TMC 204
Statistical Data Analysis with R
Unit 5
Graphical Analysis in R Part 2
Line, Pie and Bar charts

Presented By : Aditya Joshi
Asst. Professor
Department of Computer Application
Graphic Era Deemed to be University
01-05-2020

Line Plot:

The function **plot**() or **lines**() can be used to create a line plot.

R base functions: plot() and lines()

The simplified format of plot() and lines() is as follow.

plot(x, y, type = "l", lty = 1)

lines(x, y, type = "l", lty = 1)

x, y: coordinate vectors of points to join

type: character indicating the type of plotting. Allowed values are:

“p” for points

“l” for lines

“b” for both points and lines

“c” for empty points joined by lines

“o” for overplotted points and lines

“s” and “S” for stair steps

“n” does not produce any points or lines

lty: line types. Line types can either be specified as an integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings “blank”, “solid”, “dashed”, “dotted”, “dotdash”, “longdash”, or “twodash”, where “blank” uses ‘invisible lines’ (i.e., does not draw them).

Create some data

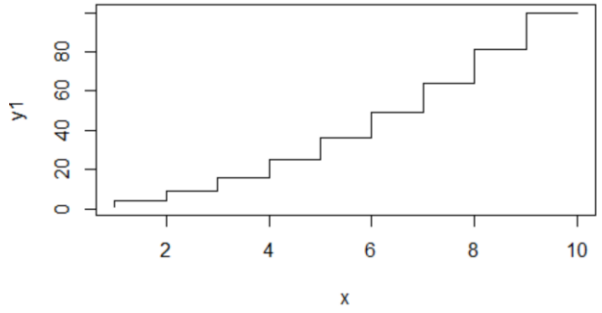
```
# Create some variables
x <- 1:10
y1 <- x*x
y2  <- 2*y1
```

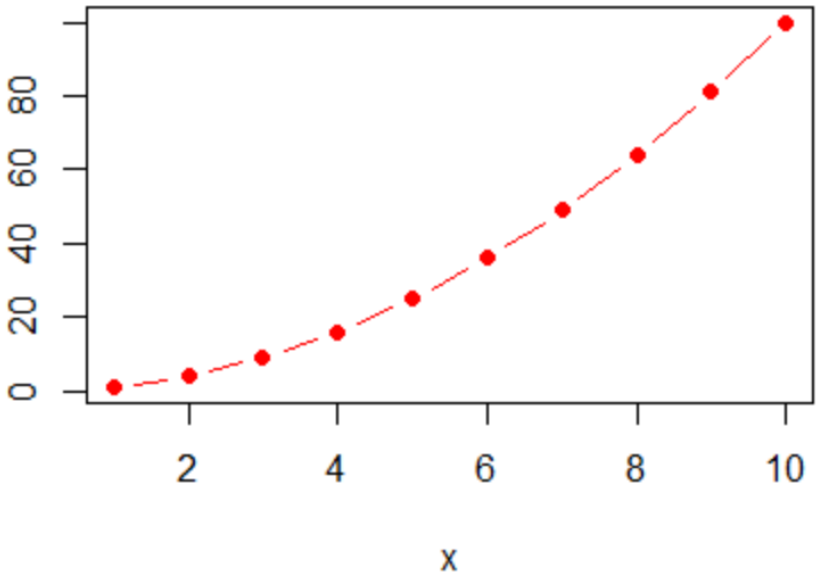
We'll plot a plot with two lines: **lines**(x, y1) and **lines**(x, y2).

Note that the function lines() can not produce a plot on its own. However, it can be used to add lines() on an existing graph. This means that, first you have to use the function plot() to create an empty graph and then use the function lines() to add lines.

