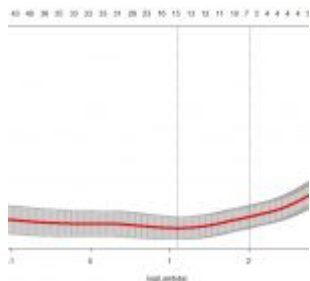


Data Visualization with googleVis exercises part 4



Adding Features to your Charts

We saw in the previous charts some basic and well-known types of charts that googleVis offers to users. Before continuing with other, more sophisticated charts in the next parts we are going to “dig a little deeper” and see some interesting features of those we already know.

Read the examples below to understand the logic of what we are going to do and then test your skills with the exercise set we prepared for you. Lets begin!

Answers to the exercises are available [here](#).

Package & Data frame

As you already know, the first thing you have to do is install and load the googleVis package with:

```
install.packages("googleVis")  
library(googleVis)
```

Secondly we will create an experimental data frame which will be used for our charts' plotting. You can create it with:

```
df=data.frame(name=c("James", "Curry", "Harden"),  
Pts=c(20,23,34),  
Rbs=c(13,7,9))
```

NOTE: The charts are created locally by your browser. In case they are not displayed at once press F5 to reload the page.

Customizing Chart

We are going to use the two-axis Line Chart we created in [part 1](#). This is the code we used, in case you forgot it:

```
LineC2 <- gvisLineChart(df, "name", c("Pts", "Rbs"),
options=list(
series="[{"targetAxisIndex: 0},
{"targetAxisIndex:1}]",
vAxes="[{"title:'Pts'}, {"title:'Rbs'}]"
))
plot(LineC2)
```

Colours

To set the color of every line we can use:

```
series="[{"color:'green', targetAxisIndex: 0,
```

Exercise 1

Change the colours of your line chart to green and yellow respectively and plot the chart.

Line Width

You can determine the line width of every line with:

```
series="[{"color:'green',targetAxisIndex: 0, lineWidth: 3},
```

Exercise 2

Change the line width of your lines to 3 and 6 respectively and plot the chart.

Dashed lines

You can transform your lines to dashed with:

```
series="[{"color:'green', targetAxisIndex: 0,
lineWidth: 1, lineDashStyle: [2, 2, 20, 2, 20, 2]}],
```

There are many styles and colours available and you can find them [here](#).



Learn more about using GoogleVis in the online course [Mastering in Visualization with R programming](#). In this course you will learn how to:

- Work extensively with the GoogleVis package and its functionality
- Learn what visualizations exist for your specific use case
- And much more

Exercise 3

Choose two different styles of dashed lines for every line of your chart from the link above and plot your chart.

Point Shape

With the `pointShape` option you can choose from a variety of shapes for your points.

We will use the scatter chart we built in [part 3](#) to see how it works. Here is the code:

```
ScatterCD <- gvisScatterChart(cars,
options=list(
  legend="none",
  pointSize=3,linewidth=2,
  title="Cars", vAxis="{title:'speed'}",
  hAxis="{title:'dist'}",
  width=600, height=300))
plot(ScatterCD)
```

Exercise 4

Change the shape of your scatter chart's points to 'square' and plot it. **HINT:** Use `pointShape`.

Exercise 5

Change the shape of your scatter chart's points to 'triangle',

their point size to 7 and plot it.

Edit Button

A really useful and easy feature that googleVis provides is the edit button which gives the user the ability to customize the chart in an automated way.

```
options=list(gvis.editor="Edit!")
```

Exercise 6

Add an edit button in the scatter chart you just created.

HINT: Use `gvis.editor`.

Chart with more options

Now let's see how we can create a chart with many features that can enhance its appearance. We will use again the 2-axis line that we used before.

```
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
options=list(
series="[{"color:'green',targetAxisIndex: 0, lineWidth: 3,
lineDashStyle: [14, 2, 2, 7]},
{"color:'yellow',targetAxisIndex:1,lineWidth: 6,
lineDashStyle: [10, 2]}]",
vAxes="[{"title:'Pts'}, {"title:'Rbs'}]"
))
plot(LineCD2)
```

Background color

You can decide the background color of your chart with:

```
backgroundColor="red",
```

Exercise 7

Set the background color of your line chart to "lightblue" and plot it. **HINT:** Use `backgroundColor`.

