

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

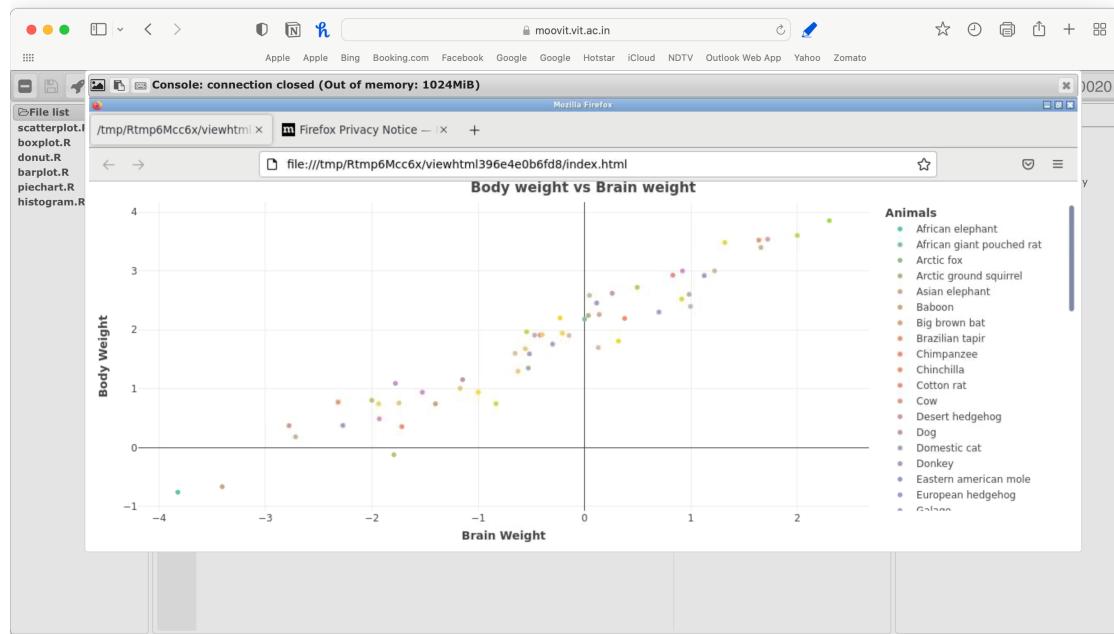
```
1 library(ggplot2) ## msleep is included in ggplot-2
2 library(plotly)
3 library(data.table)
4 library(plotly)
5 library(dplyr)
6
7
8 scatter_plot = plot_ly(data=msleep, x=-(-log10(bodywt)), y=-(-log10(brainwt)), color = ~name,
9   type='scatter', mode='markers') %>%
10   layout(
11     title= list(text = "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" contains instructions:

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

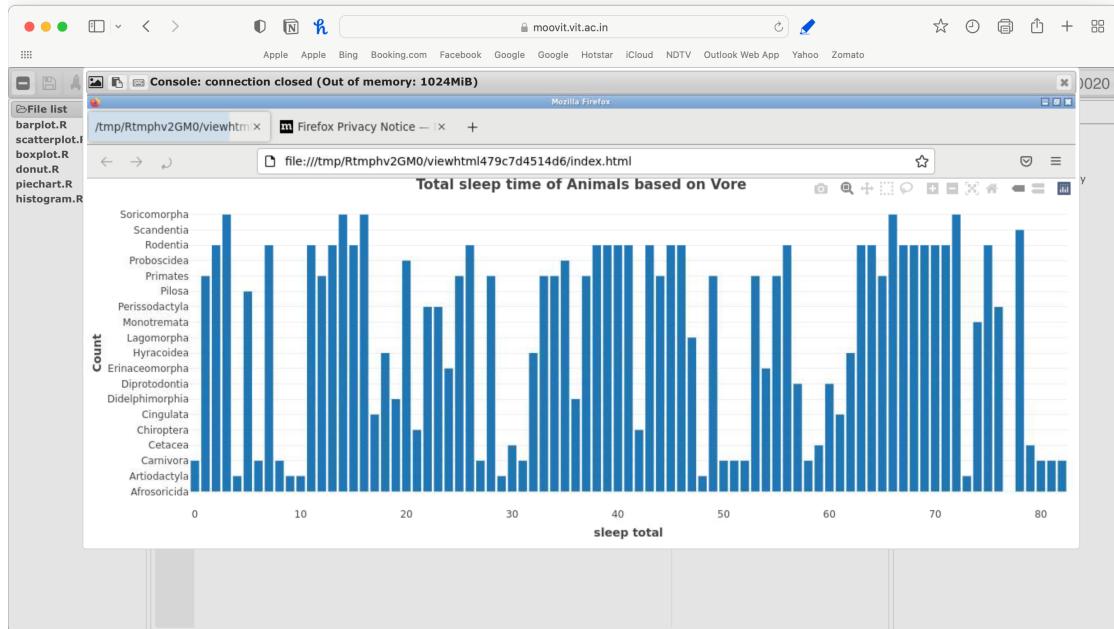
- 1) Scatter plot
- 2) Bar graph(regular, stacked, group)
