

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 ggplot2
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 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 "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

File list

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

scatterplot.R

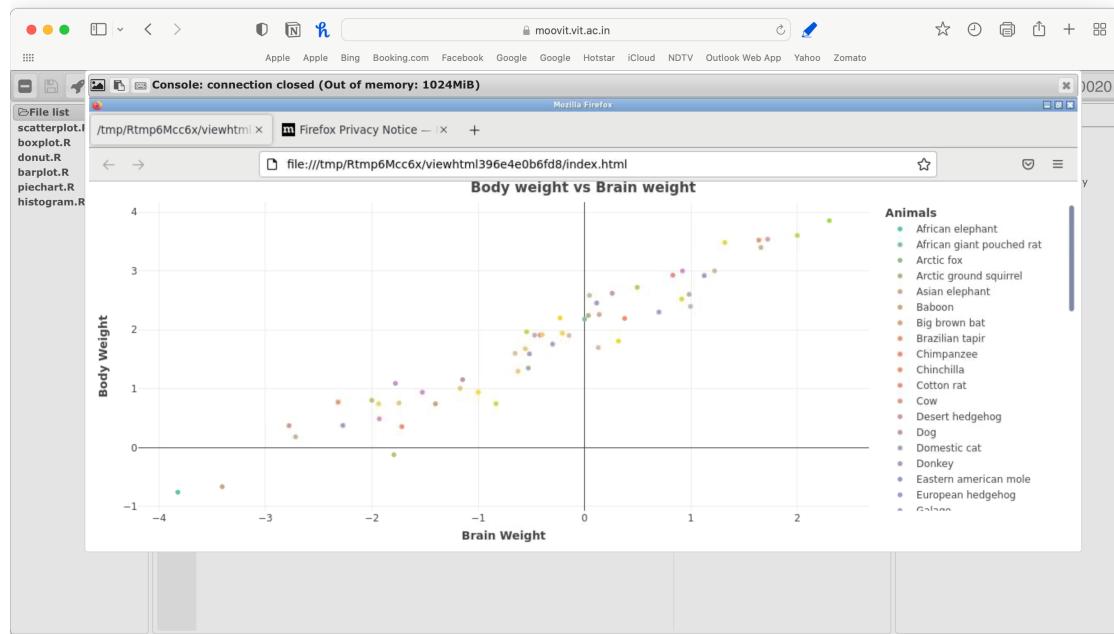
```
1 library(ggplot2) ## msleep is included in ggplot-2
2 library(plotly)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7
8 scatter_plot = plot_ly(data=msleep, x=-(-log10(bodywt)), y=-(-log10(brainwt)), color = ~name,
9   type='scatter', mode='markers') %>%
10   layout(
11     title= list(text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

The screenshot shows an RStudio interface. On the left, a file list pane contains files: scatterplot.R, boxplot.R, donut.R, barplot.R, piechart.R, and histogram.R. The main pane displays R code for creating a bar plot:

```
## Bar graph
library(ggplot2)
library(datasets)
library(data.table)
library(plotly)
library(dplyr)

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

r_stack_barplot = 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</b>"),
    legend = list(title = list(text= "<b>Animal Feature</b>")),
    xaxis = list(title = list(text = "<b>Vores</b>")),
    yaxis = list(title='Count', text=<b>Count</b>), barmode = 'stack')

r_group_barplot = 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</b>"),
    legend = list(title = list(text= "<b>Animal Feature</b>")),
    xaxis = list(title = list(text = "<b>Vores</b>")),
    yaxis = list(title='Count', text=<b>Count</b>), barmode = 'group')

