

Introduction to R Software

Swayam Prabha

Lecture 34

Partition Values, Graphics and Plots

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Slides can be downloaded from
<http://home.iitk.ac.in/~shalab/sp>



Partition values:

Such values divides the total frequency given data into required number of partitions.

Quartile: Divides the data into 4 equal parts.

Decile: Divides the data into 10 equal parts.

Percentile: Divides the data into 100 equal parts.

Partition values:

`quantile` function computes quantiles corresponding to the given probabilities.

The smallest observation corresponds to a probability of 0 and the largest to a probability of 1.

```
quantile(x, ...)
```

```
quantile(x, probs = seq(0, 1, 0.25), ...)
```

Arguments

x numeric vector whose sample quantiles are wanted,

probs numeric vector of probabilities with values in [0, 1].

Partition values:

Example: Marks of 15 students are

```
> marks <- c (99, 89, 29, 86, 37, 96, 51, 75,  
78, 56, 59, 68, 82, 63, 42)
```

```
> quantile(marks)  
 0%  25%  50%  75% 100%  
29.0 53.5 68.0 84.0 99.0
```

Partition values:

R Console

```
> marks <- c (99, 89, 29, 86, 37, 96, 51, 75, 78, 56, 59, 68, 82, 63, 42)
>
> quantile(marks)
 0%  25%  50%  75% 100%
29.0 53.5 68.0 84.0 99.0
>
```

Partition values:

Example: Marks of 15 students are

```
> marks <- c(99, 89, 29, 86, 37, 96, 51, 75,  
78, 56, 59, 68, 82, 63, 42)
```

```
> quantile(marks, probs=c(0,0.25,0.5,0.75,1))  
0%  25%  50%  75% 100%  
29.0 53.5 68.0 84.0 99.0
```

Default values

Partition values:

R Console

```
> marks <- c(99, 89, 29, 86, 37, 96, 51, 75, 78, 56, 59, 68, 82, 63, 42)
> quantile(marks, probs=c(0,0.25,0.5,0.75,1))
 0%  25%  50%  75% 100%
29.0 53.5 68.0 84.0 99.0
>
```

Partition values:

Example: Marks of 15 students are

```
> marks <- c(99, 89, 29, 86, 37, 96, 51, 75, 78,  
56, 59, 68, 82, 63, 42)
```

Defining probabilities

```
> quantile(marks, probs=c(0,0.20,0.4,0.6,0.8,1))  
0%  20%  40%  60%  80% 100%  
29.0 49.2 61.4 76.2 86.6 99.0
```


Partition values:

```
R Console  
> marks <- c(99, 89, 29, 86, 37, 96, 51, 75, 78, 56, 59, 68, 82, 63, 42)  
> quantile(marks, probs=c(0,0.20,0.4,0.6,0.8,1))  
 0%  20%  40%  60%  80% 100%  
29.0 49.2 61.4 76.2 86.6 99.0
```

Graphical tools:

Graphical tools- various type of plots

- 2D & 3D plots,**
- scatter diagram**
- Pie diagram**
- Histogram**
- Bar plot**
- Stem and leaf plot**
- Box plot ...**

Appropriate number and choice of plots in analysis provides better inferences.

Graphical tools:

In R, Such graphics can be easily created and saved in various formats.

- **Bar plot**
- **Pie chart**
- **Box plot**
- **Grouped box plot**
- **Scatter plot**
- **Coplots**
- **Histogram**
- **Normal QQ plot ...**

Bar plots:

Visualize the relative or absolute frequencies of observed values of a variable.

It consists of one bar for each category.

The height of each bar is determined by either the absolute frequency or the relative frequency of the respective category and is shown on the *y-axis*.

```
barplot(x, width = 1, space = NULL,...)
```

```
> barplot(table(x))
```

```
> barplot(table(x)/length(x))
```

Bar plots:

```
> help("barplot")
```

```
barplot(height, width = 1, space = NULL,
names.arg = NULL, legend.text = NULL, beside
= FALSE, horiz = FALSE, density = NULL, angle
= 45, col = NULL, border = par("fg"), main =
NULL, sub = NULL, xlab = NULL, ylab = NULL,
xlim = NULL, ylim = NULL, xpd = TRUE, log =
"", axes = TRUE, axisnames = TRUE, cex.axis =
par("cex.axis"), cex.names = par("cex.axis"),
inside = TRUE, plot = TRUE, axis.lty = 0,
offset = 0, add = FALSE, args.legend = NULL,
...)
```

Example:

Code the 10 persons by using, say 1 for male (M) and 2 for female (F).