- 3) Histogram
- 4) Pie chart
- 5) donut chart
- 6) Box plot

and any others.

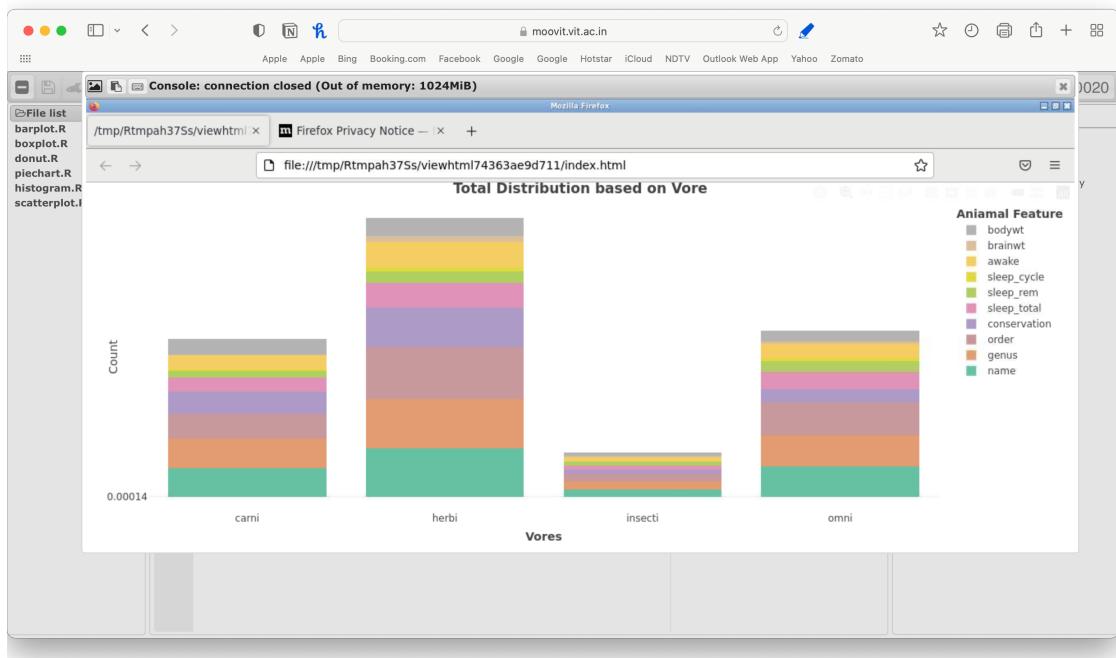
Upload this exercise on VTOP

### Executing the code one-by-one plots

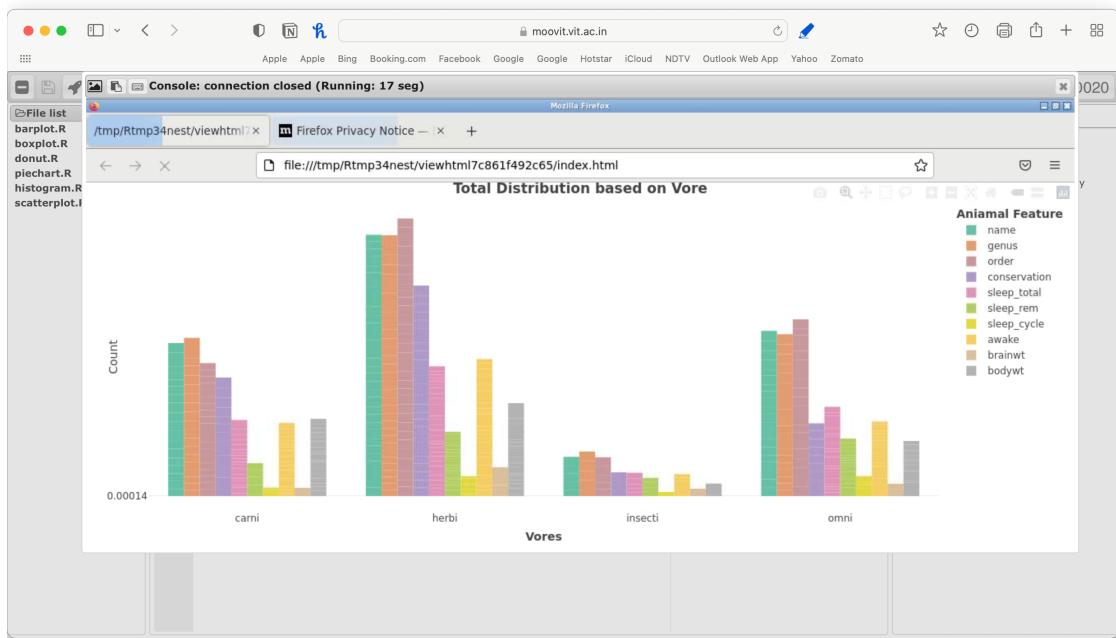
#### Bar-Chart



## Stacked Bar-Chart



## Grouped Bar-Chart



## 2) Histogram Plot

The screenshot shows a web-based R environment. On the left, a file list contains files like histogram.R, barplot.R, scatterplot.R, boxplot.R, donut.R, and piechart.R. The main 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="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')))
```

