# A Simple Demonstration of Time Series Development for Fiscal Year Expenditure Data

Le' Sean Roberts (NYC Office of Management and Budget College Aide)

## Time Data Cleaning and Time Series Analysis Intro

### Time Data Cleaning

Time data cleaning is a crucial step in time series analysis and forecasting to ensure the accuracy and reliability of the results. It involves preprocessing the time-related data to handle missing values, outliers, inconsistencies, and other anomalies that may affect the analysis. Here are some conventional issues concerning data cleaning for time series development:

1. **Handling missing values:** Missing values are common in time series data and can arise due to various reasons such as equipment failure, data collection errors, or system downtime. It's important to handle missing values appropriately before proceeding with analysis. This can involve techniques such as imputation (replacing missing values with estimated values based on surrounding data points), deletion of rows with missing values, or interpolation.
2. **Dealing with outliers:** Outliers are data points that significantly deviate from the general pattern of the data. They can distort the analysis and affect the accuracy of forecasts. Outliers should be identified and, if necessary, treated using techniques such as robust statistical methods, transformations, or trimming.
3. **Resampling and frequency adjustment:** Time series data may be collected at different frequencies (e.g., daily, weekly, monthly). In some cases, it may be necessary to resample the data to a consistent frequency or adjust the frequency to match the requirements of the analysis. This can involve aggregation, interpolation, or downsampling/upsampling techniques.
4. **Handling time zone and daylight saving time (DST):** Time zone differences and changes in daylight saving time can introduce complexities in time series analysis, especially for datasets collected from multiple locations. It's important to standardize time zones and handle DST adjustments appropriately to ensure consistency in the data.

5. **Detecting and correcting inconsistencies:** Time series data may contain inconsistencies such as duplicate entries, data entry errors, or conflicting information. Detecting and correcting these inconsistencies is essential for maintaining data integrity and ensuring accurate analysis.
6. **Accounting for seasonality and calendar effects:** Many time series datasets exhibit seasonality and calendar effects, which can influence the analysis and forecasts. It's important to identify and account for these effects appropriately using techniques such as seasonal decomposition or calendar adjustment.

In this project however, most of the above issues will not be encountered due to the goal of having a simple demonstration of time series development.

## Time Series Analysis

Time series analysis is a branch of statistics and data science that deals with analyzing and forecasting sequential data points collected over time. It is widely used in various fields such as finance, economics, weather forecasting, signal processing, and more. Here's a small introduction to time series analysis:

1. **What is a Time Series?**
   - A time series is a sequence of data points collected or recorded at regular intervals over time. Each data point in the series is associated with a specific time index or timestamp.
   - Time series data can exhibit various patterns, including trends, seasonality, cycles, and irregular fluctuations or noise.
2. **Goals of Time Series Analysis:**
   - Understand the underlying structure and patterns present in the time series data.
   - Make predictions or forecasts about future values of the time series.
   - Extract meaningful insights and relationships between variables over time.
   - Monitor and analyze changes or anomalies in the time series for decision-making purposes.
3. **Key Concepts in Time Series Analysis:**
   - **Trend:** The long-term movement or directionality of the time series data over time.
   - **Seasonality:** Periodic patterns or fluctuations that occur at regular intervals within the time series data.

- **Cycles:** Recurring patterns or fluctuations that are not strictly periodic but occur over longer time periods.
- **Autocorrelation:** The correlation between a time series and a lagged version of itself, indicating the degree of dependence between consecutive observations.
- **Stationarity:** A property of time series data where the statistical properties such as mean, variance, and autocorrelation structure remain constant over time.

4. **Methods and Techniques in Time Series Analysis:**

    - **Descriptive Analysis:** Summarizing and visualizing the time series data using statistical measures, plots, and charts.
    - **Time Series Decomposition:** Decomposing the time series into its constituent components such as trend, seasonality, and noise.
    - **Forecasting:** Making predictions about future values of the time series using statistical models, machine learning algorithms, or time series decomposition methods.
    - **Modeling:** Fitting mathematical models such as autoregressive integrated moving average (ARIMA), exponential smoothing, or seasonal decomposition of time series (STL) to capture the underlying patterns and relationships in the data.
    - **Evaluation:** Assessing the accuracy and performance of forecasting models using metrics such as mean absolute error (MAE), root mean squared error (RMSE), and others.

Time series analysis provides valuable insights into temporal data patterns, allowing analysts and decision-makers to understand past trends, anticipate future behavior, and make informed decisions based on data-driven forecasts and predictions.

As well, in this project however, most of the above analyses will not be encountered due to the goal of having a simple demonstration of time series development.

## Data Source

The applied data stems from the NYC Independent Budget Office (IBO) collection of capital expenditures since for different NYC gov't agencies 1985; annual capital expenditures, by capital budget area and agency, from FY 1985 - 2020. Through records of the Comprehensive Annual Financial Reports of the Comptroller.

