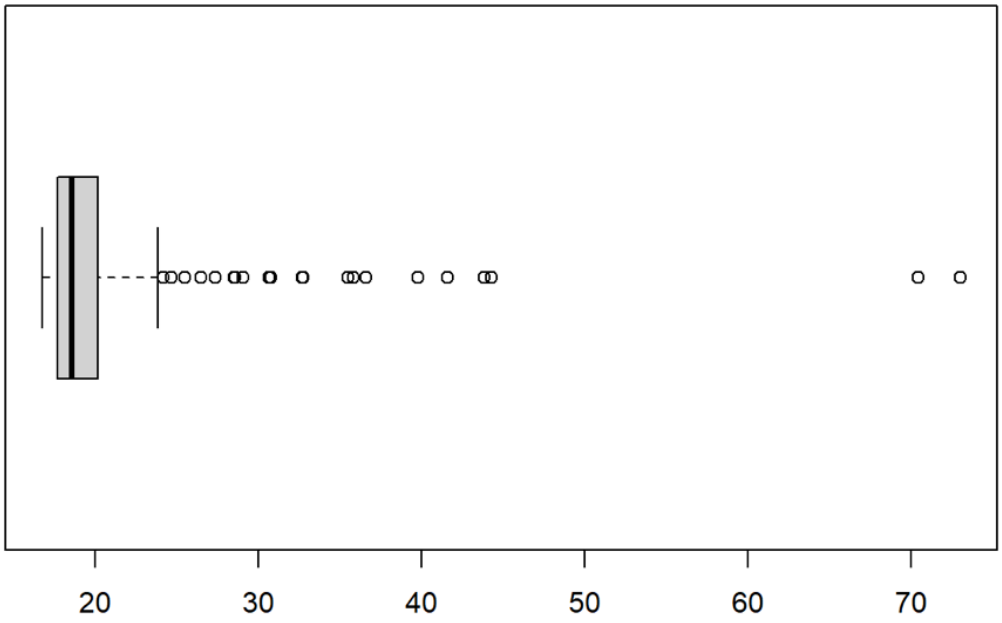
Boxplot

Box plot is based on the 5-number summary. In its simplest usage, the boxplot has a box with lines at Q_1, the Median, Q_3 and whiskers which extend to the min and max. To showcase possible outliers, a convention is adopted to shorten the whiskers to the length of 1.5 times the box length. Any points beyond that are plotted with points.

```
> boxplot(survey$Age)
```




```
> boxplot(survey$Age, horizontal = TRUE)
```

