

COS30018 – Intelligent System

Option B: Stock Price Prediction

Report v0.3

Name Phong Tran

Student Id:104334842

Class: 1-5

Tutor: Tuan Dung Lai

## Table of Contents

<b>1. Write a function using candlestick chart. ....</b>	<b>2</b>
a) Explain .....	2
b) Optional with include an option to express the data of n trading days ( $n \geq 1$ ) .....	3
Images plot demonstration below: .....	3
<b>2) Write a function display boxplot stock market financial data, and try to moving window of n consecutive trading days. ....</b>	<b>5</b>
Display the boxplot data .....	6

## 1. Write a function using candlestick chart.

### a) Explain

```
def display_stock_candlestick(data, n=1, title='Candlestick chart'):
    """
    Write a function to display stock market financial data using candlestick chart
    :param data: pd.DataFrame, stock market data with columns ['Open', 'High', 'Low', 'Close', 'Volume']
    :param n: trading days
    :param title: show the chart title
    :return:
    """

    if not isinstance(data.index, pd.DataFrame):
        data.index = pd.to_datetime(data.index)

    if n >= 1:
        # Resample the data to aggregate over 'n' trading days
        data = data.resample(f'{n}D').agg({
            'open': 'first',
            'high': 'max',
            'low': 'min',
            'close': 'last',
            'volume': 'sum'
        }).dropna()

    fplt.plot(
        data,
        type='candle',
        style='charles',
        title=title,
        ylabel='Price ($)',
        volume=True,
        ylabel_lower='Shares\nTraded'
    )
```

Check if index is not in the datetime format. Then use fplt as alternative called from mplfinance library to plot the data with parameters:

- data: retrieve data
- type: candle (also called candlestick)
- style: charles (to show color between red and green)
- title: show title in details
- volume: to see volume plot below the candlestick chart

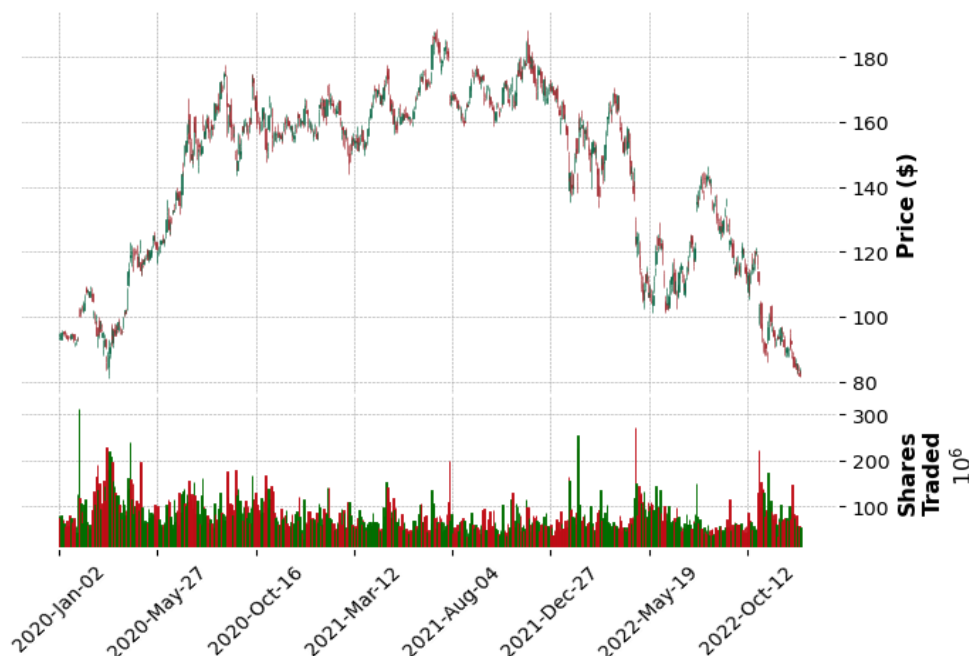
- ylabel\_lower: to change label of y-axis of the volume plot

b) Optional with include an option to express the data of n trading days (n >= 1)

```
# allow each candle stick to express the data of `n` trading days
if n >= 1:
    # Resample the data to aggregate over 'n' trading days
    data = data.resample(f'{n}D').agg({
        'open' : 'first',
        'high' : 'max',
        'low' : 'min',
        'close' : 'last',
        'volume' : 'sum'
    }).dropna()
```

If more than or equal to 1 trading days, the data will resample with corresponding days then aggregate with different function for each column. Finally, drop any rows that missing value.

