

Mam, I am taking the msleep data-set for visualization purpose

Description of the data-set

This is an updated and expanded version of the mammals sleep dataset. Updated sleep times and weights were taken from V. M. Savage and G. B. West. A quantitative, theoretical framework for understanding mammalian sleep. Proceedings of the National Academy of Sciences, 104 (3):1051-1056, 2007.

Usage

msleep

Format

A data frame with 83 rows and 11 variables:

name
common name
genus
vore
carnivore, omnivore or herbivore?
order
conservation
the conservation status of the animal
sleep_total
total amount of sleep, in hours
sleep_rem
rem sleep, in hours
sleep_cycle
length of sleep cycle, in hours
awake
amount of time spent awake, in hours
brainwt
brain weight in kilograms
bodywt
body weight in kilograms

Details

Additional variables order, conservation status and vore were added from wikipedia.

Loading the libraries and the data-set

```
## {r}
library(ggplot2) ## msleep is included in ggplot-2
library(plotly)
library(data.table)
##
```

```
## {r}
print(paste("Attributes of the data-set"))
colnames(df)
##
```

```
[1] "Attributes of the data-set"
[1] "name"      "genus"      "vore"      "order"      "conservation" "sleep_total"
[7] "sleep_rem" "sleep_cycle" "awake"     "brainwt"   "bodywt"
```

```
## {r}
str(df)
##
```

```
tibble [83 × 11] (S3: tbl_df/tbl/data.frame)
 $ name      : chr [1:83] "Cheetah" "Owl monkey" "Mountain beaver" "Greater short-tailed shrew" ...
 $ genus     : chr [1:83] "Acinonyx" "Aotus" "Aplodontia" "Blarina" ...
 $ vore      : chr [1:83] "carni" "omni" "herbi" "omni" ...
 $ order     : chr [1:83] "Carnivora" "Primates" "Rodentia" "Soricomorpha" ...
 $ conservation: chr [1:83] "lc" NA "nt" "lc" ...
 $ sleep_total : num [1:83] 12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...
 $ sleep_rem  : num [1:83] NA 1.8 2.4 2.3 0.7 2.2 1.4 NA 2.9 NA ...
 $ sleep_cycle : num [1:83] NA NA NA 0.133 0.667 ...
 $ awake      : num [1:83] 11.9 7 9.6 9.1 20 9.6 15.3 17 13.9 21 ...
 $ brainwt    : num [1:83] NA 0.0155 NA 0.00029 0.423 NA NA NA 0.07 0.0982 ...
 $ bodywt     : num [1:83] 50 0.48 1.35 0.019 600 ...
```

1) Scatter Plot

The screenshot shows a web application interface for creating a scatter plot. The interface includes a file list on the left, a code editor in the center, and a description panel on the right.

File list:

- scatterplot.R
- boxplot.R
- donut.R
- barplot.R
- piechart.R
- histogram.R

Code editor:

```
1 library(ggplot2) ## msleep is included in ggplot-2
2 library(plotly)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7
8 scatter_plot = plot_ly(data=msleep, x=~log10(bodywt), y=~log10(brainwt), color = ~name,
9                        type="scatter", mode="markers") %>%
10 layout(
11   title= list(text = "<b>Body weight vs Brain weight"),
12   legend = list(title = list(text = "<b>Animals"),
13   xaxis = list(title = list(text = "<b>Brain Weight"),
14   yaxis = list(title = list(text = "<b>Body Weight"))))
15
16
17 scatter_plot
18
19
```

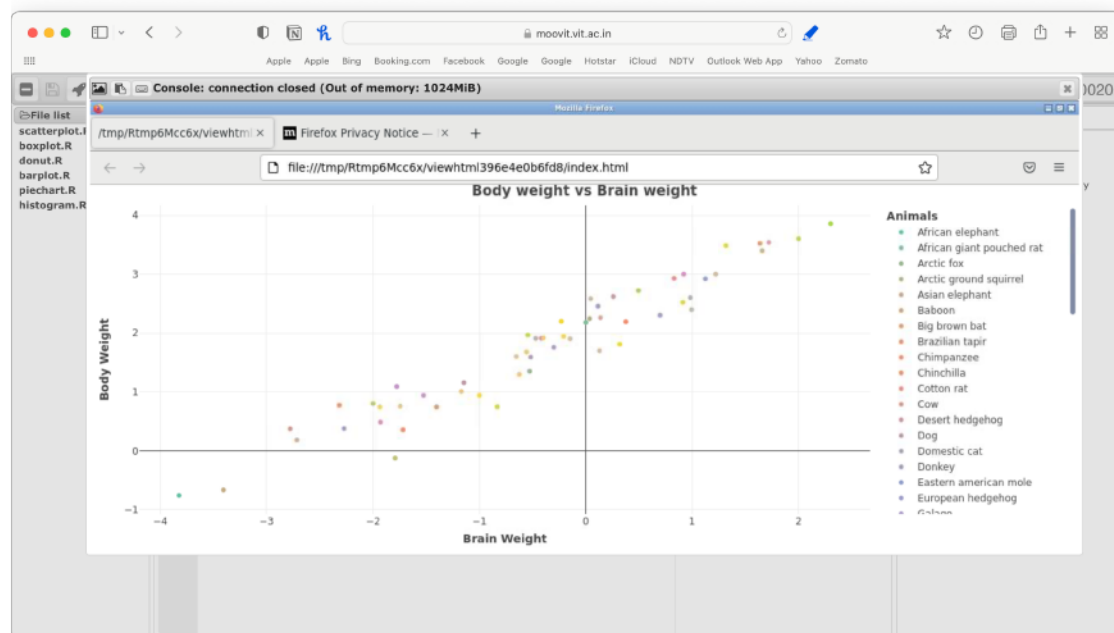
Description:

Use any of the inbuilt datasets(except iris) and try the following basic graphs using plotly

- 1) Scatter plot
- 2) Bar graph(regular, stacked, group)
- 3) Histogram
- 4) Pie chart
- 5) donut chart
- 6) Box plot

and any others.

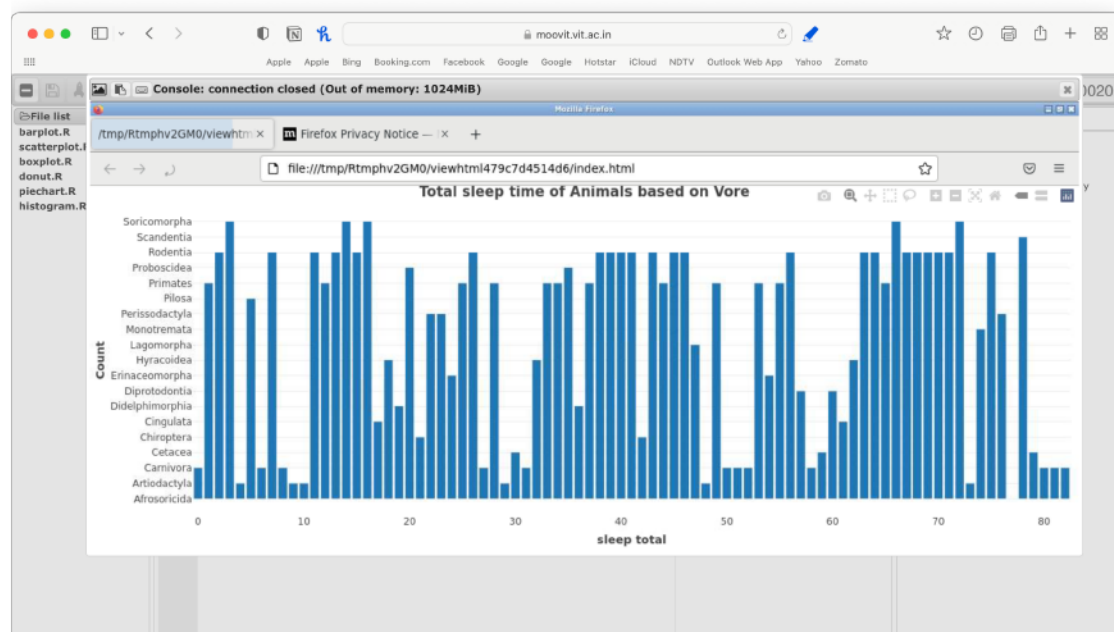
Upload this exercise on VTOP



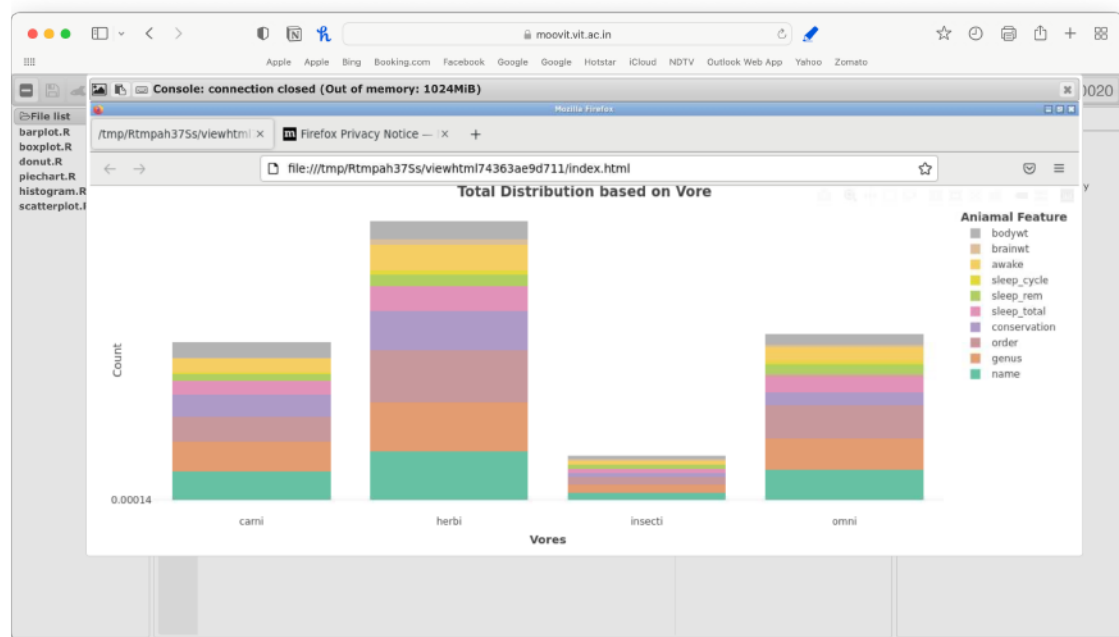
1) Bar plot

```
1 ## Bar graph
2 library(ggplot2)
3 library(datasets)
4 library(data.table)
5 library(plotly)
6 library(dplyr)
7
8 r_barchart = plot_ly(data=df, y=order, type="bar") %>%
9   layout(
10     title = list(text = "<b>Total sleep time of Animals based on Vore"),
11     legend = list(title = list(text= "<b>Vore"),
12     xaxis = list(title = list(text = "<b>sleep total", tickformat = "%"),
13     yaxis = list(title = list(text = "<b>Count")))
14
15 r_stack_barchart = data.table::melt(df, id.vars='vore') %>%
16 plot_ly(x = ~vore, y = ~value, type = "bar", name = ~variable, color = ~variable) %>%
