# AUA CS 108, Statistics, Fall 2019

## Lecture 05

Michael Poghosyan
YSU, AUA
michael@ysu.am, mpoghosyan@aua.am

04 Sep 2019

# Contents

# Last Lecture ReCap

- Define the **Empirical CDF**

# Last Lecture ReCap

- Define the **Empirical CDF**
- Why is it for?

# Last Lecture ReCap

- Define the **Empirical CDF**
- Why is it for?
- Define the **Frequency (Rel Frequency) Histogram**

# Last Lecture ReCap

- Define the **Empirical CDF**
- Why is it for?
- Define the **Frequency (Rel Frequency) Histogram**
- Define the **Density Histogram**

# Density Histograms

We assume that our 1D dataset $x_1, ..., x_n$ is numerical, coming from an either Discrete or a Continuous Variable.

# Density Histograms

We assume that our 1D dataset $x_1, ..., x_n$ is numerical, coming from an either Discrete or a Continuous Variable.

- We first take the interval $I$ to be either $[\min_i\{x_i\}, \max_i\{x_i\}]$ or any interval containing $[\min_i\{x_i\}, \max_i\{x_i\}]$;

# Density Histograms

We assume that our 1D dataset $x_1, ..., x_n$ is numerical, coming from an either Discrete or a Continuous Variable.

- We first take the interval $I$ to be either $[\min_i\{x_i\}, \max_i\{x_i\}]$ or any interval containing $[\min_i\{x_i\}, \max_i\{x_i\}]$;
- Then we take a finite partition of $I$: class intervals (bins) $I_1, I_2, ..., I_k$, i.e. $I_j$-s are disjoint, and their union is the interval $I$;

# Density Histograms

We assume that our 1D dataset $x_1, ..., x_n$ is numerical, coming from an either Discrete or a Continuous Variable.

- We first take the interval $I$ to be either $[\min_i\{x_i\}, \max_i\{x_i\}]$ or any interval containing $[\min_i\{x_i\}, \max_i\{x_i\}]$;
- Then we take a finite partition of $I$: class intervals (bins) $I_1, I_2, ..., I_k$, i.e. $I_j$-s are disjoint, and their union is the interval $I$;
- We calculate the number $n_j$ of datapoints $x_i$ lying in $I_j$:

$$n_j = \text{the number of data points in } I_j \qquad j = 0, 1, 2, ..., k.$$

# The Density or Normalized Relative Frequency Histogram

Now, we define the Density Histogram:

# The Density or Normalized Relative Frequency Histogram

Now, we define the Density Histogram:

**Definition:** The **Density Histogram** or the **Normalized Relative Frequency Histogram** of our Data $x_1, ..., x_n$ is the piecewise constant function

$$h_{dens}(x) = \frac{n_j}{n} \cdot \frac{1}{length(I_j)}, \qquad \forall x \in I_j.$$

# The Density or Normalized Relative Frequency Histogram

Now, we define the Density Histogram:

**Definition:** The **Density Histogram** or the **Normalized Relative Frequency Histogram** of our Data $x_1, ..., x_n$ is the piecewise constant function

$$h_{dens}(x) = \frac{n_j}{n} \cdot \frac{1}{length(I_j)}, \qquad \forall x \in I_j.$$

Here $length(I_j)$ is the length of the interval $I_j$. Also we define $h(x) = 0$, if $x \notin I$.

# Note

In the case (which is the mostly used one) when all intervals $I_j$ have the same length:

$$length(I_j) = h,$$

then

# Note

In the case (which is the mostly used one) when all intervals $I_j$ have the same length:

$$length(I_j) = h,$$

then

$$h_{dens}(x) = \frac{h_{relfreq}(x)}{h} = \frac{n_j}{n \cdot h}, \qquad \forall x \in I_j.$$

# Idea of the Density Histogram

The idea of dividing to the lenght of the corresponding interval, in the definition of the Density Histogram, is that in this case, the Total Area of all rectangles of our Histogram is 1.

# Idea of the Density Histogram

The idea of dividing to the lenght of the corresponding interval, in the definition of the Density Histogram, is that in this case, the Total Area of all rectangles of our Histogram is 1.

