

Line charts

Dr. Niladri Chakraborty

1. Introduction to Line Charts in ggplot2

Line charts are one of the most popular and useful visualization techniques for displaying trends over time. In this tutorial, we will learn how to create beautiful and informative line charts using ggplot2, one of the most widely used plotting packages in R.

Required Packages

We will use the built-in **economics** dataset in R for this tutorial. The economics dataset contains quarterly data on US macroeconomic variables from 1960 to 2020, including variables such as the unemployment rate, personal income, and consumer spending.

First, we need to load the necessary packages and the data. We use the 'tidyverse' package that includes all the ggplot2 tools and several other packages.

```
library(tidyverse)

## — Attaching core tidyverse packages — tidyverse
## 2.0.0 —
## ✓ dplyr      1.1.2      ✓ readr      2.1.4
## ✓ forcats   1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2    3.4.2      ✓ tibble     3.2.1
## ✓ lubridate 1.9.2      ✓ tidyr      1.3.0
## ✓ purrr     1.0.1
## — Conflicts —
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

data(economics)
head(economics)

## # A tibble: 6 × 6
##   date       pce    pop psavert uempmed unemploy
##   <date>     <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 1967-07-01  507. 198712   12.6     4.5    2944
## 2 1967-08-01  510. 198911   12.6     4.7    2945
## 3 1967-09-01  516. 199113   11.9     4.6    2958
## 4 1967-10-01  512. 199311   12.9     4.9    3143
```

```
## 5 1967-11-01 517. 199498 12.8 4.7 3066
## 6 1967-12-01 525. 199657 11.8 4.8 3018
```

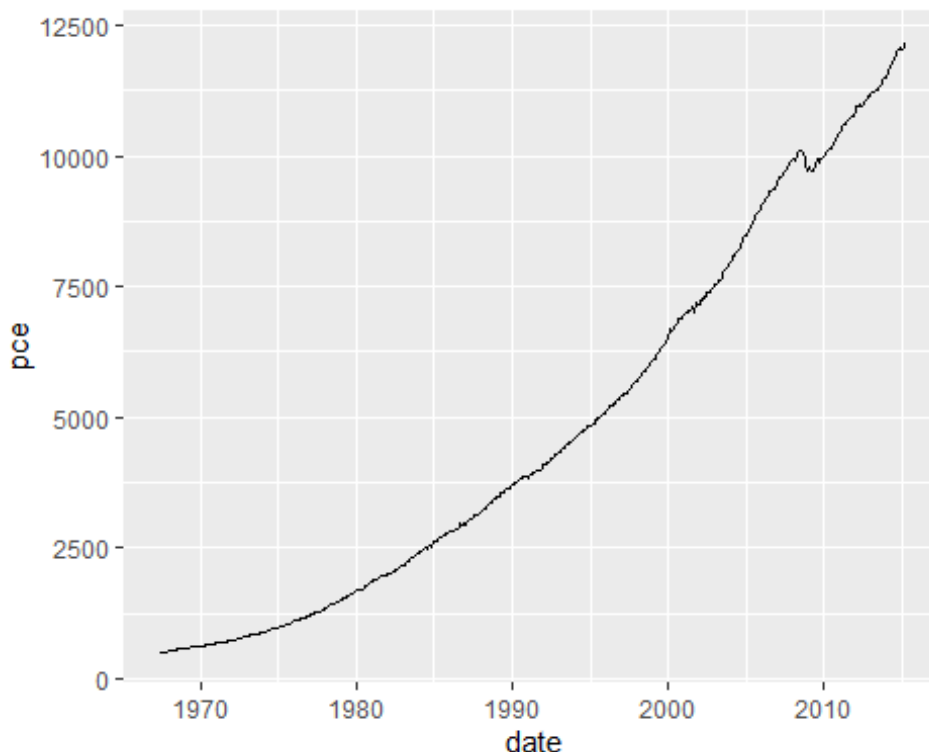
The economics dataset contains the following variables:

- date: the date in year-month format
- psavert: personal savings rate as a percentage of disposable personal income
- pce: personal consumption expenditures, in billions of dollars
- unemploy: the number of unemployed people, in thousands
- pop: the total population, in thousands
- realgdp: real gross domestic product, in billions of dollars