Title

To give a title and decide its features you can use:

```
title="Title",  
titleTextStyle="{color:'orange',  
fontName:'Courier',  
fontSize:14}",
```

Exercise 8

Give a title of your choice to the line chart and set its font to blue, Courier of size 16. **HINT:** Use `titleTextStyle`.

Curve Type & Legend

Another nice-looking choice that `googleVis` gives you is to display the lines like curves with:

```
curveType="function"
```

You can also move the legend of your chart to the bottom with:

```
legend="bottom"
```

Exercise 9

Smooth the lines of your line chart by setting the `curveType` option to `function` and move the legend to the bottom. **HINT:** Use `curveType` and `legend`.

Axes features

Finally you can “play” with your axes. This is an example:

```
vAxis="{gridlines:{color:'green', count:4}}",  
hAxis="{title:'City', titleTextStyle:{color:'red'}}",  
series="[{color:'yellow', targetAxisIndex: 0},  
{color: 'brown',targetAxisIndex:1}]",  
vAxes="[{title:'val1'}, {title:'val2'}]",
```

Exercise 10

Give the title “Name” to your `hAxis` and color it orange. Separate your `vAxis` with 3 red gridlines. **HINT:** Use `titleTextStyle` and `gridlines`

Data Visualization with googleVis solutions part 4

Below are the solutions to [these](#) exercises on visualizations with googleVis.

```
#####  
#                               #  
#   Exercise 1                 #  
#                               #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))  
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),  
                        options=list(  
series="[ {color:'green',targetAxisIndex: 0},  
{color:'yellow',targetAxisIndex:1} ]",  
                        vAxes="[ {title:'Pts'},  
{title:'Rbs'} ]"  
                        ))  
plot(LineCD2)
```

```
#####  
#                               #  
#   Exercise 2                 #  
#                               #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))  
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
```

```

options=list(
series="[ {color:'green',targetAxisIndex: 0, lineWidth: 3},
{color:'yellow',targetAxisIndex:1,lineWidth: 6}]",
vAxes="[ {title:'Pts'}",
{title:'Rbs'}]"
))

```

```
plot(LineCD2)
```

```
#####
#                               #
#   Exercise 3                   #
#                               #
#####

```

```
library(googleVis)
```

```
df=data.frame(name=c("James", "Curry", "Harden"),
Pts=c(20,23,34),
Rbs=c(13,7,9))
```

```
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
options=list(
series="[ {color:'green',targetAxisIndex: 0, lineWidth: 3,
lineDashStyle: [14, 2, 2, 7]},
{color:'yellow',targetAxisIndex:1,lineWidth: 6,
lineDashStyle: [10, 2]}]",
vAxes="[ {title:'Pts'}",
{title:'Rbs'}]"
))

```

```
plot(LineCD2)
```

```
#####
#                               #
#   Exercise 4                   #
#                               #
#####

```

```
library(googleVis)
```

```
ScatterCD <- gvisScatterChart(cars,
options=list(
legend="none",
pointSize=3,lineWidth=2,pointShape='square',
title="Cars",
```

```

vAxis="{title:'speed'}",
                                hAxis="{title:'dist'}",
                                width=600, height=300))
plot(ScatterCD)

#####
#                               #
#   Exercise 5                 #
#                               #
#####

library(googleVis)
ScatterCD <- gvisScatterChart(cars,
                              options=list(
                                legend="none",
pointSize=7,lineWidth=2,pointShape='triangle',
                                title="Cars",
vAxis="{title:'speed'}",
                                hAxis="{title:'dist'}",
                                width=600, height=300))
plot(ScatterCD)

#####
#                               #
#   Exercise 6                 #
#                               #
#####

library(googleVis)
ScatterCD <- gvisScatterChart(cars,
                              options=list(gvis.editor="Edit",
                                legend="none",
pointSize=7,lineWidth=2,pointShape='triangle',
                                title="Cars",
vAxis="{title:'speed'}",
                                hAxis="{title:'dist'}",
                                width=600, height=300))
plot(ScatterCD)

#####

