

DATA VISUALIZATION

Jyotheesh Gaddam

PIE Chart

- **Syntax**

```
pie(x, labels, radius, main, col, clockwise)
```

x is a vector containing the numeric values used in the pie chart.

labels is used to give description to the slices.

radius indicates the radius of the circle of the pie chart. (value between -1 and +1).

main indicates the title of the chart.

col indicates the color palette.

clockwise is a logical value indicating if the slices are drawn clockwise or anti clockwise.

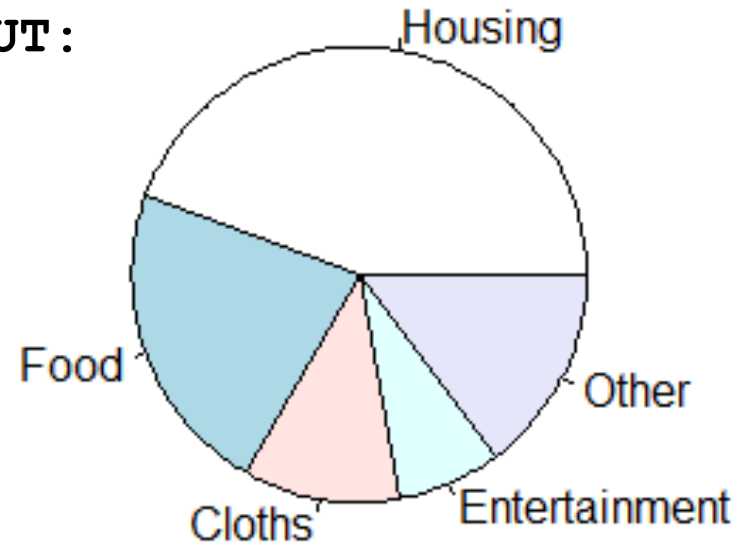
PIE Chart

- Create a **Vector** named as **slices** and assign these values **600,300,150,100,200**
- Create a **Vector** named as **label** and assign these values **Housing, Food, Cloths, Entertainment, Other**
- **#Plotting Pie Chart**
- **pie(slices, labels = label, main="Expenditure")**

PIE Chart

```
slices <- c(600,300,150,100,200)
label <- c("Housing", "Food", "Cloths",
"Entertainment", "Other")
pie(slices, labels = label, main="Expenditure")
```

OUTPUT:



PIE Chart with Percentage

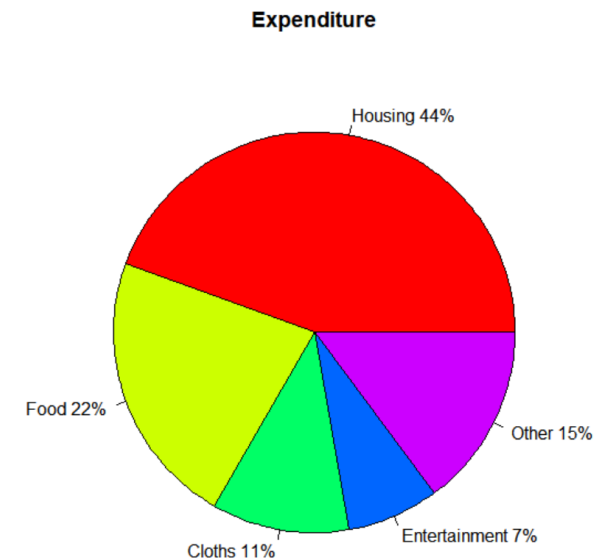
```
slices <- c(600,300,150,100,200)
label <- c("Housing", "Food", "Cloths",
"Entertainment", "Other")

percentage <- round(slices/sum(slices)*100)

label <- paste(label, percentage) # add percents to
labels

label <- paste(label,"%",sep="") # add % to labels

pie(slices, labels = label,
col=rainbow(length(label)),main="Expenditure")
```