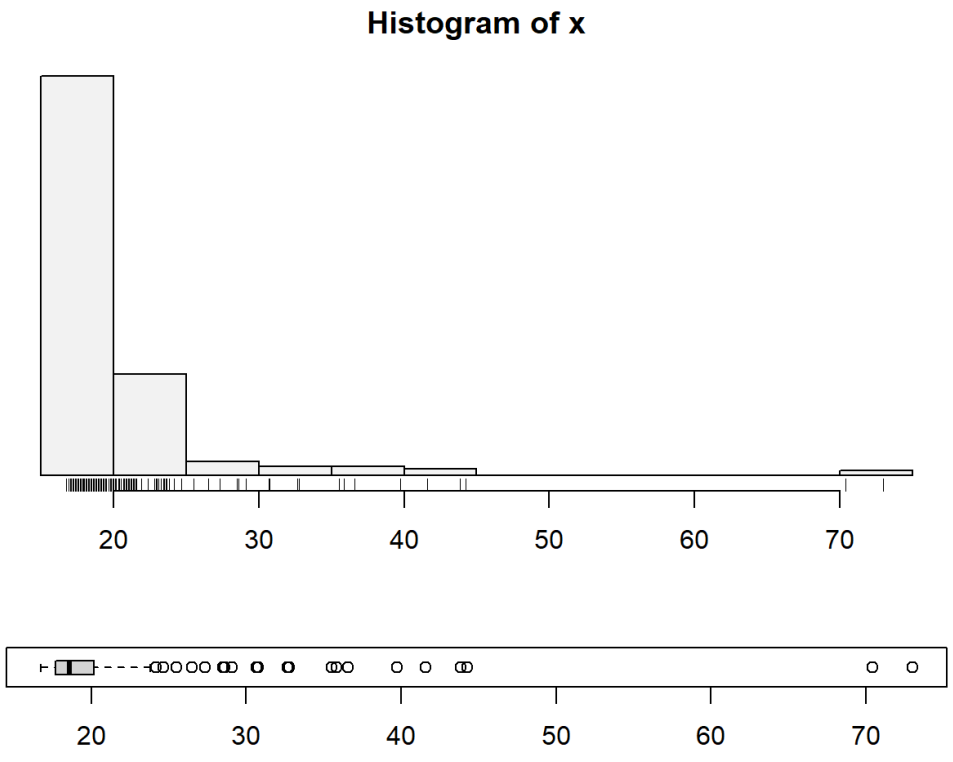


Boxplot allows us to check quickly for symmetry and outliers.

From this we can easily see that the ages of the students are with skewed distribution with a long tail.

We can also see the histogram and the boxplot together on one graphic

```
> library(UsingR)
Warning: package 'UsingR' was built under R version 4.0.3
Loading required package: HistData
Loading required package: Hmisc
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2

Attaching package: 'Hmisc'
The following objects are masked from 'package:base':

    format.pval, units

Attaching package: 'UsingR'
The following object is masked from 'package:survival':

    cancer
> simple.hist.and.boxplot(survey$Age)
```