## Time Data Cleaning & Wrangling

Process starts with fetching/loading the data of interest.

```r
library(readr)
NYC_Independent_Budget_Office_IBO_Capital_Expenditures_Since_1985_20240228 <-
  read_csv("C:/Users/verlene/Downloads/NYC_Independent_Budget_Office__IBO__Capital_Expendi
library(tidyverse)
glimpse(NYC_Independent_Budget_Office_IBO_Capital_Expenditures_Since_1985_20240228)
```

```
Rows: 63
Columns: 38
$ `AGENCY CAPITAL EXPENDITURES BY PURPOSE` <chr> "Environmental Protection and~
$ CATEGORY                                 <chr> NA, "Sewage Collection and Tr~
$ `FY 2020`                                <dbl> 1845932787, 1066265296, 72644~
$ `FY 2019`                                <dbl> 1991755815, 1034894687, 89102~
$ `FY 2018`                                <chr> "1,687,882,725", "844,046,079~
$ `FY 2017`                                <chr> "1,453,949,135", "831,878,907~
$ `FY 2016`                                <chr> "1,378,234,234", "767,563,124~
$ `FY 2015`                                <chr> "1,373,488,401", "687,002,478~
$ `FY 2014`                                <chr> "1,577,802,893", "803,228,494~
$ `FY 2013`                                <dbl> 1843947453, 953371696, 798383~
$ `FY 2012`                                <dbl> 2405599562, 1195170589, 10414~
$ `FY 2011`                                <dbl> 2824135363, 1303130966, 12809~
$ `FY 2010`                                <dbl> 2625317710, 1041023183, 14146~
$ `FY 2009`                                <dbl> 2700236388, 1342826997, 12675~
$ `FY 2008`                                <dbl> 2313038722, 1094743184, 11550~
$ `FY 2007`                                <dbl> 1948836326, 1065666287, 78979~
$ `FY 2006`                                <dbl> 1841278686, 1027913991, 71237~
$ `FY 2005`                                <dbl> 1679394109, 1026980915, 56721~
$ `FY  2004`                               <dbl> 1630607201, 1028832860, 50384~
$ `FY 2003`                                <dbl> 1301779898, 815280838, 377006~
$ `FY 2002`                                <dbl> 796865009, 485352807, 2437272~
$ `FY 2001`                                <dbl> 830009738, 465302117, 2900927~
$ `FY 2000`                                <dbl> 1036705764, 540055147, 424331~
$ `FY 1999`                                <dbl> 787928245, 469587367, 2605486~
$ `FY 1998`                                <dbl> 764767187, 408723581, 3030676~
$ `FY 1997`                                <dbl> 977555046, 403288236, 3981259~
$ `FY 1996`                                <dbl> 1004251889, 363907447, 445796~
$ `FY 1995`                                <dbl> 705398608, 387874556, 2098876~
$ `FY 1994`                                <dbl> 616392666, 393502317, 1725403~
$ `FY 1993`                                <dbl> 745534529, 498388858, 2231112~
$ `FY 1992`                                <dbl> 893639535, 717335346, 1535736~
$ `FY 1991`                                <dbl> 826139905, 639502160, 1653736~
$ `FY 1990`                                <dbl> 636874169, 455579594, 1692626~
$ `FY 1989`                                <dbl> 621770158, 431270960, 1824071~
$ `FY 1988`                                    <dbl> 567085283, 397428543, 1644212~
$ `FY 1987`                                    <dbl> 562163938, 410829792, 1464941~
$ `FY 1986`                                    <dbl> 564968375, 407948120, 1543870~
$ `FY 1985`                                    <dbl> 498475715, 356499516, 1390849~
```

Observed potentially is the “unprofessional” habit of placing commas and decimal points in currency values. As well, the common existence of NAs throughout. Will deal with the latter first.

```
# Remove NAs from rows
Expenditure_data <-
  NYC_Independent_Budget_Office_IBO_Capital_Expenditures_Since_1985_20240228[complete.case
```

Concerning the removal of characters or strings in instances, columns used for identification or characterization, namely, agency identification and the types of expenditure projects will be excluded.

```
# Define the columns to clean (excluding the first two columns)
cols_to_clean <- names(Expenditure_data)[-(1:2)]

# Remove commas and decimal points from all specified columns and convert to numeric
for (col in cols_to_clean) {
  # Remove commas
  Expenditure_data[[col]] <- gsub(",", "", Expenditure_data[[col]])
  # Remove decimal points
  Expenditure_data[[col]] <- gsub("\\.", "", Expenditure_data[[col]])
  # Convert to numeric, ignoring errors
  Expenditure_data[[col]] <-
    as.numeric(Expenditure_data[[col]], errors = "ignore")
}
```

```
Warning: NAs introduced by coercion

Warning: NAs introduced by coercion

Warning: NAs introduced by coercion

Warning: NAs introduced by coercion

Warning: NAs introduced by coercion
```

```r
# Showing first 5 Or 6 rows
head(Expenditure_data)
```

```
# A tibble: 6 x 38
  AGENCY CAPITAL EXPENDITURES~1 CATEGORY `FY 2020` `FY 2019` `FY 2018` `FY 2017`
  <chr>                          <chr>        <dbl>     <dbl>     <dbl>     <dbl>
1 Environmental Protection and~ Sewage ~    1.07e9    1.03e9 844046079 831878907
2 Environmental Protection and~ Water S~    7.26e8    8.91e8 778809329 554826868
3 Environmental Protection and~ Equipme~    5.32e7    6.58e7  65027317  67243360
4 Department of Sanitation      Equipme~    1.32e8    1.70e8 151393589 150146378
5 Department of Sanitation      Garages     4.12e7    4.12e7  42131402  22723297
6 Department of Sanitation      Waste D~    2.97e7    3.14e7  95995252 150863652
# i abbreviated name: 1: `AGENCY CAPITAL EXPENDITURES BY PURPOSE`
# i 32 more variables: `FY 2016` <dbl>, `FY 2015` <dbl>, `FY 2014` <dbl>,
#   `FY 2013` <dbl>, `FY 2012` <dbl>, `FY 2011` <dbl>, `FY 2010` <dbl>,
#   `FY 2009` <dbl>, `FY 2008` <dbl>, `FY 2007` <dbl>, `FY 2006` <dbl>,
#   `FY 2005` <dbl>, `FY  2004` <dbl>, `FY 2003` <dbl>, `FY 2002` <dbl>,
#   `FY 2001` <dbl>, `FY 2000` <dbl>, `FY 1999` <dbl>, `FY 1998` <dbl>,
#   `FY 1997` <dbl>, `FY 1996` <dbl>, `FY 1995` <dbl>, `FY 1994` <dbl>, ...
```