Installing Packages

Package
name: **plotrix**

Environment is empty

Files Plots Packages Help Viewer

Install Update

Name	Description	Version
System Library		
<input checked="" type="checkbox"/> base		
<input type="checkbox"/> boot		
<input type="checkbox"/> class		
<input type="checkbox"/> cluster		
<input type="checkbox"/> codetools		
<input type="checkbox"/> compiler		
<input checked="" type="checkbox"/> datasets		
<input type="checkbox"/> foreign		

Files Plots Packages Help Viewer

Install Update

Name	Description	Version
System Library		
<input checked="" type="checkbox"/> base		
<input type="checkbox"/> boot		
<input type="checkbox"/> class		
<input type="checkbox"/> cluster		
<input type="checkbox"/> codetools		
<input type="checkbox"/> compiler		
<input checked="" type="checkbox"/> datasets		
<input type="checkbox"/> foreign		
<input checked="" type="checkbox"/> graphics		
<input checked="" type="checkbox"/> grDevices		
<input type="checkbox"/> grid	The Grid Graphics Package	3.6.1
<input type="checkbox"/> KernSmooth	Functions for Kernel Smoothing Supporting Wand &	2.23-15

Install Packages

Install from: [Configuring Repositories](#)

Repository (CRAN)

Packages (separate multiple with space or comma):

Install to Library:

\\jgaddam.homes.deakin.edu.au/my-home/My Documents/R/v

☒ Install dependencies

Install Cancel

3D PIE Chart

```
library(plotrix)

slices <- c(600,300,150,100,200)

label <- c("Housing", "Food", "Cloths",
"Entertainment", "Other")

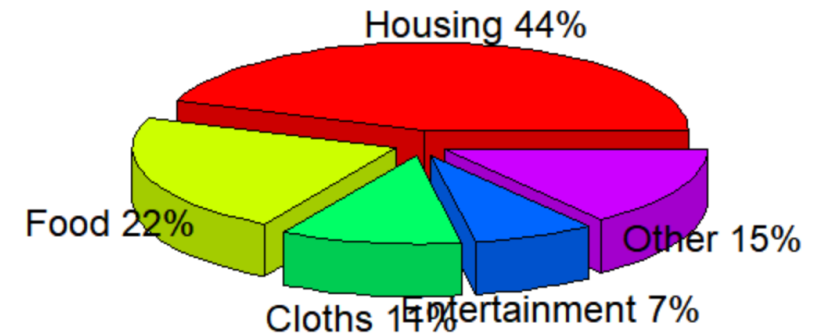
percentage <- round(slices/sum(slices)*100)

label <- paste(label, percentage) # add percents to
labels

label <- paste(label,"%",sep="") # ad % to labels

pie3D(slices,labels = label,explode=0.1,
col=rainbow(length(label)), main="Expenditure")
```

Expenditure



Save PIE chart as image

```
slices <- c(600,300,150,100,200)
label <- c("Housing", "Food", "Cloths",
"Entertainment", "Other")

#providing file name
png(file = "pie_chart.jpg")
pie(slices, labels = label, main="Expenditure")

#save the file
dev.off()
```


BAR Chart

Syntax:

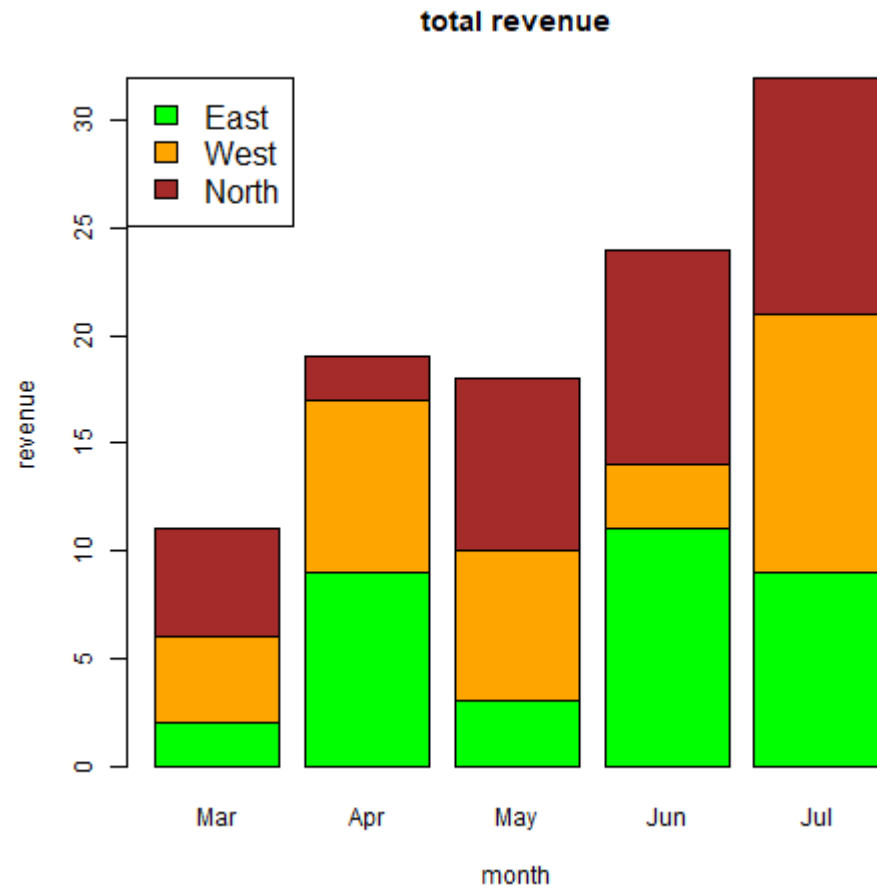
```
barplot(H,xlab,ylab,main, names.arg,col)
```

- **H** is a vector or matrix containing numeric values used in bar chart.
- **xlab** is the label for x axis.
- **ylab** is the label for y axis.
- **main** is the title of the bar chart.
- **names.arg** is a vector of names appearing under each bar.
- **col** is used to give colors to the bars in the graph.

BAR Chart

- Create a **Vector** named as **H** and assign these values **7,12,28,3,41**
- Create a **Vector** named as **M** and assign these values **Mar, Apr, May, Jun, Jul**
- `png(file = "barchart_months_revenue.png")`
- `# Plot the bar chart`
- `barplot(H,names.arg=M,xlab="Month",ylab="Revenue",col="blue", main="Revenue chart",border="red")`
- `dev.off()`

Stacked Bar Chart



Stacked Bar Chart

```
# Create the input vectors.
colors = c("green", "orange", "brown")
months <- c("Mar", "Apr", "May", "Jun", "Jul")
regions <- c("East", "West", "North")

# Create the matrix of the values.
Values <- matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11), nrow =
3, ncol = 5, byrow = TRUE)

# Give the chart file a name
png(file = "barchart_stacked.png")

# Create the bar chart
barplot(Values, main = "total revenue", names.arg = months,
xlab = "month", ylab = "revenue", col = colors)

# Add the legend to the chart
legend("topleft", regions, cex = 1.3, fill = colors)

# Save the file
dev.off()
```