17   layout(
18     title = list(text = "<b>Total Distribution based on Vore"),
19     legend = list(title = list(text= "<b>Animal Feature"),
20     xaxis = list(title = list(text = "<b>Vores"),
21     yaxis = list(title = list(text = "<b>Count", text="<b>Count"), barmode = 'stack')
22
23
24 r_group_barchart = data.table::melt(df, id.vars='vore') %>%
25 plot_ly(x = ~vore, y = ~value, type = "bar", name = ~variable, color = ~variable) %>%
26   layout(
27     title = list(text = "<b>Total Distribution based on Vore"),
28     legend = list(title = list(text= "<b>Animal Feature"),
29     xaxis = list(title = list(text = "<b>Vores"),
30     yaxis = list(title = list(text = "<b>Count", text="<b>Count"), barmode = 'group')
31
32 r_barchart
33 r_stack_barchart
34 r_group_barchart
35
```

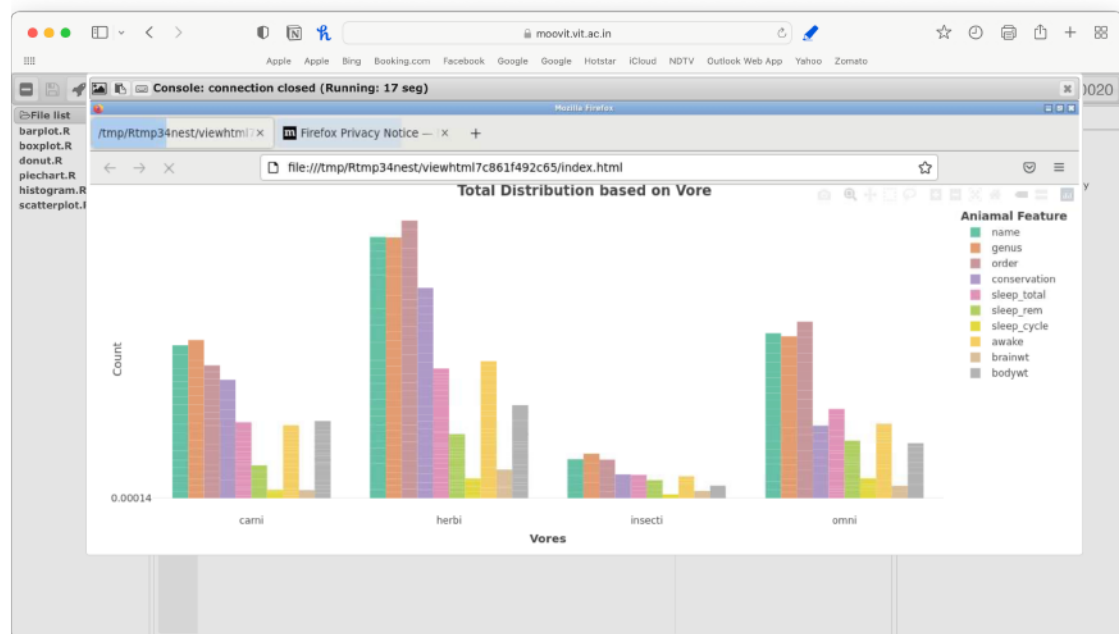
Executing the code one-by-one plots
Bar-Chart