Basic line plots

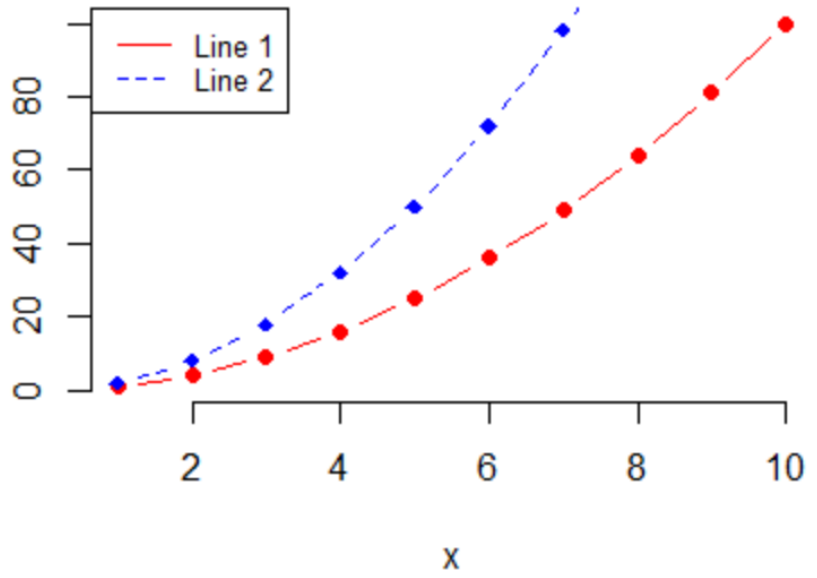
```
# Create a basic stair steps plot
plot(x, y1, type = "S")
# Show both points and line
plot(x, y1, type = "b", pch = 19,
     col = "red", xlab = "x", ylab = "y")
```





Plots with multiple lines

```
# Create a first line
plot(x, y1, type = "b", frame = FALSE, pch = 19,
     col = "red", xlab = "x", ylab = "y")
# Add a second line
lines(x, y2, pch = 18, col = "blue", type = "b", lty = 2)
# Add a legend to the plot
legend("topleft", legend=c("Line 1", "Line 2"),
       col=c("red", "blue"), lty = 1:2, cex=0.8)
```



Pie Chart:

The R base function **pie**() can be used for this.

Create some data

```
df <- data.frame(
      group = c("Male", "Female", "Child"),
      value = c(25, 25, 50)
)
df
> df
  group value
1   Male    25
2 Female    25
3  Child    50
```

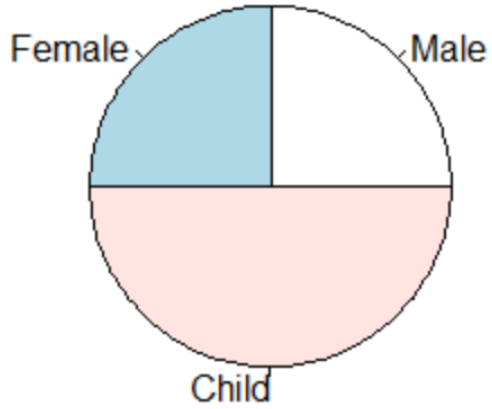
Create basic pie charts: pie()

The function **pie**() can be used to draw a **pie chart**.

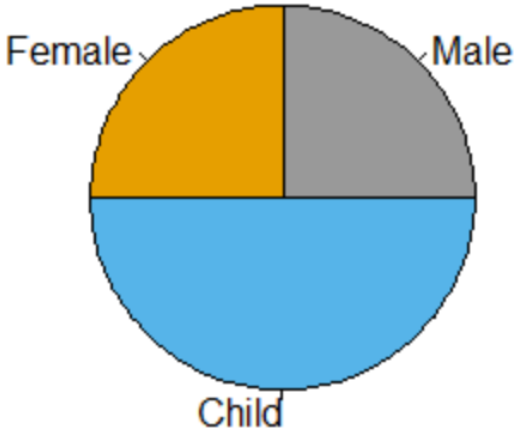
pie(x, labels = names(x), radius = 0.8)

x: a vector of non-negative numerical quantities. The values in x are displayed as the areas of pie slices.
labels: character strings giving names for the slices.
radius: radius of the pie circle. If the character strings labeling the slices are long it may be necessary to use a smaller radius.

