EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

Date	Rabbit late by
Jan 1, 2016	40 mins
Jan 2, 2016	30 mins
Jan 3, 2016	20 mins
Jan 4, 2016	45 mins
Jan 5, 2016	60 mins
Jan 6, 2016	120 mins
Jan 7, 2016	35 mins
Jan 8, 2016	40 mins
Jan 9, 2016	50 mins
Jan 10, 2016	55 mins
Jan 11, 2016	53 mins
Jan 12, 2016	22 mins
Jan 13, 2016	27 mins
Jan 14, 2016	48 mins
Jan 15, 2016	62 mins
Jan 16, 2016	33 mins
Jan 17, 2016	35 mins
Jan 18, 2016	40 mins
Jan 19, 2016	45 mins

FIRST WE NEED A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
```

THE C() FUNCTION COMBINES ALL THESE VALUES INTO A VARIABLE OF TYPE VECTOR

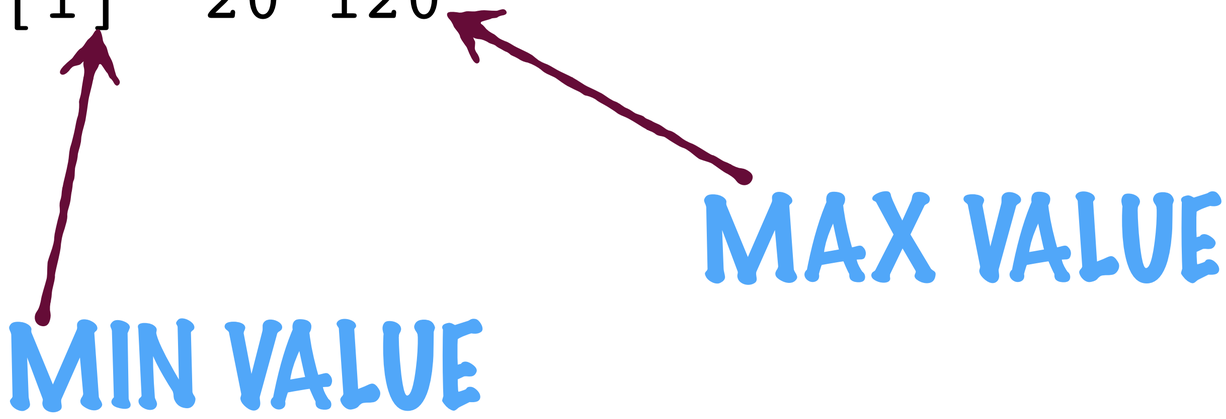
VECTORS ARE SPECIAL DATA STRUCTURES IN R, WE'LL LEARN A LOT MORE ABOUT THEM LATER

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
```

A VARIABLE THAT HOLDS THE DATA

LET'S FIND THE RANGE OF THE VALUES

```
range(rabbitLateness)
[1]   20 120
```



A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)

range(rabbitLateness)
[1]  20 120
```

LET'S CREATE A FREQUENCY TABLE

FIRST CREATE INTERVALS
FROM THE VALUES

20-29 mins
30-39 mins
40-49 mins
50-59 mins
60-69 mins
70-79 mins
80-89 mins
90-99 mins
100-109 mins
110 -120 mins

EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
range(rabbitLateness)
[1]  20 120
```

LET'S CREATE A FREQUENCY TABLE

```
bins <- seq(20,130,by=10)
bins
```

```
bins 20  30  40  50  60  70  80  90 100 110 120 130
```

WE CREATE A SEQUENCE OF VALUES TO REPRESENT THE START OF EACH INTERVAL

20-29 mins
30-39 mins
40-49 mins
50-59 mins
60-69 mins
70-79 mins
80-89 mins
90-99 mins
100-109 mins
110 -120 mins

EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
range(rabbitLateness)
[1]  20 120
```

LET'S CREATE A FREQUENCY TABLE

```
bins <- seq(20,130,by=10)
bins
[1]  20  30  40  50  60  70  80  90 100 110 120 130
intervals <- cut(rabbitLateness,bins,right=FALSE)
```

20-29 mins
30-39 mins
40-49 mins
50-59 mins
60-69 mins
70-79 mins
80-89 mins
90-99 mins
100-109 mins
110 -120 mins

THE CUT() FUNCTION WILL USE THIS SEQUENCE TO CREATE THE INTERVALS WE WANT

EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
range(rabbitLateness)
[1]  20 120
```

