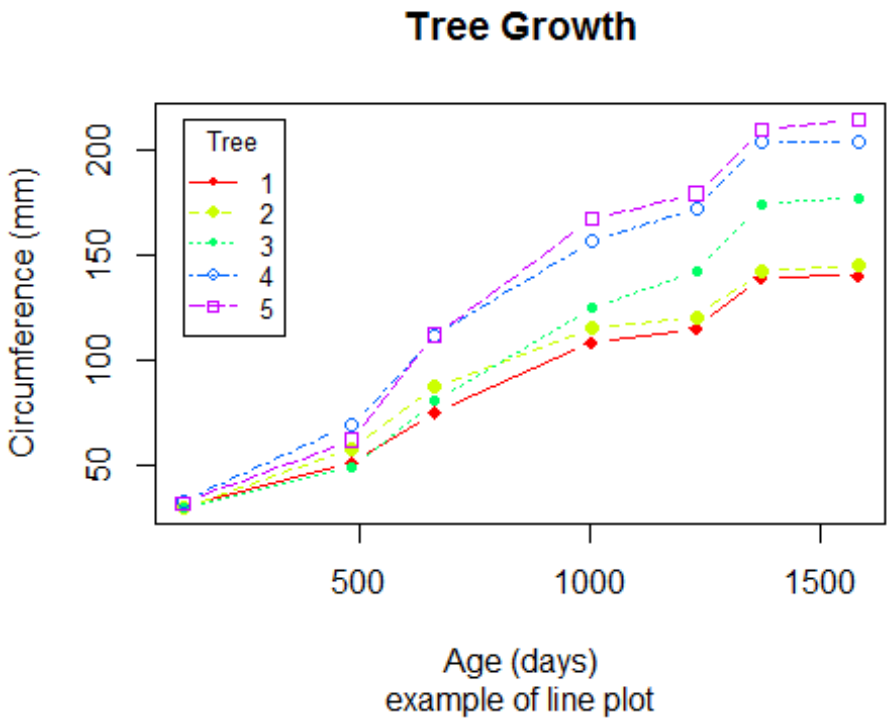
Chart Families : Data Visualization

SAGAR KUMAR : 194161013

Line chart

```
# Create Line Chart

# convert factor to numeric for convenience
Orange$Tree <- as.numeric(Orange$Tree)
ntrees <- max(Orange$Tree)

# get the range for the x and y axis
xrange <- range(Orange$age)
yrange <- range(Orange$circumference)

# set up the plot
plot(xrange, yrange, type="n", xlab="Age (days)",
   ylab="Circumference (mm)" )
colors <- rainbow(ntrees)
linetype <- c(1:ntrees)
plotchar <- seq(18,18+ntrees,1)

# add lines
for (i in 1:ntrees) {
  tree <- subset(Orange, Tree==i)
  lines(tree$age, tree$circumference, type="b", lwd=1.5,
    lty=linetype[i], col=colors[i], pch=plotchar[i])
}

# add a title and subtitle
title("Tree Growth", "example of line plot")

# add a legend
legend(xrange[1], yrange[2], 1:ntrees, cex=0.8, col=colors,
   pch=plotchar, lty=linetype, title="Tree")
```

