ctapply - An R package to measure central tendency and spread

Anthony McCofie - anthonymccofie@usf.edu

April 23, 2022

ctapply: **Object Summaries**

Description:

ctapply is a package used to get summaries of basic statistics, such as central tendency(mean, median..) and standard deviation (spread). The package uses specific methods to produce the result. The user have option to print the result.
The package is supported by four additional functions that return graphical visualization.

ctbarplot() return a bar graph. Usage example **ctbarplot(electricdata$cost)**

cthistgraph return histogram. Usage example **cthistgraph(electricdata$cost)**

ctpairplot return a plot combinations for the electricdata variables. Usage example **ctpairplot(electricdata)**

ctscattergraph return scatter graph for the data. Usage example **ctscattergraph(electricdata***cost*, *electricdata***fuel)**

The sample data used for this package is: **Cost Function of Electricity Producers (1955)**

Below are the fields for the sample data used for the package

```
##   X  cost output labor laborshare capital capitalshare fuel fuelshare
## 1 1 0.082      2  2.09     0.3164     183       0.4521 17.9    0.2315
## 2 2 0.661      3  2.05     0.2073     174       0.6676 35.1    0.1251
## 3 3 0.990      4  2.05     0.2349     171       0.5799 35.1    0.1852
## 4 4 0.315      4  1.83     0.1152     166       0.7857 32.2    0.0990
## 5 5 0.197      5  2.12     0.2300     233       0.3841 28.6    0.3859
## 6 6 0.098      9  2.12     0.1881     195       0.2926 28.6    0.5193
```

Usage:

- You can use the package in two ways:
  - Either with the data that comes with the package **electricdata** as shown below

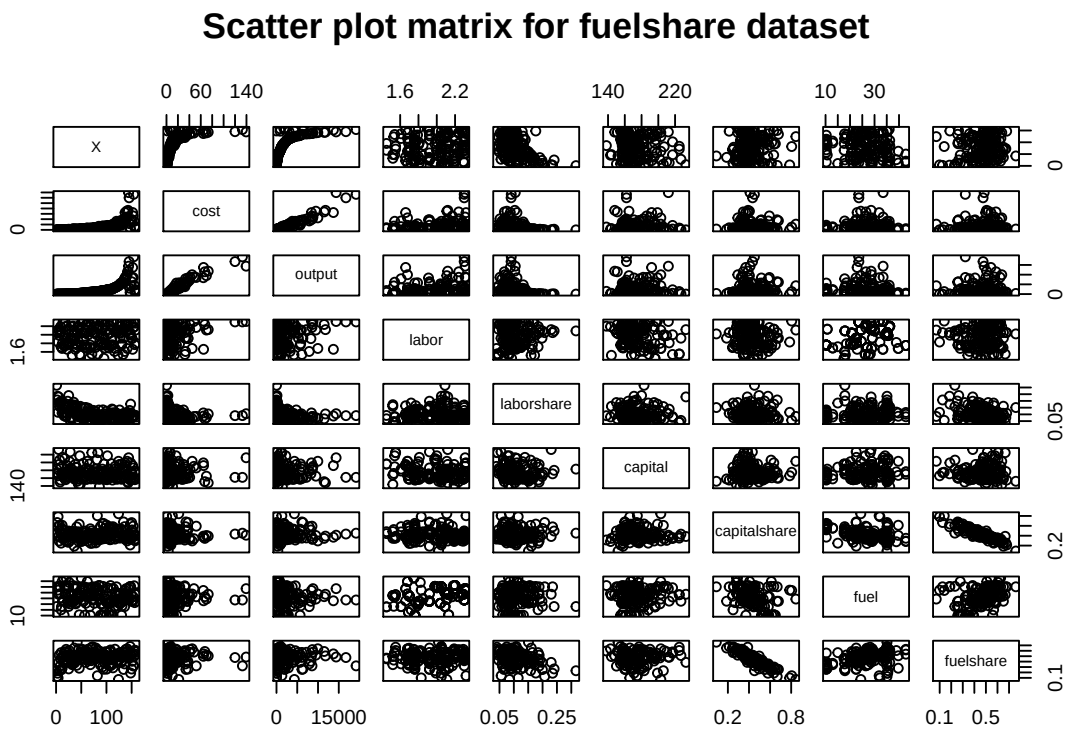
```
ctapply(electricdata$fuel, options=2)
```

```
## OPTION: 2
```

```
##
## Call:
## lm(formula = log(cost/fuel) ~ log(output) + log(labor/fuel) +
##     log(capital/fuel), data = electricdata)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.02819 -0.23068 -0.01754  0.15501  1.79910
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -4.772086   0.880175  -5.422 2.22e-07 ***
## log(output)        0.722676   0.016859  42.865  < 2e-16 ***
## log(labor/fuel)    0.561425   0.203908   2.753   0.0066 **
## log(capital/fuel) -0.001553   0.188565  -0.008   0.9934
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3998 on 155 degrees of freedom
## Multiple R-squared:  0.9299,   Adjusted R-squared:  0.9286
## F-statistic: 685.6 on 3 and 155 DF,  p-value: < 2.2e-16
```

```
ctpairplot(electricdata)
```