HISTOGRAM

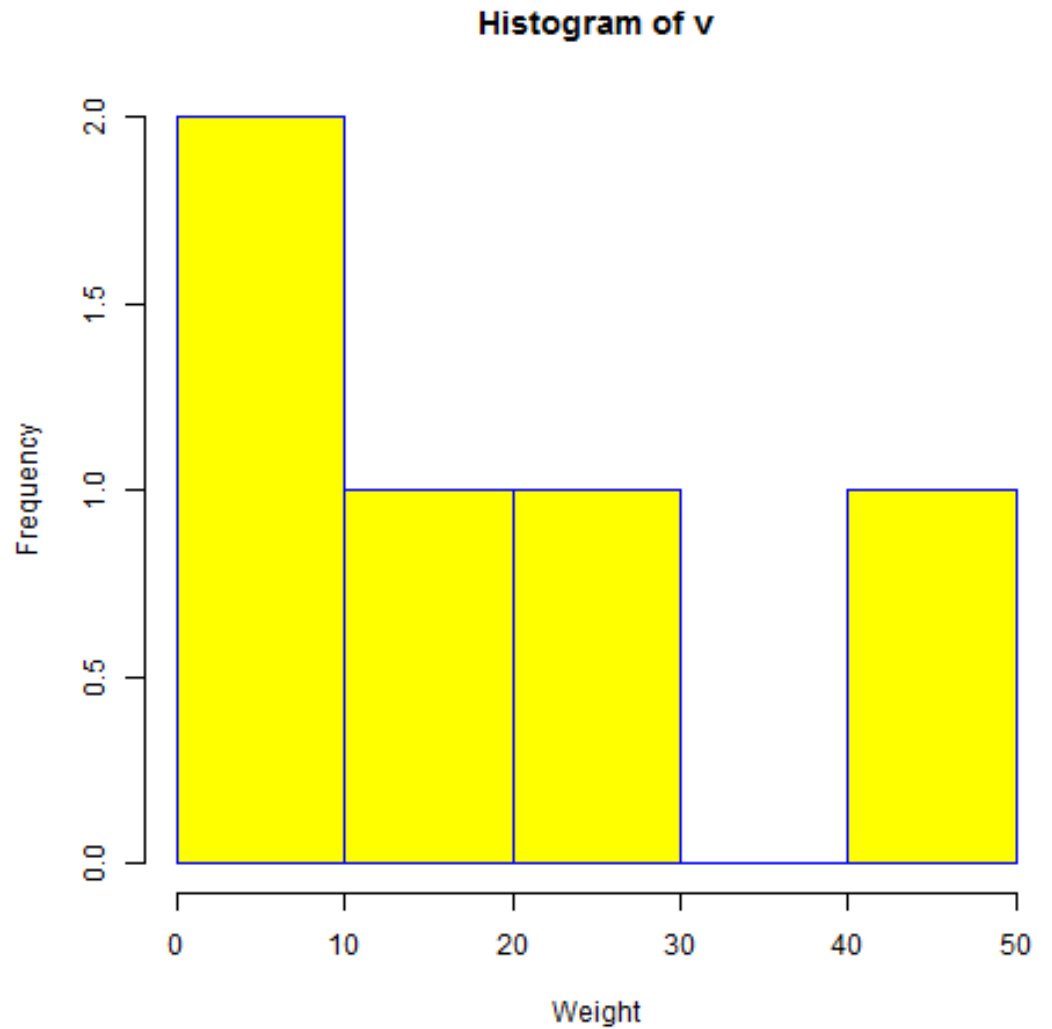
- **Syntax**

```
hist(v,main,xlab,xlim,ylim,breaks,col,border)
```

- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to mention the width of each bar.