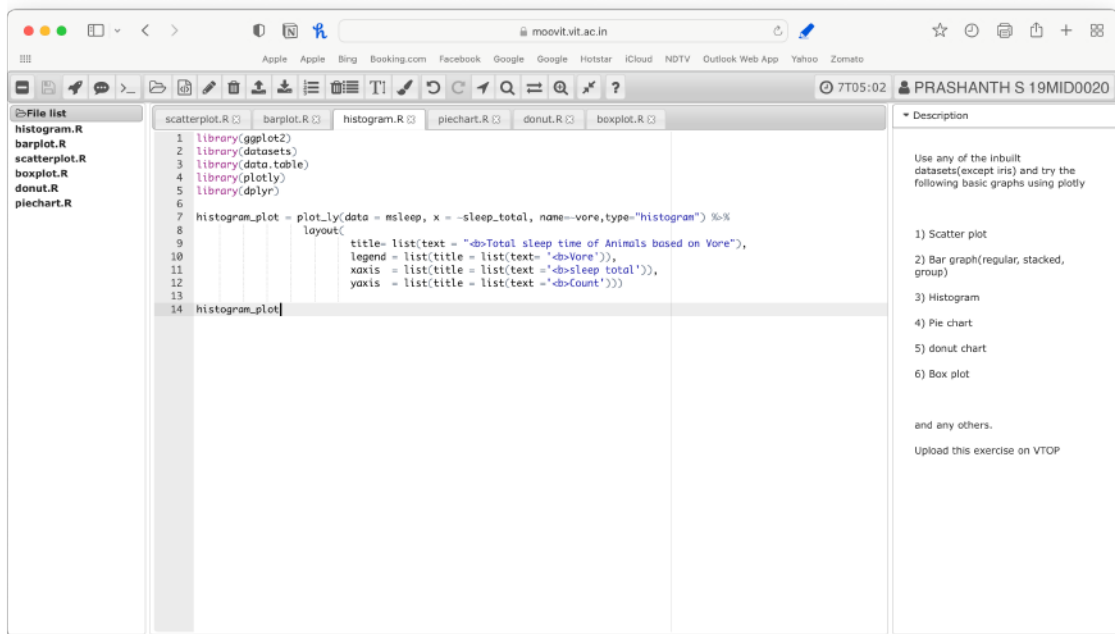
Stacked Bar-Chart



Grouped Bar-Chart



2) Histogram Plot



The screenshot shows a web-based R IDE interface. The top bar includes a browser address bar with 'moovit.vit.ac.in' and a toolbar with various icons. Below the toolbar is a file explorer on the left showing a list of files: 'histogram.R', 'barplot.R', 'scatterplot.R', 'boxplot.R', 'donut.R', and 'piechart.R'. The main editor area displays the following R code:

```
1 library(ggplot2)
2 library(datasets)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7 histogram_plot = plot_ly(data = msleep, x = ~sleep_total, name=~vore,type="histgram") %>%
8   layout(
9     title= list(text = "<div>Total sleep time of Animals based on Vore"),
10    legend = list(title= "<div>Vore"),
11    xaxis = list(title = list(text = "<div>sleep total")),
12    yaxis = list(title = list(text = "<div>Count"))
13  )
14 histogram_plot
```

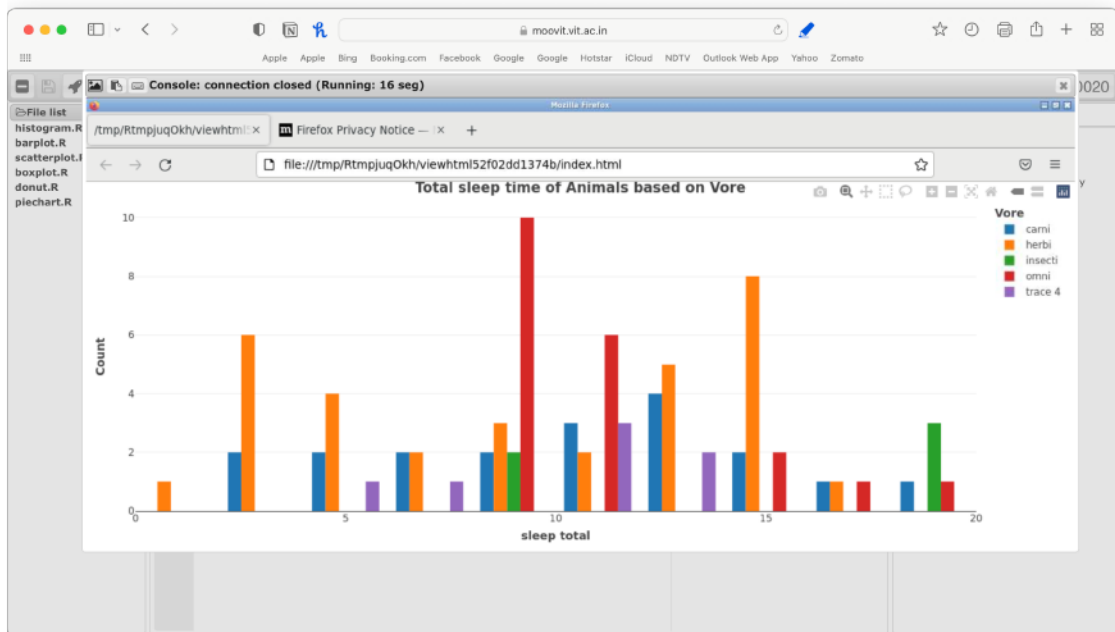
On the right side, there is a 'Description' panel with the following text:

Use any of the inbuilt datasets(except iris) and try the following basic graphs using plotly

- 1) Scatter plot
- 2) Bar graph(regular, stacked, group)
- 3) Histogram
- 4) Pie chart
- 5) donut chart
- 6) Box plot

and any others.