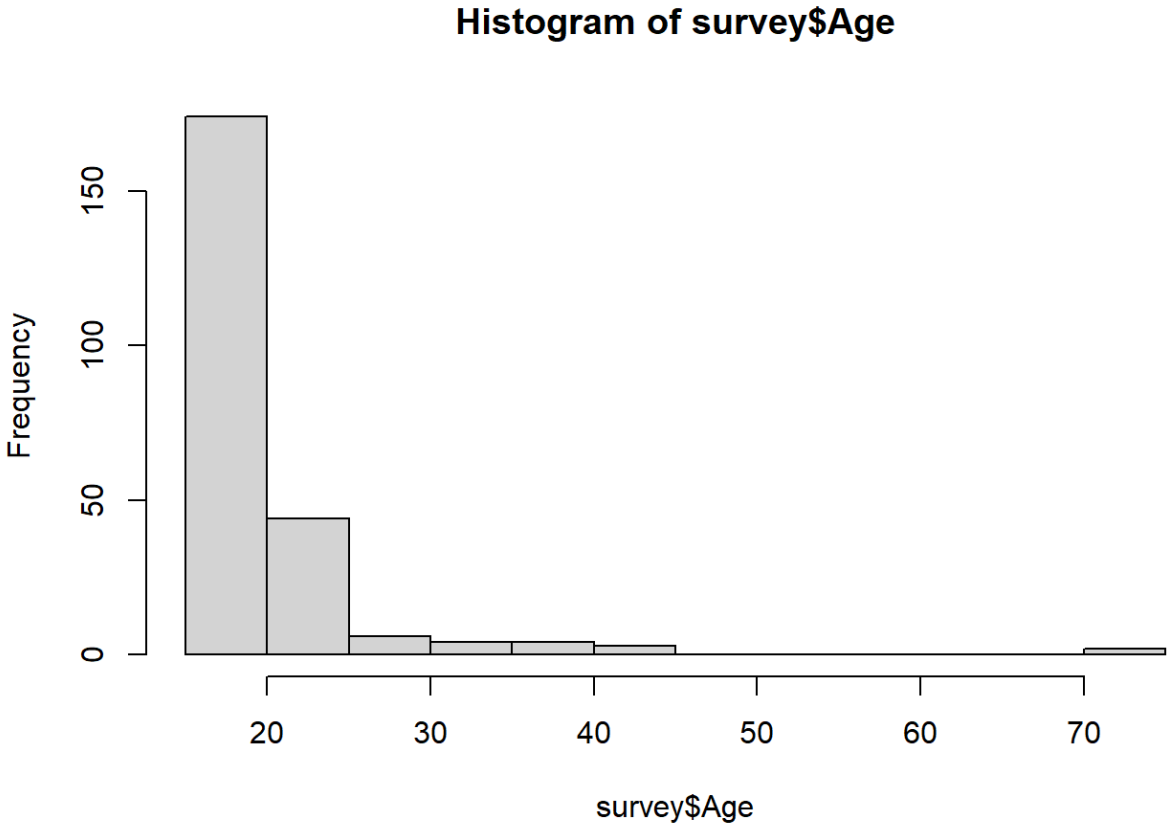


Frequency polygons

Frequency polygon represent the histogram information in a different way. Rather than draw a rectangle for each bin, polygon puts a point at the top of the rectangle and then connect these points with straight lines.

One way to generate the polygon in R is to use the attributes of the histogram.

```
> h <- hist(survey$Age)
```



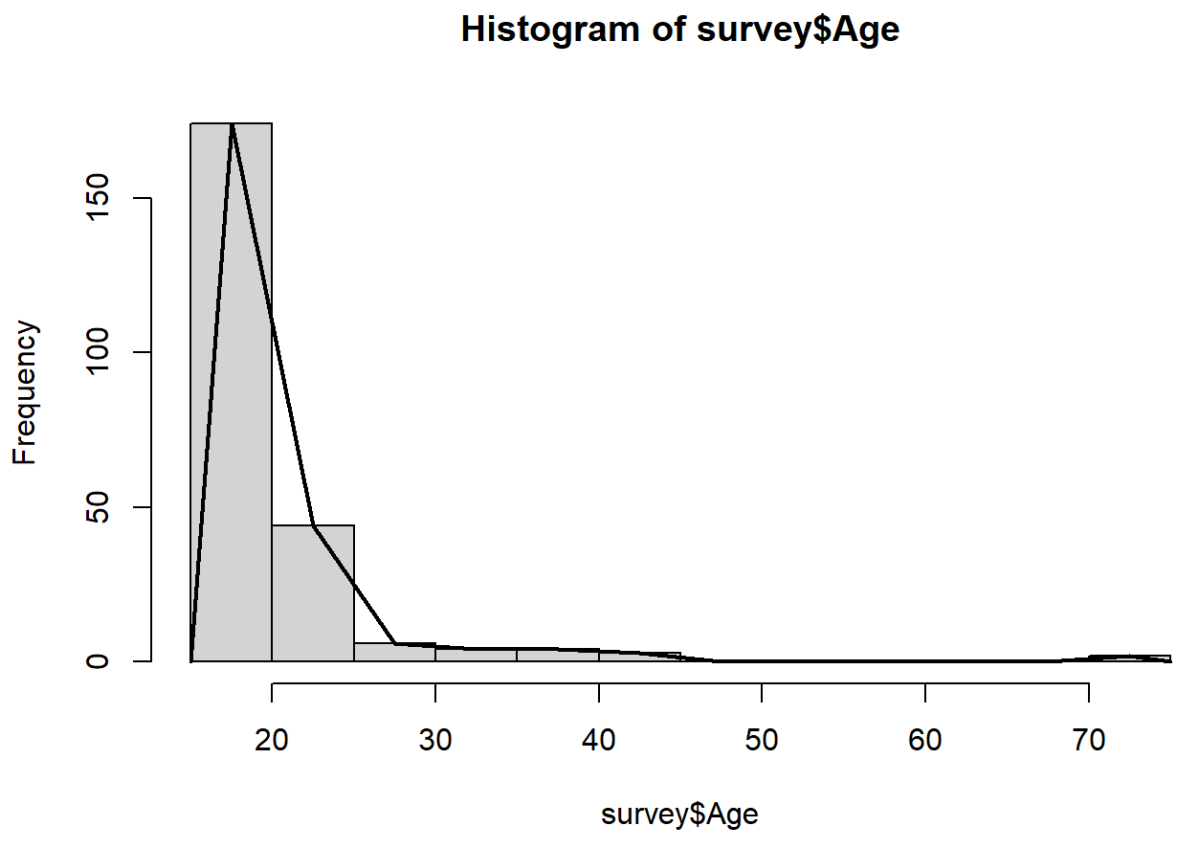

```
> attributes(h)
$names
[1] "breaks"   "counts"   "density"  "mids"     "xname"
"equidist"

$class
[1] "histogram"
```