```



```

#           #
#   Exercise 7   #
#           #
#####

library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
options=list(backgroundColor="lightblue",
series="[{color:'green',targetAxisIndex: 0, lineWidth: 3,
          lineDashStyle: [14, 2, 2, 7]},
{color:'yellow',targetAxisIndex:1,lineWidth: 6,
          lineDashStyle: [10, 2]})",
          vAxes="[{title:'Pts'},
{title:'Rbs'}]"))
plot(LineCD2)

```

```

#####
#           #
#   Exercise 8   #
#           #
#####

library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
options=list(backgroundColor="lightblue",
              title="Line Chart",
titleTextStyle="{color:'blue',
                  fontName:'Courier',
                  fontSize:16}",
series="[{color:'green',targetAxisIndex: 0, lineWidth: 3,
          lineDashStyle: [14, 2, 2, 7]},
{color:'yellow',targetAxisIndex:1,lineWidth: 6,
          lineDashStyle: [10, 2]})",
          vAxes="[{title:'Pts'},

```

```

{title:'Rbs'}}]"
))
plot(LineCD2)

#####
#                               #
#   Exercise 9                   #
#                               #
#####

library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
options=list(backgroundColor="lightblue",
              title="Line Chart",
titleTextStyle="{color:'blue',
                  fontName:'Courier',
                  fontSize:16}",
                  curveType="function",
series="[ {color:'green',targetAxisIndex: 0, lineWidth: 3,
           lineDashStyle: [14, 2, 2, 7]},
          {color:'yellow',targetAxisIndex:1,lineWidth: 6,
           lineDashStyle: [10, 2]}]\"",
              vAxes="[ {title:'Pts'},
{title:'Rbs'}}]"
))
plot(LineCD2)

```

```

#####
#                               #
#   Exercise 10                  #
#                               #
#####

library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
LineCD2 <- gvisLineChart(df, "name", c("Pts","Rbs"),

```


going to do and then test your skills with the exercise set we prepared for you. Let's begin!

Answers to the exercises are available [here](#).

Package Installation

As you already know, the first thing you have to do is install and load the googleVis package with:

```
install.packages("googleVis")  
library(googleVis)
```

NOTE: The charts are created locally by your browser. In case they are not displayed at once press F5 to reload the page.

Scatter chart

It is quite simple to create a scatter chart with googleVis. We will use the cars dataset. Look at the example below:

```
ScatterC <- gvisScatterChart(cars)  
plot(ScatterC)
```

Exercise 1

Create a list named "ScatterC" and pass to it the cars dataset as a scatter chart. **HINT:** Use gvisScatterChart().

Exercise 2

Plot the scatter chart. **HINT:** Use plot().

Titles

It is time to learn how to enhance the appearance of our googleVis charts. We shall give a title to the chart and also name hAxis and vAxis. Look at the example:

```
options=list(title="Cars", vAxis="{title:'speed'}",  
hAxis="{title:'dist'}" )
```

Exercise 3

Name your chart "Cars", your chart's vAxis "speed", your chart's hAxis "dist" and plot the chart. **HINT:** Use list().

Size

You can adjust the size with width and height.

Exercise 4

Set your chart's width to 600 and height to 300.

Legend

You can deactivate your chart's legend if you set it to "none".

Exercise 5

Deactivate your chart's legend.



Learn more about using GoogleVis in the online course [Mastering in Visualization with R programming](#). In this course you will learn how to:

- Work extensively with the GoogleVis package and its functionality
- Learn what visualizations exist for your specific use case
- And much more

Point size & Line width

You can determine the size of the chart's points with `pointsize` and also choose to unite them with line with `linewidth`. For example:

```
pointSize=4,linewidth=3
```

Exercise 6

Set point size to 3 and line width to 2.