pie(df$value, labels = df$group, radius = 1)



```
# Change colors
pie(df$value, labels = df$group, radius = 1,
    col = c("#999999", "#E69F00", "#56B4E9"))
```

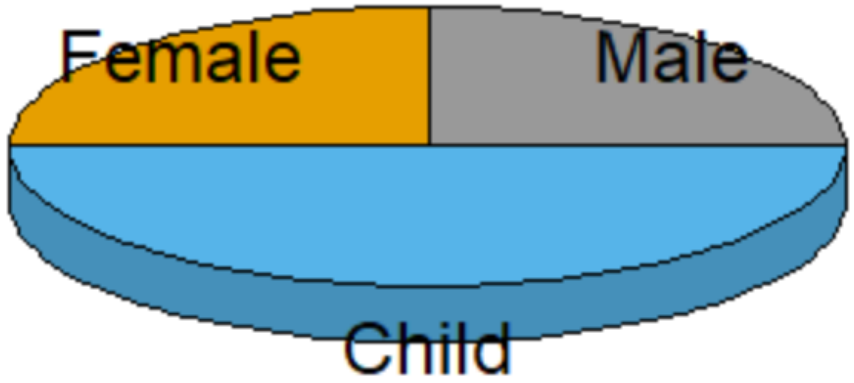


Create 3D pie charts: plotix::pie3D()
Te function pie3D()[in plotrix package] can be used to draw a 3D pie chart.

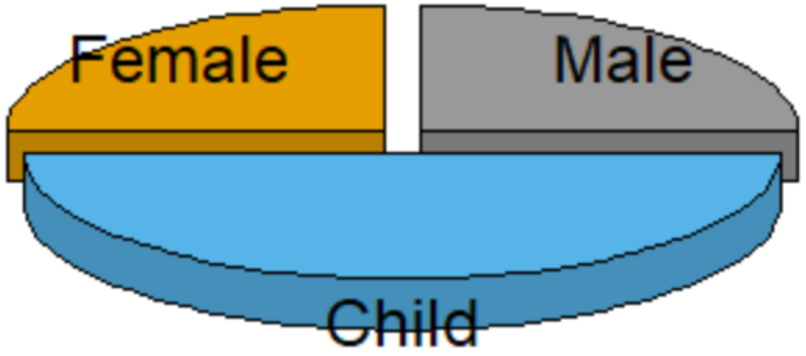
Install plotrix package:

```
install.packages("plotrix")
```

```
# 3D pie chart
library("plotrix")
pie3D(df$value, labels = df$group, radius = 1.5,
      col = c("#999999", "#E69F00", "#56B4E9"))
```



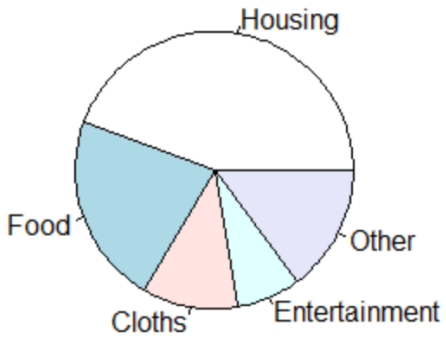

```
# Explode the pie chart
pie3D(df$value, labels = df$group, radius = 1.5,
      col = c("#999999", "#E69F00", "#56B4E9"),
      explode = 0.1)
```



```
expenditure <- data.frame(
      group = c("Housing", "Food",
"Cloths","Entertainment","Other"),value = c(600, 300, 150,
100, 200)
)
expenditure
```