Where `breaks` gives the histogram cells boundaries, `counts` gives the number observations in the bound, `mids` gives the histogram cell midpoints

```
> h$breaks
 [1] 15 20 25 30 35 40 45 50 55 60 65 70 75
> h$counts
 [1] 174  44   6   4   4   3   0   0   0   0   0   2
> h$mids
 [1] 17.5 22.5 27.5 32.5 37.5 42.5 47.5 52.5 57.5 62.5
67.5 72.5
```

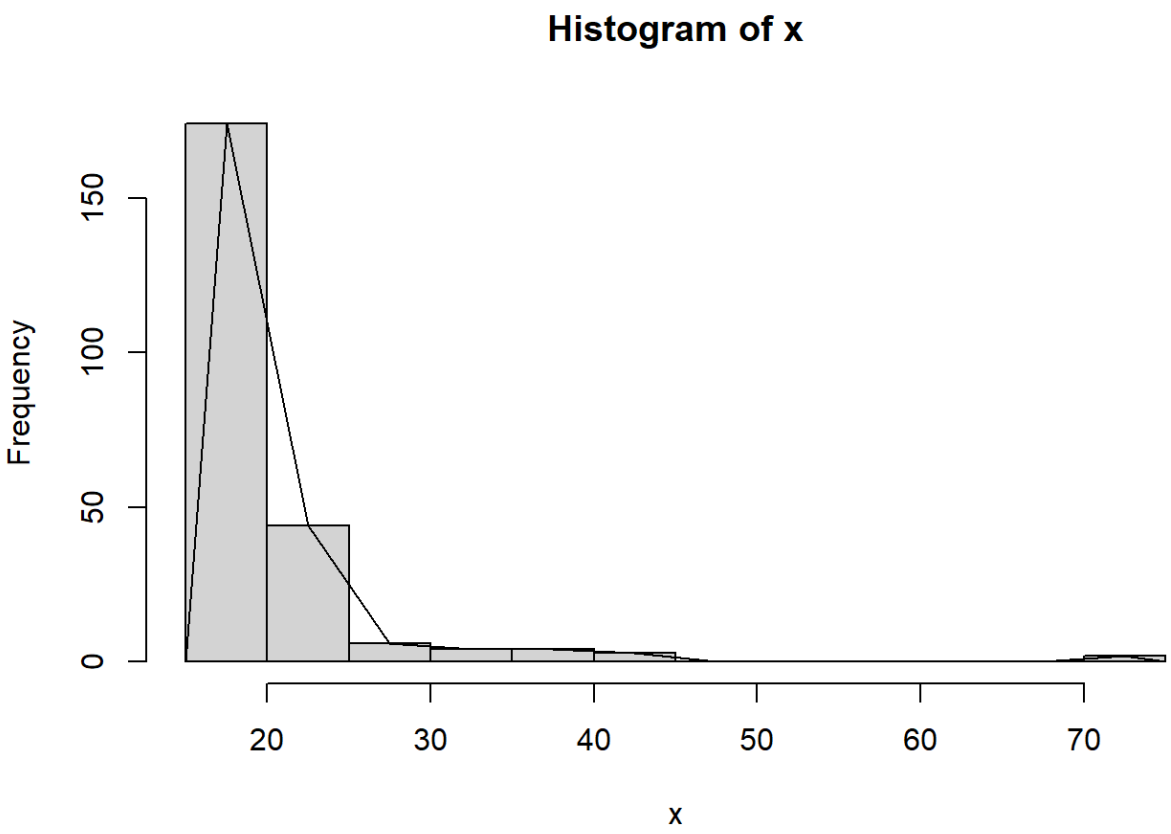
Using this data we can draw the polygon with the `line` function, giving its `x` and `y` coordinates

```
> hist(survey$Age)
> lines(x = c(min(h$breaks), h$mids, max(h$breaks)),
+       y = c(0, h$counts, 0),
+       type = "l",
+       lwd = 2)
```