r_barplot
r_stack_barplot
r_group_barplot
```

To the right, a panel titled "Description" provides instructions:

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

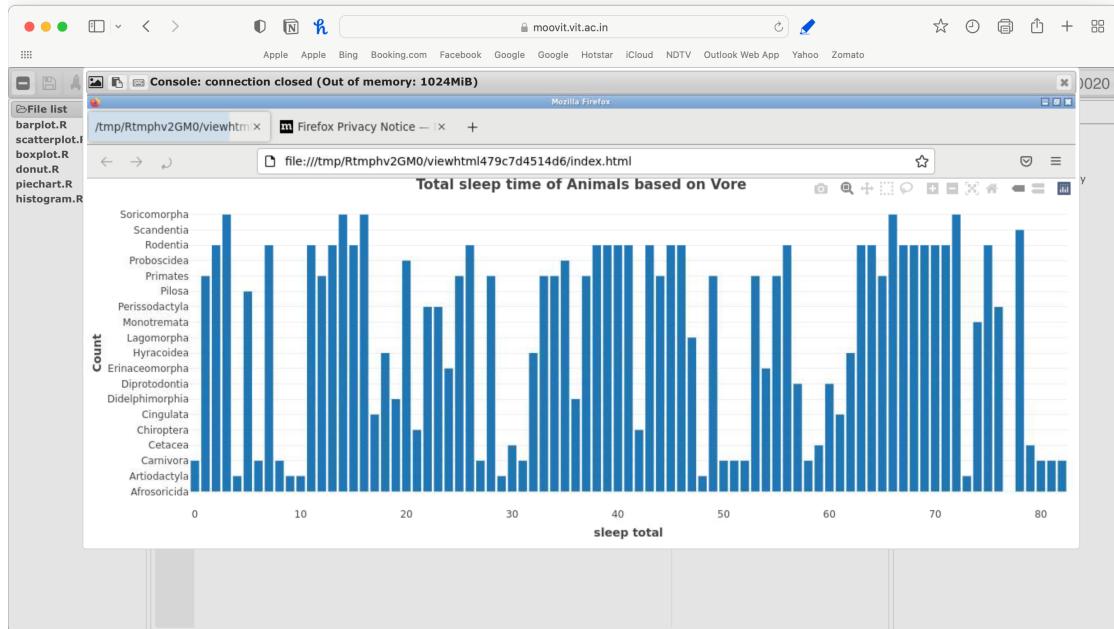
- 1) Scatter plot
- 2) Bar graph(regular, stacked, group)
- 3) Histogram
- 4) Pie chart
- 5) donut chart
- 6) Box plot

and any others.

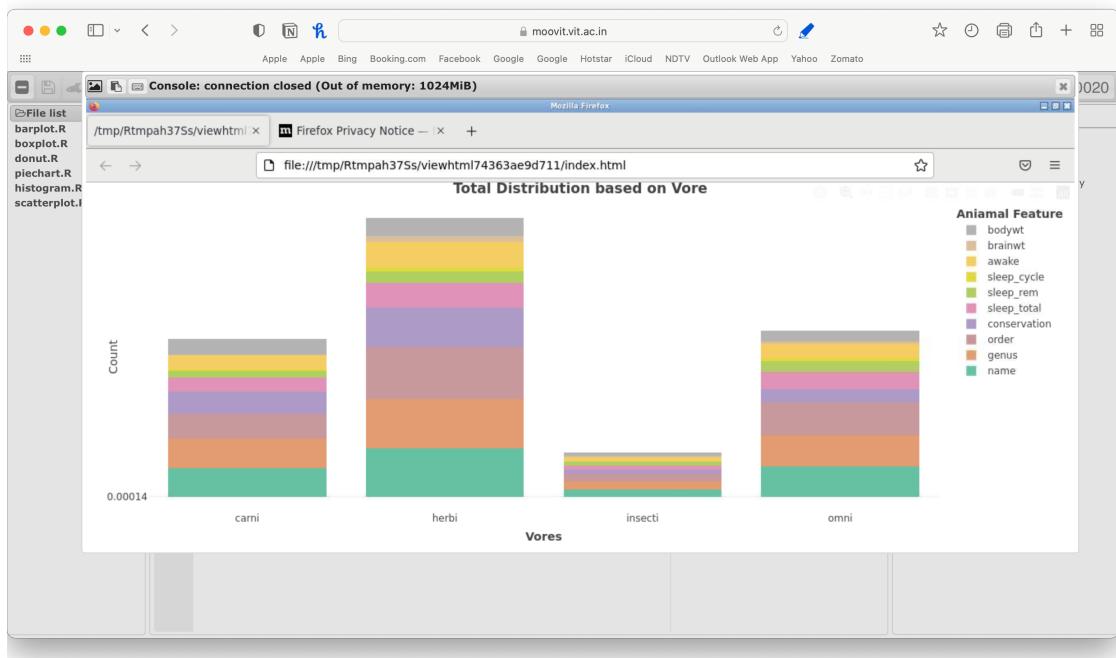
Upload this exercise on VTOP

Executing the code one-by-one plots

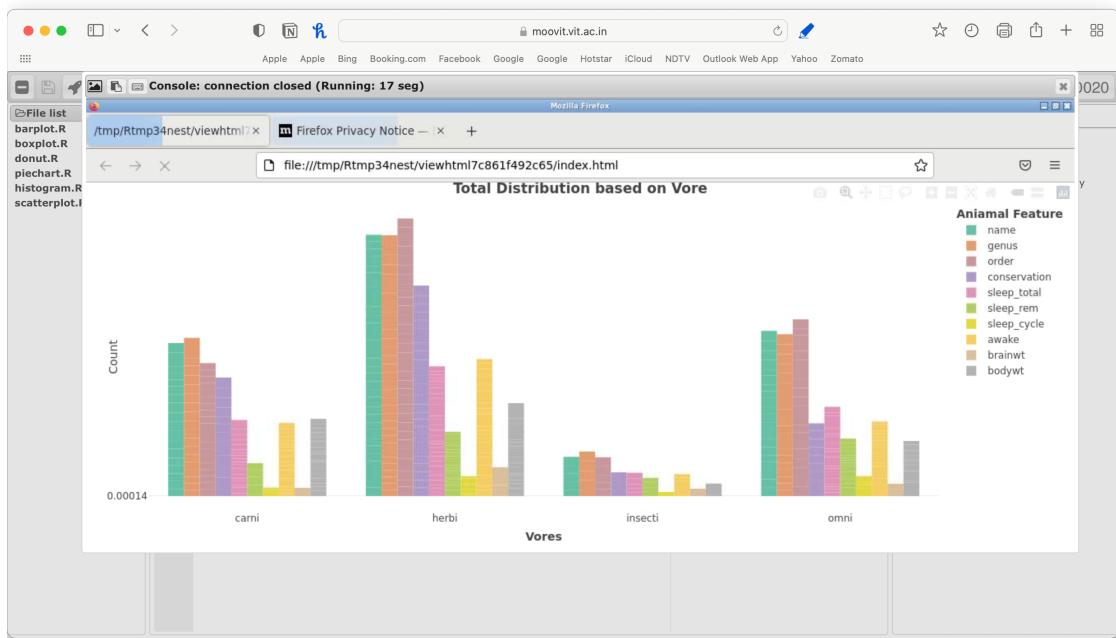
Bar-Chart



Stacked Bar-Chart



Grouped Bar-Chart



2) Histogram Plot