Creating a Basic Line Chart

To create a basic line chart in ggplot2, we first need to specify the data we want to use and the variables we want to map to the x and y axes. In this case, we want to plot the date on the x-axis and the GDP on the y-axis. We can use the following code:

```
ggplot(economics, aes(x = date, y = pce)) +  
  geom_line()
```



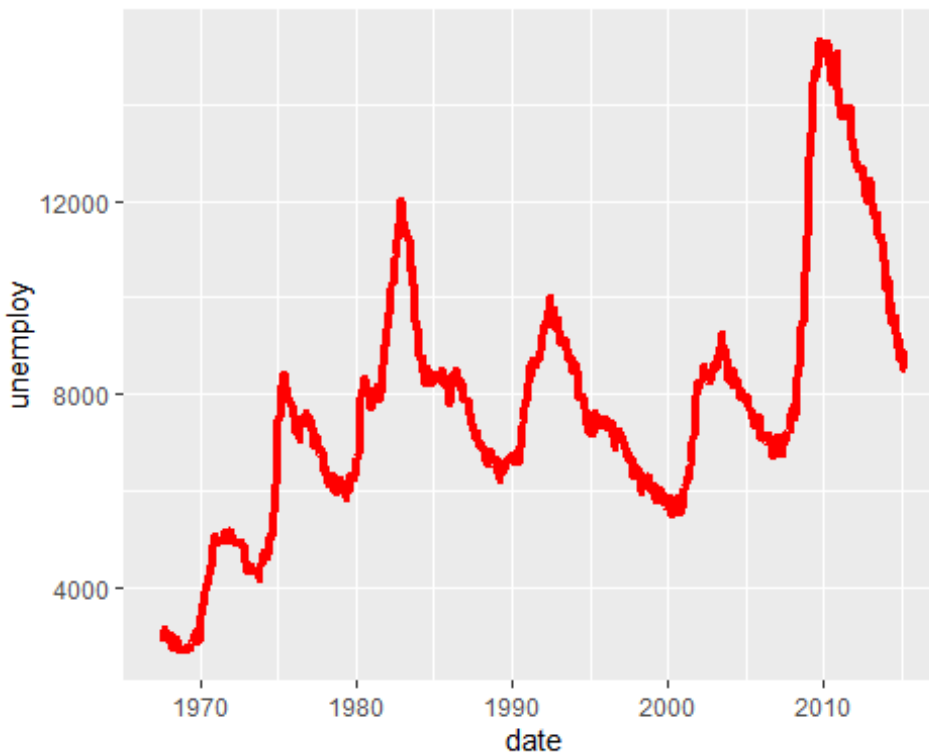
The `ggplot()` function initializes the plot, and the `aes()` function specifies the aesthetics, or the variables we want to map to the x and y axes. The `geom_line()` function creates the line chart.

Customizing Line Chart

Changing Line Color and Width

We can change the color and the width of the line by adding color and size arguments to `geom_line()`. Let's change the color of the line to red and the size of the line to 1.5.

