**3 Ways to Visualize Time Series You May Not Know**

**A ready-to-run tutorial on how to visualize a time series in Python and Altair**

Photo by [Anton Maksimov 5642.su](https://unsplash.com/@juvnsky?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

A time series is a sequence of data points, typically measured at discrete time intervals. We can represent time series data using various methods, including line graphs, bar graphs, and scatter plots. In Python, we can use libraries like matplotlib, seaborn, and Altair to create time series plots.

There are several ways to display time series data, such as line charts and bar charts. Line charts are the most common type of time series chart and are used to show the trend over time. Bar charts are used to show comparisons between different time periods. In this article, we will describe three alternative approaches to visualizing time series:

* Calendar heatmap
* Box plot
* Cycle plot

To demonstrate these three types of graphs, we’ll implement a practical example using Python and Altair, but we can generalize the approach to other libraries.

**Dataset Loading**

As an example, we use the [Global Temperature Time Series](https://datahub.io/core/global-temp), released under the [Open Data Commons](https://opendatacommons.org/) license.

First, we load the dataset as a Pandas DataFrame

import pandas as pd  
  
df = pd.read\_csv( '../Datasets/monthly\_temperature.csv' , parse\_dates=[ 'Date' ])   
df.head()

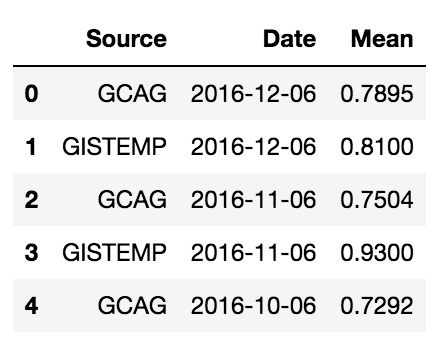


Image by Author

Next, we focus on the GCAG source, so we apply some filters:

df = df[df[ 'Source' ] == 'GCAG' ]   
df.drop(labels=[ 'Source' ], axis= 1 , inplace= True )

We draw a simple line to show the classic trendline.

import altair come alt   
  
alt.Chart(df).mark\_line().encode(   
 x= 'Data' ,   
 y= 'Media'  
 )

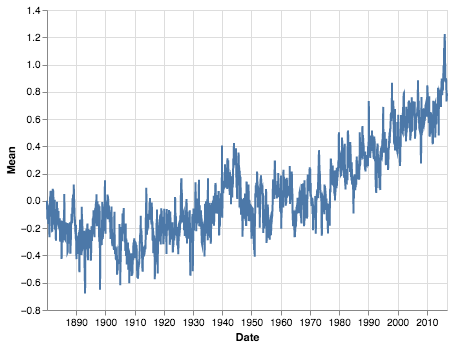


Image by Author

**Calendar heatmap**

A calendar heatmap is a great way to visualize a time series. It shows the distribution of values ​​over time and makes it easier to see patterns and outliers. Let’s build a calendar heatmap showing years on the x-axis and months on the y-axis.

First, we extract months and years from the dates.

df[ 'Mese' ] = pd.to\_datetime(df[ 'Data' ]).dt.month   
df[ 'Anno' ] = pd.to\_datetime(df[ 'Data' ]).dt.year

Then, we build the graph.

alt.Chart(df).mark\_rect().encode(   
 x= 'Anno:O' ,   
 y= 'Mese:O' ,   
 color=alt.Color( 'Mean:Q' , scale=alt.Scale( range =['blue','green', 'yellow','orange','red']))   
).properties(width= 800 )

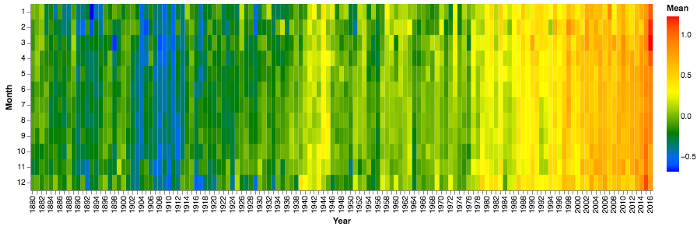


Image by Author

A calendar heatmap is very useful for showing recurring patterns over months and years. In our scenario, we only see an increase in temperature over the years.