Upload this exercise on VTOP



3) Pie-Chart

moovit.vit.ac.in

Apple Bing Booking.com Facebook Google Hotstar iCloud NDTV Outlook Web App Yahoo Zomato

7:05:01 PRASHANTH S 19MID0020

File list: piechart.R, histogram.R, barplot.R, scatterplot.R, boxplot.R, donut.R

```
1 library(ggplot2)
2 library(datasets)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7 ## pie-chart
8 df_order = data.frame(table(msleep$order))
9 pie_chart = plot_ly(type="pie", labels=df_order$Var1, values=df_order$Freq,
10                    textinfo="label+percent", insidetextorientation="radial") %>%
11                    layout(
12                      title = list(text = "<b>Order Distributions"),
13                      legend = list(title = list(text = "<b>Order")))
14
15 pie_chart
16
```

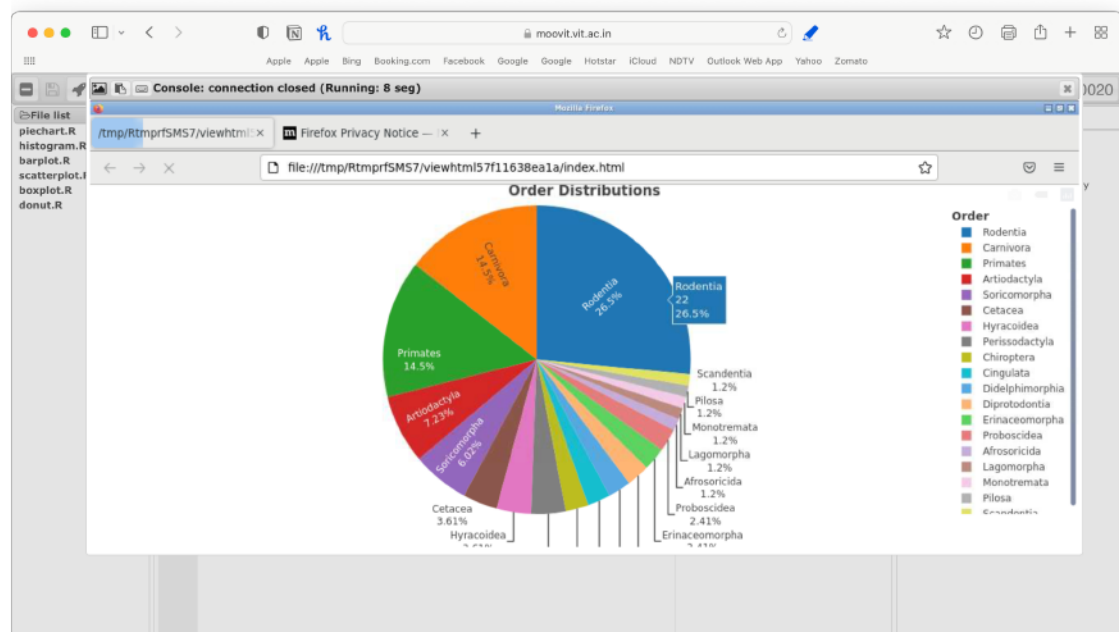
Description

Use any of the inbuilt datasets(except iris) and try the following basic graphs using plotly

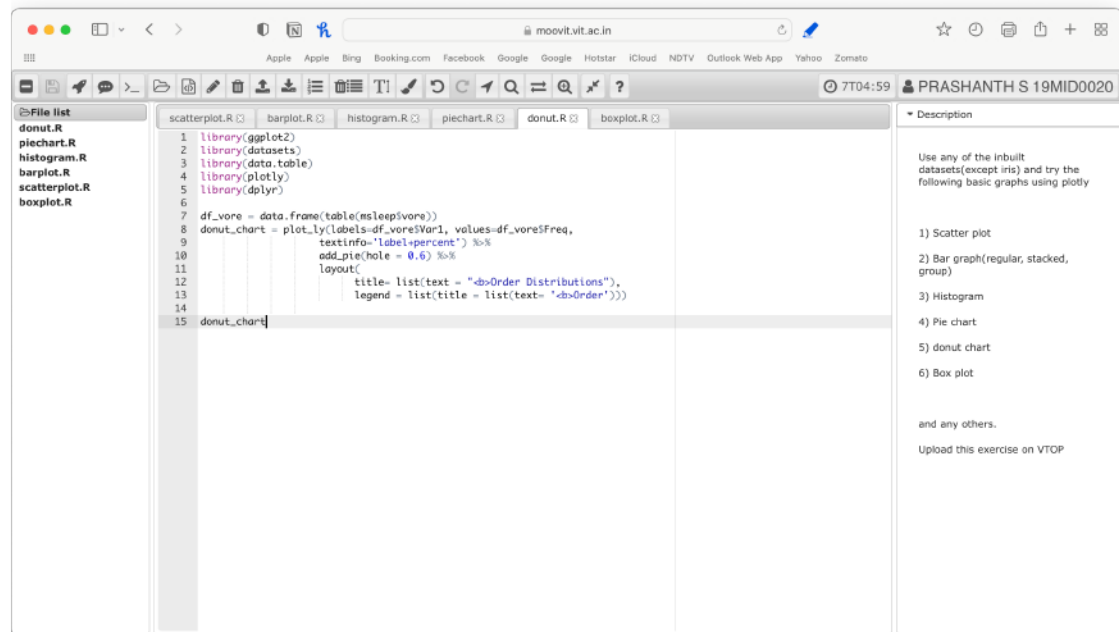
- 1) Scatter plot
- 2) Bar graph(regular, stacked, group)
- 3) Histogram
- 4) Pie chart
- 5) donut chart
- 6) Box plot

and any others.