**1-day trading candlestick chart**



Images plot demonstration below:

1 day

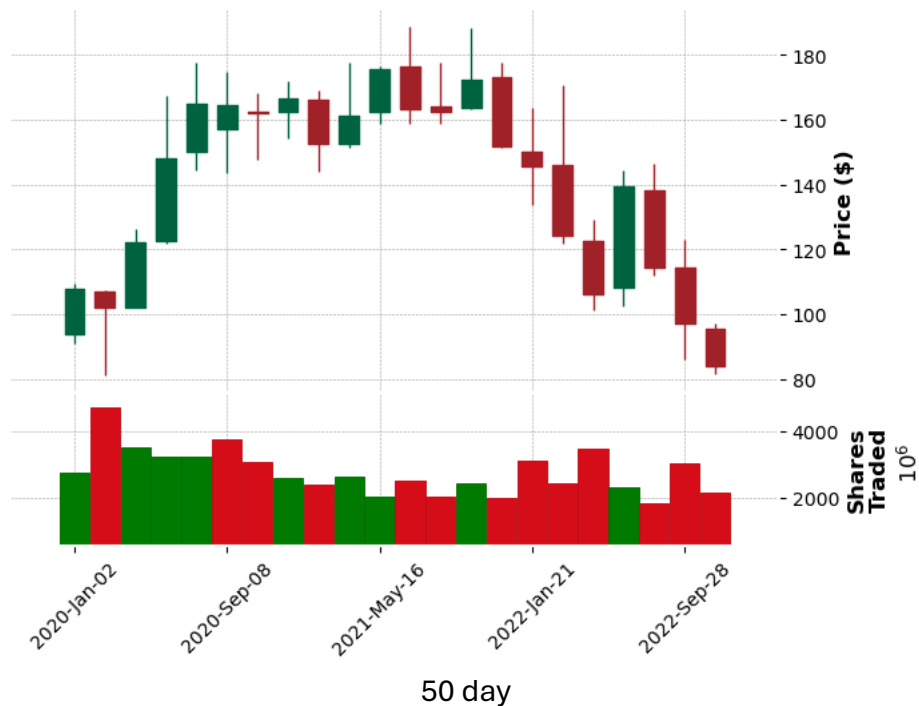
**2-day trading candlestick chart**



**10-day trading candlestick chart**



50-day trading candlestick chart



2) Write a function display boxplot stock market financial data, and try to moving window of n consecutive trading days.

```
# Check data is in dataframe format
if not isinstance(data, pd.DataFrame):
    print("Data must be DataFrame")
    return
```