Bubble Chart

Another amazing type of chart that googleVis provides is the bubble chart. You can create a simple Bubble Chart of the Fruits dataset like this:

```
BubbleC <- gvisBubbleChart(Fruits)
plot(BubbleC)
```

Exercise 7

Create a list named "BubbleC" and pass to it the Fruits dataset as a bubble chart. **HINT:** Use gvisBubbleChart().

Exercise 8

Plot the chart. **HINT:** Use plot().

Bubble Chart's Features

As you can see, you created a bubble chart but it seems to be useless. In order to make it useful you should pass to it some of your dataset's variables as features. It depends on what you want to be displayed and how. If you type head(Fruits) you can easily recognize the numeric variables of your dataset. Then you can use them like this:

```
BubbleC <- gvisBubbleChart(Fruits,idvar="VAR1",
xvar="VAR2", yvar="VAR3",
colorvar="VAR4", sizevar="VAR5")
```

Exercise 9

Find the numeric variables of Fruits, then set "Fruit" as idvar, "Sales" as xvar, "Expenses" as yvar, "Year" as colorvar and "Profit" as sizevar and plot your chart. **HINT:** Use head().

Data range

You can also adjust the minimum and maximum number of hAxis and vAxis that you want to be displayed. Look at the example below:

```
options=list(
hAxis='{minValue:50, maxValue:150}')
```

Exercise 10

Set your hAxis range from 70 to 130 and your vAxis range from 50 to 100.

Data visualization with googleVis solutions part 3

Below are the solutions to [these](#) exercises on visualizations with googleVis.

```
#####
#                               #
#   Exercise 1                 #
#                               #
#####
```

```
library(googleVis)
ScatterC <- gvisScatterChart(cars)
```

```
#####
#                               #
#   Exercise 2                 #
#                               #
#####
```

```
library(googleVis)
ScatterC <- gvisScatterChart(cars)
plot(ScatterC)
```

```
#####
#                               #
```



```

width=600, height=300))
plot(ScatterC)

#####
#                               #
#   Exercise 6                   #
#                               #
#####

library(googleVis)
ScatterC <- gvisScatterChart(cars,
                             options=list(
                               legend="none",
                               pointSize=3,lineWidth=2,
                               title="Cars",
vAxis="{title:'speed'}",
hAxis="{title:'dist'}",
width=600, height=300))
plot(ScatterC)

```

```

#####
#                               #
#   Exercise 7                   #
#                               #
#####

```

```

library(googleVis)
BubbleC <- gvisBubbleChart(Fruits)

```

```

#####
#                               #
#   Exercise 8                   #
#                               #
#####

```

```

library(googleVis)
BubbleC <- gvisBubbleChart(Fruits)
plot(BubbleC)

```

```

#####
#                               #

```

```

# Exercise 9 #
# #
#####

library(googleVis)
head(Fruits)
BubbleC <- gvisBubbleChart(Fruits,idvar="Fruit",
                           xvar="Sales", yvar="Expenses",
                           colorvar="Year", sizevar="Profit")

plot(BubbleC)

```

```

#####
# #
# Exercise 10 #
# #
#####

```

```

library(googleVis)
BubbleC <- gvisBubbleChart(Fruits,idvar="Fruit",
                           xvar="Sales", yvar="Expenses",
                           colorvar="Year", sizevar="Profit",
                           options=list(
                               hAxis='{minValue:70,
maxValue:130}',
                               vAxis='{minValue:50,
maxValue:100}'))
plot(BubbleC)

```

[Bonus: Statistical graphics solutions](#)

Below are the solutions to [these](#) exercises on Statistical graphics.

```

#####