Further NA issues to be dealt with:

```r
Expenditure_data <- Expenditure_data |>
  na.omit()
```

```r
str(Expenditure_data)
```

```
tibble [19 x 38] (S3: tbl_df/tbl/data.frame)
 $ AGENCY CAPITAL EXPENDITURES BY PURPOSE: chr [1:19] "Environmental Protection and Sanitati
 $ CATEGORY                              : chr [1:19] "Sewage Collection and Treatment" "Wate
 $ FY 2020                               : num [1:19] 1.07e+09 7.26e+08 5.32e+07 1.32e+08 4.
 $ FY 2019                               : num [1:19] 1.03e+09 8.91e+08 6.58e+07 1.70e+08 4.
 $ FY 2018                               : num [1:19] 8.44e+08 7.79e+08 6.50e+07 1.51e+08 4.
 $ FY 2017                               : num [1:19] 8.32e+08 5.55e+08 6.72e+07 1.50e+08 2.
 $ FY 2016                               : num [1:19] 7.68e+08 5.55e+08 5.58e+07 1.56e+08 3.
 $ FY 2015                               : num [1:19] 6.87e+08 6.05e+08 8.12e+07 1.09e+08 6.
 $ FY 2014                               : num [1:19] 8.03e+08 7.08e+08 6.65e+07 7.77e+07 8.
 $ FY 2013                               : num [1:19] 9.53e+08 7.98e+08 9.22e+07 1.29e+08 1.
 $ FY 2012                               : num [1:19] 1.20e+09 1.04e+09 1.69e+08 1.09e+08 8.
 $ FY 2011                               : num [1:19] 1.30e+09 1.28e+09 2.40e+08 6.06e+07 6.
 $ FY 2010                               : num [1:19] 1.04e+09 1.41e+09 1.70e+08 1.48e+08 1.5
 $ FY 2009                               : num [1:19] 1.34e+09 1.27e+09 8.98e+07 1.73e+08 5.2
 $ FY 2008                               : num [1:19] 1.09e+09 1.16e+09 6.33e+07 1.08e+08 6.8
 $ FY 2007                               : num [1:19] 1.07e+09 7.90e+08 9.34e+07 4.40e+07 8.2
 $ FY 2006                               : num [1:19] 1.03e+09 7.12e+08 1.01e+08 2.10e+07 6.1
 $ FY 2005                               : num [1:19] 1.03e+09 5.67e+08 8.52e+07 5.26e+07 7.1
 $ FY  2004                              : num [1:19] 1.03e+09 5.04e+08 9.79e+07 4.02e+07 8.0
 $ FY 2003                               : num [1:19] 8.15e+08 3.77e+08 1.09e+08 5.80e+07 3.5
 $ FY 2002                               : num [1:19] 4.85e+08 2.44e+08 6.78e+07 2.81e+07 8.0
 $ FY 2001                               : num [1:19] 4.65e+08 2.90e+08 7.46e+07 1.32e+08 2.9
 $ FY 2000                               : num [1:19] 5.40e+08 4.24e+08 7.23e+07 1.43e+08 3.8
 $ FY 1999                               : num [1:19] 4.70e+08 2.61e+08 5.78e+07 2.56e+07 1.8
 $ FY 1998                               : num [1:19] 4.09e+08 3.03e+08 5.30e+07 1.71e+07 7.1
 $ FY 1997                               : num [1:19] 4.03e+08 3.98e+08 1.76e+08 1.76e+07 5.9
 $ FY 1996                               : num [1:19] 3.64e+08 4.46e+08 1.95e+08 4.02e+07 1.5
 $ FY 1995                               : num [1:19] 3.88e+08 2.10e+08 1.08e+08 5.30e+07 1.4
 $ FY 1994                               : num [1:19] 3.94e+08 1.73e+08 5.04e+07 2.92e+07 7.4
 $ FY 1993                               : num [1:19] 4.98e+08 2.23e+08 2.40e+07 2.13e+07 1.1
 $ FY 1992                               : num [1:19] 7.17e+08 1.54e+08 2.27e+07 2.92e+07 6.6
 $ FY 1991                               : num [1:19] 6.40e+08 1.65e+08 2.13e+07 7.87e+07 3.5
 $ FY 1990                               : num [1:19] 4.56e+08 1.69e+08 1.20e+07 4.81e+07 7.3
 $ FY 1989                               : num [1:19] 4.31e+08 1.82e+08 8.09e+06 1.00e+08 3.8
 $ FY 1988                               : num [1:19] 3.97e+08 1.64e+08 5.24e+06 4.42e+07 5.1
 $ FY 1987                               : num [1:19] 4.11e+08 1.46e+08 4.84e+06 6.44e+07 2.6
 $ FY 1986                               : num [1:19] 4.08e+08 1.54e+08 2.63e+06 3.04e+07 7.2
 $ FY 1985                               : num [1:19] 3.56e+08 1.39e+08 2.89e+06 6.25e+07 0.0
 - attr(*, "na.action")= 'omit' Named int [1:3] 12 21 22
  ..- attr(*, "names")= chr [1:3] "12" "21" "22"
```