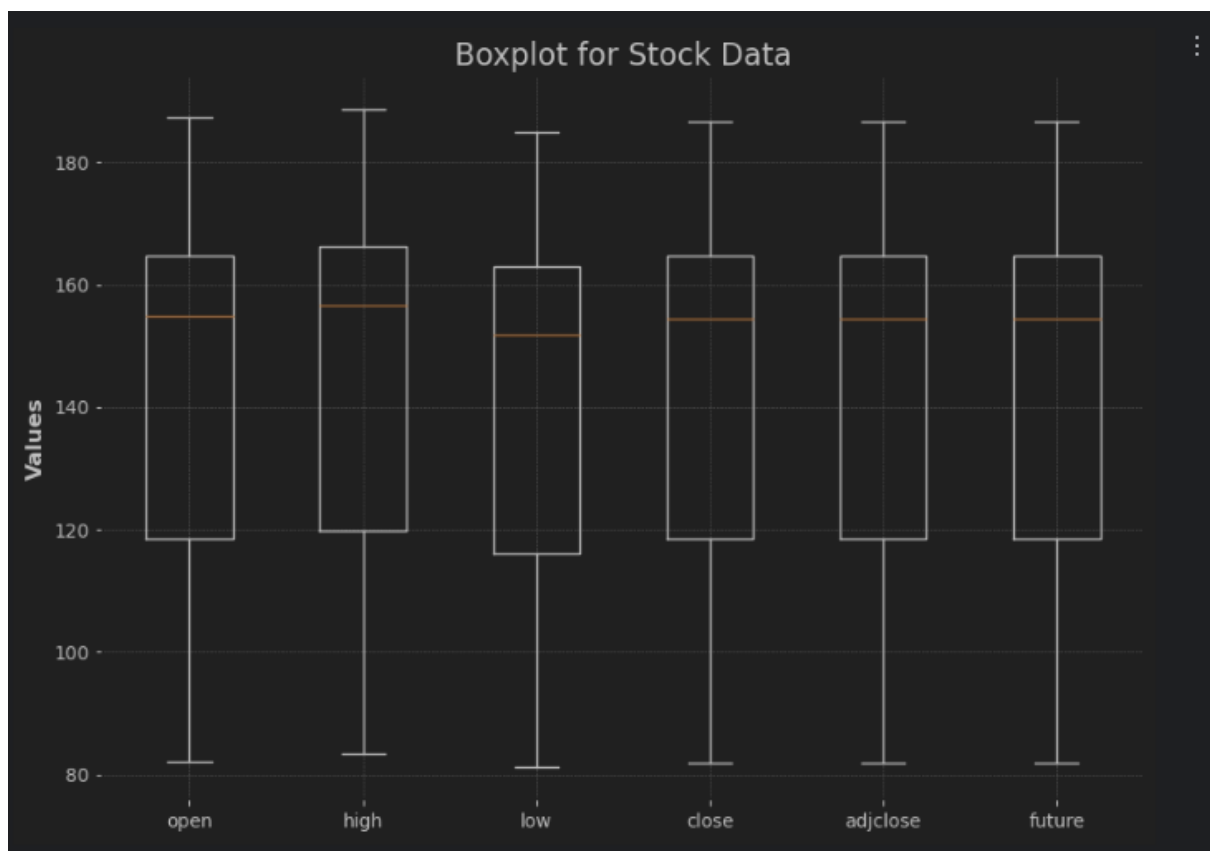
Check the condition to print and return at the function if satisfy the following conditions:

- check data if it in the dataframe type

```
data_show = data[columns]
plt.figure(figsize=figsize)
plt.boxplot(data_show.values, labels=columns)
plt.title(title, fontsize=16)
plt.ylabel('Values')
plt.show()
```

Using boxplot show each columns, and each of them display their own data.

## Display the boxplot data



In the boxplot chart, there are five things to indicate:

- Minimum: The highest data point (the point in the middle of the short line at the top)
- Maximum (the point in the middle of the short line at the bottom)
- Median (The middle value of the interquartile range)
- Q1 – Quartile 1: The bottom value in the Interquartile range
- Q2 – Quartile 2: The top value of the Interquartile range

```

def plot_moving_window_boxplot(data, columns, n, figsize=(10, 7), title='Boxplot for Stock Data'):
    """
    This function generates a boxplot for the specified stock data over a moving window of n days.

    Parameters:
    data (DataFrame): The entire dataset containing stock data.
    columns (list): The list of column names for which the boxplot will be generated (e.g., 'open', 'high', 'low').
    n (int): The window size (number of days) to use for the boxplot. Must be between 1 and the number of rows in the data.
    figsize (tuple): Optional; the size of the plot (default is (10, 7)).
    title (str): the title of the boxplot (default is 'Boxplot for Stock Data').
    """

    # Check if the window size 'n' is within valid bounds.
    if n < 1 or n > len(data):
        raise ValueError(f"The window size n must be >= 1 and <= the number of rows: {len(data)} in the data.")

    # Extract the first 'n' rows of data for the selected columns to create the moving window.
    window_data = data[columns].iloc[:n]

    # Create a new figure with the specified size for the boxplot.
    plt.figure(figsize=figsize)

    # Generate a boxplot using the values of the selected columns within the moving window.
    plt.boxplot(window_data.values, labels=columns)

    # Set the title of the boxplot, incorporating the window size 'n'.
    plt.title(f'{title} with {n} days', fontsize=16)

    # Label the y-axis as 'Values'.
    plt.ylabel('Values')

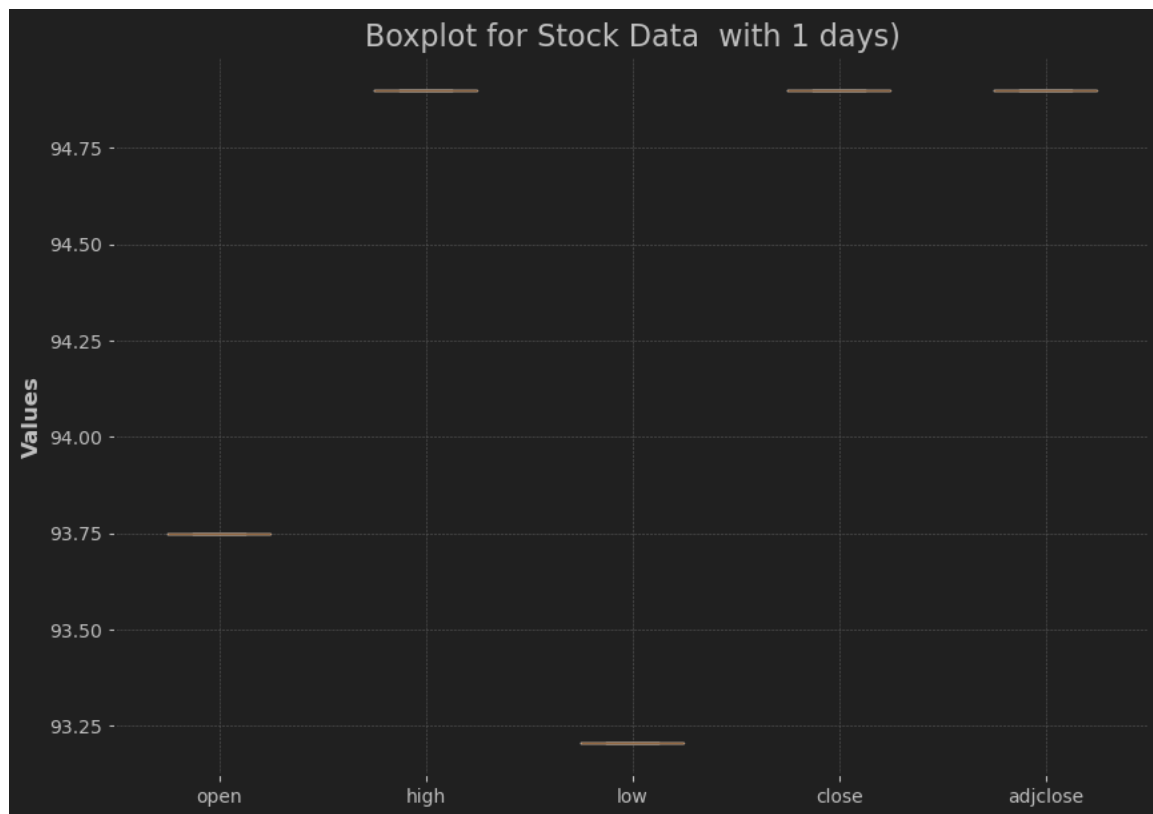
    # Display the plot.
    plt.show()

# Example usage:
columns = ['open', 'high', 'low', 'close', 'adjclose']

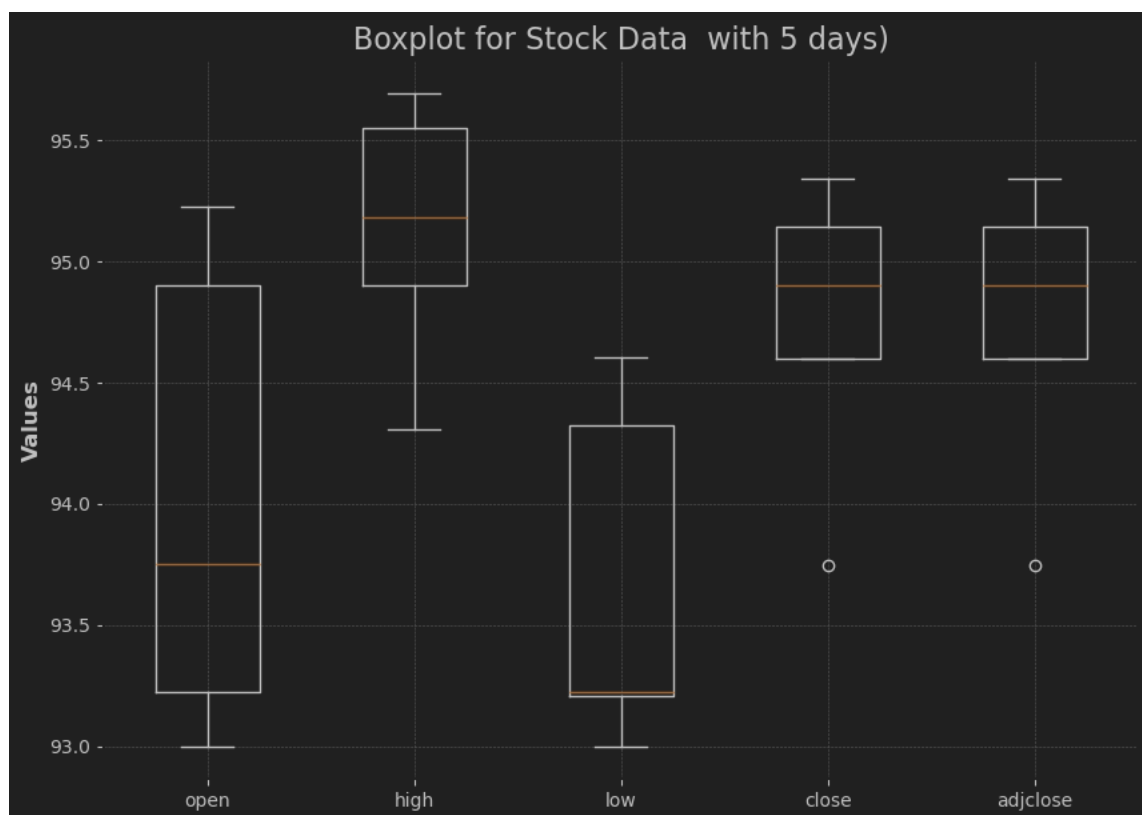
```