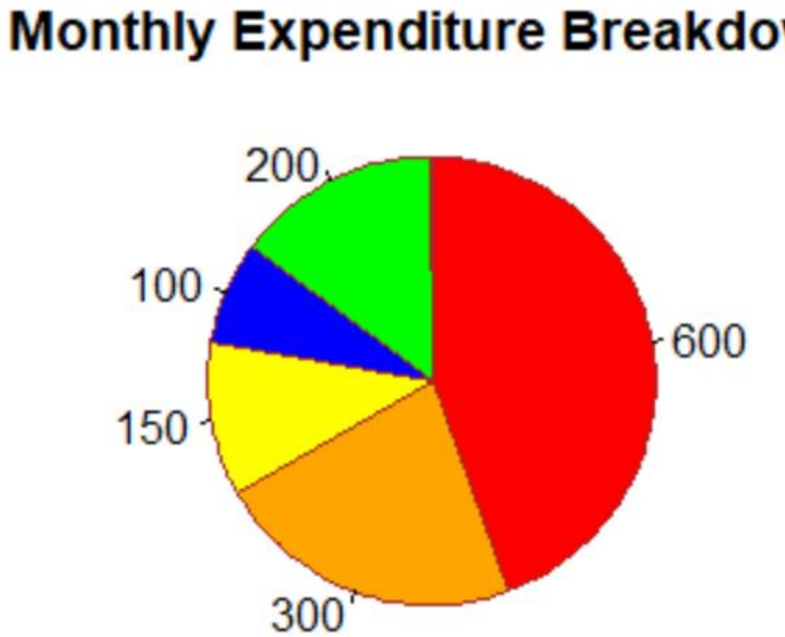
```
> expenditure
         group value
1      Housing   600
2         Food   300
3       Cloths   150
4 Entertainment   100
5        Other   200
```

```
pie(expenditure$value, labels = expenditure$group, radius =
1)
```



With additional parameters

```
pie(expenditure$value, labels = expenditure$value, radius =
1,col=c("red","orange","yellow","blue","green"),main="Mont
hly Expenditure Breakdown",border="brown",
   clockwise=TRUE)
```



Bar plots:

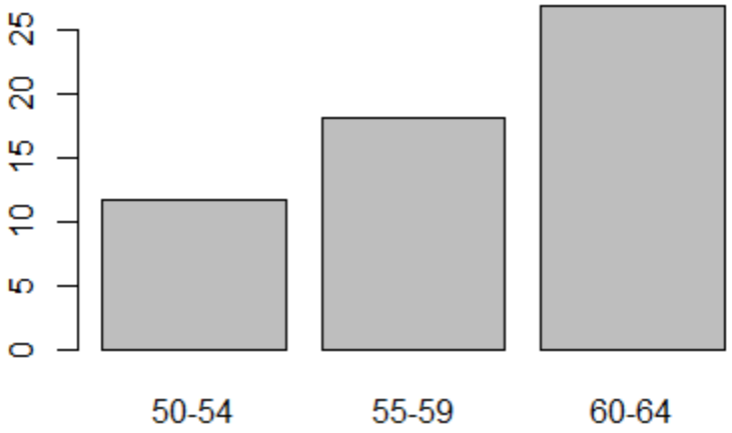
The function **barplot**() can be used to create a **bar plot** with vertical or horizontal bars.

Here, we'll use the R built-in **VADeaths** data set.