HISTOGRAM

```
v <- c(9,13,21,8,36,22,12,41,31,33,19)
```

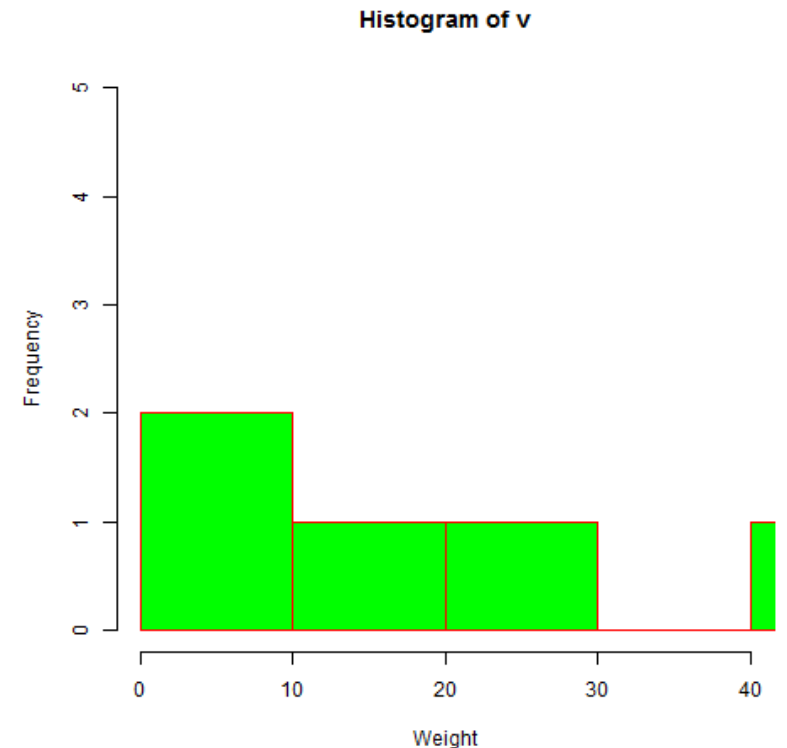


HISTOGRAM

```
# Create data for the graph.  
v <- c(9,13,21,8,36,22,12,41,31,33,19)  
# Give the chart file a name.  
png(file = "histogram.png")  
# Create the histogram.  
hist(v,xlab = "Weight",col = "yellow",border =  
"blue")  
# Save the file.  
dev.off()
```

HISTOGRAM

```
# Create data for the graph.  
v <- c(9,13,21,8,36,22,12,41,31,33,19)  
  
# Give the chart file a name.  
png(file = "histogram_lim_breaks.png")  
  
# Create the histogram.  
hist(v,xlab = "Weight",col = "green",border = "red",  
xlim = c(0,40), ylim = c(0,5), breaks = 5)  
  
# Save the file.  
dev.off()
```



Line Graph

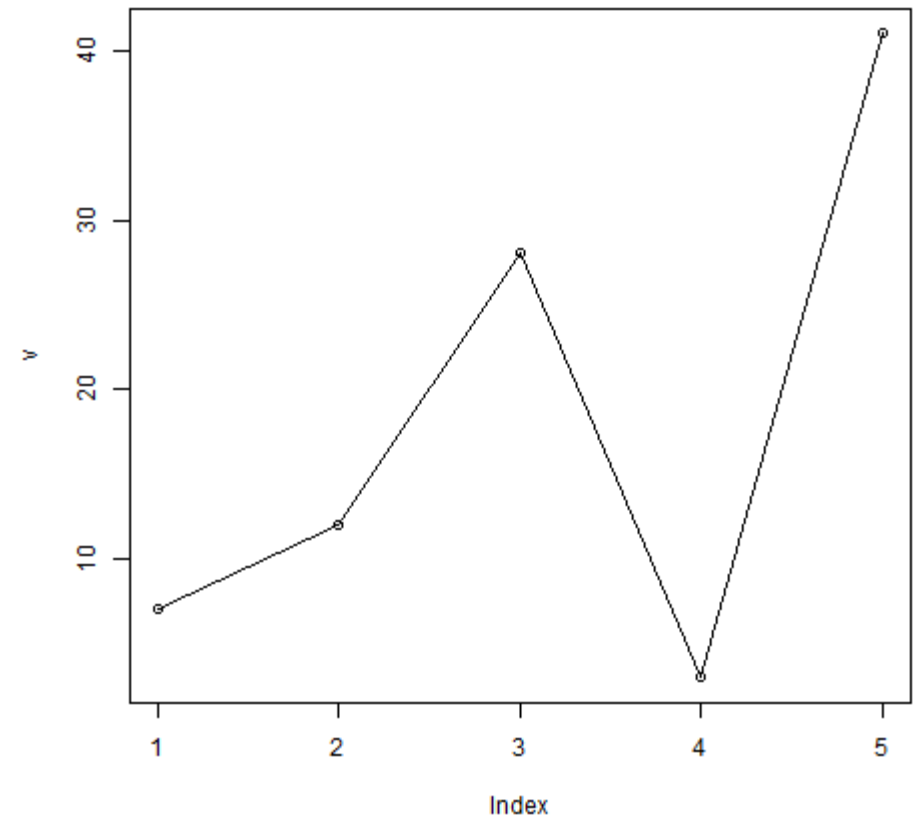
- **Syntax**

```
plot(v,type,col,xlab,ylab)
```

- **v** is a vector containing the numeric values.
- **type** takes the value "p" to draw only the points, "l" to draw only the lines and "o" to draw both points and lines.
- **xlab** is the label for x axis.
- **ylab** is the label for y axis.
- **main** is the Title of the chart.
- **col** is used to give colors to both the points and lines.