The screenshot shows the RStudio interface. On the left, the 'File list' pane contains files: histogram.R, barplot.R, scatterplot.R, boxplot.R, donut.R, and piechart.R. The main code editor pane displays the following R code:

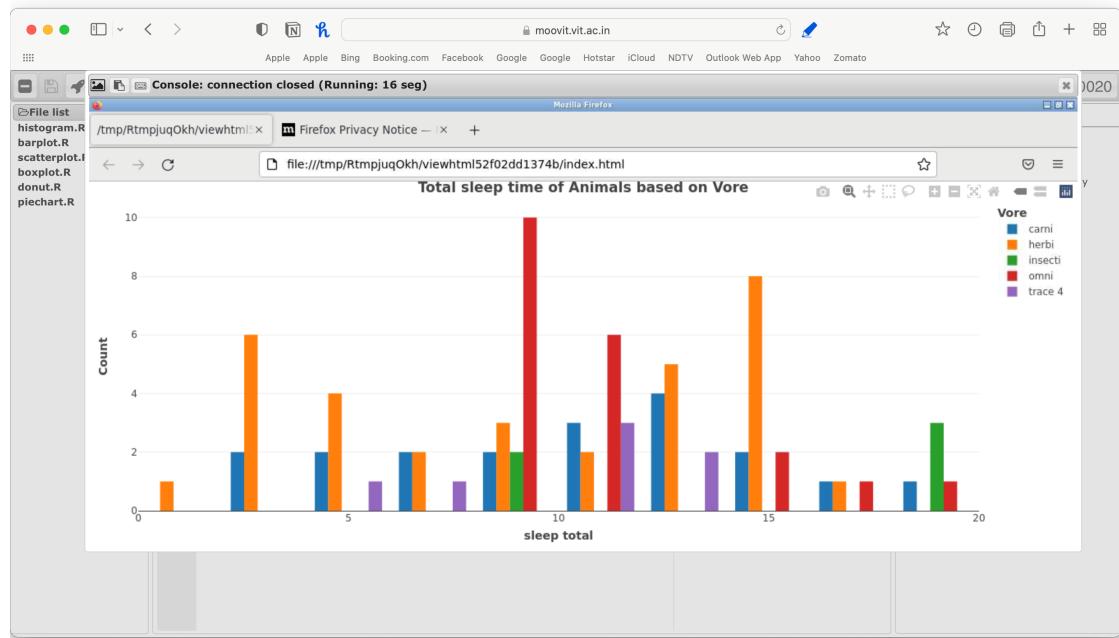
```
1 library(ggplot2)
2 library(datasets)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7 histogram_plot = plot_ly(data = msleep, x = ~sleep_total, name=vore, type="histogram") %>%
8   layout(
9     title= list(text = "<b>Total sleep time of Animals based on Vore"),
10    legend = list(title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