```

```
# #
# Exercise 1 #
# #
#####
```

```
x <- rnorm(100)
hist(x)
```

```
#####
# #
# Exercise 2 #
# #
#####
```

```
br <- c(,5,10,16,17,18,20,25,60,80)
```

```
#####
# #
# Exercise 3 #
# #
#####
```

```
mid.point <- c(2.5,7.5,13,16.5,17.5,19,22.5,44.5,70.5)
player.count <- c(39,57,69,31,42,75,160,327,114)
point.acc <- rep(mid.point,pla.count)
br <- c(,5,10,16,17,18,20,25,60,80)
hist(point.acc,breaks=br)
```

```
#####
# #
# Exercise 4 #
# #
#####
```

```
library(ISwR)
boxplot(IgM)
```

```
#####
# #
# Exercise 5 #
```

```
# #
#####

library(ISwR)
par(mfrow=c(1,2))
boxplot(IgM)
boxplot(log(IgM))

#####
# #
# Exercise 6 #
# #
#####

library(ISwR)
par(mfcol=c(2,1))
boxplot(IgM)
boxplot(log(IgM))

#####
# #
# Exercise 7 #
# #
#####

library(ISwR)
par(mfrow=c(2,2))
boxplot(IgM)
boxplot(log(IgM))

#####
# #
# Exercise 8 #
# #
#####

library(ISwR)
par(mfrow=c(1,3))
boxplot(IgM)
boxplot(log(IgM))
```

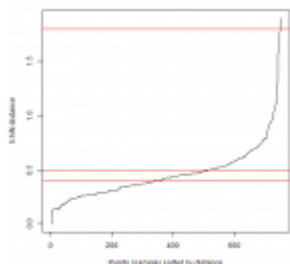
```
#####  
#                               #  
#   Exercise 9                   #  
#                               #  
#####
```

```
library(ISwR)  
par(mfcol=c(2,2))  
boxplot(IgM)  
boxplot(log(IgM))
```

```
#####  
#                               #  
#   Exercise 10                  #  
#                               #  
#####
```

```
library(ISwR)  
par(mfcol=c(3,1))  
boxplot(IgM)  
boxplot(log(IgM))
```

Bonus: Statistical graphics exercises



Hello everybody! This is a bonus set of exercises that r-exercises.com kindly provides.

In this part we are going to see two useful statistical

graphics in R and some of their features.

Answers to the exercises are available [here](#).

To run these examples we have to create an artificial dataset of 100 normally distributed observations. Notice that you will not find the same results exactly every time you run the code because you generate random numbers every single time. For this purpose we will use `rnorm()` and attach it to a variable `x` like this:

```
x <- rnorm(100)
```

Histogram

Histograms are considered to be very useful in distribution recognition and can be created with:

```
hist(x)
```

Exercise 1

Create a histogram for the variable "x".

By specifying breaks in the `hist()`, you can decide the number of the bars in the histogram. It is preferable to specify breaks as a vector. To run this example let's create a random experimental dataset.

```
mid.point <- c(2.5,7.5,13,16.5,17.5,19,22.5,44.5,70.5)
```

```
player.count <- c(39,57,69,31,42,75,160,327,114)
```

```
point.acc <- rep(mid.point,pla.count)
```

To create the breaks' vector and attach it to histogram you can use:

```
br <- c(0,5,10,16,17,18,20,25,60,80)
```

```
hist(point.acc,breaks=br)
```

Exercise 2

Create a vector to set the histogram's breaks of your choice.

Exercise 3

Attach the breaks' vector to `point.acc` and create its histogram.

Boxplots

Now let's see how to create a boxplot. For this example we will need the the "IgM" dataset of the "ISwR" package. In order to install and call the package use:

```
install.packages("ISwR")
library(ISwR)
```

To create a simple boxplot you just need the `boxplot()` function.



Learn more about using graphics for statistics in the online course [Mastering in Visualization with R programming](#). In this course you will learn how to:

- Work extensively with the different basic graphics like histogram, clustering and density plots
- Learn what visualizations exist for your specific use case
- And much more

Exercise 4

Draw the boxplot of the "IgM" dataset.

In order to produce a layout with two plots side by side use the `mfrow` graphical parameter. There is also an `mfcol` parameter to plot columnwise.

```
#row
par(mfrow=c(1,2))
boxplot(IgM)
boxplot(log(IgM))
#column
par(mfcol=c(2,1))
```

```
boxplot(IgM)
boxplot(log(IgM))
```

Exercise 5

Produce a layout with the an “IgM” bloxpot side by side with a log(IgM) bloxpot. **HINT:** Use mfrow.