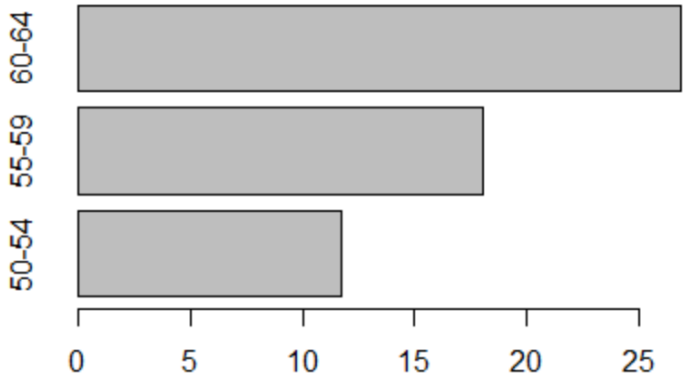
```
> VADeaths
      Rural Male Rural Female Urban Male Urban Female
50-54       11.7          8.7       15.4          8.4
55-59       18.1         11.7       24.3         13.6
60-64       26.9         20.3       37.0         19.3
65-69       41.0         30.9       54.6         35.1
70-74       66.0         54.3       71.1         50.0

# Subset
x <- VADeaths[1:3, "Rural Male"]
x
> x
50-54 55-59 60-64
 11.7  18.1  26.9
```