LET'S CREATE A FREQUENCY TABLE

```
bins <- seq(20,130,by=10)
bins
[1]  20  30  40  50  60  70  80  90 100 110 120 130
intervals <- cut(rabbitLateness,bins,right=FALSE)
```

20-29 mins
30-39 mins
40-49 mins
50-59 mins
60-69 mins
70-79 mins
80-89 mins
90-99 mins
100-109 mins
110 -120 mins

FIRST IT TAKES THE SEQUENCE AND CREATES INTERVALS

EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
range(rabbitLateness)
[1]  20 120
```

LET'S CREATE A FREQUENCY TABLE

```
bins <- seq(20,130,by=10)
bins
 [1]  20  30  40  50  60  70  80  90 100 110 120 130
intervals <- cut(rabbitLateness,bins,right=FALSE)
intervals
 [1] [40,50)   [30,40)   [20,30)   [,50)   [60,70)   [120,130) [30,40)
 [8] [40,50)   [50,60)   [50,60)   [50,60)   [20,30)   [20,30)   [40,50)
[15] [60,70)   [30,40)   [30,40)   [40,50)   [40,50)
11 Levels: [20,30) [30,40) [40,50) [50,60) [60,70) [70,80) ... [120,130)
```

20-29 mins
30-39 mins
40-49 mins
50-59 mins
60-69 mins
70-79 mins
80-89 mins
90-99 mins
100-109 mins
110 -120 mins

THEN IT ASSIGNS EACH ELEMENT TO THE CORRESPONDING INTERVAL

EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
range(rabbitLateness)
[1]  20 120
```

LET'S CREATE A FREQUENCY TABLE

```
bins <- seq(20,130,by=10)
intervals <- cut(rabbitLateness,bins,right=FALSE)

table(intervals)
```

NEXT, COUNT HOW OFTEN EACH INTERVAL OCCURS

20-29 mins	3
30-39 mins	4
40-49 mins	6
50-59 mins	3
60-69 mins	2
70-79 mins	0
80-89 mins	0
90-99 mins	0
100-109 mins	0
110 -120 mins	0
	1

A VARIABLE THAT HOLDS THE DATA

```
rabbitLateness <- c(40, 30, 20, 45, 60, 120, 35, 40,
                    50, 55, 53, 22, 27, 48, 62, 33, 35, 40 ,45)
range(rabbitLateness)
[1]  20 120
```

LET'S CREATE A FREQUENCY TABLE

```
bins <- seq(20,130,by=10)
intervals <- cut(rabbitLateness,bins,right=FALSE)
```

```
table(intervals)
intervals
 [20,30)  [30,40)  [40,50)  [50,60)  [60,70)  [70,80)  [80,90)
       3        4        6        3        2        0        0
[90,100) [100,110) [110,120) [120,130)
       0         0         0         1
```

20-29 mins	3
30-39 mins	4
40-49 mins	6
50-59 mins	3
60-69 mins	2
70-79 mins	0
80-89 mins	0
90-99 mins	0
100-109 mins	0
110 -120 mins	0
	1