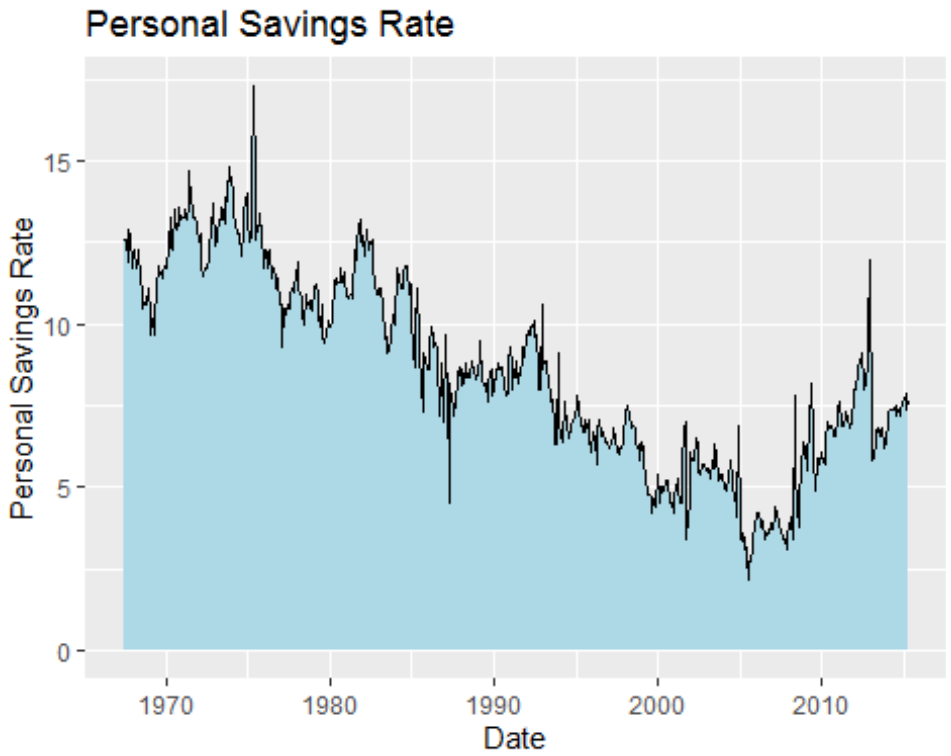


A line chart displays how quantitative values have changed over time for different categorical items.

Above plot shows the growth of 5 orange trees over time. Each tree will have its own distinctive line. The data come from the dataset Orange(inbuilt in R).

Area Chart

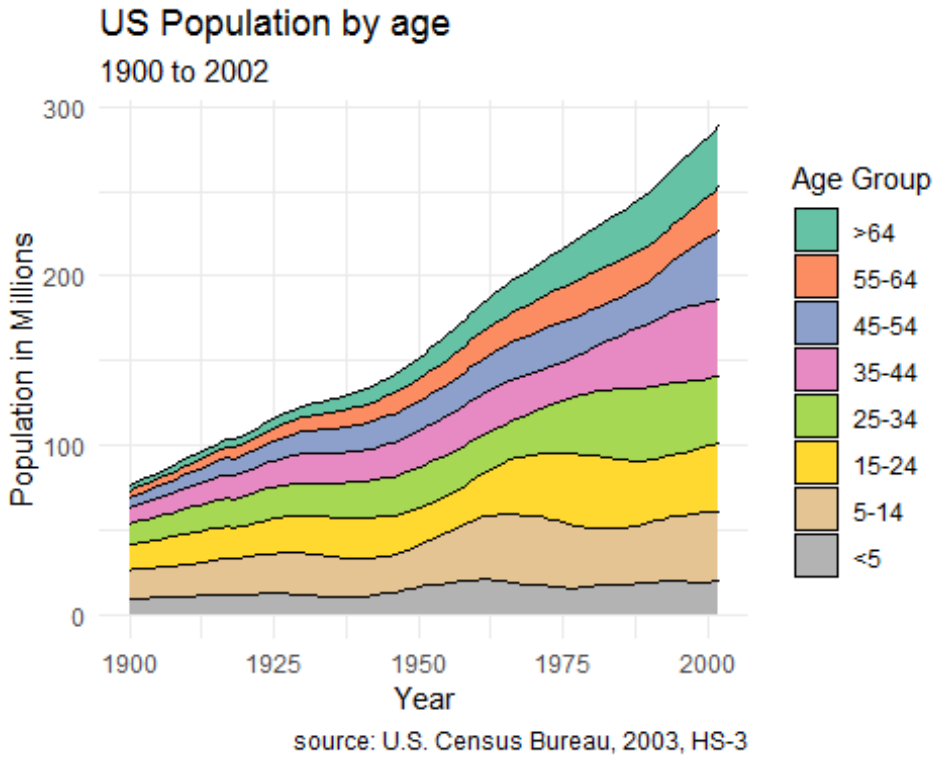
```
# basic area chart
ggplot(economics, aes(x = date, y = psavert)) +
  geom_area(fill="lightblue", color="black") +
  labs(title = "Personal Savings Rate",
       x = "Date",
       y = "Personal Savings Rate")
```