Upload this exercise on VTOP

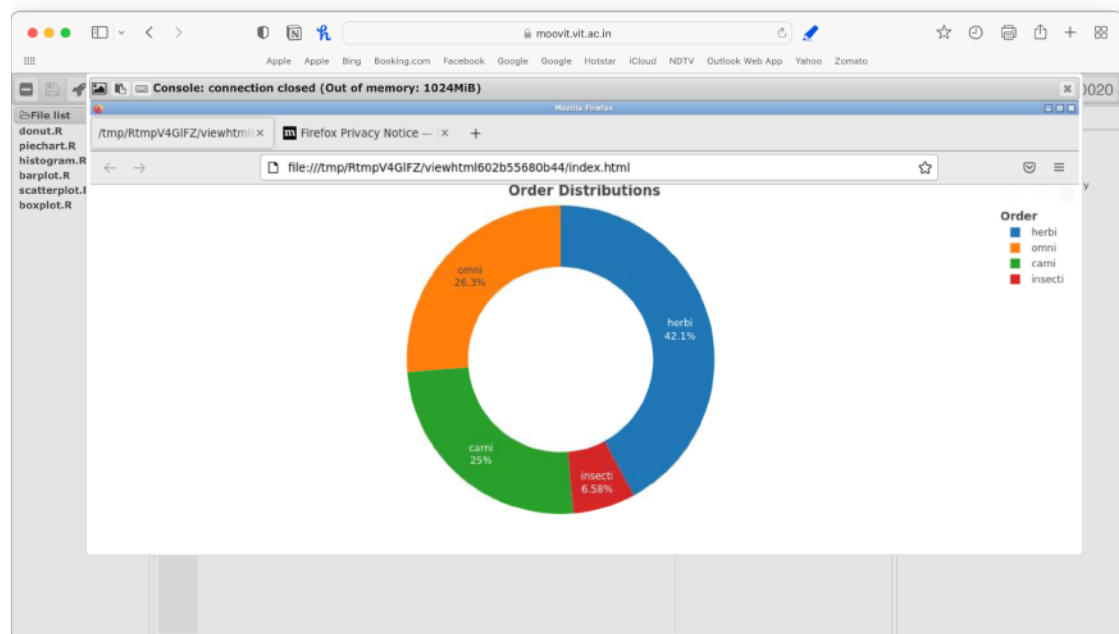


4) Donut chart



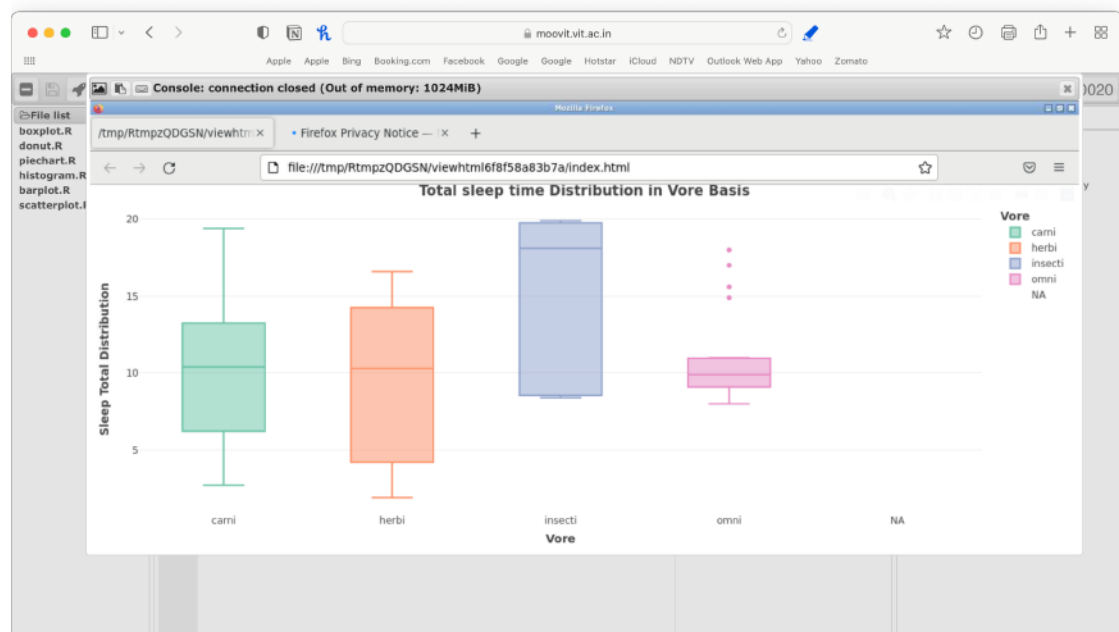
The screenshot shows a web-based R IDE interface. The top bar includes a browser address bar with 'moovit.vit.ac.in' and a toolbar with various icons. Below the toolbar is a file explorer on the left showing a list of files: 'donut.R', 'piechart.R', 'histogram.R', 'barplot.R', 'scatterplot.R', and 'boxplot.R'. The main editor area displays the R code for creating a donut chart. The code uses 'ggplot2' and 'plotly' libraries, loads the 'msleep' dataset, and creates a donut chart titled 'Order Distributions' with a legend. The right sidebar contains a 'Description' section with instructions on how to use the built-in datasets and a list of exercises: 1) Scatter plot, 2) Bar graph (regular, stacked, group), 3) Histogram, 4) Pie chart, 5) donut chart, 6) Box plot, and 'and any others.' It also mentions 'Upload this exercise on VTOP'.

```
1 library(ggplot2)
2 library(datasets)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7 df_vore = data.frame(table(msleep$vore))
8 donut_chart = plot_ly(labels=df_vore$Var1, values=df_vore$Freq,
9                       textinfo="label+percent") %>%
10                      add_pie(hole = 0.6) %>%
11                      layout(
12                        title= list(text = "<b>Order Distributions"),
13                        legend = list(title = list(text= "<b>Order")))
14
15 donut_chart
```