Another way is to use the function `simple.freqpoly` from the package `UsingR`

```
> simple.freqpoly(survey$Age)
```



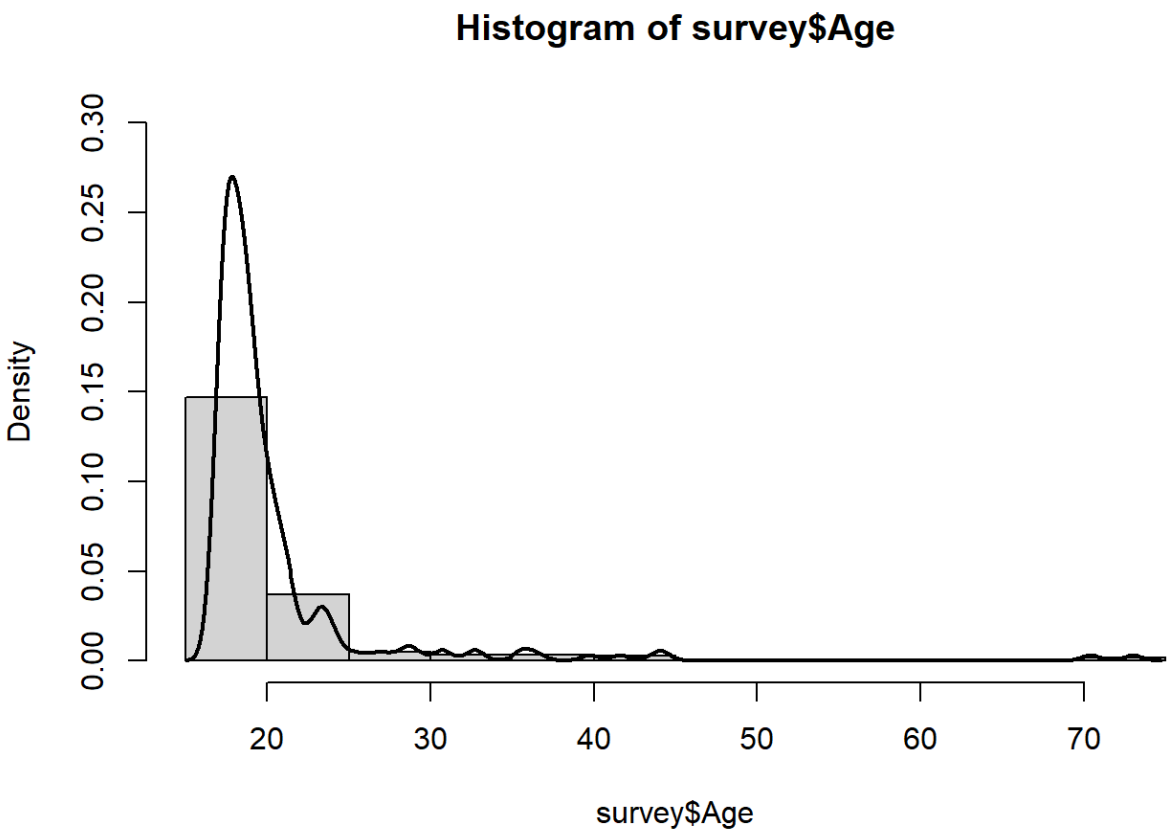
The idea of the polygon is to give you an overall idea of the data distribution, so you can tie it up with the probability density of the parent population

Densities

`density` gives more sophisticated presentation of the data distribution. The basic idea is for each point to take some kind of an average for the points nearby and based on this to give an estimate for the density.

```
> density(survey$Age)

Call:
    density.default(x = survey$Age)

Data: survey$Age (237 obs.);    Bandwidth 'bw' = 0.5625

       x                y            
 Min.   :15.06   Min.   :0.000000  
 1st Qu.:29.97   1st Qu.:0.000000  
 Median :44.88   Median :0.001214  
 Mean   :44.88   Mean   :0.016753  
```