M, F, M, F, M, M, M, F, M, M

1, 2, 1, 2, 1, 1, 1, 2, 1, 1

```
> gender <- c(1, 2, 1, 2, 1, 1, 1, 2, 1, 1)
```

```
> gender
```

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

R Console

```
> gender <- c(1, 2, 1, 2, 1, 1, 1, 2, 1, 1)
```

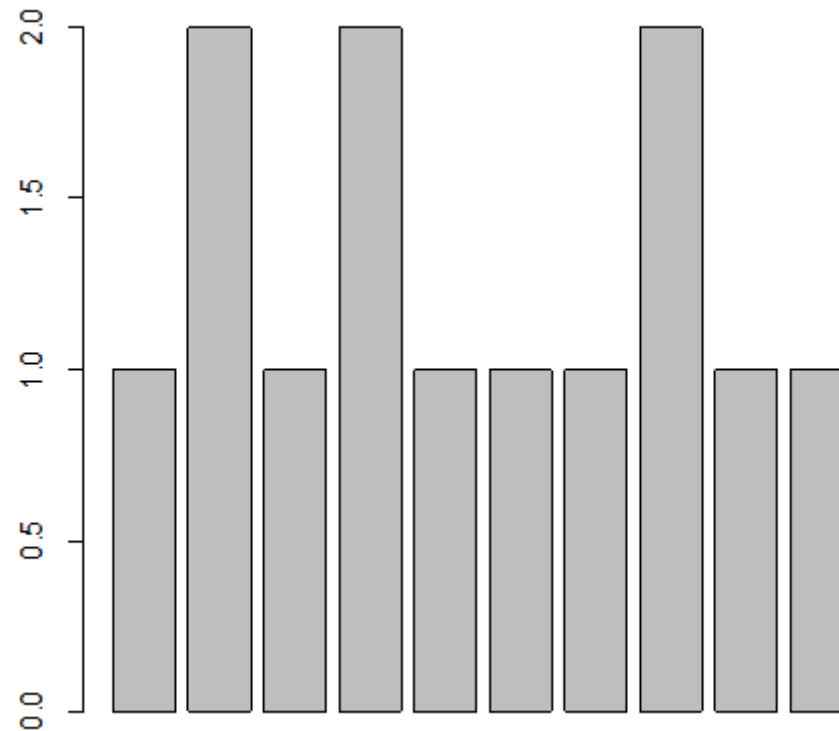
```
> gender
```

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

Bar plots: Example

```
> gender <- c(1, 2, 1, 2, 1, 1, 1, 2, 1, 1)
> gender
[1] 1 2 1 2 1 1 1 2 1 1
> barplot(gender)
```

Do you want this?



Bar plots:

Example

```
> gender <- c(1, 2, 1, 2, 1, 1, 1, 2, 1, 1)
```

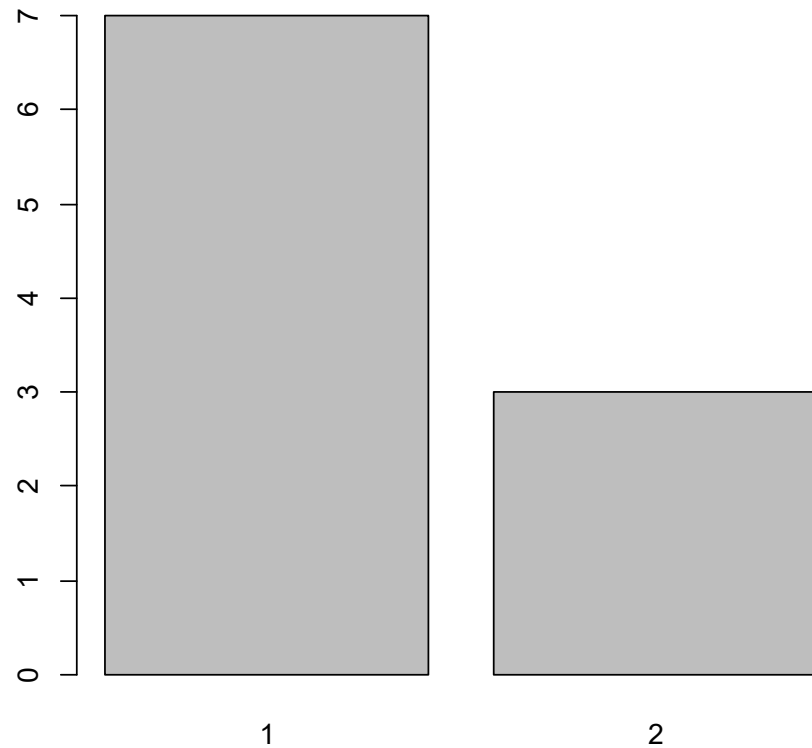
```
> table(gender)
```

```
gender
```

```
1 2
```

```
7 3
```

```
> barplot(table(gender))
```