5) Box Plot

```
1 library(ggplot2)
2 library(datasets)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7 box_plot=plot_ly(data=msleep, y=msleep$total_sleep_time, color=msleep$vore, type='box') %>%
8   layout(
9     title= list(text = "Total sleep time Distribution in Vore Basis"),
10    legend = list(title = list(text= "Vore"),
11      xaxis = list(title = list(text = "Vore")),
12      yaxis = list(title = list(text = "Sleep Total Distribution")))
13
14 box_plot
```



Mam, I am also including the work done on <https://moovit.vit.ac.in/>

19MID0020 Plots

Code

```
library(ggplot2) ## msleep is included in ggplot2
library(plotly)
```

Hide

Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':

last_plot

The following object is masked from 'package:stats':

filter

The following object is masked from 'package:graphics':

layout

Hide

```
library(data.table)
```

data.table 1.14.2 using 1 threads (see ?getDTthreads). Latest news: r-datatable.com

This installation of data.table has not detected OpenMP support. It should still work but in single-threaded mode.
This is a Mac. Please read https://mac.r-project.org/openmp/. Please engage with Apple and ask them for support.
Check r-datatable.com for updates, and our Mac instructions here: https://github.com/Rdatatable/data.table/wiki/Installation.
After several years of many reports of installation problems on Mac, it's time to gingerly point out that there have been no similar problems on Windows or Linux.

Hide

df = msleep

Hide

```
print(paste("Attributes of the data-set"))
```

[1] "Attributes of the data-set"

Hide

colnames(df)

[1] "name" "genus" "vore" "order" "conservation" "sleep_total"
[7] "sleep_cycle" "sleep_rem" "awake" "brainwt" "bodywt"

Hide

str(df)

```
tibble [83 x 11] (S3: tbl_df/tbl/data.frame)
 $ name      : chr [1:83] "Cheetah" "Owl monkey" "Mountain beaver" "Greater short-tailed shrew" ...
 $ genus     : chr [1:83] "Acinonyx" "Aotus" "Aplodontia" "Blarina" ...
 $ vore      : chr [1:83] "carni" "omni" "herbi" "omni" ...
 $ order     : chr [1:83] "Carnivora" "Primates" "Rodentia" "Soricomorpha" ...
 $ conservation: chr [1:83] "lc" NA "lc" NA "lc" NA "lc" NA ...
 $ sleep_total : num [1:83] 12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...
 $ sleep_rem  : num [1:83] NA NA NA 0.133 0.667 ...
 $ sleep_cycle : num [1:83] 11.9 7 9 9.1 28 9 8 18.2 17 13.9 21 ...
 $ awake     : num [1:83] NA 0.0155 NA 0.0802 0.425 NA NA 0.07 0.0982 ...
 $ brainwt   : num [1:83] 50 0.48 1.35 0.019 600 ...
 $ bodywt    : num [1:83] 50 0.48 1.35 0.019 600 ...
```

1) Scatter Plot

Hide

```
scatter_plot = plot_ly(data=msleep, x=~(-log10(bodywt)), y=~(-log10(brainwt)), color = ~name,
  type='scatter',mode='markers') %>%
  layout(
    title= list(text = "<b>Body weight vs Brain weight"),
    legend = list(title = list(text = "<b>Animals"),
      xaxis = list(title = list(text = "<b>Brain Weight"),
        yaxis = list(title = list(text = "<b>Body Weight"))
```

Hide

Warning: Ignoring 27 observations
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors

Warning: Ignoring 27 observations
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors

Warning: Ignoring 27 observations
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors

Warning: Ignoring 27 observations
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors

2) Barchart

Hide

```
head(df)
```

name genus vore order conservation sleep_total sleep_rem
<chr> <chr> <chr> <chr> <chr> <dbl> <dbl>
Cheetah Acinonyx carni Carnivora lc 12.1 NA
Owl monkey Aotus omni Primates NA 17.0 1.8
Mountain beaver Aplodontia herbi Rodentia nt 14.4 2.4
Greater short-tailed shrew Blarina omni Soricomorpha lc 14.9 2.3
Cow Bos herbi Artiodactyla domesticated 4.0 0.7
Three-toed sloth Bradypus herbi Pilosa NA 14.4 2.2
6 rows [1-7 of 11 columns]