Line Graph

```
# Create the data for the chart.  
v <- c(7,12,28,3,41)  
  
# Give the chart file a name.  
png(file = "line_chart.jpg")  
  
# Plot the bar chart.  
plot(v,type = "o")  
  
# Save the file.  
dev.off()
```



Line Graph

#Different Plot Types

```
plot(v, type = "o")
```

```
plot(v, type = "l")
```

```
plot(v, type = "b")
```

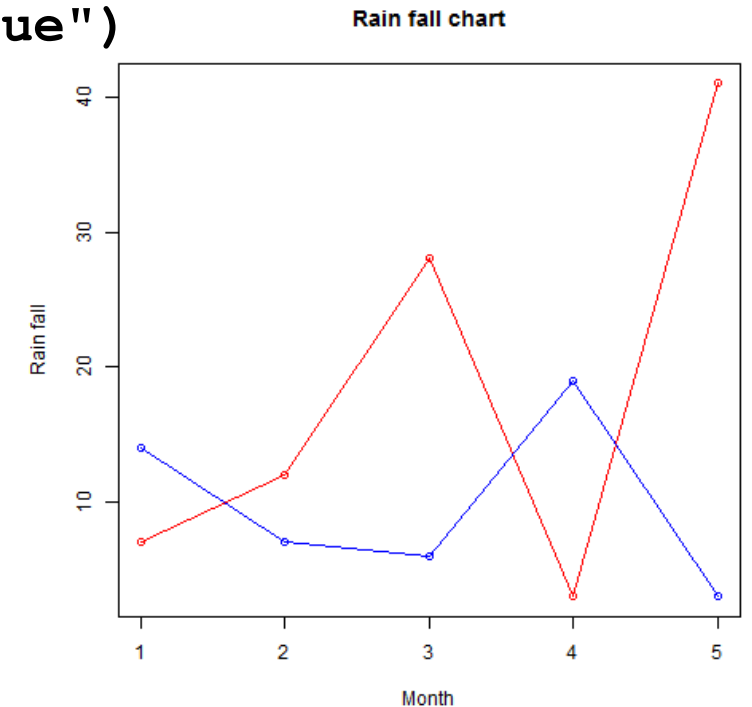
```
plot(v, type = "p")
```

```
plot(v, type = "c")
```

```
plot(v, type = "h")
```

Line Graph

```
# Create the data for the chart.  
v <- c(7,12,28,3,41)  
t <- c(14,7,6,19,3)  
  
png(file = "line_chart_2_lines.jpg")  
plot(v,type = "o",col = "red", xlab = "Month", ylab =  
"Rain fall",main = "Rain fall chart")  
  
#second line  
lines(t, type = "o", col = "blue")  
dev.off()
```



Scatter Plot

- **Syntax**

`plot(x, y, main, xlab, ylab, xlim, ylim, axes)`

- **x** is the data set whose values are the horizontal coordinates.
- **y** is the data set whose values are the vertical coordinates.
- **main** is the title of the graph.
- **xlab** is the label in the horizontal axis.
- **ylab** is the label in the vertical axis.
- **xlim** is the limits of the values of x used for plotting.
- **ylim** is the limits of the values of y used for plotting.
- **axes** indicates whether both axes should be drawn on the plot.