Basic bar plots

# Bar plot of one variable
barplot(x)

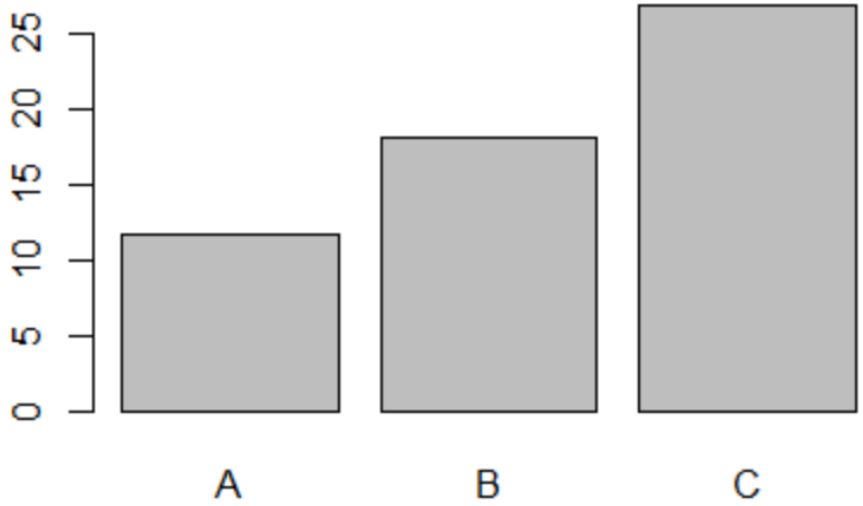


# Horizontal bar plot
barplot(x, horiz = TRUE)

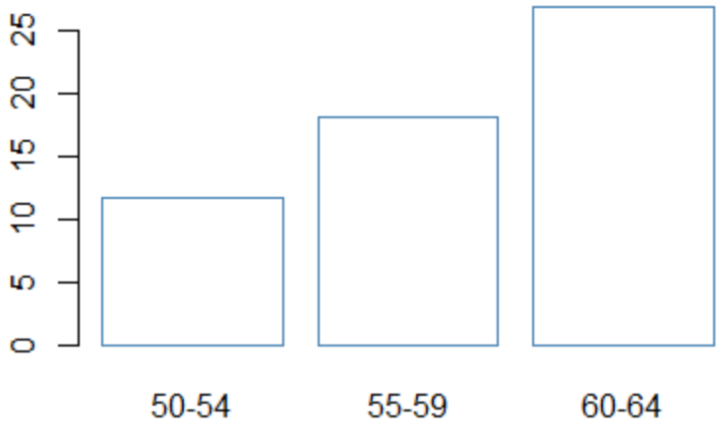


Change group names

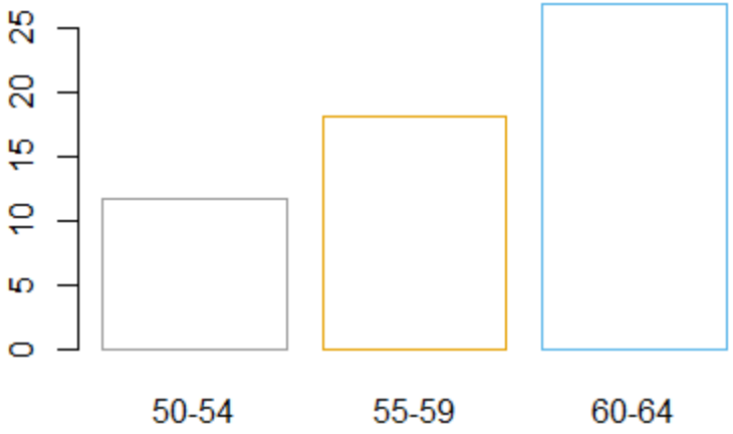
barplot(x, names.arg = c("A", "B", "C"))



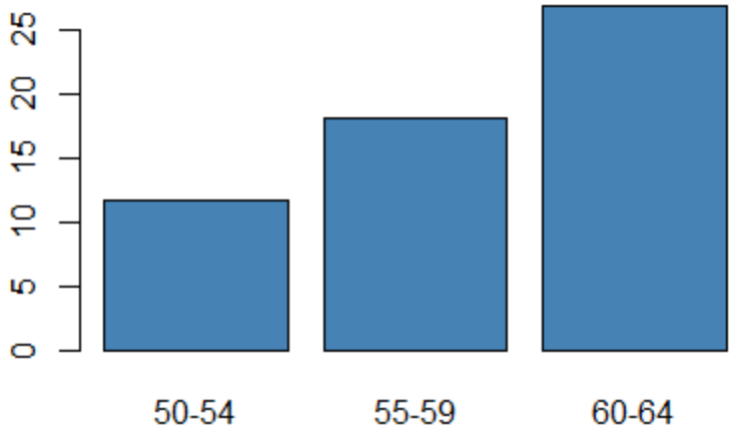
# Change border and fill color using one single color
barplot(x, col = "white", border = "steelblue")



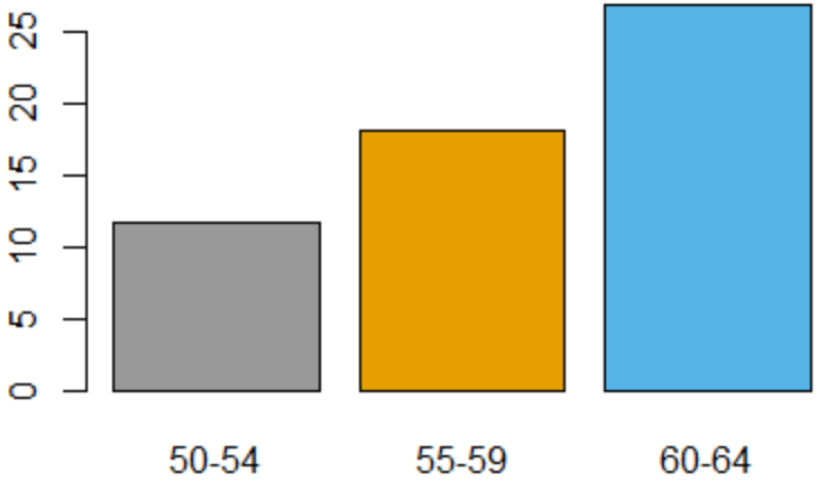
Change the color of border.
Use different colors for each group
barplot(x, col = "white",
border = c("#999999", "#E69F00", "#56B4E9"))