Hide

2) a) Regular Barchart

Hide

```
r_barchart = plot_ly(data=df, y=order, type="bar") %>%
  layout(
    title= list(text = "<b>Total sleep time of Animals based on Vore"),
    legend = list(title = list(text= '<b>Vore'),
      xaxis = list(title = list(text = "<b>sleep total", tickformat = "%"),
        yaxis = list(title = list(text = "<b>Count"))
```

Hide

2) b) Stacked Barchart

Hide

```
r_stack_barchart = data.table::melt(df, id.vars="vore") %>%
  plot_ly(x = ~vore, y = ~value, type = 'bar', name = ~variable, color = ~variable) %>%
  layout(
    title= list(text = "<b>Total Distribution based on Vore"),
    legend = list(title = list(text= '<b>Vores'),
      xaxis = list(title = list(text = "<b>Vores"),
        yaxis = list(title = list(text = "<b>Count"), barnode = 'stack')
```

Hide

Warning in data.table::melt(df, id.vars = "vore") :
The melt generic in data.table has been passed a tbl_df and will attempt to redirect to the relevant reshape2 method; please note that reshape2 is deprecated, and this redirection is now deprecated as well. To continue using melt methods from reshape2 while both libraries are attached, e.g. melt.list, you can prepend the namespace like reshape2::melt(df). In the next version, this warning will become an error.

Hide

```
r_stack_barchart
```

Warning: Ignoring 184 observations
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors

2) c) Grouped Barchart

Hide

```
r_group_barchart = data.table::melt(df, id.vars="vore") %>%
  plot_ly(x = ~vore, y = ~value, type = 'bar', name = ~variable, color = ~variable) %>%
  layout(
    title= list(text = "<b>Total Distribution based on Vore"),
    legend = list(title = list(text= '<b>Animal Feature'),
      xaxis = list(title = list(text = "<b>Vores"),
        yaxis = list(title = list(text = "<b>Count"), barnode = 'group')
```

Hide

Warning in data.table::melt(df, id.vars = "vore") :
The melt generic in data.table has been passed a tbl_df and will attempt to redirect to the relevant reshape2 method; please note that reshape2 is deprecated, and this redirection is now deprecated as well. To continue using melt methods from reshape2 while both libraries are attached, e.g. melt.list, you can prepend the namespace like reshape2::melt(df). In the next version, this warning will become an error.

Hide

```
r_group_barchart
```

Warning: Ignoring 184 observations
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
Warning in RColorBrewer::brewer.pal(N, "Set2") :
n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors

3) Histogram

Hide

```
# Define labels for the bars
labs = c('herbi'='Herbivore',
  'carni'='Carnivore',
  'omni'='Omnivore',
  'insecti'='Insectivore')
```

Hide

Hide

```
histogram_plot = plot_ly(data = df, x = ~sleep_total, name=~vore,type="histogram") %>%
  layout(
    title= list(text = "<b>Total sleep time of Animals based on Vore"),
    legend = list(title = list(text= '<b>Vore'),
      xaxis = list(title = list(text = "<b>sleep total"),
        yaxis = list(title = list(text = "<b>Count"))
```

Hide

4) Pie-Chart

Hide

```
df_order = data.frame(table(msleep$order))
df_order
```

Var1 Freq
<chr> <int>
Afrosoricida 1
Artiodactyla 6
Carnivora 12
Cetacea 3
Chiroptera 2
Cingulata 2
Didelphimorphia 2
Diprotodontia 2
Eumecometomorphia 2
Hyracoida 3
1-10 of 19 rows Previous 1 2 Next

Hide

Hide

```
pie_chart = plot_ly(type="pie", labels=df_order$Var1, values=df_order$Freq,
  textinfo="label+percent",insidetextorientation="radial") %>%
  layout(
    title= list(text = "<b>Order Distributions"),
    legend = list(title = list(text= '<b>Order'))
```

Hide

5) Donut Chart / Open Pie-Chart

Hide

```
df_vore = data.frame(table(msleep$vore))
df_vore
```

Var1 Freq
<chr> <int>
carni 19
herbi 32
insecti 5
omni 20
4 rows

Hide

Hide

```
donut_chart = plot_ly(labels=df_vore$Var1, values=df_vore$Freq,
  textinfo="label+percent") %>%
  add_pie(hole = 0.6) %>%
  layout(
    title= list(text = "<b>Order Distributions"),
    legend = list(title = list(text= '<b>Order'))
```

Hide

6) Box Plot

Hide

```
box_plot=plot_ly(data=msleep, y=~sleep_total, color=~vore, type='box') %>%
  layout(
    title= list(text = "<b>Total sleep time Distribution in Vore Basis"),
    legend = list(title = list(text= '<b>Vore'),
      xaxis = list(title = list(text = "<b>Vore"),
        yaxis = list(title = list(text = "<b>Sleep Total Distribution"))
```

Hide