Bar plots:

Example

```
> gender <- c(1, 2, 1, 2, 1, 1, 1, 2, 1, 1)
```

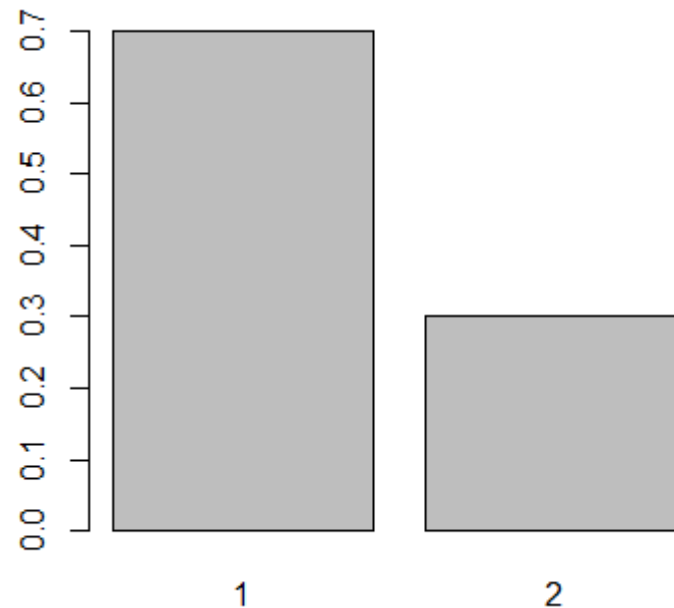
```
> table(gender)/length(gender)
```

```
gender
```

```
 1    2
```

```
0.7 0.3
```

```
> barplot(table(gender)/length(gender))
```



Example

Consider a data set on home delivery of grocery items. The home delivery is centrally managed over phone and delivered by one of the three branches (East- denoted as 1, West- denoted as 2, Central- denoted as 3) of the shop and the 100 data values are recorded on the directions where the grocery items are delivered.