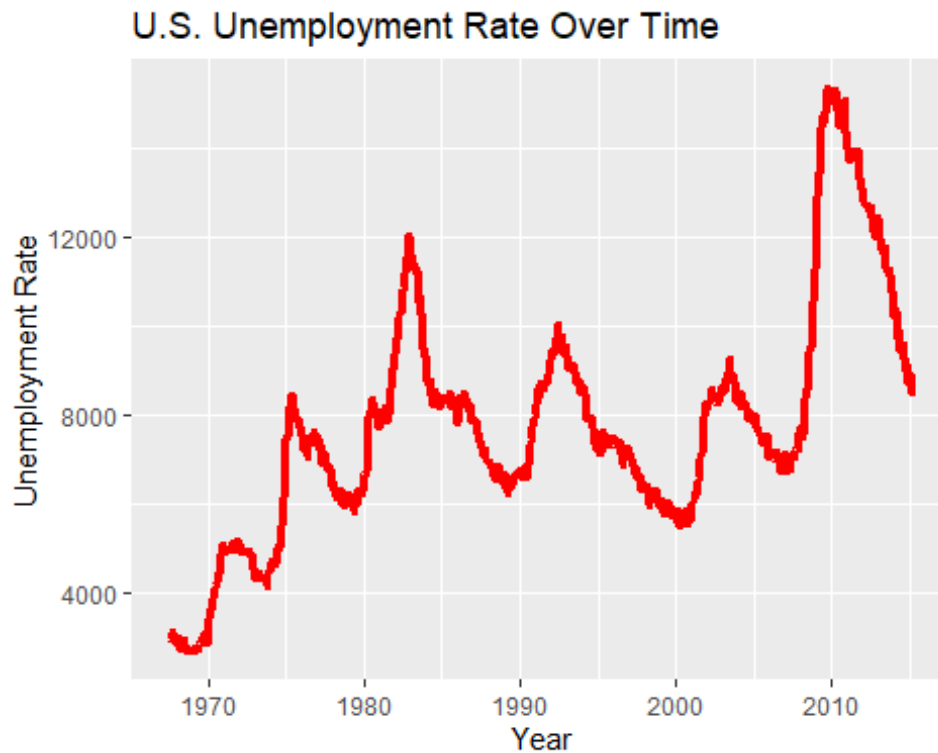
```
ggplot(data = economics, aes(x = date, y = unemploy)) +  
  geom_line(color = "red", size = 1.5)  
  
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## i Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```



Adding Labels and Titles

We can add a title to the plot by adding the `labs()` function to the plot. We can also add labels to the x-axis and y-axis by adding the `xlab()` and `ylab()` functions, respectively.

```
ggplot(data = economics, aes(x = date, y = unemploy)) +  
  geom_line(color = "red", size = 1.5) +  
  labs(title = "U.S. Unemployment Rate Over Time", x = "Year", y =  
        "Unemployment Rate")
```

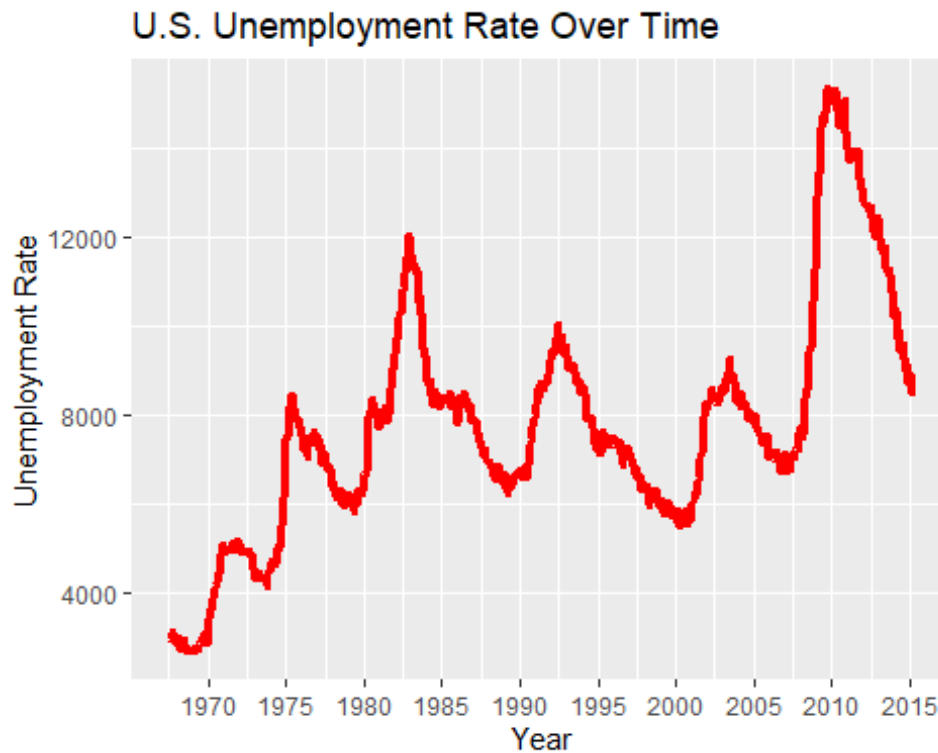


In `ggplot()` function, we can save the plot created by `ggplot()` function in a variable. Let us save the above plot in a variable 'chart'.