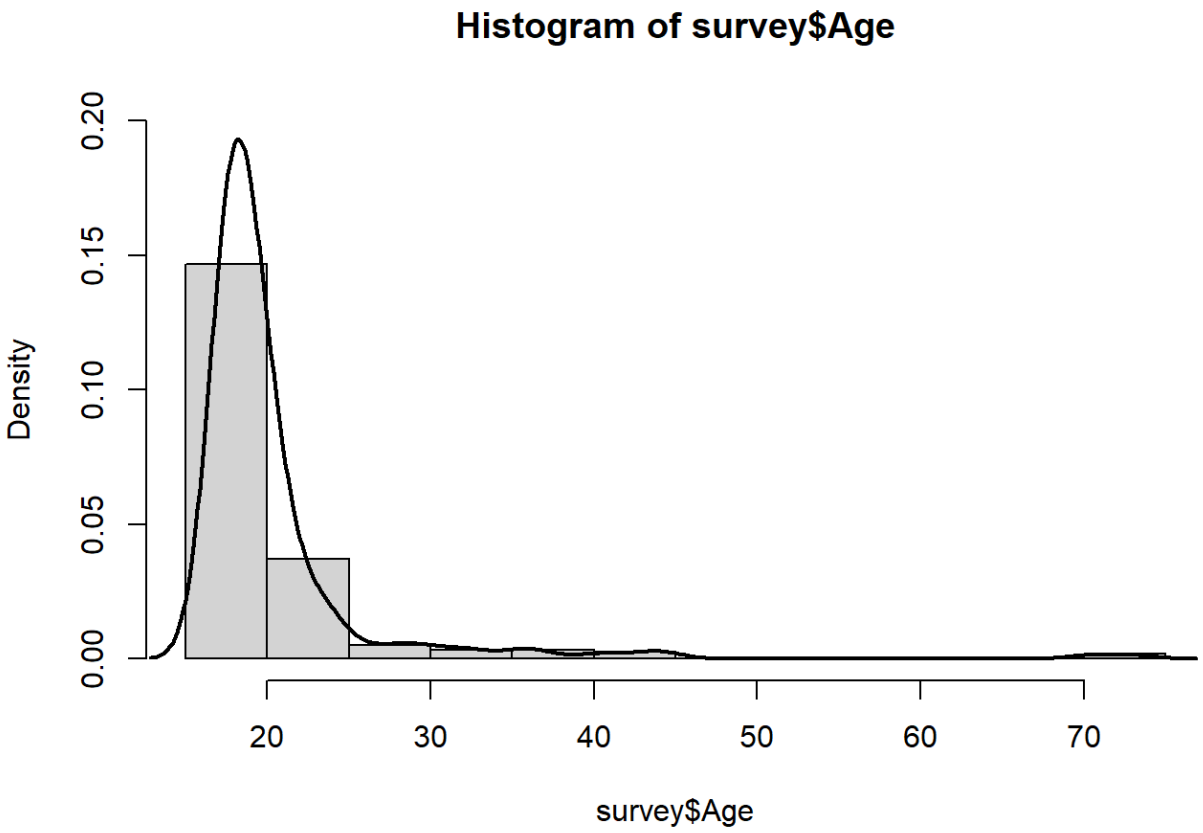
```
 3rd Qu.:59.78   3rd Qu.:0.004858
 Max.   :74.69   Max.   :0.270452
> hist(survey$Age, probability = TRUE, ylim = c(0, 0.3))
> lines(density(survey$Age), lwd = 2)
```



Here we can change the bandwidth `bw` of the interval from which we are taking the average of the points, by default it is `SJ`.