**Boxplot**

A boxplot is a type of graph that shows the distribution of a data set. It’s a good way to visualize a time series, as it shows the distribution of values ​​over time. When creating a boxplot, make sure the axes are scaled correctly. Otherwise, the plot will be misleading. For example, if the y-axis is not scaled correctly, it will appear that there are more outliers than there actually are.

Once we’ve created our boxplot, we can use it to identify patterns in the data. For example, we notice that there is a lot of variation in the data, or that there are certain times of day when the values ​​tend to be higher or lower than others.

Write the following code in Altair to build a boxplot:

alt.Chart(df).mark\_boxplot().encode(  
 x='Month',  
 y='Mean'  
)

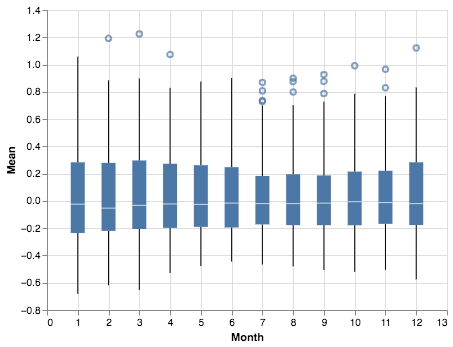


Image by Author

**Cycle plot**

A cycle plot is a graphical tool used to visualize the cyclical nature of time series data. This type of plot is particularly useful for identifying seasonality in data. The x-axis of a cycle plot is divided into equal intervals, each representing a complete cycle of the data. The y-axis shows the magnitude of the data at each point in the cycle.

Cycle plots can be used to identify periodic trends in data, such as monthly or yearly cycles. They can also be used to identify one-time events, such as spikes in data that occur at irregular intervals.

To build a cycle plot in Altair, we must concatenate twelve different plots, one for each month.

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May','Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec',]  
  
charts =alt.hconcat(spacing=0)  
for i in range(1,13):  
 if i == 1:  
 axis = alt.Axis(grid=False)  
 else:  
 axis = None  
 chart = alt.Chart(df).mark\_line().encode(  
 x=alt.X('Year:O', title=months[i-1]),  
 y=alt.Y('Mean', axis=axis)  
 ).transform\_filter(  
 alt.FieldEqualPredicate(field='Month', equal=i)  
 ).properties(  
 width=70,  
 height=300  
 )  
 charts |= chart

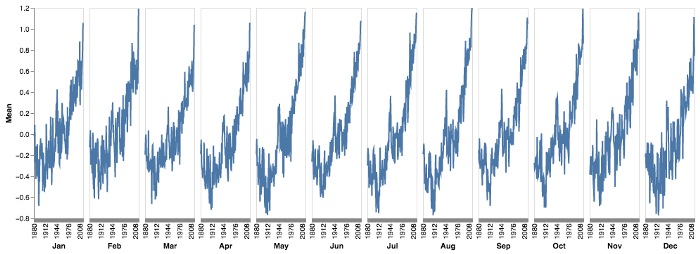


Image by Author

**Summary**

Congratulations! You have just learned three new ways to visualize time series! There are many ways to visualize time series data, and in this article, we’ve explored three that you may not be familiar with. Calendar heatmaps, box plots, and cycle plots can all offer valuable insights into your data.

So next time you’re looking at a time series, don’t forget to try out these different visualization techniques — you might just uncover something new about your data.

Download [here](https://github.com/alod83/data-science/blob/master/TimeSeriesAnalysis/Types%20of%20Graphs.ipynb) the code of the examples described in this article to play with your datasets.

**You may also be interested in….**

* [**Understanding the Seasonal Order of the SARIMA Model**](https://towardsdatascience.com/understanding-the-seasonal-order-of-the-sarima-model-ebef613e40fa)— A quick overview and a ready-to-run code to understand the (D, P, Q,M) seasonal order of the SARIMA model of the Python statsmodels library.
* [**4 different approaches for Time Series Analysis**](https://towardsdatascience.com/4-different-approaches-for-time-series-analysis-7e2364fadcb9) — A ready-to-run Python code including different strategies and libraries for Time Series Analysis

**Do you want to learn more about time series?**

Read Chapter 11 of the book [Comet for Data Science](http://www.cometfordatascience.com/) published by Packt Ltd.