```
# Create a line chart of the 'unemploy' variable over time
chart = ggplot(data = economics, aes(x = date, y = unemploy)) +
  geom_line(color = "red", size = 1.5) +
  labs(title = "U.S. Unemployment Rate Over Time", x = "Year", y =
"Unemployment Rate")
```

Customize the x-axis tick marks and labels

```
chart <- chart + scale_x_date(date_labels = "%Y", date_breaks = "5 years")
chart
```



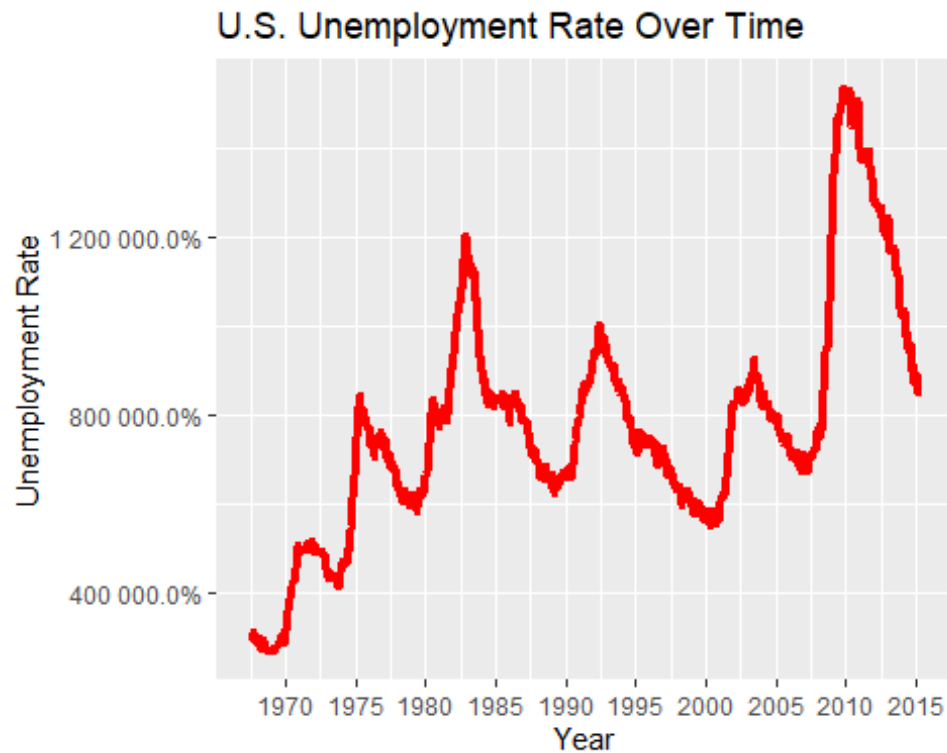
`scale_x_date` is a function in the `ggplot2` package in R that allows you to customize the x-axis scale of a plot when working with dates.

`date_labels` is an argument of `scale_x_date` that specifies the format in which the dates will be displayed as text labels. In this case, “%Y” indicates that only the year of the date will be displayed.

`date_breaks` is another argument of `scale_x_date` that specifies the interval at which the tick marks will be displayed on the x-axis. In this case, “5 years” indicates that the tick marks will be displayed every 5 years.

Customize the y-axis tick marks and labels