Now, to isolate a specific agency w.r.t. project considered. The following case concerns “Sewage Collection and Treatment”.

```
EP_Sewage <- Expenditure_data |>
  filter(CATEGORY == "Sewage Collection and Treatment")
EP_Sewage
```

```
# A tibble: 1 x 38
  AGENCY CAPITAL EXPENDITURES~1 CATEGORY `FY 2020` `FY 2019` `FY 2018` `FY 2017`
  <chr>                         <chr>        <dbl>     <dbl>     <dbl>     <dbl>
1 Environmental Protection and~ Sewage ~    1.07e9    1.03e9 844046079 831878907
# i abbreviated name: 1: `AGENCY CAPITAL EXPENDITURES BY PURPOSE`
# i 32 more variables: `FY 2016` <dbl>, `FY 2015` <dbl>, `FY 2014` <dbl>,
#   `FY 2013` <dbl>, `FY 2012` <dbl>, `FY 2011` <dbl>, `FY 2010` <dbl>,
#   `FY 2009` <dbl>, `FY 2008` <dbl>, `FY 2007` <dbl>, `FY 2006` <dbl>,
#   `FY 2005` <dbl>, `FY  2004` <dbl>, `FY 2003` <dbl>, `FY 2002` <dbl>,
#   `FY 2001` <dbl>, `FY 2000` <dbl>, `FY 1999` <dbl>, `FY 1998` <dbl>,
#   `FY 1997` <dbl>, `FY 1996` <dbl>, `FY 1995` <dbl>, `FY 1994` <dbl>, ...
```

```
EP_Water <- Expenditure_data |>
  filter(CATEGORY == "Water Supply and Distribution")
EP_Water
```

```
# A tibble: 1 x 38
  AGENCY CAPITAL EXPENDITURES~1 CATEGORY `FY 2020` `FY 2019` `FY 2018` `FY 2017`
  <chr>                         <chr>        <dbl>     <dbl>     <dbl>     <dbl>
1 Environmental Protection and~ Water S~ 726443381 891027669 778809329 554826868
# i abbreviated name: 1: `AGENCY CAPITAL EXPENDITURES BY PURPOSE`
# i 32 more variables: `FY 2016` <dbl>, `FY 2015` <dbl>, `FY 2014` <dbl>,
#   `FY 2013` <dbl>, `FY 2012` <dbl>, `FY 2011` <dbl>, `FY 2010` <dbl>,
#   `FY 2009` <dbl>, `FY 2008` <dbl>, `FY 2007` <dbl>, `FY 2006` <dbl>,
#   `FY 2005` <dbl>, `FY  2004` <dbl>, `FY 2003` <dbl>, `FY 2002` <dbl>,
#   `FY 2001` <dbl>, `FY 2000` <dbl>, `FY 1999` <dbl>, `FY 1998` <dbl>,
#   `FY 1997` <dbl>, `FY 1996` <dbl>, `FY 1995` <dbl>, `FY 1994` <dbl>, ...
```

The Department of Sanitation “Equipment” initiative example:

```
Dos_Equip <- Expenditure_data |>
  filter(CATEGORY == "Equipment",
         `AGENCY CAPITAL EXPENDITURES BY PURPOSE`== "Department of Sanitation")
Dos_Equip
```

```
# A tibble: 1 x 38
  AGENCY CAPITAL EXPENDITURES~1 CATEGORY `FY 2020` `FY 2019` `FY 2018` `FY 2017`
  <chr>                         <chr>        <dbl>     <dbl>     <dbl>     <dbl>
1 Department of Sanitation      Equipme~ 131518334 170257950 151393589 150146378
# i abbreviated name: 1: `AGENCY CAPITAL EXPENDITURES BY PURPOSE`
# i 32 more variables: `FY 2016` <dbl>, `FY 2015` <dbl>, `FY 2014` <dbl>,
#   `FY 2013` <dbl>, `FY 2012` <dbl>, `FY 2011` <dbl>, `FY 2010` <dbl>,
#   `FY 2009` <dbl>, `FY 2008` <dbl>, `FY 2007` <dbl>, `FY 2006` <dbl>,
#   `FY 2005` <dbl>, `FY  2004` <dbl>, `FY 2003` <dbl>, `FY 2002` <dbl>,
#   `FY 2001` <dbl>, `FY 2000` <dbl>, `FY 1999` <dbl>, `FY 1998` <dbl>,
#   `FY 1997` <dbl>, `FY 1996` <dbl>, `FY 1995` <dbl>, `FY 1994` <dbl>, ...
```