Exercise 6

Produce a layout with the an “IgM” bloxpot on the same column with the log(IgM) bloxpot. **HINT:** Use mfcol()

Exercise 7

Change the parameters of the mfrow parameter of exercise 5 from (1,2) to (2,2) and notice the difference.

Exercise 8

Change the parameters of the mfrow parameter of exercise 5 from (1,2) to (1,3) and notice the difference.

Exercise 9

Change the parameters of the mfcol parameter of exercise 7 from (2,1) to (2,2) and notice the difference.

Exercise 10

Change the parameters of the mfcol parameter of exercise 7 from (2,1) to (3,1) and notice the difference.

Data Visualization with

googleVis exercises part 2



Area, Stepped & Combo charts

In the second part of our series we are going to meet three more googleVis charts. More specifically these charts are **Area Chart**, **Stepped Area Chart** and **Combo Chart**.

Read the examples below to understand the logic of what we are going to do and then test your skills with the exercise set we prepared for you. Let's begin!

Answers to the exercises are available [here](#).

Package & Data frame

As you already know, the first thing you have to do is install and load the googleVis package with:

```
install.packages("googleVis")  
library(googleVis)
```

Secondly we will create an experimental data frame which will be used for our charts' plotting. You can create it with:

```
df=data.frame(name=c("James", "Curry", "Harden"),  
Pts=c(20,23,34),  
Rbs=c(13,7,9))
```

NOTE: The charts are created locally by your browser. In case they are not displayed at once press F5 to reload the page.

Area chart

It is quite simple to create an area chart with googleVis with:

```
AreaC <- gvisBarChart(df)
plot(AreaC)
```

Exercise 1

Create a list named "AreaC" and pass to it the "df" data frame you just created as an area chart. **HINT:** Use `gvisAreaChart()`.

Exercise 2

Plot the area chart. **HINT:** Use `plot()`.

Stepped Area chart

Creating a stepped area chart is a little different than the area chart. You have to set the X and Y variables and also make it stacked in order to display the values correctly. Here is an example:

```
SteppedAreaC <- gvisSteppedAreaChart(df, xvar="name",
yvar=c("val1", "val2"),
options=list(isStacked=TRUE))
plot(SteppedAreaC)
```

Exercise 3

Create a list named "SteppedAreaC" and pass to it the "df" data frame you just created as a stepped area chart. You should also set the X variable as the players' names and Y variable as their values. **HINT:** Use `gvisSteppedAreaChart()`, `xvar` and `yvar`.

Exercise 4

Plot the stepped area chart. **HINT:** Use `plot()`.



Learn more about using GoogleVis in the online course [Mastering in Visualization with R programming](#). In this course you will learn how to:

- Work extensively with the GoogleVis package and its functionality
- Learn what visualizations exist for your specific use case
- And much more

Exercise 5

Now transform your stepped area chart to stacked to correct it and plot it.

Combo chart

The next chart we are going to meet is the combination of lines and bars chart known as combo chart. You can produce it like this:

```
ComboC <- gvisComboChart(df, xvar="country",
yvar=c("val1", "val2"),
options=list(seriesType="bars",
series='{1: {type:"line"}}')
plot(ComboC)
```

Exercise 6

Create a list named “ComboC” and pass to it the “df” data frame you just created as a combo chart. You should also set the X variable as the players’ names and Y variable as their values. **HINT:** Use `gvisComboChart()`, `xvar` and `yvar`.

Exercise 7

Plot the chart. What kind of chart do you see? **HINT:** Use `plot()`.

In order to add the bars we have to set it as the example below.

```
options=list(seriesType="bars",
series='{1: {type:"line"}}')
```

Exercise 8

Transform the chart you just created into a combo chart with bars and lines and plot it. **HINT:** Use `list()`.

Exercise 9

In the previous exercise “Pts” are represented by the bars and “Rbs” by the lines. Try to reverse them.

You can easily transform your combo chart into a column chart or a line chart just by setting `series='{1: {type:"line"}}'` to 2

Exercise 10

Transform the combo chart into a column chart and then into a line chart.