```
chart <- chart + scale_y_continuous(labels = scales::percent_format(accuracy = 0.1))
chart
```



`scale_y_continuous` is a function in the `ggplot2` package in R that allows you to customize the y-axis scale of a plot.

`labels` is an argument of `scale_y_continuous` that allows you to specify the text labels to be displayed on the y-axis ticks.

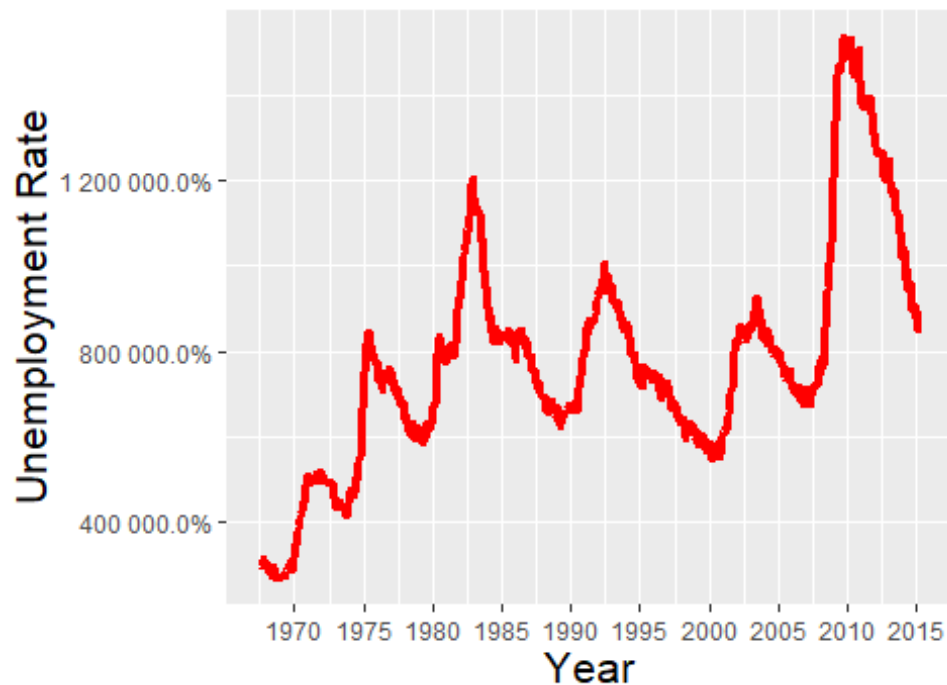
`scales::percent_format` is a function in the `scales` package in R that formats numeric values as percentages.

`accuracy = 0.1` is an argument of `scales::percent_format` that specifies the number of decimal places to be displayed in the percentage. In this case, it is set to 0.1, which means that one decimal point will be displayed in the percentage value.

Change the font size of the title and axis labels

```
chart <- chart + theme(plot.title = element_text(size = 20),  
                        axis.title = element_text(size = 16))  
chart
```

U.S. Unemployment Rate Over

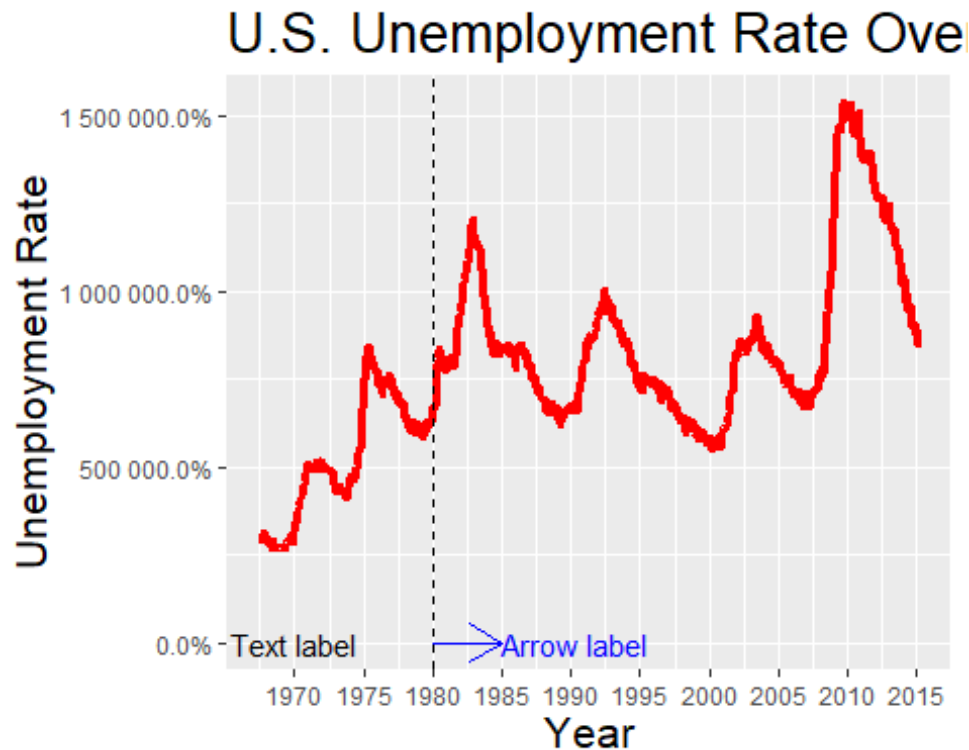


Add annotations to the chart