To the right of the code editor is a sidebar with the following content:

- 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

Screenshot of RStudio showing the code for generating a pie chart and its execution results.

The code in the script pane:

```
library(ggplot2)
library(datasets)
library(data.table)
library(plotly)
library(dplyr)

## pie-chart
df_order<-data.frame(table(msleep$order))
pie_chart = plot_ly(type='pie', labels=df_order$Var1, values=df_order$Freq,
textinfo="label+percent", insidetextorientation='radial') %>%
layout(
    title= list(text = "<b>Order Distributions</b>"),
    legend = list(title = list(text= '<b>Order</b>')))

pie_chart
```

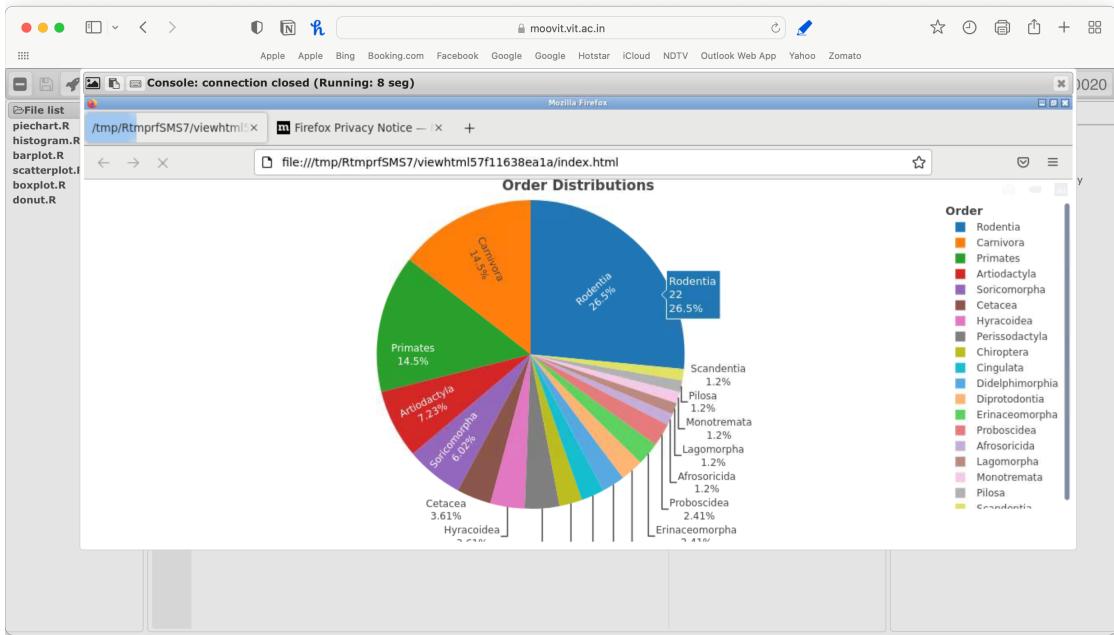
The results pane shows the generated pie chart titled "Order Distributions".

Description pane:

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

Screenshot of a web-based R environment showing a donut chart exercise.

The browser title bar says "moovit.vit.ac.in". The address bar also shows "moovit.vit.ac.in". The top menu bar includes links for Apple, Bing, Booking.com, Facebook, Google, Hotstar, iCloud, NDTV, Outlook Web App, Yahoo, and Zomato. The top right corner shows the time "7T04:59" and a user profile "PRASHANTH S 19MID020".

The left sidebar lists files: scatterplot.R, barplot.R, histogram.R, piechart.R, scatterplot.R, donut.R, boxplot.R.