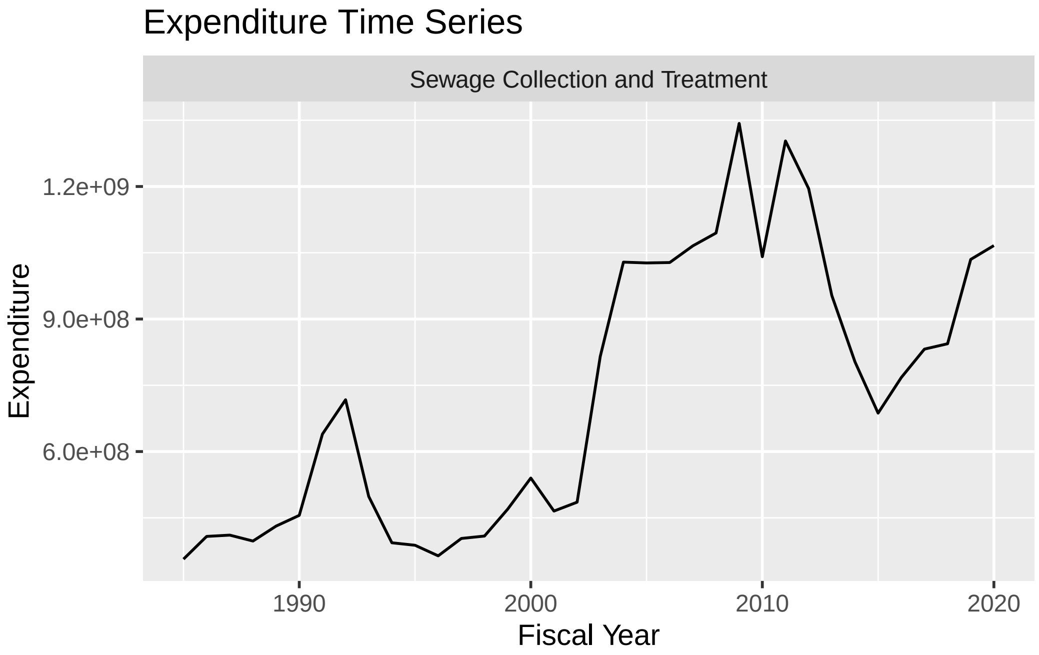
Now, observed in prior cases, the time series data is expressed in wide format which each time period assigned a column; something that's contrary to customary time series data in a data frame dictated by time duration in row format. Hence, to now resolve the matter and provide a time series exhibition.

```r
library(tidyr)
library(ggplot2)

# Convert from wide to long format
EP_Sewage_tseries <-
  pivot_longer(EP_Sewage,
               cols = -c(`AGENCY CAPITAL EXPENDITURES BY PURPOSE`
                                           , CATEGORY),
                                 names_to = "fiscal_year",
                                 values_to = "expenditure")

# Extract fiscal year from column names
EP_Sewage_tseries$fiscal_year <-
  as.numeric(gsub("FY ", "", EP_Sewage_tseries$fiscal_year))

# Plot the time series
ggplot(EP_Sewage_tseries, aes(x = fiscal_year, y = expenditure)) +
  geom_line() +
  labs(x = "Fiscal Year", y = "Expenditure",
       title = "Expenditure Time Series") +
  facet_wrap(~ CATEGORY, scales = "free_y")
```



### Time Series Analysis

#### Summary Statistics

Calculate and display summary statistics such as mean, median, standard deviation, minimum, and maximum values of the time series data. These statistics provide insights into the central tendency, variability, and range of the data.