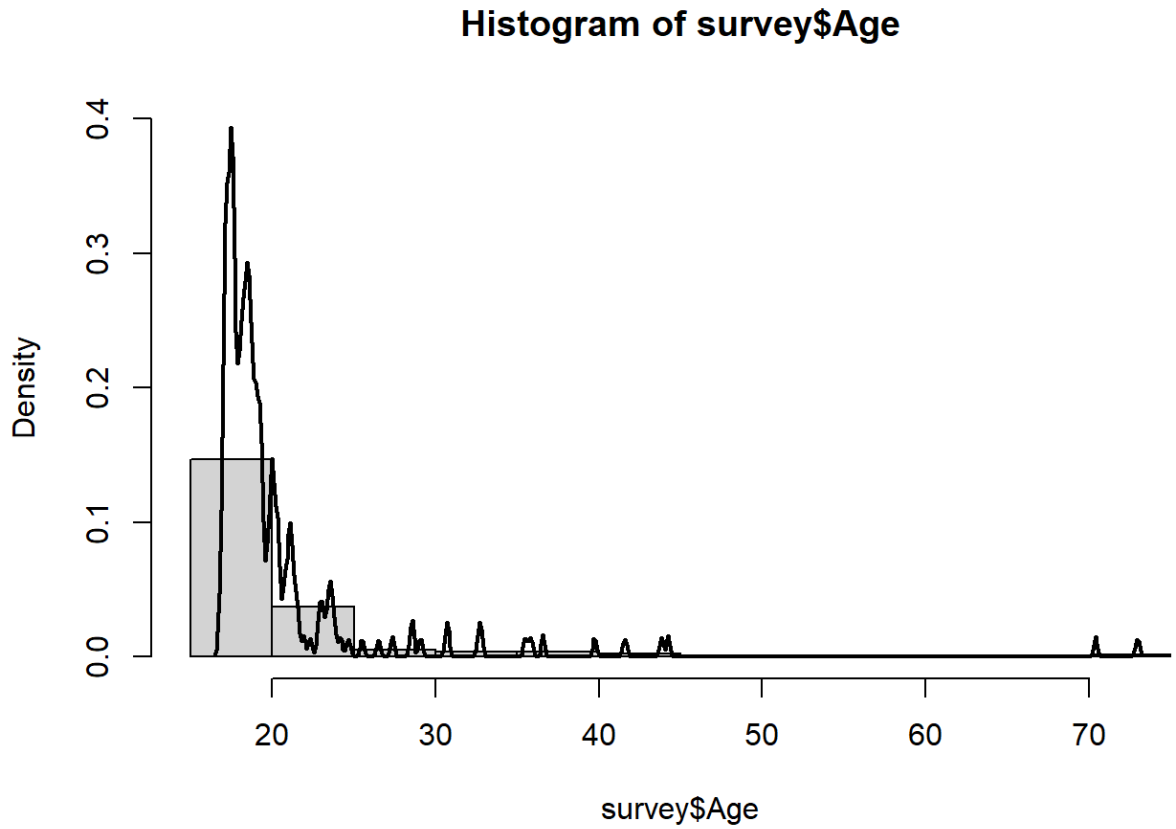
```
> density(survey$Age, bw = 1.3)

Call:
    density.default(x = survey$Age, bw = 1.3)

Data: survey$Age (237 obs.);    Bandwidth 'bw' = 1.3

       x                y
 Min.   :12.85   Min.   :0.000e+00
 1st Qu.:28.86   1st Qu.:8.100e-07
 Median :44.88   Median :1.581e-03
 Mean   :44.88   Mean   :1.560e-02
 3rd Qu.:60.89   3rd Qu.:4.986e-03
 Max.   :76.90   Max.   :1.934e-01
> hist(survey$Age, probability = TRUE, ylim = c(0, 0.2))
```

```
> lines(density(survey$Age, bw = 1.3), lwd = 2)
```