The main code editor window contains the following R code:

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

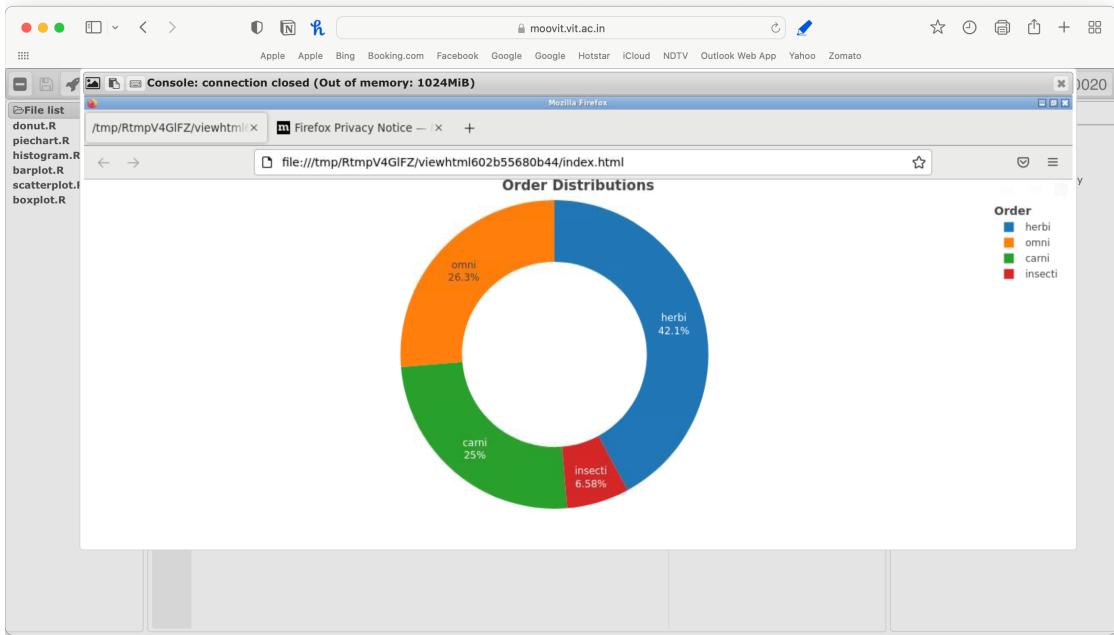
The right panel contains a "Description" section with instructions:

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



5) Box Plot

The screenshot shows an RStudio interface. On the left, a file list pane contains files: scatterplot.R, barplot.R, histogram.R, piechart.R, donut.R, boxplot.R, and scatterplot.R. The main code editor pane displays the following R code:

```
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=sleep_total, color=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|
```

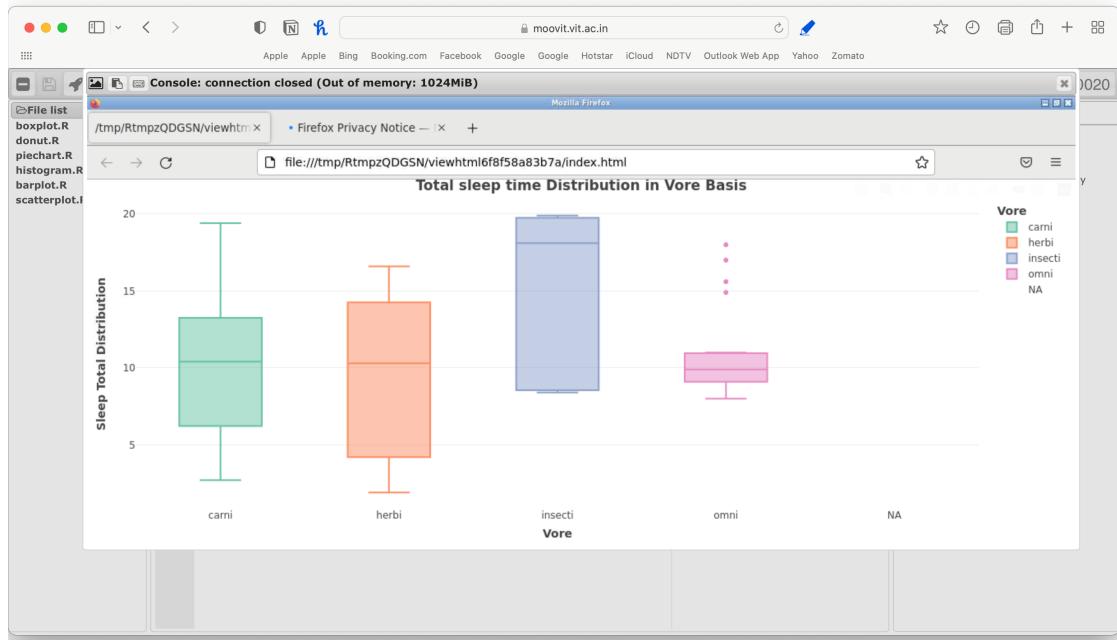
To the right of the code editor is a panel titled "Description" with the following text:

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



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

19MID0020 Plots

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

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

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

data.table 1.14.2 using 1 threads (see ?getDTHThreads). Latest news: r-data.table.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-data.table.com for updates, and our Mac instructions here: <https://github.com/Rdatatable/data.table/wiki/I-nstallation>. 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.

```
df = msleep
```

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

```
[1] "Attributes of the data-set"
```

```
colnames(df)
```

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

```
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"        "nt"         "lc"        ...
 $ sleep_total: num [1:83] 17.0 14.4 14.4 8.7 7.1 1.3 ...
 $ sleep_rem: num [1:83] NA 1.2 2.1 2.7 6.7 2.2 1.4 NA 2.9 NA ...
 $ sleep_cycle: num [1:83] 11.9 7.9 6.1 15.3 17.1 13.9 21 ...
 $ awake : num [1:83] 0.0155 NA 0.0029 0.423 NA NA 0.07 0.0982 ...
 $ brainwt: num [1:83] 50 0.48 1.35 0.619 600 ...
 $ bodywt: num [1:83] 50 0.48 1.35 0.619 600 ...
```

1) Scatter Plot

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

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 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 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

Body weight vs Brain weight

Animals

- African elephant
- African giant pouched rat
- Arctic fox
- Arctic ground squirrel
- Asian elephant
- Baboon
- Big brown bat
- Brazilian tapir
- Chimpanzee
- Chinchilla
- Cotton rat
- Cow
- Desert hedgehog
- Dog
- Domestic cat
- Donkey
- Eastern american mole
- European hedgehog
- Galago
- Genet
- Giant armadillo
- Goat

2) Barchart

```
head(df)
```

| name | genus | vore | order | conservation | sleep_total | sleep_rem |
|----------------------------|------------|-------|--------------|--------------|-------------|-----------|
| 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 |

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

Total sleep time of Animals based on Vore

2) a) Regular Barchart

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

Total sleep time of Animals based on Vore

```
r_stack_barchart = data.table::melt(df, id.vars='vore', value=TRUE, variable, color = ~variable) %>%
  layer(
    title = list(text = ">>Total Distribution based on Vore"),
    legend = list(title = list(text = ">>Animal Feature")),
    xaxis = list(title = list(text = ">>Vore")),
    yaxis = list(title = list('Count', text = ">>Count"), barmode = 'stack'))
```

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.

```
r_stack_barchart
```

Total Distribution based on Vore

Animal Feature

- bodywt
- brainwt
- awake
- sleep_cycle
- sleep_rem
- sleep_total
- conservation
- order
- genus
- name

2) b) Stacked Barchart

```
r_stack_barchart = data.table::melt(df, id.vars='vore', value=TRUE, variable, color = ~variable) %>%
  layer(
    title = list(text = ">>Total Distribution based on Vore"),
    legend = list(title = list(text = ">>Animal Feature")),
    xaxis = list(title = list(text = ">>Vore")),
    yaxis = list(title = list('Count', text = ">>Count"), barmode = 'stack'))
```

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.

```
r_stack_barchart
```

Total Distribution based on Vore

Animal Feature

- name
- genus
- order
- conservation
- sleep_total
- sleep_rem
- sleep_cycle
- awake
- brainwt
- bodywt

2) c) Grouped Barchart

```
r_group_barchart = data.table::melt(df, id.vars='vore', value=TRUE, variable, color = ~variable) %>%
  layer(
    title = list(text = ">>Total Distribution based on Vore"),
    legend = list(title = list(text = ">>Animal Feature")),
    xaxis = list(title = list(text = ">>Vore")),
    yaxis = list(title = list('Count', text = ">>Count"), group = 'group'))
```

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.

```
r_group_barchart
```

Total Distribution based on Vore

Animal Feature

- name
- genus
- order
- conservation
- sleep_total
- sleep_rem
- sleep_cycle
- awake
- brainwt
- bodywt

3) Histogram

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

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

Total sleep time of Animals based on Vore

Vore

Count

Step