```
# Add annotations to the chart
chart +
  # Add a vertical line at a specific date
  geom_vline(xintercept = as.numeric(as.Date("1980-01-01")), linetype =
"dashed") +

  # Add a text label to the chart
  annotate("text", x = as.Date("1970-01-01"), y = 16, label = "Text label") +

  # Add an arrow and text label to the chart
  annotate("text", x = as.Date("1985-01-01"), y = 20, label = "Arrow label",
    hjust = 0, vjust = 0.5, size = 4, color = "blue") +
  annotate("segment", x = as.Date("1980-01-01"), y = 19, xend =
as.Date("1985-01-01"), yend = 20,
    arrow = arrow(length = unit(0.5, "cm")), color = "blue")
```



The first annotation is adding a text label to the plot using the following arguments:

- `text`: This specifies the type of annotation to add, which in this case is text.
- `x = as.Date("1985-01-01")`: This specifies the x-coordinate of the text label on the plot, which is converted to a date format using the `as.Date()` function.
- `y = 20`: This specifies the y-coordinate of the text label on the plot.
- `label = "Arrow label"`: This specifies the text that should be displayed as the label.
- `hjust = 0, vjust = 0.5`: These arguments specify the horizontal and vertical justification of the text label.
- `size = 4`: This specifies the font size of the text label.
- `color = "blue"`: This specifies the color of the text label.
- Horizontal justification (`hjust`) specifies how the annotation should be aligned horizontally relative to its x-coordinate. A value of 0 aligns the annotation to the right of its x-coordinate, a value of 1 aligns it to the left, and a value of 0.5 aligns it to the center.
- Vertical justification (`vjust`) specifies how the annotation should be aligned vertically relative to its y-coordinate. A value of 0 aligns the annotation to the bottom of its y-coordinate, a value of 1 aligns it to the top, and a value of 0.5 aligns it to the center.

The second annotation is adding a segment with an arrow to the plot using the following arguments:

- “segment”: This specifies the type of annotation to add, which in this case is a line segment with an arrow.
- `x = as.Date("1980-01-01")`: This specifies the starting x-coordinate of the line segment on the plot.
- `y = 19`: This specifies the starting y-coordinate of the line segment on the plot. `xend = as.Date("1985-01-01")`: This specifies the ending x-coordinate of the line segment on the plot.
- `yend = 20`: This specifies the ending y-coordinate of the line segment on the plot.
- `arrow = arrow(length = unit(0.5, "cm"))`: This specifies that an arrow should be added to the end of the line segment, with a length of 0.5 cm.
- `color = "blue"`: This specifies the color of the line segment and arrow.

```
# Add annotations to the chart
chart +
  # Add a vertical line at a specific date
  geom_vline(xintercept = as.numeric(as.Date("1980-01-01")), linetype =
"dashed") +

  # Add a text label to the chart
  annotate("text", x = as.Date("1970-01-01"), y = 16, label = "Text label") +

  # Add an arrow and text label to the chart
  annotate("text", x = as.Date("1985-01-01"), y = 20, label = "Arrow label",
    hjust = 1, vjust = 0.5, size = 4, color = "blue") +
  annotate("segment", x = as.Date("1980-01-01"), y = 19, xend =
as.Date("1985-01-01"), yend = 20,
    arrow = arrow(length = unit(0.5, "cm")), color = "blue")
```

U.S. Unemployment Rate Over