# Change fill color : single color
barplot(x, col = "steelblue")

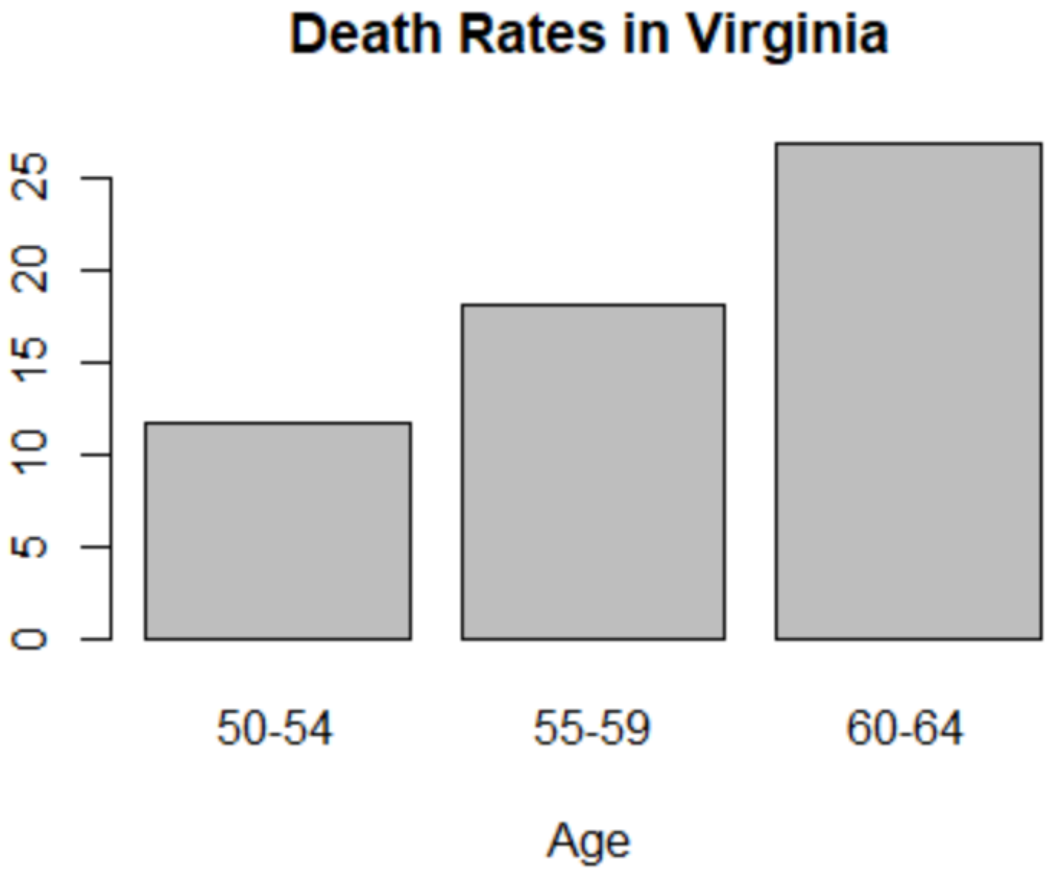


# Change fill color: multiple colors
barplot(x, col = c("#999999", "#E69F00", "#56B4E9"))



Change main title and axis labels

```
# Change axis titles
# Change color (col = "gray") and remove frame
barplot(x, main = "Death Rates in Virginia",
        xlab = "Age", ylab = "Rate")
```