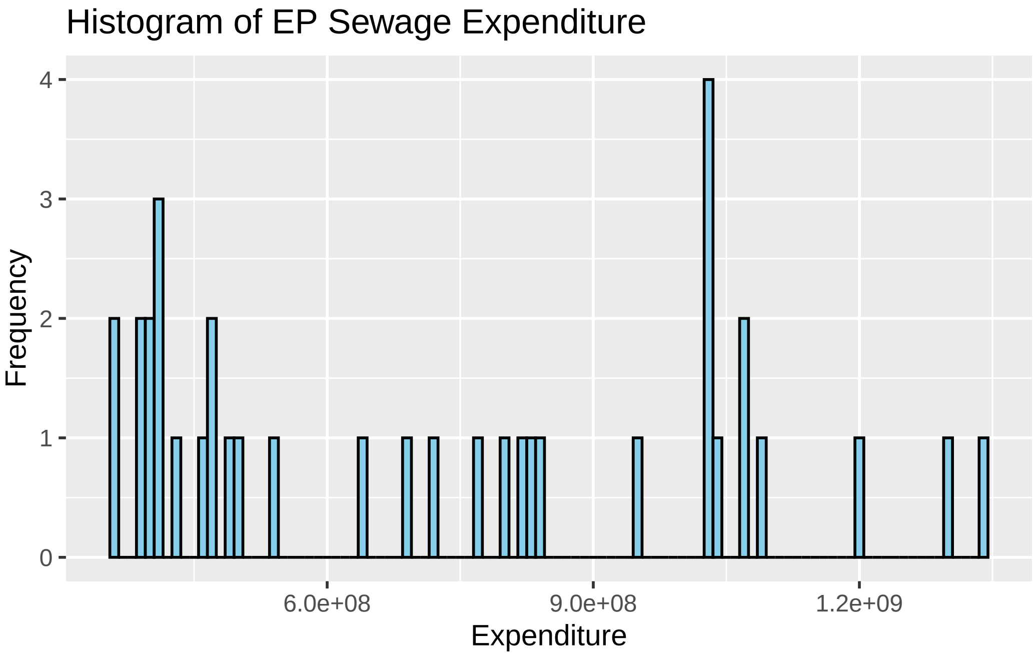
```
# Summary Statistics
summary(EP_Sewage_tseries$expenditure)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
3.565e+08 4.262e+08 7.022e+08 7.267e+08 1.028e+09 1.343e+09 
```

#### Histogram

Create a histogram to visualize the distribution of the expenditure values. This helps in understanding the frequency distribution of the data and identifying any potential patterns or outliers.

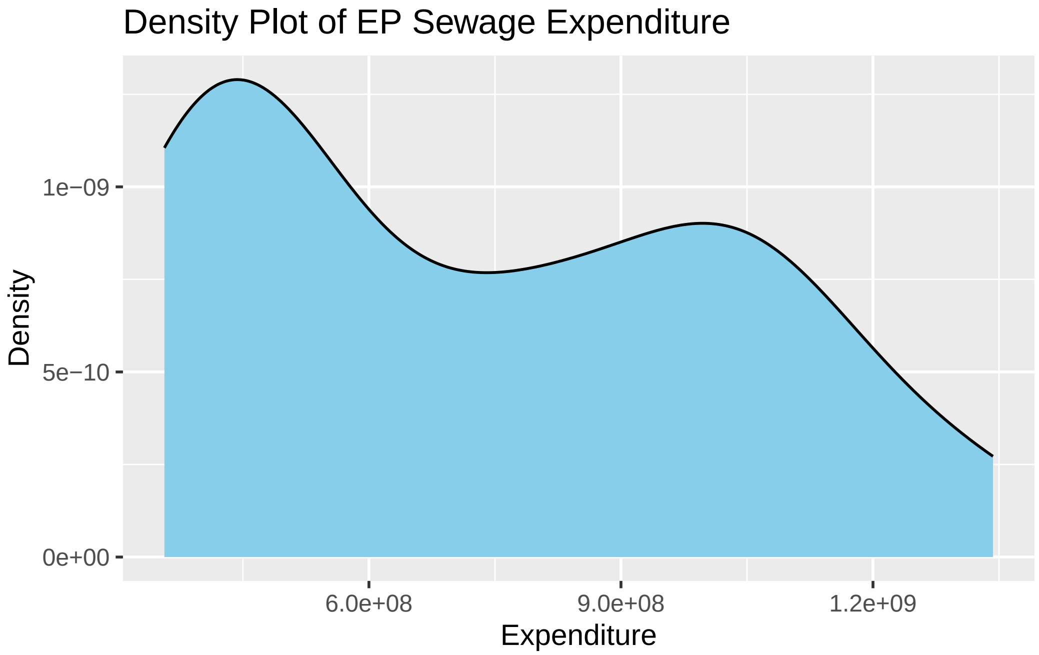
```
# Histogram
ggplot(EP_Sewage_tseries, aes(x = expenditure)) +
  geom_histogram(binwidth = 10000000, fill = "skyblue", color = "black") +
  labs(x = "Expenditure", y = "Frequency",
       title = "Histogram of EP Sewage Expenditure")
```



#### Density Plot

Plot a kernel density estimate (KDE) or density plot to visualize the probability density function of the expenditure values. This provides a smoother representation of the distribution compared to the histogram.

```
# Density plot
ggplot(EP_Sewage_tseries, aes(x = expenditure)) +
  geom_density(fill = "skyblue", color = "black") +
  labs(x = "Expenditure", y = "Density",
       title = "Density Plot of EP Sewage Expenditure")
```