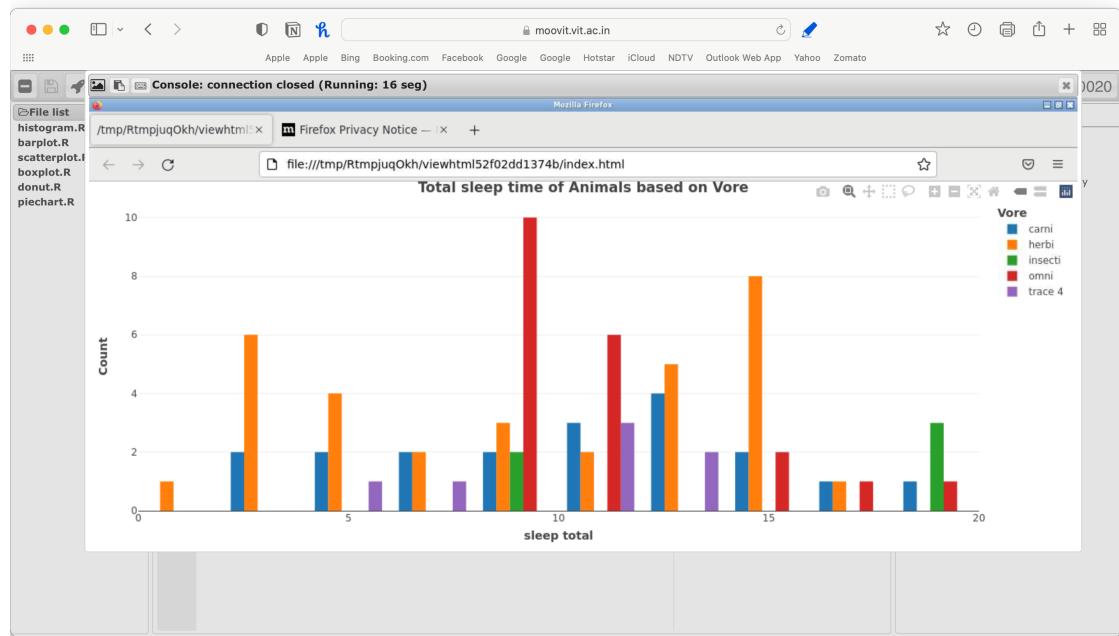
To the right, a "Description" panel provides instructions:

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



### 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
```

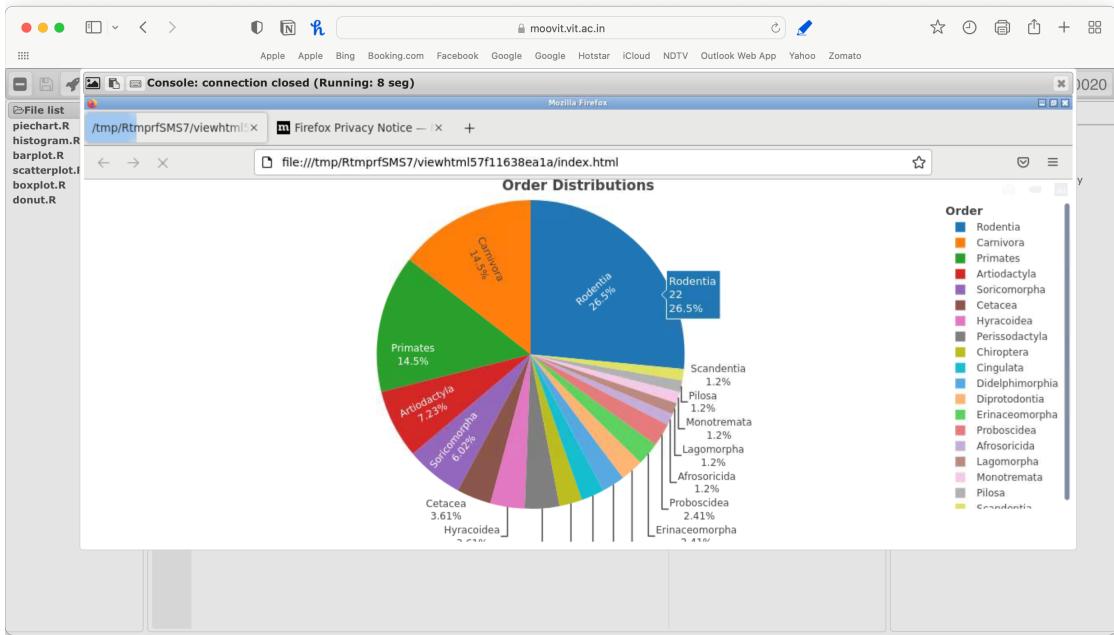
The output pane shows the generated pie chart titled "Order Distributions".

The sidebar on the right contains 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 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   labels=df_vore$label, 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>')))
```