– Or use your own data such as creating a numeric vector shown below:

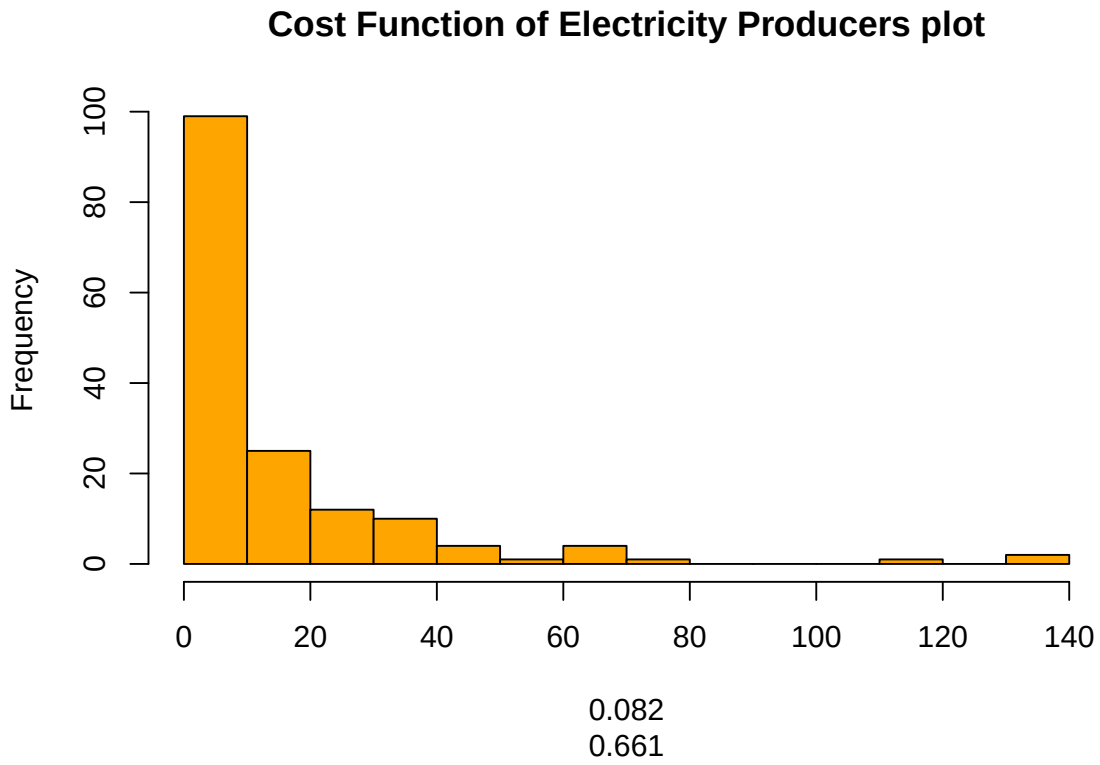
```
data1 <- c(4,5,6,5,6,5,6,4,6,5,6,4,5,4,5,6)
ctapply(data1, options=2)
```

```
## OPTION: 2

##
## Call:
## lm(formula = log(cost/fuel) ~ log(output) + log(labor/fuel) +
```

```
##     log(capital/fuel), data = electricdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02819 -0.23068 -0.01754  0.15501  1.79910
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -4.772086   0.880175  -5.422 2.22e-07 ***
## log(output)        0.722676   0.016859  42.865  < 2e-16 ***
## log(labor/fuel)    0.561425   0.203908   2.753   0.0066 **
## log(capital/fuel) -0.001553   0.188565  -0.008   0.9934
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3998 on 155 degrees of freedom
## Multiple R-squared:  0.9299,   Adjusted R-squared:  0.9286
## F-statistic: 685.6 on 3 and 155 DF,  p-value: < 2.2e-16
```

```
cthistgraph(electricdata$cost)
```



Arguments

object : An object for which a summary is desired. This could be a numeric vector or a data frame dataset.

np : A boolean value **(TRUE/FALSE)**. The statistics to use **(Mean, Median,. . .)**.

print : A boolean value **(TRUE/FALSE)**. This provide an option for users to print result.

options : Options takes 1, 2 or 3 as an argument **(options =1, 2, 3)**. This provide additional output.