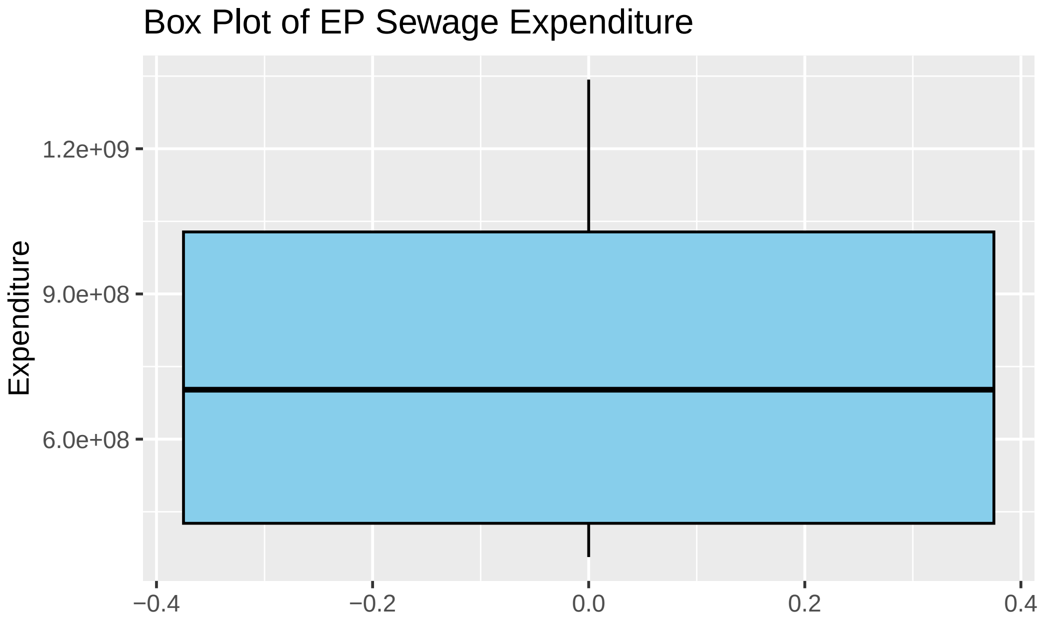


What’s observed above definitely isn’t close to normal distribution.

### Box Plot

Create a box plot to visualize the distribution of expenditure values, including measures such as median, quartiles, and potential outliers.

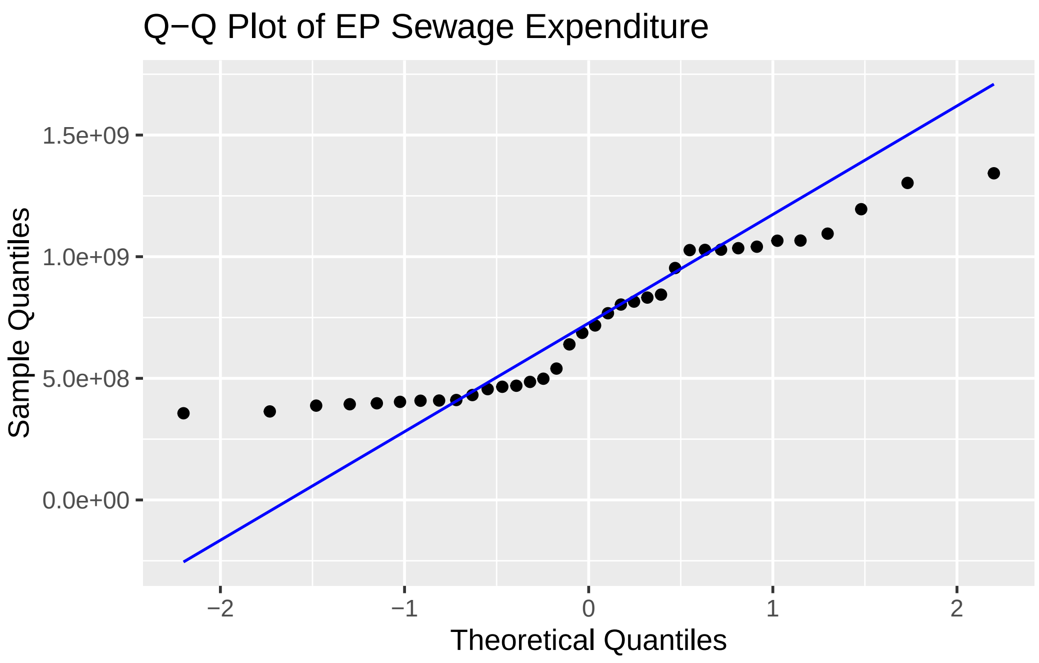
```
# Box plot
ggplot(EP_Sewage_tseries, aes(y = expenditure)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(x = "", y = "Expenditure",
       title = "Box Plot of EP Sewage Expenditure")
```



### Quantile-Quantile Plot

A Q-Q plot, short for quantile-quantile plot, is a graphical tool used in statistics to assess whether a set of data follows a specific probability distribution, such as the normal distribution. The purpose of a Q-Q plot is to visually compare the quantiles of the empirical data with the quantiles of a theoretical distribution.

```
# Q-Q plot
ggplot(EP_Sewage_tseries, aes(sample = expenditure)) +
  stat_qq() +
  stat_qq_line(color = "blue") +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "Q-Q Plot of EP Sewage Expenditure")
```



By using the statistical measures and prior three visualization techniques, one can effectively convey the distribution of the time series data and gain insights into its characteristics, central tendency, variability, and potential outliers. Adjust the parameters and aesthetics of the plots as needed to enhance clarity and interpretability.