Data visualizations with googleVis solutions part 2

Below are the solutions to [these](#) exercises on visualizations with googleVis.

```
#####  
#                               #  
#   Exercise 1                 #  
#                               #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```
AreaC <- gvisAreaChart(df)
```

```
#####  
#           #  
#   Exercise 2   #  
#           #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```
AreaC <- gvisAreaChart(df)  
plot(AreaC)
```

```
#####  
#           #  
#   Exercise 3   #  
#           #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```
SteppedAreaC <- gvisSteppedAreaChart(df, xvar="name",  
                                     yvar=c("Pts", "Rbs"))
```

```
#####  
#           #  
#   Exercise 4   #  
#           #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```
SteppedAreaC <- gvisSteppedAreaChart(df, xvar="name",  
                                     yvar=c("Pts", "Rbs"))  
plot(SteppedAreaC)
```

```
#####  
#           #  
#   Exercise 5   #  
#           #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))  
SteppedAreaC <- gvisSteppedAreaChart(df, xvar="name",  
                                     yvar=c("Pts", "Rbs"),  
options=list(isStacked=TRUE))  
plot(SteppedAreaC)
```

```
#####  
#           #  
#   Exercise 6   #  
#           #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))  
ComboC <- gvisComboChart(df, xvar="name",  
                          yvar=c("Pts", "Rbs")  
                          )
```

```
#####  
#           #  
#   Exercise 7   #  
#           #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```

ComboC <- gvisComboChart(df, xvar="name",
                        yvar=c("Pts", "Rbs")
                        )
plot(ComboC)

```

```

#####
#                               #
#   Exercise 8                 #
#                               #
#####

```

```

library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))

```

```

ComboC <- gvisComboChart(df, xvar="name",
                        yvar=c("Pts", "Rbs"),
                        options=list(seriesType="bars",
                                    series='{1:
{type:"line"}}'))
plot(ComboC)

```

```

#####
#                               #
#   Exercise 9                 #
#                               #
#####

```

```

library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))

```

```

ComboC <- gvisComboChart(df, xvar="name",
                        yvar=c("Pts", "Rbs"),
                        options=list(seriesType="line",
                                    series='{1:
{type:"bars"}}'))
plot(ComboC)

```

```
#####  
#           #  
#   Exercise 10   #  
#           #  
#####
```

```
#column chart  
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```
ComboC <- gvisComboChart(df, xvar="name",  
                        yvar=c("Pts", "Rbs"),  
                        options=list(seriesType="bars",  
                                    series='{2:  
{type:"line"}}'))  
plot(ComboC)
```

```
#line chart  
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```
ComboC <- gvisComboChart(df, xvar="name",  
                        yvar=c("Pts", "Rbs"),  
                        options=list(seriesType="line",  
                                    series='{2:  
{type:"bars"}}'))  
plot(ComboC)
```

Data Visualization with

googleVis exercises part 1



Line, Bar and Column Charts

Hello everybody! This is the first part of an exciting data visualization series r_exercises.com is going to provide you.

Data visualization involves the creation and study of the visual representation of data. The increasing amounts of data created by Internet activity make the “data visualization” tool totally necessary for data scientists.

One of the best packages that R language provides for this purpose is **googleVis** and guess what, we are going to see its amazing features together. In this first part we are going to see basic stuff of three useful types of charts (**Line Chart**, **Bar Chart** and **Column Chart**) before “digging deeper” in the next parts.

NOTE: The charts are created locally by your browser. In case they are not displayed at once press F5 to reload the page.

Read the examples below to understand the logic of what we are going to do and then test your skills with the exercise set we prepared for you. Lets begin!

Answers to the exercises are available [here](#).

Install the Package.

Exercise 1

Install and load the package googleVis.

Create a data frame

First of all let's create an experimental data.frame to use for all our plots. This is an example:

```
dfr=data.frame(name=c("GRE", "ARG", "BRA"),
val1=c(20,32,19),
val2=c(25,52,12))
```

Exercise 2

Create a data frame named "df". Give as variables the "Pts" (Points) and "Rbs" (Rebounds) of three NBA players. Names and values are up to you.