Recall that all PDF functions integrate to 1.

# Idea of the Density Histogram

The idea of dividing to the lenght of the corresponding interval, in the definition of the Density Histogram, is that in this case, the Total Area of all rectangles of our Histogram is 1.

Recall that all PDF functions integrate to 1. And the Density Histogram is approximating (estimating) the unknown PDF behind our Data!

# Example

To draw the Density Histogram, we will use the *freq=FALSE* parameter in the *hist* command.

# Example

To draw the Density Histogram, we will use the *freq=FALSE* parameter in the *hist* command.

We use here the *discoveries* Standard Dataset from **R**, which gives us the numbers of "great" inventions and scientific discoveries in each year from 1860 to 1959:

# Example

To draw the Density Histogram, we will use the *freq=FALSE* parameter in the *hist* command.

We use here the *discoveries* Standard Dataset from **R**, which gives us the numbers of "great" inventions and scientific discoveries in each year from 1860 to 1959:

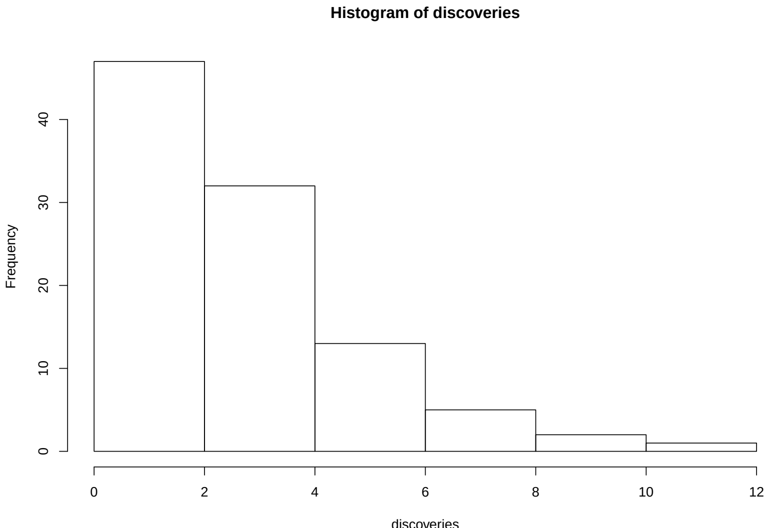
```
discoveries
```

```
## Time Series:
## Start = 1860
## End = 1959
## Frequency = 1
##   [1]  5  3  0  2  0  3  2  3  6  1  2  1  2  1  3  3
##  [26] 12  3 10  9  2  3  7  7  2  3  3  6  2  4  3  5
##  [51]  3  6  5  8  3  6  6  0  5  2  2  2  6  3  4  4
##  [76]  2  2  1  3  4  2  2  1  1  1  2  1  4  4  3  2
```

# Example

First, the Frequency Histogram:

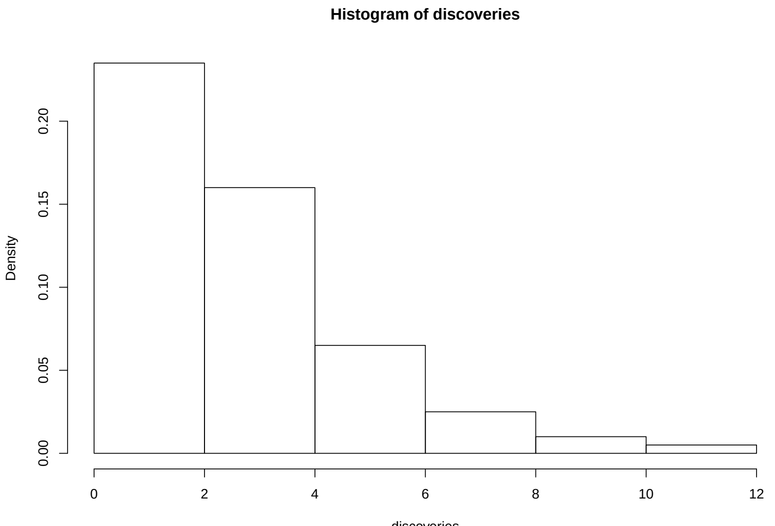
```r
hist(discoveries)
```



# Example

Now, the Density Histogram:

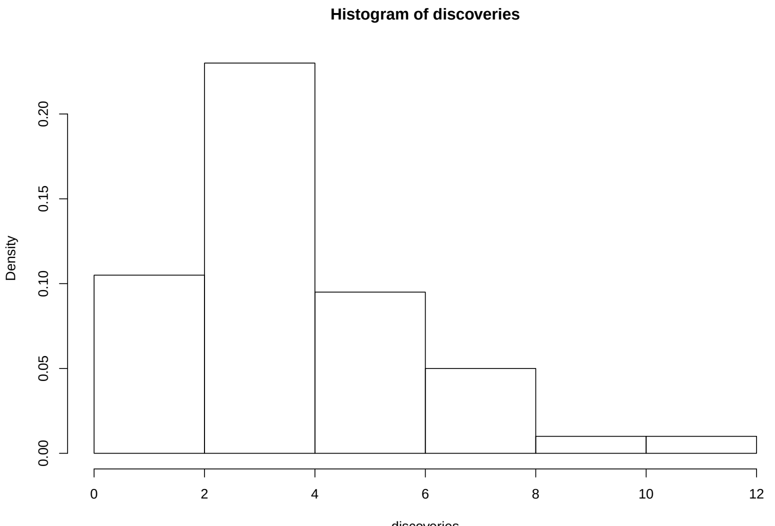
```
hist(discoveries, freq = FALSE)
```



# Example

Finally, the Density Histogram with the Bins left-endpoints included:

```r
hist(discoveries, freq = FALSE, right = FALSE)
```



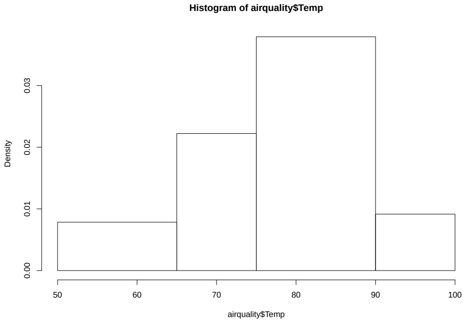
# Example

Now let us change the default bins for a Histogram.

# Example

Now let us change the default bins for a Histogram. We can use the following - first define the vector of our class interval (Bins) endpoints: (note that you need to cover all Datapoints!)

```
bins.endpoitns <- c(50, 65, 75, 90, 100)
hist(airquality$Temp, breaks = bins.endpoitns)
```



# Notes

- By default, if we give custom bins with non-equal lengths, **R** is plotting the Density Histogram!

# Notes

- By default, if we give custom bins with non-equal lengths, **R** is plotting the Density Histogram!
- You can give the *breaks* parameter either the vector of Bins' endpoints or the number of (equal-length) intervals

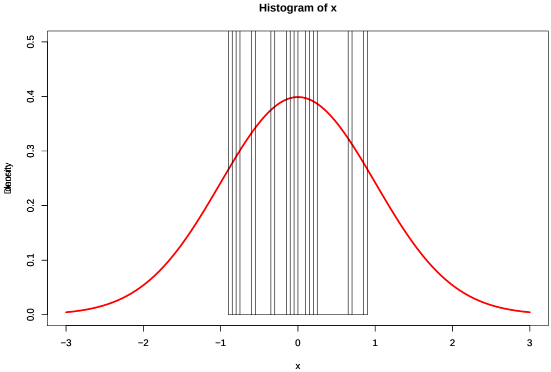
## Estimation of the PDF through the Density Histogram

As it was stated above, the Density Histogram is an approximation (estimate) of the PDF of the Data unknown Distribution. To check this, let us take a synthetic Dataset from the Distribution we know:

# Estimation of the PDF through the Density Histogram

As it was stated above, the Density Histogram is an approximation (estimate) of the PDF of the Data unknown Distribution. To check this, let us take a synthetic Dataset from the Distribution we know:

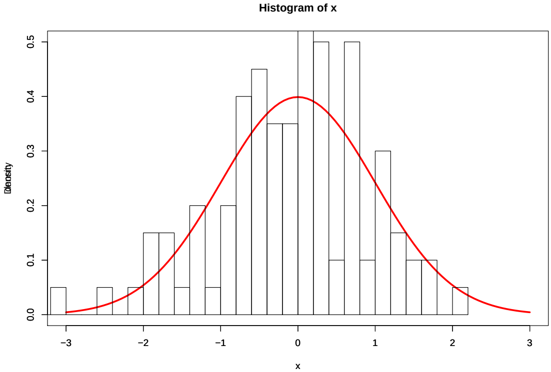
```r
plot(dnorm, lwd = 3, col= "red", xlim=c(-3,3), ylim=c(0,0.5))
x <- rnorm(10)
par(new = TRUE)
hist(x, breaks = 40, freq = FALSE, xlim=c(-3,3), ylim=c(0,0.5))
```



# Estimation of the PDF through the Density Histogram

As it was stated above, the Density Histogram is an approximation (estimate) of the PDF of the Data unknown Distribution. To check this, let us take a synthetic Dataset from the Distribution we know:

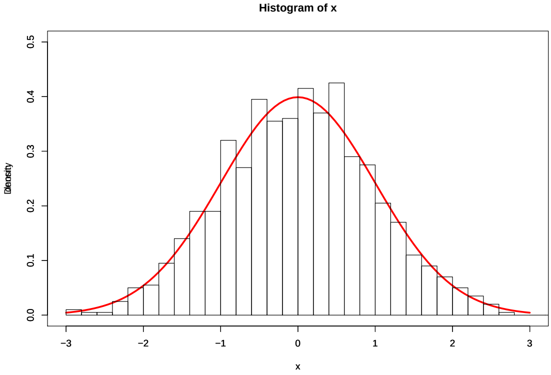
```
plot(dnorm, lwd = 3, col= "red", xlim=c(-3,3), ylim=c(0,0.5))
x <- rnorm(100)
par(new = TRUE)
hist(x, breaks = 40, freq = FALSE, xlim=c(-3,3), ylim=c(0,0.5))
```



# Estimation of the PDF through the Density Histogram

As it was stated above, the Density Histogram is an approximation (estimate) of the PDF of the Data unknown Distribution. To check this, let us take a synthetic Dataset from the Distribution we know:

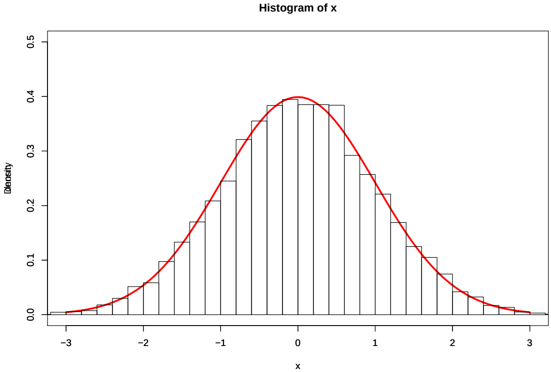
```
plot(dnorm, lwd = 3, col= "red", xlim=c(-3,3), ylim=c(0,0.5))
x <- rnorm(1000)
par(new = TRUE)
hist(x, breaks = 40, freq = FALSE, xlim=c(-3,3), ylim=c(0,0.5))
```



# Estimation of the PDF through the Density Histogram

As it was stated above, the Density Histogram is an approximation (estimate) of the PDF of the Data unknown Distribution. To check this, let us take a synthetic Dataset from the Distribution we know:

```
plot(dnorm, lwd = 3, col= "red", xlim=c(-3,3), ylim=c(0,0.5))
x <- rnorm(10000)
par(new = TRUE)
hist(x, breaks = 40, freq = FALSE, xlim=c(-3,3), ylim=c(0,0.5))
```



# Choosing Bin sizes correctly

It is important to choose the Bin sizes (lengths of the Bin, class, intervals) wisely. Otherwise you will skip some info or you will not get any valuable info.

# Choosing Bin sizes correctly

It is important to choose the Bin sizes (lengths of the Bin, class, intervals) wisely. Otherwise you will skip some info or you will not get any valuable info.