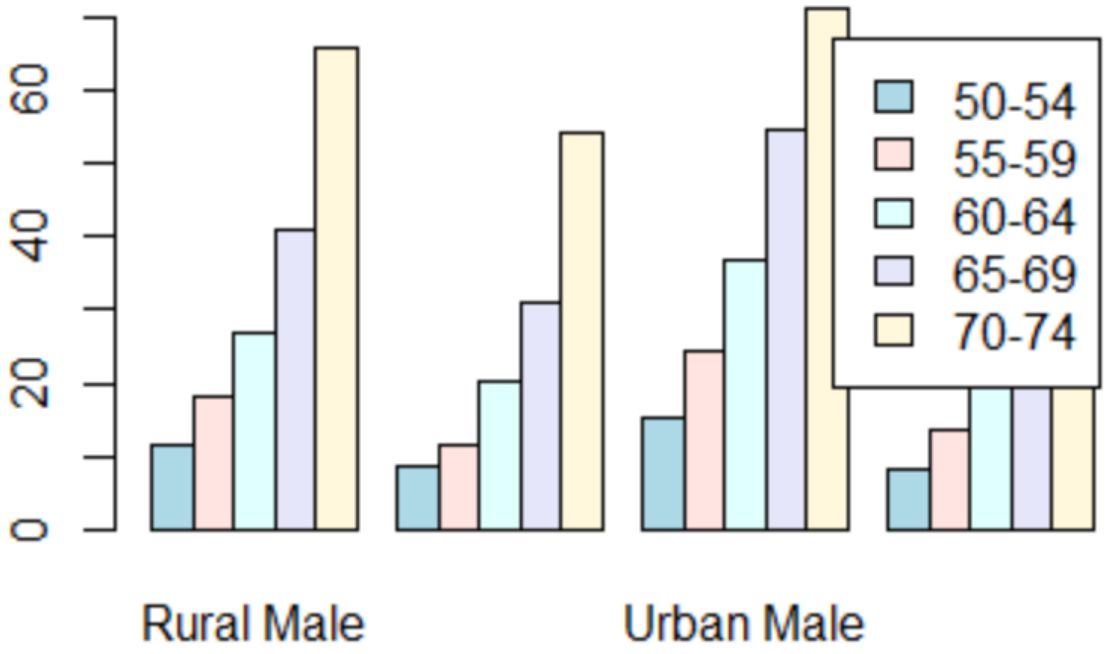


Stacked bar plots

```
barplot(VADeaths,
        col = c("lightblue", "mistyrose", "lightcyan",
                "lavender", "cornsilk"),
        legend = rownames(VADeaths))
```

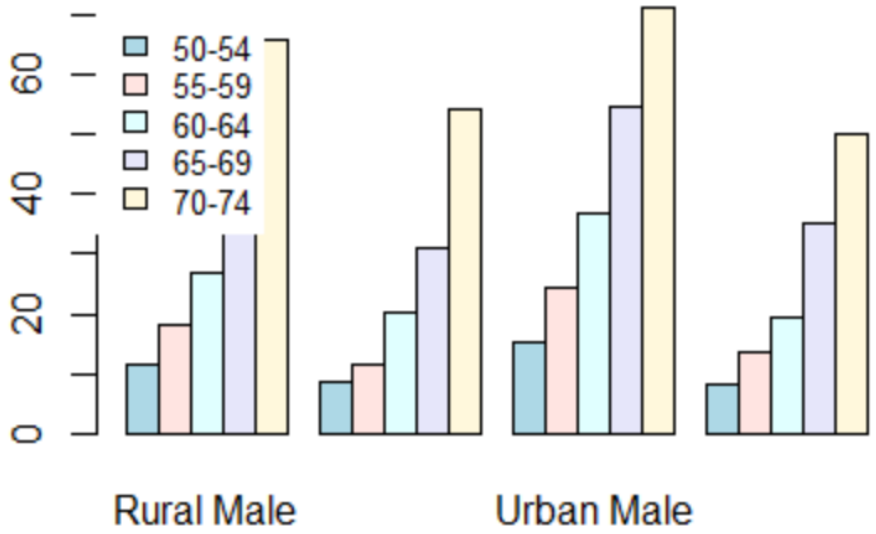
Grouped bar plots

```
barplot(VADeaths,
        col = c("lightblue", "mistyrose", "lightcyan",
                "lavender", "cornsilk"),
      legend = rownames(VADeaths), beside = TRUE)
```



It's also possible to add legends to a plot using the function **legend**() as follow.

```
# Define a set of colors
my_colors <- c("lightblue", "mistyrose", "lightcyan",
               "lavender", "cornsilk")
# Bar plot
barplot(VADeaths, col = my_colors, beside = TRUE)
# Add legend
legend("topleft", legend = rownames(VADeaths),
       fill = my_colors, box.lty = 0, cex = 0.8)
```



box.lty = 0: Remove the box around the legend
cex = 0.8: legend text size