Line Chart

To produce a Line Chart you can use:

```
LineC <- gvisLineChart(df)
plot(LineC)
```

Exercise 3

Create a list named "LineC" and pass to it the "df" data frame you just created as a line chart. **HINT:** Use gvisLineChart().

Exercise 4

Plot the line chart. **HINT:** Use plot().

Line chart with two axis

Below there is an example of this type of Line Chart:

```
LineC2 <- gvisLineChart(df, "name", c("val1","val2"),
options=list(
series="[{"targetAxisIndex: 0},
{"targetAxisIndex:1}]",
vAxes="[{"title:'val1'}, {"title:'val2'}]"
))
plot(LineC2)
```

As you can see you create a list (options) that corresponds

values 1 & 2 in the two axes respectively.

Exercise 5

Create a single axis Line chart that displays only the "Pts" of the "df" data frame.

Exercise 6

Now create a two axis line chart that displays both "Pts" and "Rbs" of the "df" data frame. **HINT:** Use `list()`.

Bar Chart

It is quite simple to create a bar chart with `googleVis` with:

```
BarC <- gvisBarChart(df)
plot(BarC)
```

Exercise 7

Create a list named "BarC" and pass to it the "df" data frame you just created as a bar chart. **HINT:** Use `gvisBarChart()`.

Exercise 8

Plot the bar chart. **HINT:** Use `plot()`.

Column Chart

Column charts could be considered as vertical bar charts and are quite simple to be created too.

```
ColumnC <- gvisColumnChart(df)
plot(ColumnC)
```

Exercise 9

Create a list named "ColumnC" and pass to it the "df" data frame you just created as a column chart. **HINT:** Use `gvisColumnChart()`.

Exercise 10

Plot the column chart. **HINT:** Use plot().

Data visualization with googleVis solutions part 1

Below are the solutions to [these](#) exercises on googleVis Visualizations.

```
#####  
#           #  
# Exercise 1 #  
#           #  
#####
```

```
install.packages("googleVis")  
library(googleVis)
```

```
#####  
#           #  
# Exercise 2 #  
#           #  
#####
```

```
library(googleVis)  
df=data.frame(name=c("James", "Curry", "Harden"),  
              Pts=c(20,23,34),  
              Rbs=c(13,7,9))
```

```
#####  
#           #  
# Exercise 3 #  
#           #  
#####
```

```
library(googleVis)
```

```
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
LineC <- gvisLineChart(df)
```

```
#####
#                               #
#   Exercise 4                 #
#                               #
#####
```

```
library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
LineC <- gvisLineChart(df)
plot(LineC)
```

```
#####
#                               #
#   Exercise 5                 #
#                               #
#####
```

```
library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
LineC2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
                        )
plot(LineC2)
```

```
#####
#                               #
#   Exercise 6                 #
#                               #
#####
```

```
library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
```

```
      Rbs=c(13,7,9))
LineC2 <- gvisLineChart(df, "name", c("Pts","Rbs"),
                        options=list(
                          series="[{targetAxisIndex: 0},
                          {targetAxisIndex:1}]",
                          vAxes="[{title:'Pts'},
                          {title:'Rbs'}]"]
                        ))
plot(LineC2)
```

```
#####
#                               #
#   Exercise 7                 #
#                               #
#####
```

```
library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
BarC <- gvisBarChart(df)
```

```
#####
#                               #
#   Exercise 8                 #
#                               #
#####
```

```
library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
BarC <- gvisBarChart(df)
plot(BarC)
```

```
#####
#                               #
#   Exercise 9                 #
#                               #
#####
```

```
library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
ColumnC <- gvisColumnChart(df)
```

```
#####
#                               #
#   Exercise 10                 #
#                               #
#####
```

```
library(googleVis)
df=data.frame(name=c("James", "Curry", "Harden"),
              Pts=c(20,23,34),
              Rbs=c(13,7,9))
ColumnC <- gvisColumnChart(df)
plot(ColumnC)
```