Let us use another **R** standard dataset to show the effect of the choice of the bin size: *precip*. This Dataset shows the average amount of precipitation (rainfall) in inches for each of 70 United States (and Puerto Rico) cities.

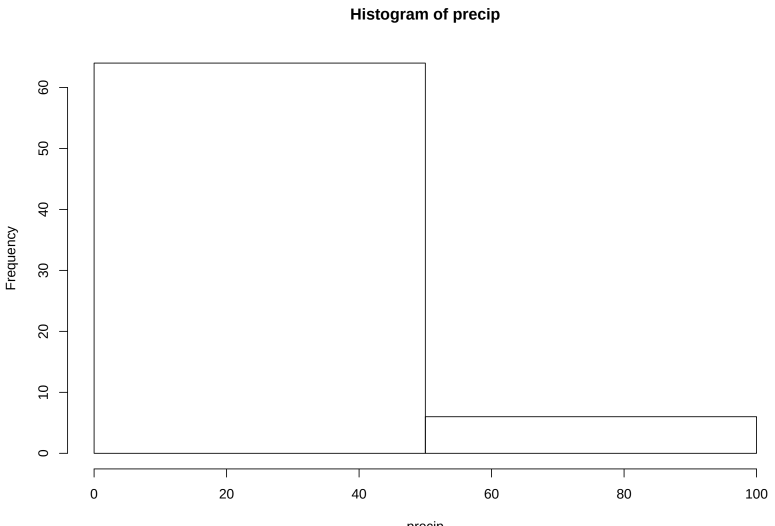
```
head(precip)
```

```
##       Mobile       Juneau      Phoenix Little Rock Los Ange
##         67.0         54.7          7.0        48.5        1
```

# Version 1, Small bins

Here, we just use 2 bins:

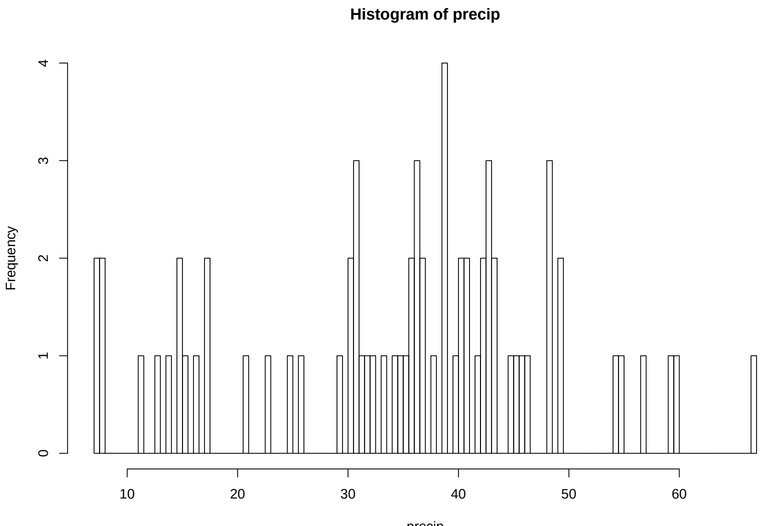
```r
hist(precip, breaks = 2)
```



# Version 2, large bins

Here, we use 200 bins:

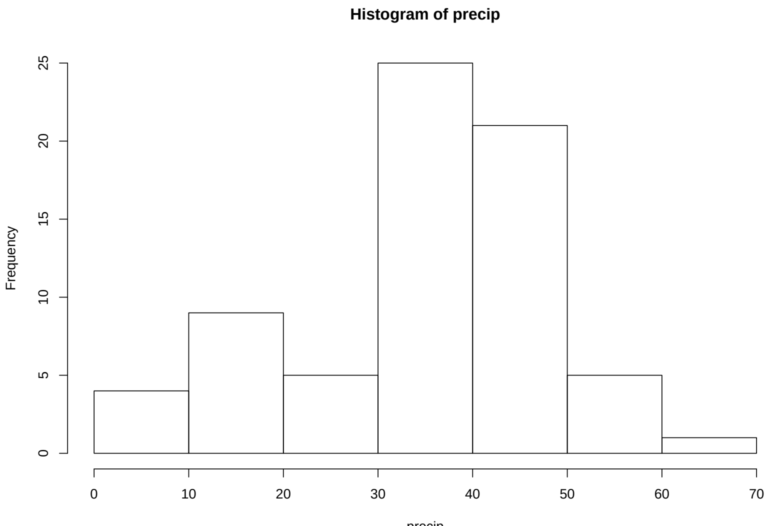
```
hist(precip, breaks = 200)
```



# Version 2, large bins

Now, the default:

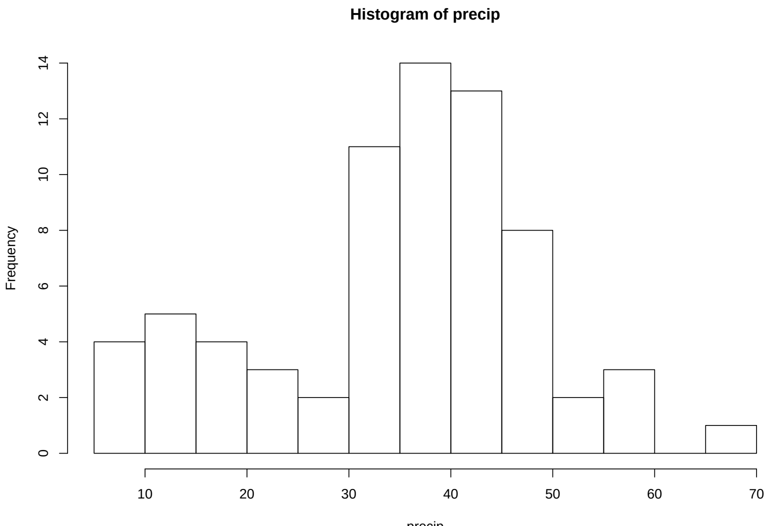
```
hist(precip)
```



# Version 3

Now, let us change to 20 bin intervals:

```
hist(precip, breaks = 20)
```



# Choosing the Bin Length

In fact, choosing the correct Bin width is not an easy job. See, for example, the Histogram Wiki page.

# Differences between the Barplot and Histogram

- Can you give some differences?

# Differences between the Barplot and Histogram

- Can you give some differences?

Here are some:

- *Barplot*'s rectangles widths are arbitrary, do not mean anything, rectangles are not adjacent; *Histogram*'s rectangles are adjacent, and the choice of the Bin widths is changing the graph

## Differences between the Barplot and Histogram

- Can you give some differences?

Here are some:

- *Barplot*'s rectangles widths are arbitrary, do not mean anything, rectangles are not adjacent; *Histogram*'s rectangles are adjacent, and the choice of the Bin widths is changing the graph
- *Barplot* is for a categorical or Discrete Data, *Histogram* is for both Discrete and Continuous

# Differences between the Barplot and Histogram

- Can you give some differences?

Here are some:

- *Barplot*'s rectangles widths are arbitrary, do not mean anything, rectangles are not adjacent; *Histogram*'s rectangles are adjacent, and the choice of the Bin widths is changing the graph
- *Barplot* is for a categorical or Discrete Data, *Histogram* is for both Discrete and Continuous
- We can exactly reconstruct the Dataset from the *Barplot*, but not the *Histogram*

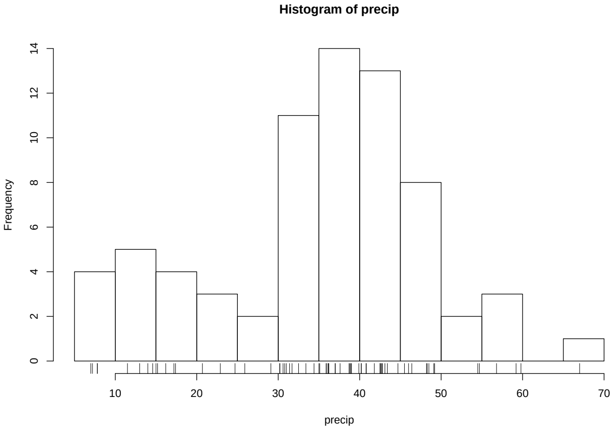
# Addition to the Histogram

Nice addition to your Histogram Plot is to add, in some way, the Datapoints:

# Addition to the Histogram

Nice addition to your Histogram Plot is to add, in some way, the Datapoints:

```r
hist(precip, breaks = 20)
rug(precip)
```



# What we can see from the Histogram

If we will not look at the Histogram as being an estimate for the unknown Distribution behind the Data, and if we will just try to get some info about our Dataset, Histogram is helping us to say if the Data:

# What we can see from the Histogram

If we will not look at the Histogram as being an estimate for the unknown Distribution behind the Data, and if we will just try to get some info about our Dataset, Histogram is helping us to say if the Data:

- is symmetric about some point or is skewed to the left or right

# What we can see from the Histogram

If we will not look at the Histogram as being an estimate for the unknown Distribution behind the Data, and if we will just try to get some info about our Dataset, Histogram is helping us to say if the Data:

- is symmetric about some point or is skewed to the left or right
- is spread out or concentrated at some point

# What we can see from the Histogram

If we will not look at the Histogram as being an estimate for the unknown Distribution behind the Data, and if we will just try to get some info about our Dataset, Histogram is helping us to say if the Data:

- is symmetric about some point or is skewed to the left or right
- is spread out or concentrated at some point
- has some gaps

# What we can see from the Histogram

If we will not look at the Histogram as being an estimate for the unknown Distribution behind the Data, and if we will just try to get some info about our Dataset, Histogram is helping us to say if the Data:

- is symmetric about some point or is skewed to the left or right
- is spread out or concentrated at some point
- has some gaps
- has values far apart from others, has outliers (anomalies)

# What we can see from the Histogram

If we will not look at the Histogram as being an estimate for the unknown Distribution behind the Data, and if we will just try to get some info about our Dataset, Histogram is helping us to say if the Data:

- is symmetric about some point or is skewed to the left or right
- is spread out or concentrated at some point
- has some gaps
- has values far apart from others, has outliers (anomalies)
- is unimodal, bimodal or multimodal

# KDE

Another estimate for the unknown Distribution PDF is the **Kernel Density Estimator**, KDE.

# KDE

Another estimate for the unknown Distribution PDF is the **Kernel Density Estimator**, KDE. It is, in some sense, the smoothed version of the Histogram: Histogram is a piecewise-constant function, with jumps, so it is not a smooth function.

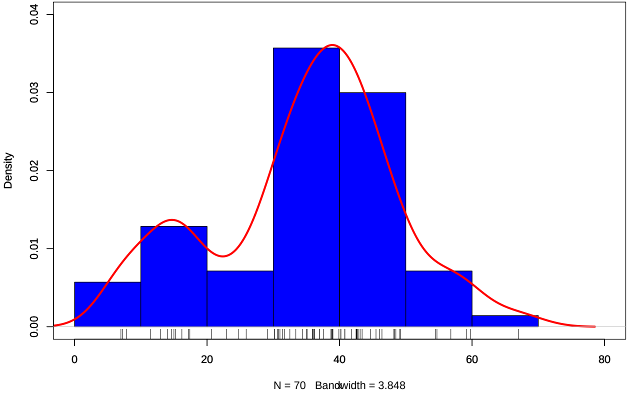
# KDE

Another estimate for the unknown Distribution PDF is the **Kernel Density Estimator**, KDE. It is, in some sense, the smoothed version of the Histogram: Histogram is a piecewise-constant function, with jumps, so it is not a smooth function.

You will find the Definition of the KDE in the Lecture Notes (and in different books), and here I will give the **R** code to construct the KDE:

# KDE Example

```
x <- precip; d <- density(x)
hist(x, freq = FALSE, xlim = c(0, 80), ylim = c(0,0.04),
     col = "blue", main = "")
rug(x); par(new = TRUE)
plot(d, lwd = 3, col = "red", xlim = c(0,80), ylim = c(0,0.04),
     main = "")
```



# Stem-n-Leaf Plot

Another method to visualize a (not-so-large) 1D Dataset is to give the Stem-and-Leaf plot:

Assume we have a 1D Dataset $x_1, x_2, ..., x_n$. We represent each number $x_k$ in the form

$$Stem \mid Leaf$$

# Stem-n-Leaf Plot

Another method to visualize a (not-so-large) 1D Dataset is to give the Stem-and-Leaf plot:

Assume we have a 1D Dataset $x_1, x_2, ..., x_n$. We represent each number $x_k$ in the form

$$Stem \mid Leaf$$

The *Leaf* need to consist only of 1 digit. The rest is in Stem.