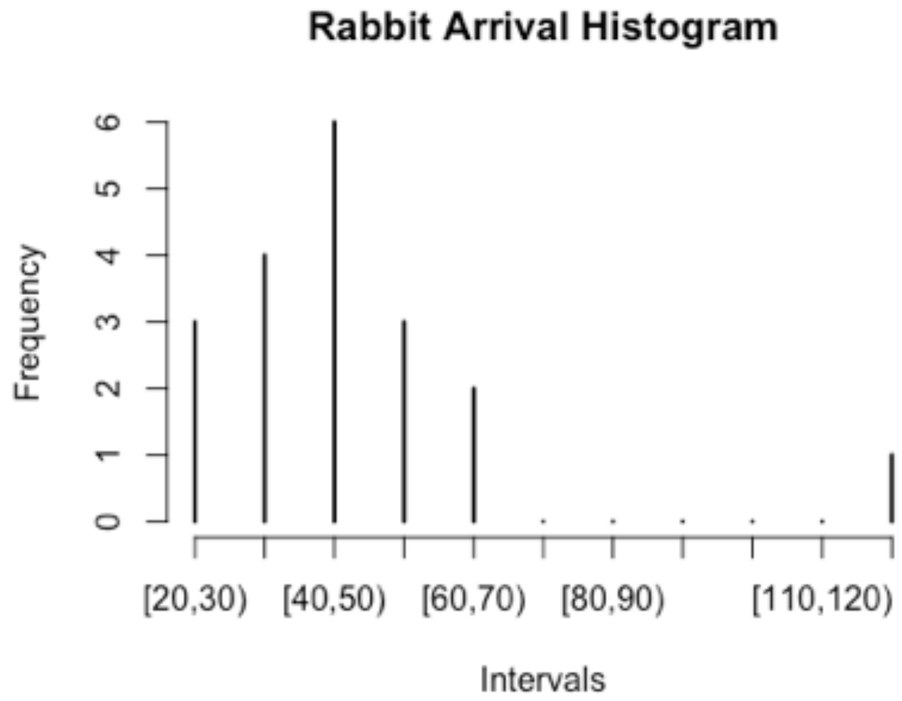
EXAMPLE 1: RANGE AND FREQUENCY TABLE IN R

EXAMPLE 2: DRAWING A HISTOGRAM

LET'S PLOT A HISTOGRAM

```
plot(table(intervals), type ="h", main = "Rabbit Arrival Histogram", xlab =
      "Intervals", ylab = "Frequency" )
```