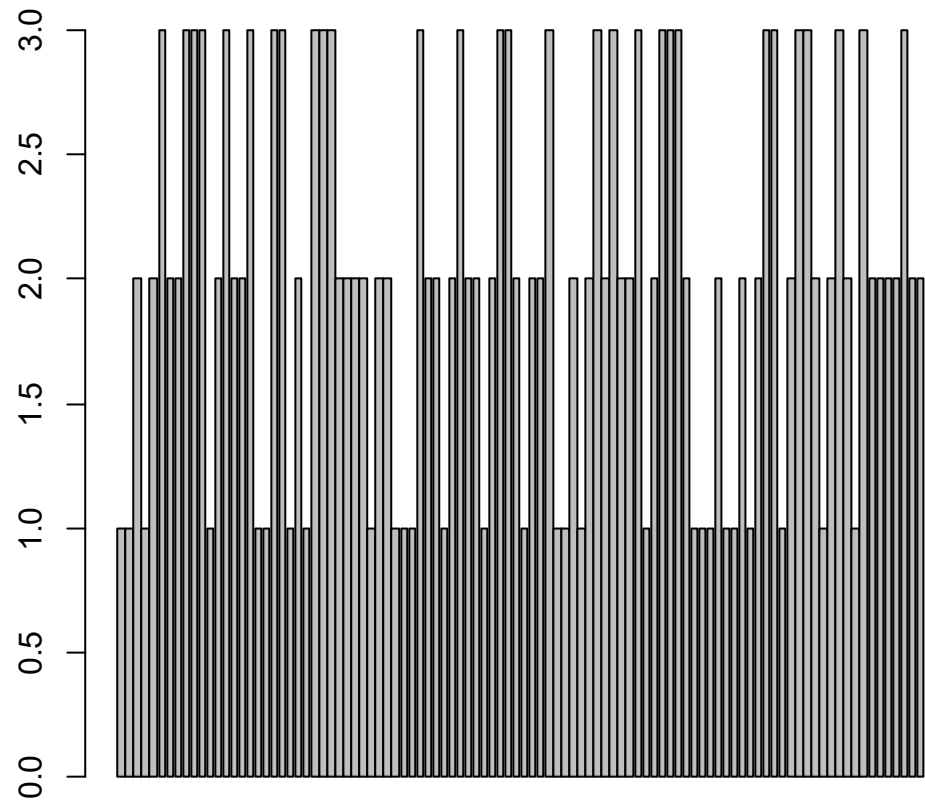
```
direction <-  
c(1,1,2,1,2,3,2,2,3,3,3,1,2,3,2,2,3,1,  
1,3,3,1,2,1,3,3,3,2,2,2,2,1,2,2,1,1,1,3,2,2,1,2  
,3,2,2,1,2,3,3,2,1,2,2,3,1,1,2,1,2,3,2,3,2,2,3,  
1,2,3,3,3,2,1,1,1,2,1,1,2,1,2,3,3,1,2,3,3,2,1,2  
,3,2,1,3,2,2,2,2,3,2,2)
```

Bar plots:

Example

```
> barplot(direction)
```

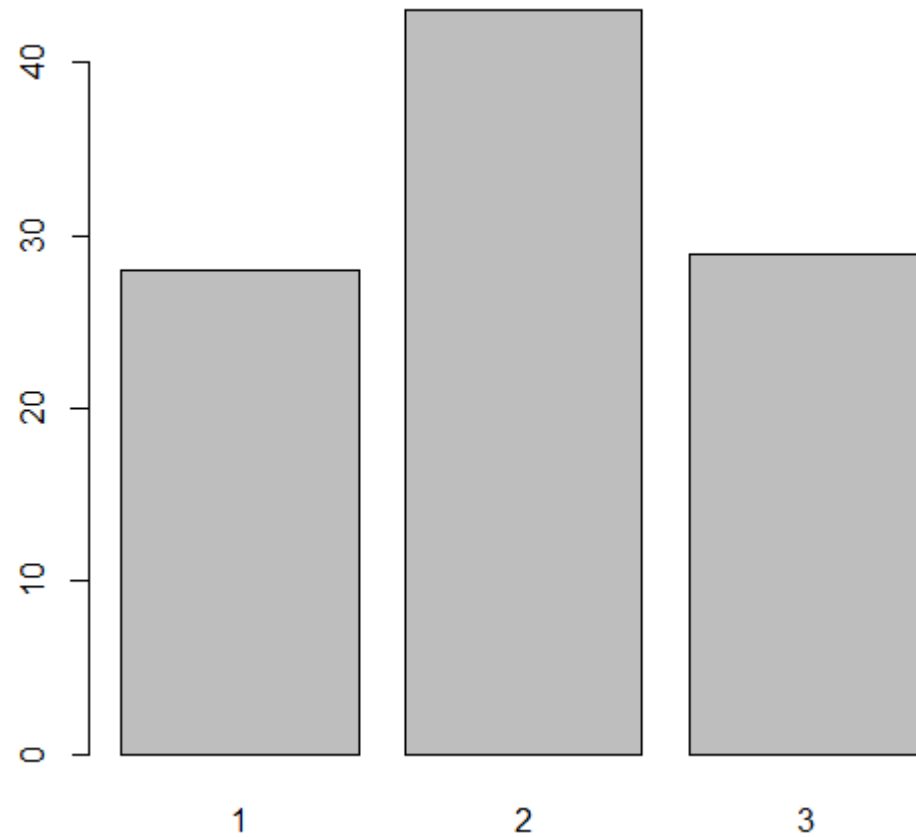
Do you want this?



Bar plots:

Example

```
> barplot(table(direction))
```



Bar plots:

Example

```
> barplot(table(direction)/length(direction))
```