# Stem-n-Leaf Plot

Another method to visualize a (not-so-large) 1D Dataset is to give the Stem-and-Leaf plot:

Assume we have a 1D Dataset $x_1, x_2, ..., x_n$. We represent each number $x_k$ in the form

$$Stem \mid Leaf$$

The *Leaf* need to consist only of 1 digit. The rest is in Stem. Sometimes, we do a rounding before making the S-n-L Plot, but, for simplicity, let's assume we are not doing any roundings.

# Example, S-n-L Plot

**Example:** Assume our Dataset is:

$$x : 14, 23, 5, 16, 32, 22$$

# Example, S-n-L Plot

**Example:** Assume our Dataset is:

$$x : 14, 23, 5, 16, 32, 22$$

Now, for 14, the Leaf is the last digit, 4, and the rest is the Stem, i.e., the Stem is 1. So we represent 14 as

$$1 \mid 4$$

# Example, S-n-L Plot

**Example:** Assume our Dataset is:

$$x : 14, 23, 5, 16, 32, 22$$

Now, for 14, the Leaf is the last digit, 4, and the rest is the Stem, i.e., the Stem is 1. So we represent 14 as

$$1 \mid 4$$

Similarly, 23 will give

$$2 \mid 3$$

and 5 will give

# Example, S-n-L Plot

**Example:** Assume our Dataset is:

$$x : 14, 23, 5, 16, 32, 22$$

Now, for 14, the Leaf is the last digit, 4, and the rest is the Stem, i.e., the Stem is 1. So we represent 14 as

$$1 \mid 4$$

Similarly, 23 will give

$$2 \mid 3$$

and 5 will give

$$0 \mid 5$$

# Example, S-n-L Plot

Next, 16 will be

$$1 \mid 6$$

# Example, S-n-L Plot

Next, 16 will be

$$1 \mid 6$$

and we combine this with the S-n-L representation of 14 (because they both starts by 1) to write

$$1 \mid 46$$

# Example, S-n-L Plot

Next, 16 will be

$$1 \mid 6$$

and we combine this with the S-n-L representation of 14 (because they both starts by 1) to write

$$1 \mid 46$$

Finally, our Dataset's S-n-L Plot will be

```r
x <- c(14, 23, 5, 16, 32, 22)
stem(x)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   0 | 5
##   1 | 46
##   2 | 23
##   3 | 2
```

# Notes

- Sometimes **R** will do some roundings before S-n-L Plotting

# Notes

- Sometimes **R** will do some roundings before S-n-L Plotting
- Usually, Stems are ordered, and Leafs are sorted in the increasing order (ordered SnL Plot)

# Notes

- Sometimes **R** will do some roundings before S-n-L Plotting
- Usually, Stems are ordered, and Leafs are sorted in the increasing order (ordered SnL Plot)
- The top row, the explanation about the position of |, is the **key**, is to recover the dataset.

# Example, SnL Plot

Here is another example: we use again the *airquality* Dataset, but now, the *Wind* Variable:

```
x <- airquality$Wind
stem(x)
```

```
##
##   The decimal point is at the |
##
##    1 | 7
##    2 | 38
##    3 | 4
##    4 | 016666
##    5 | 111777
##    6 | 33333333999999999
##    7 | 4444444444
##    8 | 0000000000066666666
##    9 | 2222222277777777777
##   10 | 3333333333399999999
##   11 | 555555555555555
##   12 | 0000666
##   13 | 2288888
##   14 | 33333399999999
##   15 | 555
##   16 | 1666
##   17 |
##   18 | 4
##   19 |
##   20 | 17
```

# Example, SnL Plot

Let's draw the Histogram of the same Dataset:

```
x <- airquality$Wind
hist(x, breaks = 15)
```