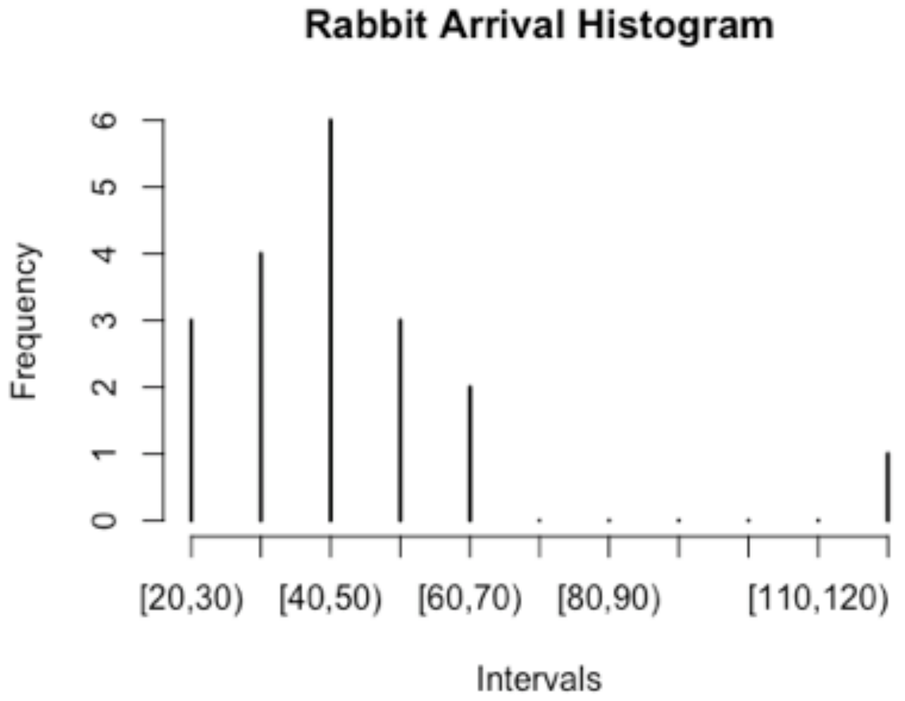
THE PLOT FUNCTION CAN TAKE A FREQUENCY TABLE AS INPUT



LET'S PLOT A HISTOGRAM

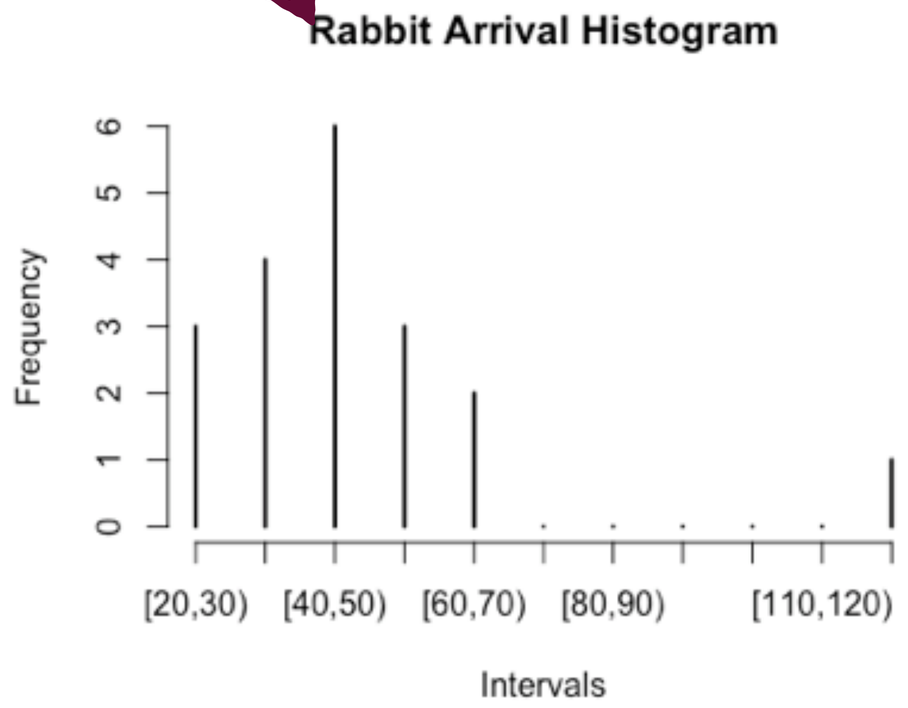
```
plot(table(intervals), type ="h", main = "Rabbit Arrival Histogram", xlab =
     "Intervals", ylab = "Frequency" )
```

THIS SPECIFIES THAT THE PLOT IS A HISTOGRAM



LET'S PLOT A HISTOGRAM

```
plot(table(intervals), type ="h", main = "Rabbit Arrival Histogram", xlab =
     "Intervals", ylab = "Frequency" )
```