A simple area chart is basically a line graph, with a fill from the line to the x-axis.

stacked area chart

```
# stacked area chart
data(uspopage, package = "gcookbook")
ggplot(uspopage, aes(x = Year,
                     y = Thousands/1000,
                     fill = forcats::fct_rev(AgeGroup))) +
  geom_area(color = "black") +
  labs(title = "US Population by age",
       subtitle = "1900 to 2002",
       caption = "source: U.S. Census Bureau, 2003, HS-3",
       x = "Year",
       y = "Population in Millions",
       fill = "Age Group") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```



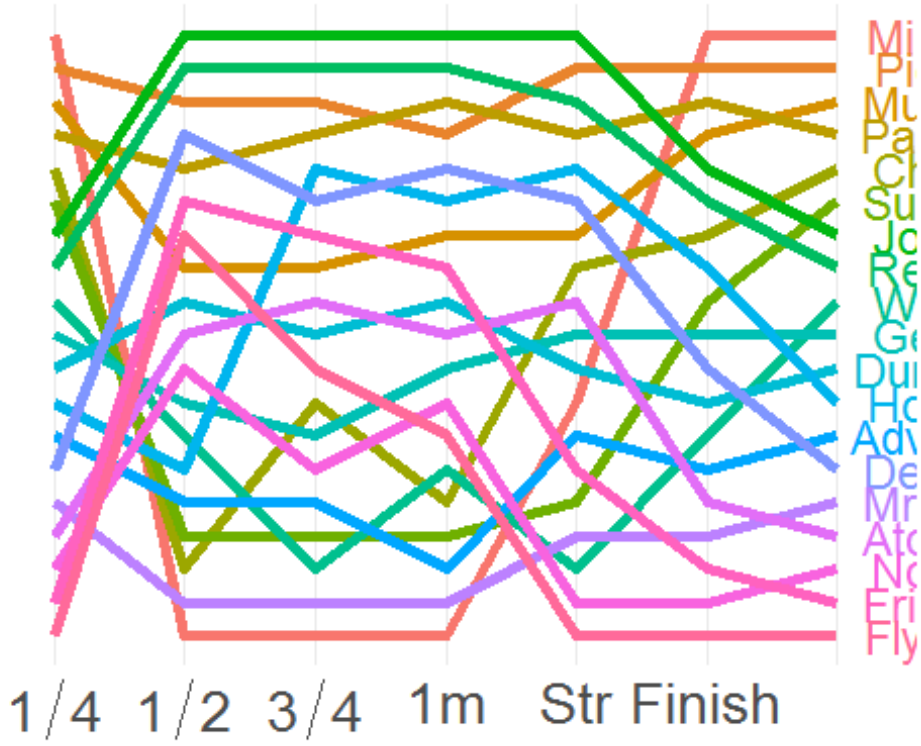
A stacked area chart can be used to show differences between groups over time. Consider the uspopage dataset from the gcookbook package. We'll plot the age distribution of the US population from 1900 and 2002.

Apparently, the number of young children have not changed very much in the past 100 years.

Stacked area charts are most useful when interest is on both (1) group change over time and (2) overall change over time. Place the most important groups at the bottom. These are the easiest to interpret in this type of plot.

Bump Chart

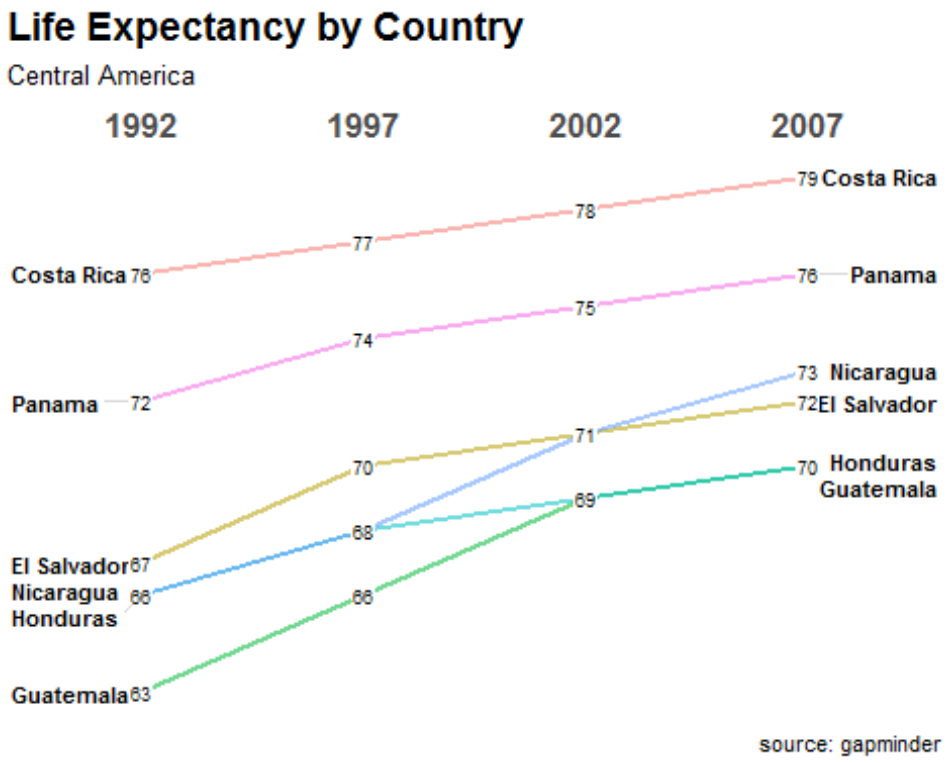
```
df=read.csv("bump_chart.csv")
#df
library(reshape2)
library(ggplot2)
df$Horse <- with(df, reorder(Horse, Finish))