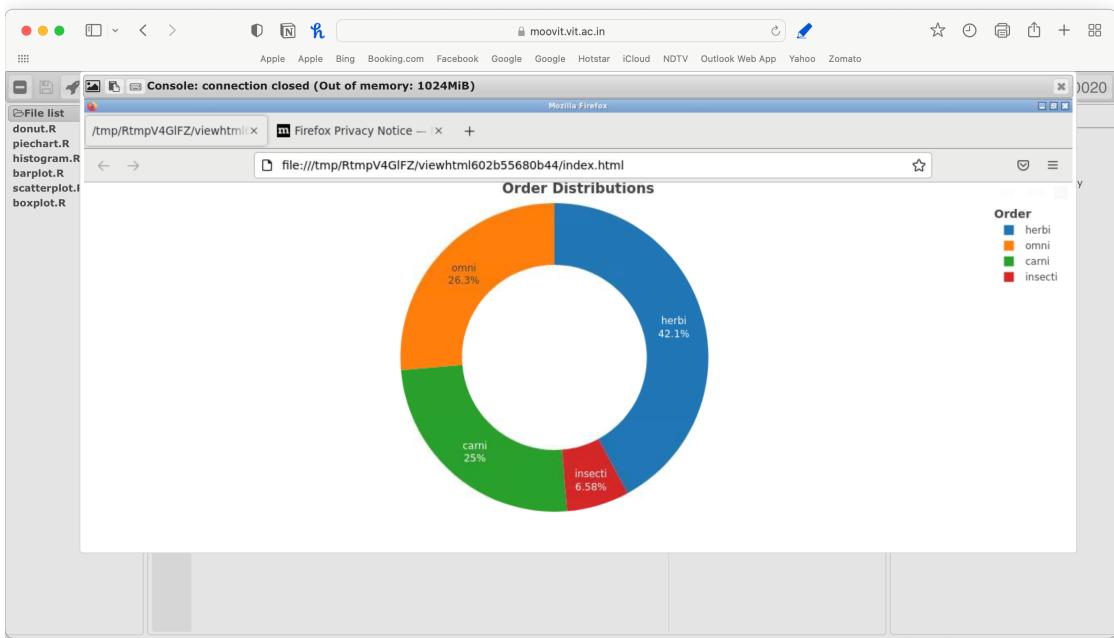
The right panel contains a "Description" section with instructions:

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

Screenshot of a RStudio interface showing a boxplot.R script and its output.

The script content:

```
library(ggplot2)
library(datasets)
library(dplyr)
library(plyr)

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

The output pane shows the generated box plot:

The box plot displays the distribution of total sleep time for different dietary categories (Vore). The y-axis represents 'Sleep Total Distribution' ranging from 0 to 20. The x-axis represents 'Vore' categories: carn, herbi, insecti, omni, and NA. The legend indicates the colors for each category: carn (teal), herbi (orange), insecti (blue), omni (pink), and NA (grey).

| Vore Category | Median Sleep (approx.) | Q1 (approx.) | Q3 (approx.) | Min (approx.) | Max (approx.) |
|---------------|------------------------|--------------|--------------|---------------|---------------|
| carn          | 10                     | 6            | 13           | 4             | 19            |
| herbi         | 10                     | 5            | 14           | 3             | 16            |
| insecti       | 8                      | 7            | 19           | 7             | 20            |
| omni          | 10                     | 8            | 11           | 8             | 16            |
| NA            | N/A                    | N/A          | N/A          | N/A           | N/A           |