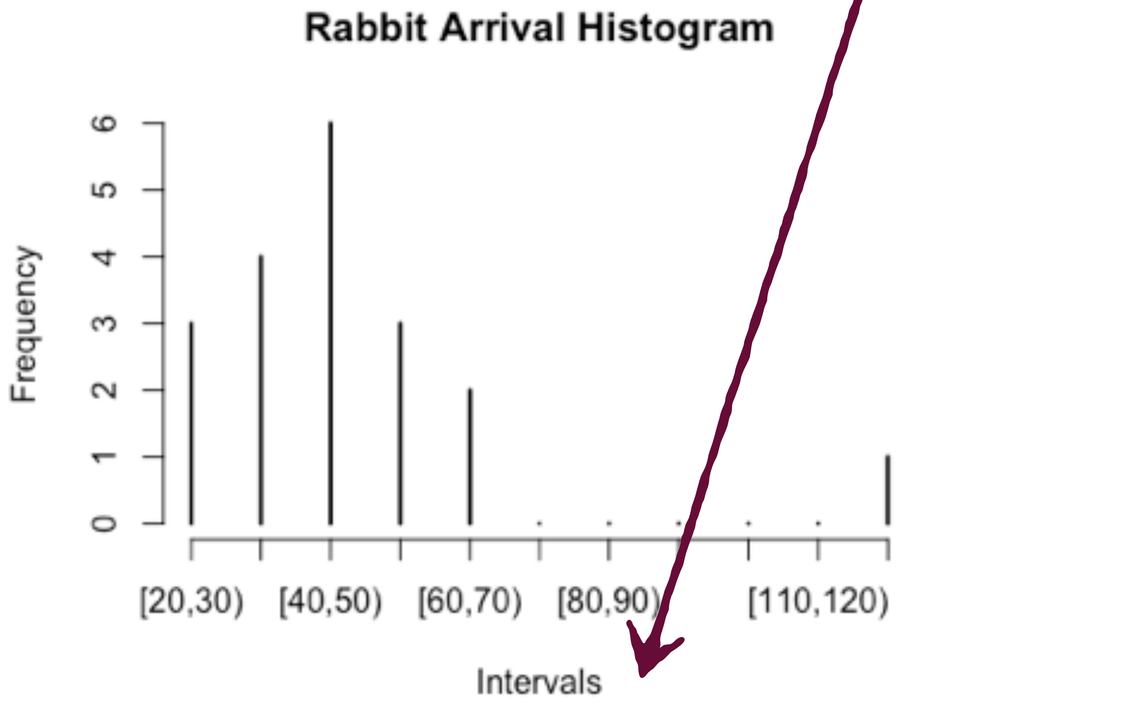


THE TITLE OF THE PLOT

LET'S PLOT A HISTOGRAM

```
plot(table(intervals), type ="h", main = "Rabbit Arrival Histogram", xlab =
     "Intervals", ylab = "Frequency" )
```

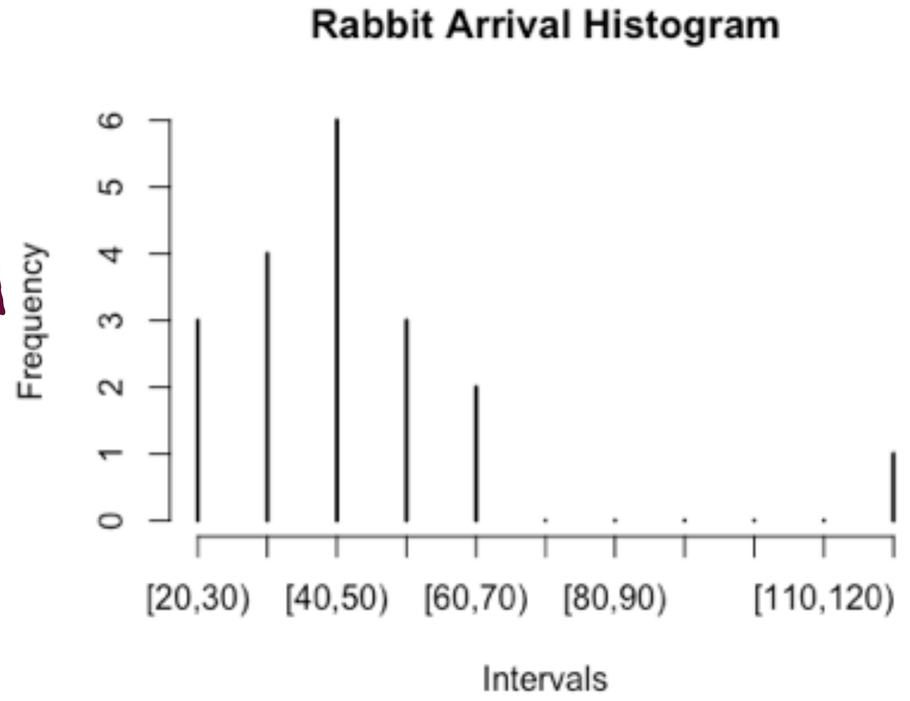
THE X-AXIS LABEL



LET'S PLOT A HISTOGRAM

```
plot(table(intervals), type ="h", main = "Rabbit Arrival Histogram", xlab =
     "Intervals", ylab = "Frequency" )
```

THE Y-AXIS LABEL



LET'S PLOT A HISTOGRAM

```
plot(table(intervals), type ="h", main = "Rabbit Arrival Histogram", xlab =
     "Intervals", ylab = "Frequency" )
```

FIRST WE CREATED INTERVALS FROM THE VALUES

THEN WE ASSIGNED EACH ELEMENT TO THE CORRESPONDING INTERVAL

NEXT, WE COUNTED HOW OFTEN EACH INTERVAL OCCURS

LAST, WE PLOTTED THE COUNTS

LET'S PLOT A HISTOGRAM

```
plot(table(intervals), type ="h", main = "Rabbit Arrival Histogram", xlab =
     "Intervals", ylab = "Frequency" )
```