**NOTE:** Normal distribution in nature should not be implied. As well, outliers are not necessarily bugs or inconveniences, rather they well define "extremes" or potential "absolutes"; quite useful to identify potential high density regions of false positives with classification techniques in machine learning; dealing with diseases and other things.

### Exponential Smoothing

Since the data is annual and there's interest to perform forecasting, to try using a forecasting method that doesn't rely on seasonal components, such as exponential smoothing. Exponential smoothing is a popular time series forecasting method used to make predictions based on the weighted average of past observations, with more recent observations typically given greater weight. It's a simple and efficient method that can effectively capture trends and patterns in the data.

The basic idea behind exponential smoothing is to assign exponentially decreasing weights to past observations. The forecast for the next time period is then obtained by combining the weighted average of past observations with a smoothing parameter (often denoted as  ) that controls the rate at which older observations are discounted.

```
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
# Perform exponential smoothing forecasting
expenditure_EP_Sewage_ts <-
  ts(EP_Sewage_tseries$expenditure,
                               start = min(EP_Sewage_tseries$fiscal_year), frequency = 1)
exp_model <- forecast::ets(expenditure_EP_Sewage_ts)  # Exponential smoothing
forecast_values <- forecast(exp_model, h = 3) # Forecast for 2021, 2022, 2023

# Print the forecasted values
print(forecast_values)
```

```
     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
2021      356504662 276937348 436071975 234816956 478192367
2022      356504662 243135241 469874082 183121094 529888229
2023      356504662 216604505 496404818 142545835 570463488
```

The summary results of the forecast provide useful information about the forecasted values and the accuracy of the forecast. Let's interpret the summary results:

1. **Point forecasts (Forecast):** These are the estimated values for each forecast horizon (2021, 2022, 2023). These values represent the most likely outcome according to the forecasting model.
2. **Prediction intervals (Lo 80, Hi 80, Lo 95, Hi 95):** These intervals provide a range within which the actual future values are expected to fall with a certain level of confidence. The "Lo 80" and "Hi 80" columns represent the 80% prediction interval, while the "Lo 95" and "Hi 95" columns represent the 95% prediction interval. For example, the "Lo 80" and "Hi 80" columns provide a range within which 80% of the actual future values are expected to fall.
3. **Standard errors (SE):** These values represent the standard errors associated with the point forecasts. Smaller standard errors indicate higher precision in the forecasts.

4. **Mean absolute error (MAE):** This is a measure of the average absolute difference between the forecasted values and the actual values. A lower MAE indicates better accuracy of the forecasts.

5. **Root mean squared error (RMSE):** This is a measure of the average magnitude of the forecast errors. It penalizes larger errors more heavily compared to MAE. Like MAE, a lower RMSE indicates better accuracy of the forecasts.

6. **Residual standard error (RSE):** This is an estimate of the standard deviation of the forecast errors. It provides a measure of the variability of the forecast errors around the mean.

7. **AIC (Akaike Information Criterion):** This is a measure of the goodness of fit of the forecasting model. Lower AIC values indicate better model fit, considering the trade-off between goodness of fit and model complexity.

The summary results provide information about the forecasted values, prediction intervals, accuracy measures (MAE, RMSE), standard errors, and goodness of fit (AIC) of the forecasting model. These measures help assess the reliability and accuracy of the forecasts and guide decision-making processes.

Time Series case studies for other NYC agencies based on the developed "Expenditure_data" data frame can also be done in similar fashion.

## Conclusion

In conclusion, time data cleaning and time series analysis are essential steps in the process of extracting meaningful insights and making accurate predictions from time-stamped data.

Time data cleaning involves preprocessing the data to handle missing values, outliers, inconsistencies, and other anomalies that may affect the analysis. By addressing these issues, time data cleaning ensures the accuracy, reliability, and integrity of the data, laying a solid foundation for further analysis.

Time series analysis, on the other hand, involves exploring, modeling, and forecasting sequential data points collected over time. It aims to understand the underlying patterns, trends, and relationships present in the data, as well as make predictions about future values. Through methods such as descriptive analysis, time series decomposition, forecasting, and modeling, analysts can gain valuable insights into temporal data patterns, anticipate future behavior, and make informed decisions based on data-driven forecasts and predictions.

Overall, time data cleaning and time series analysis are integral parts of the data analysis pipeline, enabling analysts and decision-makers to extract actionable insights, identify trends and anomalies, and make informed decisions based on the analysis of time-stamped data. By applying sound data cleaning practices and leveraging appropriate analytical techniques, organizations can unlock the full potential of their time series data and derive actionable insights to drive business success.

## References

(IBO), N. I. B. O. (2021, July 12). *NYC Independent Budget Office (IBO) Capital Expenditures since 1985: NYC open data.* NYC Independent Budget Office (IBO) Capital Expenditures Since 1985 | NYC Open Data. https://data.cityofnewyork.us/City-Government/NYC-Independent-Budget-Office-IBO-Capital-Expendit/hukm-snmq/about_data