dfm <- melt(df)

## Using Horse as id variables

p <- ggplot(dfm, aes(variable, value,
                     group = Horse, colour = Horse, label = Horse))
p1 <- p + geom_line(size=2) + geom_text(data = subset(dfm,variable ==
"Finish"), size=6,
                                        aes(x = variable, hjust = -0.1))

labels <- c(expression(1/4), expression(1/2),
            expression(3/4), "1m", "Str", "Finish",
            "")

p1 + theme_bw() + theme(legend.position = "none", panel.border =
element_blank(), axis.ticks = element_blank()) +
  scale_x_discrete(breaks = c(levels(dfm$variable), ""), labels = labels) +
  scale_y_continuous(breaks = NULL,trans = "reverse") + xlab(NULL) +
ylab(NULL)+
  theme(axis.text=element_text(size=21))
```



Bump chart is an alternative to Line chart.
A Bump Chart is a special form of a line plot designed for exploring changes in rank over time.
This chart helps compare the position / performance / ranking of multiple observations rather than the actual values itself

slope graph

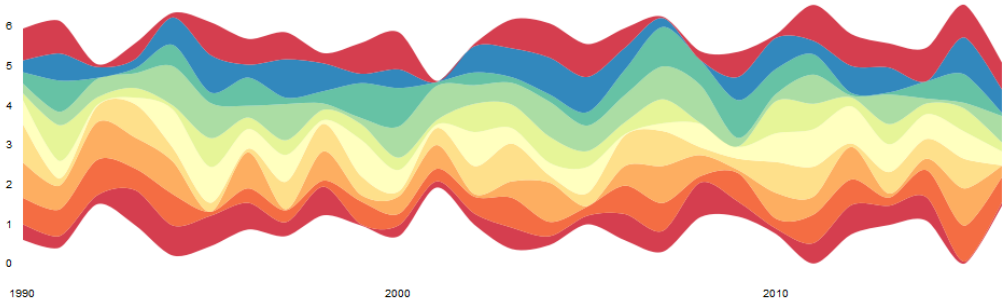
```
library(knitr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

 library(CGPfunctions)

## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car

 library(gapminder)
 library(magrittr)
 # Select Central American countries data
 # for 1992, 1997, 2002, and 2007

 df <- filter(gapminder,year %in% c(1992, 1997, 2002, 2007) &
            country %in% c("Panama", "Costa Rica",
                           "Nicaragua", "Honduras",
                           "El Salvador", "Guatemala",
                           "Belize")) %>%
  mutate(year = factor(year,ordered = TRUE),
         lifeExp = round(lifeExp))



# create slope graph
#
 newggslopegraph(df, year, lifeExp, country) +
   labs(title="Life Expectancy by Country",
        subtitle="Central America",
        caption="source: gapminder")
```



When there are several groups and several time points, a slope graph can be helpful. Aove plot Shows life expectancy for six Central American countries in 1992, 1997, 2002, and 2007

streamgraph

```
library(streamgraph)
library(dplyr)

data <- data.frame(
  year=rep(seq(1990,2016) , each=10),
  name=rep(letters[1:10] , 27),
  value=sample( seq(0,1,0.0001) , 270)
)

pp <- streamgraph(data, key="name", value="value", date="year",
height="300px", width="1000px")
pp
```



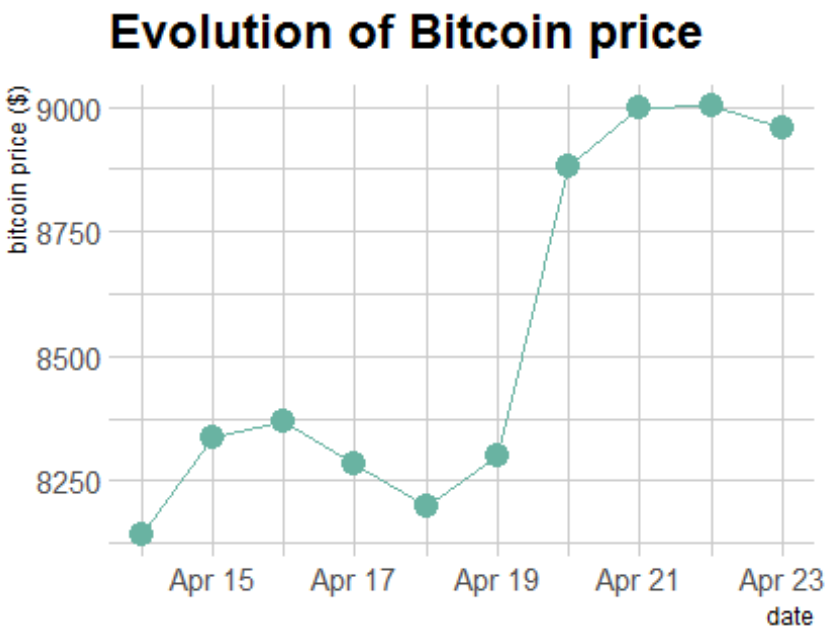
Stream Graphs display the changes in data over time of different category items.

connected scatter plot

Connected scatter plot is an extension of joined version of line chart and scatter plot.

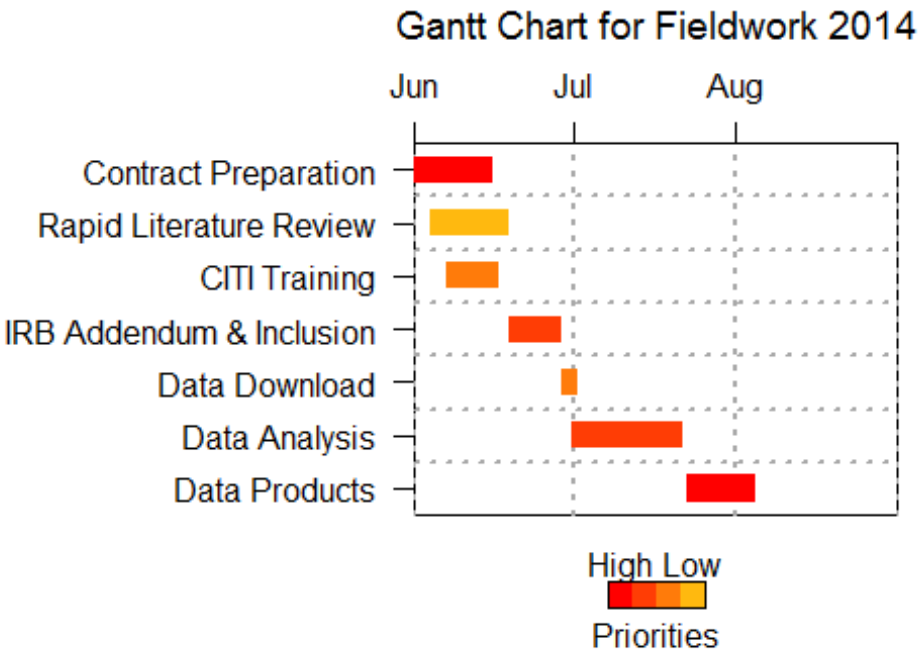
```
library(tidyverse)