DO ALL OF THIS IN ONE STEP

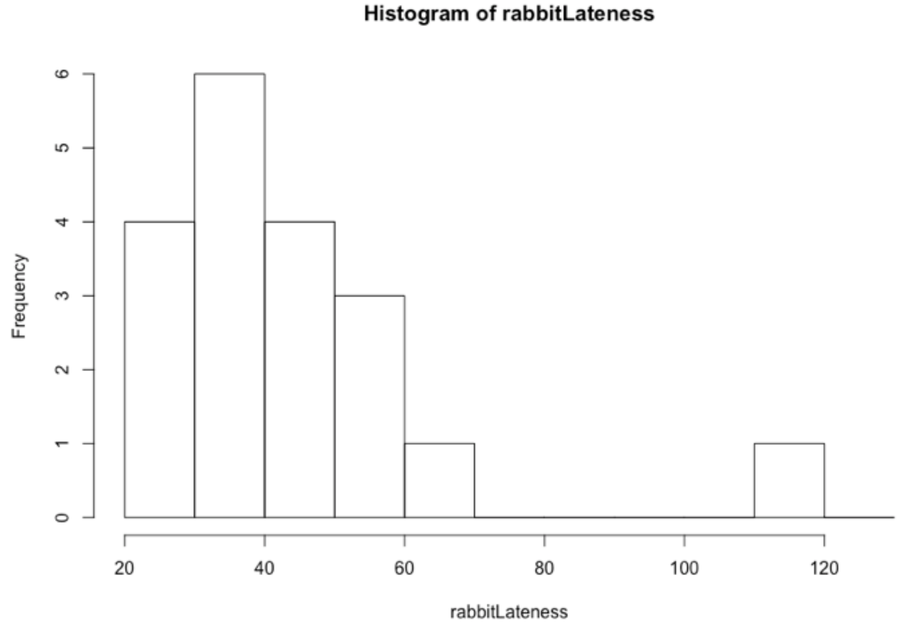
FIRST WE CREATED INTERVALS FROM THE VALUES

THEN WE ASSIGNED EACH ELEMENT TO THE CORRESPONDING INTERVAL

NEXT, WE COUNTED HOW OFTEN EACH INTERVAL OCCURS

LAST, WE PLOTTED THE COUNTS

```
hist(rabbitLateness,breaks = bins)
```



EXAMPLE 3: FINDING THE MEAN, MEDIAN AND MODE

LET'S FIND THE MEAN, MEDIAN AND MODE

```
mean(rabbitLateness)
```

```
[1] 45.26316
```

```
median(rabbitLateness)
```

```
[1] 40
```

THERE ARE BUILT-IN FUNCTIONS FOR MEAN AND MEDIAN IN R

```
mean(rabbitLateness)
[1] 45.26316
median(rabbitLateness)
[1] 40
```

LET'S FIND THE MEAN, MEDIAN AND MODE

```
sort(table(rabbitLateness),decreasing = TRUE)[1]
```

THERE ISN'T A BUILT-IN FUNCTION FOR MODE BUT YOU CAN CALCULATE IT WITH A FEW SHENANIGANS

```
mean(rabbitLateness)
[1] 45.26316
median(rabbitLateness)
[1] 40
```

LET'S FIND THE MEAN, MEDIAN AND MODE

```
sort(table(rabbitLateness),decreasing = TRUE)[1]
```

A TABLE WITH COUNTS FOR EACH VALUE IN THE DATASET

```
mean(rabbitLateness)
[1] 45.26316
median(rabbitLateness)
[1] 40
```

LET'S FIND THE MEAN, MEDIAN AND MODE

```
sort(table(rabbitLateness),decreasing = TRUE)[1]
```

A TABLE WITH COUNTS FOR EACH VALUE IN THE DATASET

SORTED IN DESCENDING ORDER

```
mean(rabbitLateness)
[1] 45.26316
median(rabbitLateness)
[1] 40
```

LET'S FIND THE MEAN, MEDIAN AND MODE

```
sort(table(rabbitLateness),decreasing = TRUE)[1]
40
 3
```

A TABLE WITH COUNTS FOR EACH VALUE IN THE DATASET

SORTED IN DESCENDING ORDER

THE FIRST ROW IN THAT TABLE

```
mean(rabbitLateness)
[1] 45.26316
median(rabbitLateness)
[1] 40
```

LET'S FIND THE MEAN, MEDIAN AND MODE

```
sort(table(rabbitLateness),decreasing = TRUE)[1]
40
 3
```

THE MODE

AND THE NUMBER OF TIMES THE MODE APPEARS

A TABLE WITH COUNTS FOR EACH VALUE IN THE DATASET

SORTED IN DESCENDING ORDER

THE FIRST ROW IN THAT TABLE