Executed at 2024.09.30 11:42:54 in 3ms

- **Functionality:** check the window size that in the range of the data length, before locate first number of window size length that we have specified at the data with array of columns into the variable. After that, display the boxplot with their variable's values, alongside columns variable that we specified at the parameter.

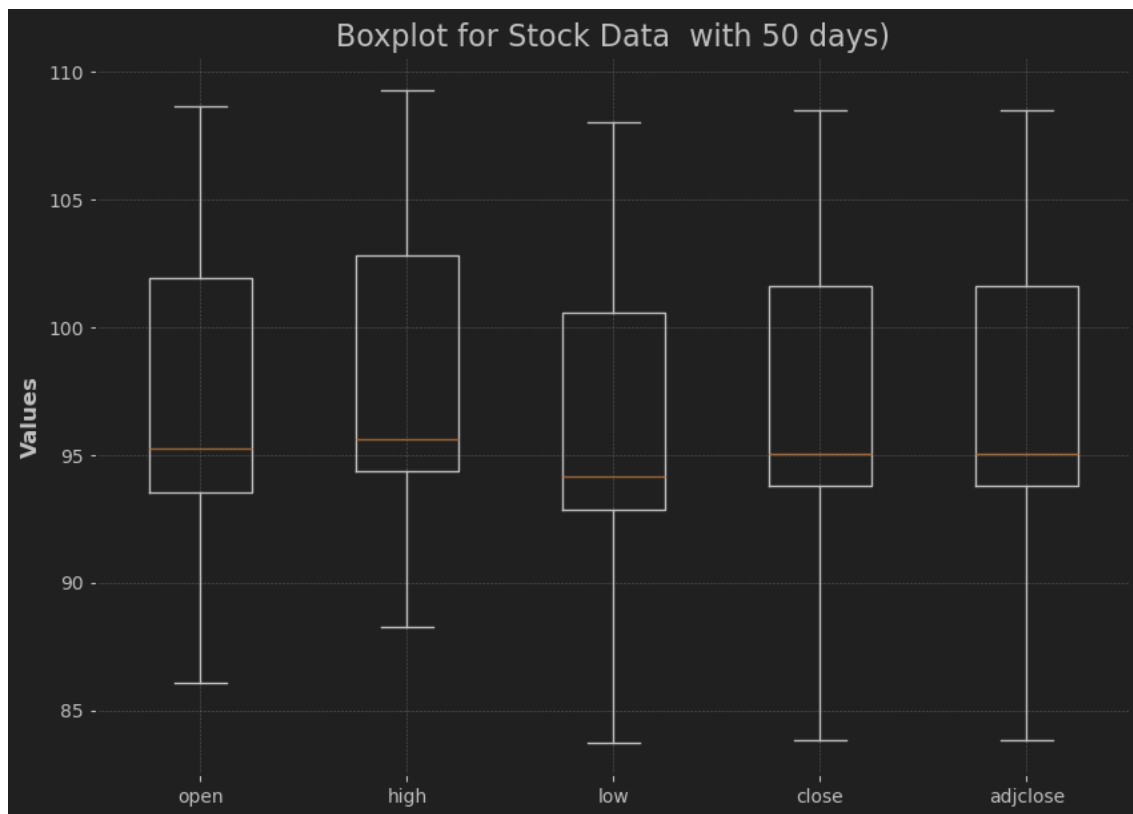


First day moving window in the plot

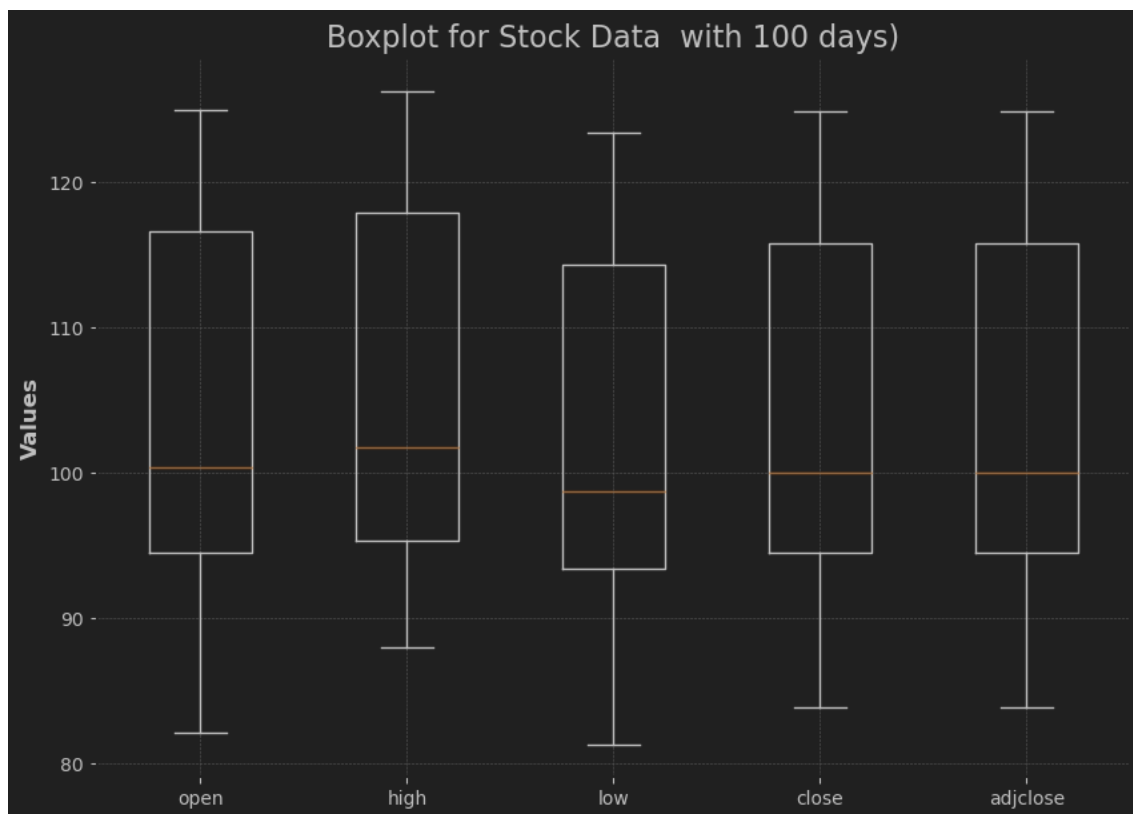


This display boxplot in the first 5 days moving window, with outliers circle in the column close and adjclose. That means there are observation points is distant from other observation.





This display boxplot in the first 50 days moving window, which shows not much difference among the previous boxplot for 1 and 5 days.



This display boxplot in the first 100 days moving window, which shows that the Interquartile range have longer than other boxplot with lower days.