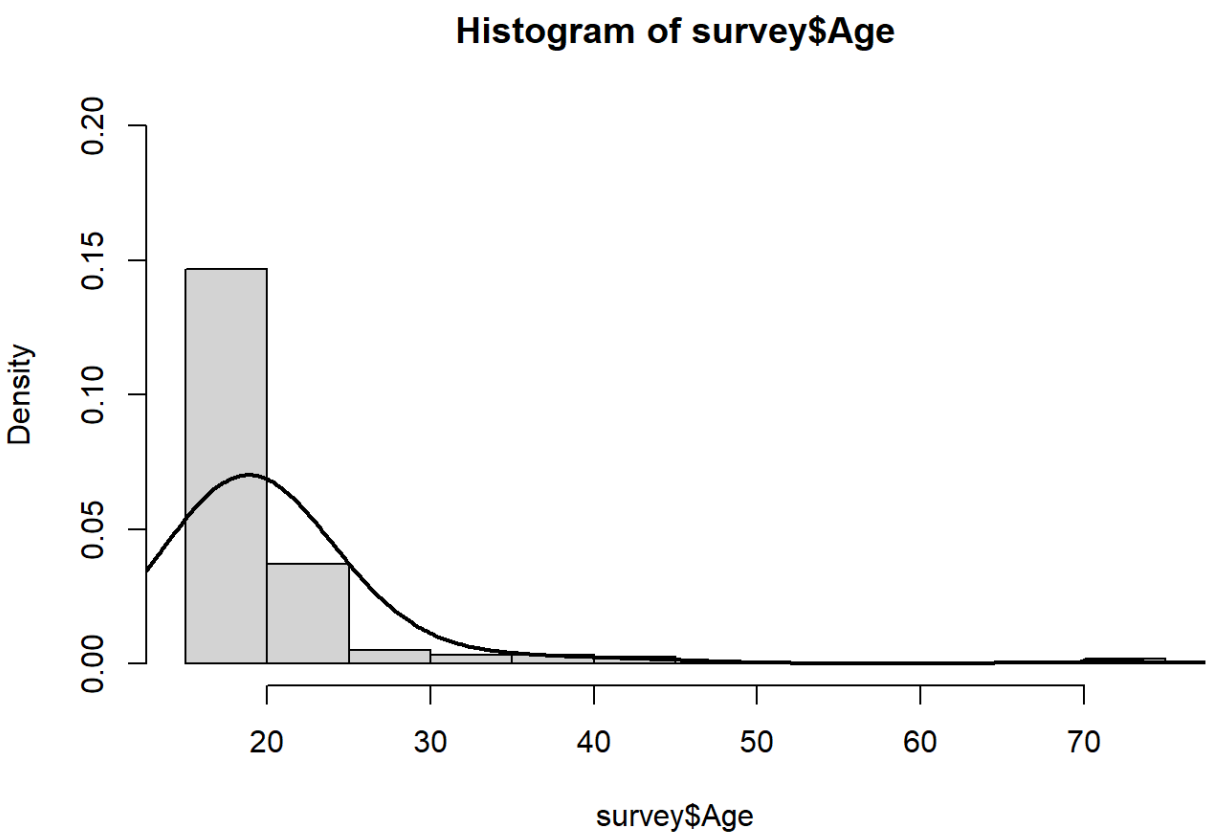


If the bandwidth is too small, the result is too jagged, and if it is too big the result is too smooth

```
> density(survey$Age, bw = 0.1)

Call:
    density.default(x = survey$Age, bw = 0.1)

Data: survey$Age (237 obs.);    Bandwidth 'bw' = 0.1

       x                y
 Min.   :16.45   Min.   :0.000000
 1st Qu.:30.66   1st Qu.:0.000000
 Median :44.88   Median :0.000000
 Mean   :44.88   Mean   :0.017571
 3rd Qu.:59.09   3rd Qu.:0.004739
 Max.   :73.30   Max.   :0.393992
> hist(survey$Age, probability = TRUE, ylim = c(0, 0.4))
> lines(density(survey$Age, bw = 0.1), lwd = 2)
```




```
> density(survey$Age, bw = 5)

Call:
    density.default(x = survey$Age, bw = 5)

Data: survey$Age (237 obs.);    Bandwidth 'bw' = 5

       x                y
 Min.   : 1.75   Min.   :4.490e-06
 1st Qu.:23.31   1st Qu.:2.491e-04
 Median :44.88   Median :9.966e-04
 Mean   :44.88   Mean   :1.158e-02
 3rd Qu.:66.44   3rd Qu.:1.056e-02
 Max.   :88.00   Max.   :7.021e-02
> hist(survey$Age, probability = TRUE, ylim = c(0, 0.2))
> lines(density(survey$Age, bw = 5), lwd = 2)
```


Histogram of survey$Age
Density
0.00
0.05
0.10
0.15
0.20
20
30
40
50
60
70
survey$Age