Scatter Plot

```
wt<-c(2.620,2.875,2.320,3.215,3.440,3.460)
mpg<-c(21.0,21.0,22.8,21.4,18.7,18.1)
cor(wt,mpg)
plot(wt, mpg,
      xlab = "Weight",
      ylab = "Milage",
      xlim = c(2.5,5),
      ylim = c(15,30),
      main = "Weight vs Milage"
)
```

CSV Files

Resources Week 3 > **input_file**

```
setwd("#location")
```

```
data<-read.csv("input_file.csv")
```

```
print(data)
```

	id,	name,	salary,	start_date,	dept
1	1	Rick	623.30	2012-01-01	IT
2	2	Dan	515.20	2013-09-23	Operations
3	3	Michelle	611.00	2014-11-15	IT
4	4	Ryan	729.00	2014-05-11	HR
5	NA	Gary	843.25	2015-03-27	Finance
6	6	Nina	578.00	2013-05-21	IT
7	7	Simon	632.80	2013-07-30	Operations
8	8	Guru	722.50	2014-06-17	Finance

Data Visualisation from CSV File

```
pie(data$salary, labels=data$name, main="Employees  
Salary")
```

```
barplot(data$salary, names.arg = data$name, xlab =  
"Employee", ylab="Salary", main = "Employees Salary")
```

```
hist(data$salary, xlab = "Salary", col = "yellow",  
border = "blue")
```

```
plot(data$salary, type = "o")
```


CSV Files

Get the details of the person with max salary

```
# Get the max salary from data frame.
```

```
sal <- max(data$salary)
```

```
print(sal)
```

```
# Get the person detail having max salary.
```

```
retval <- subset(data, salary == max(salary))
```

```
print(retval)
```

CSV File

Get all the people working in IT department

```
retval <- subset( data, dept == "IT")  
print(retval)
```

	id	name	salary	start_date	dept
1	1	Rick	623.3	2012-01-01	IT
3	3	Michelle	611.0	2014-11-15	IT
6	6	Nina	578.0	2013-05-21	IT

Get the persons in IT department whose salary is greater than 600

```
info <- subset(data, salary > 600 & dept == "IT")  
print(info)
```

	id	name	salary	start_date	dept
1	1	Rick	623.3	2012-01-01	IT
3	3	Michelle	611.0	2014-11-15	IT

Writing into CSV File

```
# Create a data frame.
```

```
data <- read.csv("input_file.csv")
```

```
retval <- subset(data, as.Date(start_date) >  
as.Date("2014-01-01"))
```

```
# Write filtered data into a new file.
```

```
write.csv(retval, "output.csv")
```

```
newdata <- read.csv("output.csv")
```

```
print(newdata)
```

	X	id	name	salary	start_date	dept
1	3	3	Michelle	611.00	2014-11-15	IT
2	4	4	Ryan	729.00	2014-05-11	HR
3	5	5	Gary	843.25	2015-03-27	Finance
4	8	8	Guru	722.50	2014-06-17	Finance

Writing into CSV File

```
# Create a data frame.
```

```
data <- read.csv("input_file.csv")
```

```
retval <- subset(data, as.Date(start_date) >  
as.Date("2014-01-01"))
```

```
# Write filtered data into a new file.
```

```
write.csv(retval, "output.csv")
```

```
newdata <- read.csv("output.csv")
```

```
print(newdata)
```

	X	id	name	salary	start_date	dept
1	3	3	Michelle	611.00	2014-11-15	IT
2	4	4	Ryan	729.00	2014-05-11	HR
3	5	5	Gary	843.25	2015-03-27	Finance
4	8	8	Guru	722.50	2014-06-17	Finance

Writing into CSV File

```
# Create a data frame.
```

```
data <- read.csv("input_file.csv")
```

```
retval <- subset(data, as.Date(start_date) >  
as.Date("2014-01-01"))
```

```
# Write filtered data into a new file.
```

```
write.csv(retval, "output.csv", row.names = FALSE )
```

```
newdata <- read.csv("output.csv")
```

```
print(newdata)
```

	id	name	salary	start_date	dept
1	3	Michelle	611.00	2014-11-15	IT
2	4	Ryan	729.00	2014-05-11	HR
3	5	Gary	843.25	2015-03-27	Finance
4	8	Guru	722.50	2014-06-17	Finance