## -- Attaching packages ---------------------------------------------------
------------------ tidyverse 1.3.0 --

## v tibble  2.1.3     v purrr   0.3.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------
------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()

library(hrbrthemes)

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use
these themes.

##       Please use hrbrthemes::import_roboto_condensed() to install Roboto
Condensed and

##       if Arial Narrow is not on your system, please see
https://bit.ly/arialnarrow

#library(plotly)
library(patchwork)
library(babynames)
library(viridis)

## Loading required package: viridisLite

# Load dataset from github
data <-
read.table("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Examp
le_dataset/3_TwoNumOrdered.csv", header=T)
data$date <- as.Date(data$date)

# Plot
data %>%
  tail(10) %>%
  ggplot( aes(x=date, y=value)) +
  geom_line(color="#69b3a2") +
  geom_point(color="#69b3a2", size=4) +
  ggtitle("Evolution of Bitcoin price") +
```

```
  ylab("bitcoin price ($)") +
  theme_ipsum()
```



Gantt chart

```
library(plotrix)
Ymd.format<-"%Y/%m/%d"
gantt.info<-list(labels=c("Contract Preparation","Rapid Literature
Review","CITI Training","IRB Addendum & Inclusion","Data Download","Data
Analysis","Data Products"),

starts=as.POSIXct(strptime(c("2014/06/01","2014/06/04","2014/06/07","2014/06/
19","2014/06/29","2014/07/01","2014/07/23"),
                                          format=Ymd.format)),

ends=as.POSIXct(strptime(c("2014/06/16","2014/06/19","2014/06/17","2014/06/29
","2014/07/02","2014/07/22","2014/08/05"),
                                        format=Ymd.format)),
                priorities=c(1,4,3,2,3,2,1))



months <- seq(as.Date("2014/06/01", "%Y/%m/%d"), by="month", length.out=8)
monthslab <- format(months, format="%b")

vgridpos<-as.POSIXct(months,format=Ymd.format)
vgridlab<-monthslab

colfunc <- colorRampPalette(c("red", "darkgoldenrod1"))

timeframe <- as.POSIXct(c("2014/06/01","2014/09/01"),format=Ymd.format)



gantt.chart(gantt.info, taskcolors=colfunc(4),xlim=timeframe, main="Gantt
Chart for Fieldwork 2014",

priority.legend=TRUE,vgridpos=vgridpos,vgridlab=vgridlab,hgrid=TRUE)
```



Gantt chart is used to visualize sub-tasks/sub-events of an event or a project over time.