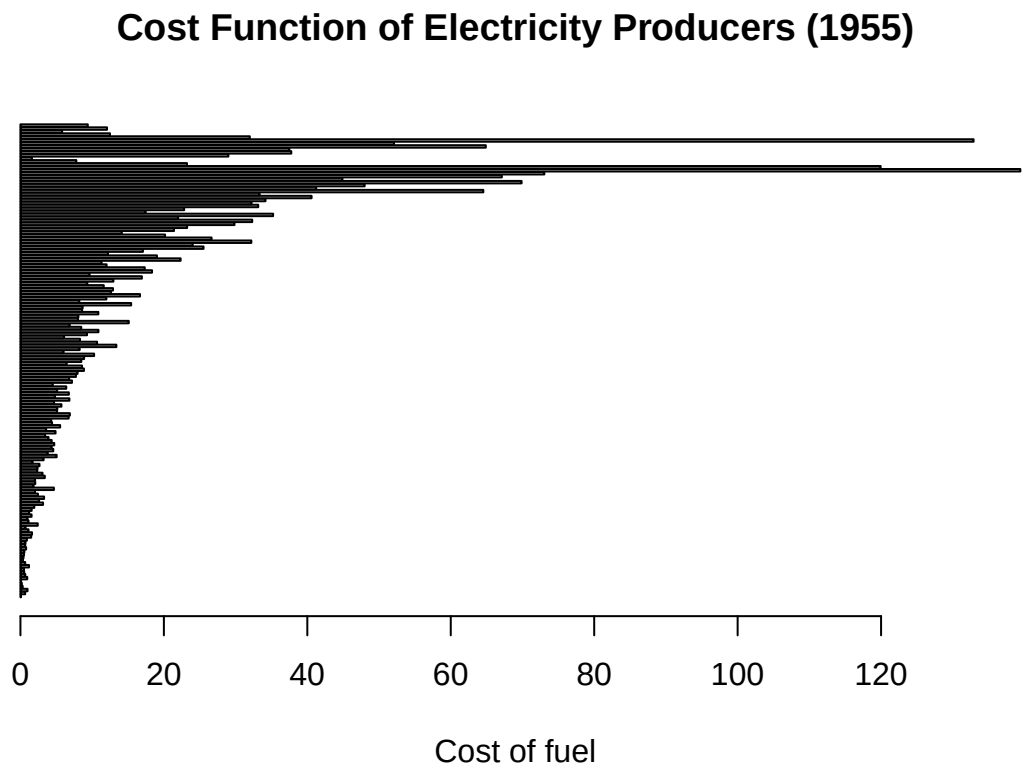
The full syntax usage:

```
data1 <- c(4,5,6,5,6,5,6,4,6,5,6,4,5,4,5,6)
ctapply(data1, np=TRUE, print=TRUE, options = 3)
```

```
## OPTION: 3

##
## Call:
## lm(formula = log(cost/fuel) ~ log(output) + I(log(output)^2) +
##     log(labor/fuel) + log(capital/fuel), data = electricdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43550 -0.13369  0.01093  0.12171  1.12369
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.711506   0.689013  -5.387 2.64e-07 ***
## log(output)        0.130496   0.059302   2.201  0.02926 *
## I(log(output)^2)   0.051788   0.005059  10.237  < 2e-16 ***
## log(labor/fuel)    0.462589   0.158102   2.926  0.00395 **
## log(capital/fuel)  0.071738   0.146109   0.491  0.62413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3094 on 154 degrees of freedom
## Multiple R-squared:  0.9583, Adjusted R-squared:  0.9572
## F-statistic: 884.7 on 4 and 154 DF,  p-value: < 2.2e-16
```

```
ctbarplot(electricdata$cost)
```



Who should use this package

Why use **ctapply**? Use **ctapply** to quickly summarize data and identify what looks normal and what looks odd. The distribution of a variable shows what values the variable takes and how often the variable takes these values.

Analytics in a true sense is leveraged only through visualizations. R, as a statistical tool, offers strong visualization capabilities. So, the numerous options associated with charts is what makes them special. Each of the charts has its own application and the chart should be studied prior to applying it to a problem.