Allianze SE, Share Price Dashboard with Bokeh

In [1]:
```
# pip install yfinance
# pip install bokeh
```

In [2]:
```
import pandas as pd
import yfinance as yf
import numpy as np
```

In [3]:
```
from bokeh.io import curdoc, show, output_notebook
from bokeh.models import ColumnDataSource, Select, DataTable, TableColumn
from bokeh.layouts import column, row
from bokeh.plotting import figure,show
from bokeh.io import output_notebook
from bokeh.resources import INLINE
output_notebook(INLINE)
from bokeh.io import curdoc

curdoc().clear()
from bokeh.application import Application
from bokeh.application.handlers import FunctionHandler
from bokeh.models import HoverTool
from bokeh.transform import factor_cmap
from bokeh.palettes import Blues8
```


Loading data from Yahoo Finance

In [4]:
```
df0 = yf.download("ALV.DE")
```

```
[**********************100%**********************]  1 of 1 completed
```

In [5]:
```
df0 = df0[df0.index > '2016-12-31']
```

In [6]:
```
df0
```

Out[6]:

Date	Open	High	Low	Close	Adj Close	Volume
2017-01-02	156.300003	157.500000	155.500000	157.300003	124.379189	855478
2017-01-03	158.000000	160.100006	157.550003	159.550003	126.158310	1623857
2017-01-04	160.000000	160.949997	159.550003	160.399994	126.830391	1221389
2017-01-05	161.000000	162.000000	159.600006	160.699997	127.067635	1469189
2017-01-06	160.149994	161.500000	159.550003	161.100006	127.383911	965147
...	...	...	...	...	...	...
2021-10-12	197.600006	198.720001	196.660004	197.800003	197.800003	780292
2021-10-13	197.559998	197.759995	194.279999	195.080002	195.080002	953889
2021-10-14	195.520004	197.440002	194.039993	196.399994	196.399994	720604
2021-10-15	198.600006	199.880005	197.619995	198.600006	198.600006	1071015
2021-10-19	198.419998	198.899994	197.020004	198.860001	198.860001	209015

1215 rows × 6 columns

Getting DataFrame

In [7]:
```
def getting_data(tick):
    if tick == 'All':
        df = df0
    else:
        df = df0[df0.index.year == int(tick)]
    return df.dropna()
```

Tickers for Drop down

In [8]:
```
tickers = ['2017','2018','2019','2020', '2021', 'All']
```

In [9]:
```
ticker = Select(title = "Year", value= 'All',
                options = tickers)
```

Source Data

In [10]:
```
data = getting_data(ticker.value)
source = ColumnDataSource(data=data)
```

Create a line figure with several tools

In [11]:
```
tools = 'pan, wheel_zoom, xbox_select, reset'


ts1 = figure(width=900, height=300, tools=tools,
             x_axis_type = 'datetime',
             active_drag="xbox_select", title = 'Allianze SE: Share Price')
ts1.line("Date", "Close", source=source,
         line_color=Blues8[3], width = 1.5)
ts1.circle("Date", "Close", size=1,source=source, color=None, selection_color='firebrick')
ts1.xgrid.grid_line_color = None
# ts1.ygrid.grid_line_color = None
ts1.ygrid.grid_line_alpha = 0.5
ts1.ygrid.grid_line_dash = [6, 4]
```

In [12]:
```
hover = HoverTool()
hover.tooltips = """
<div>
<div><strong>Date: </strong>@Date{%F}</div>
<div><strong>Share Price: </strong>@Close</div>
<div><strong>Volume: </strong>@Volume</div>
</div>

"""
hover.formatters={"@Date":"datetime"}

ts1.tools.append(hover)
```

Descriptive Statistics

In [13]:
```
stats = round(data.describe().reset_index(), 2)
stats_source = ColumnDataSource(data=stats)
stat_cols = [TableColumn(field=col, title=col) for col in stats.columns]
data_table = DataTable(source=stats_source, columns= stat_cols,
                       width = 400, height =300, index_position = None)
```

Create Histogram with some tools

In [14]:
```
def hist_df_maker(data_fr):
    measure = data_fr['Close']
    hist, edges = np.histogram(measure, density=False, bins=20)
    df_hist = pd.DataFrame({'arr_ranges': hist,
                            'left': edges[:-1],
                            'right': edges[1:]})
    return df_hist

hist_df = hist_df_maker(data)
hist_source = ColumnDataSource(data=hist_df)

p = figure(plot_height = 350, plot_width = 450,
               title = 'Allianze SE: Histogram of Share Price',
#               x_axis_label = 'Number of Shares',
               y_axis_label = 'Number of Shares',
           tools = tools,
#             background_fill_color="#fafafa",
           active_drag="xbox_select", border_fill_color = None)

    # Add a quad glyph
p.quad(source = hist_source, top='arr_ranges', bottom=0, left='left', right='right',
               fill_color=Blues8[0], line_color="white", alpha=0.4)
p.xgrid.grid_line_color = None
# p.ygrid.grid_line_color = None
p.ygrid.grid_line_alpha = 0.5
p.ygrid.grid_line_dash = [6, 4]
# p.border_fill_color = None
```

In [15]:
```
hoverh = HoverTool()
hoverh.tooltips = """
<div>
<div><strong>Volume: </strong>@arr_ranges</div>
<div><strong>Between: </strong>@left - @right</div>
</div>

"""

p.tools.append(hoverh)
```

Update function after every ticker change

In [16]:
```
def update(attrname, old, new):
    t1 = ticker.value
    df = getting_data(t1)
    source.data = df
    ts1.title.text = 'Allianz SE: Share Price, ' + t1
    stats_source.data = round(df.describe().reset_index(), 2)
    hist_source.data = hist_df_maker(df)
```

In [17]:
```
ticker.on_change('value', update)
```

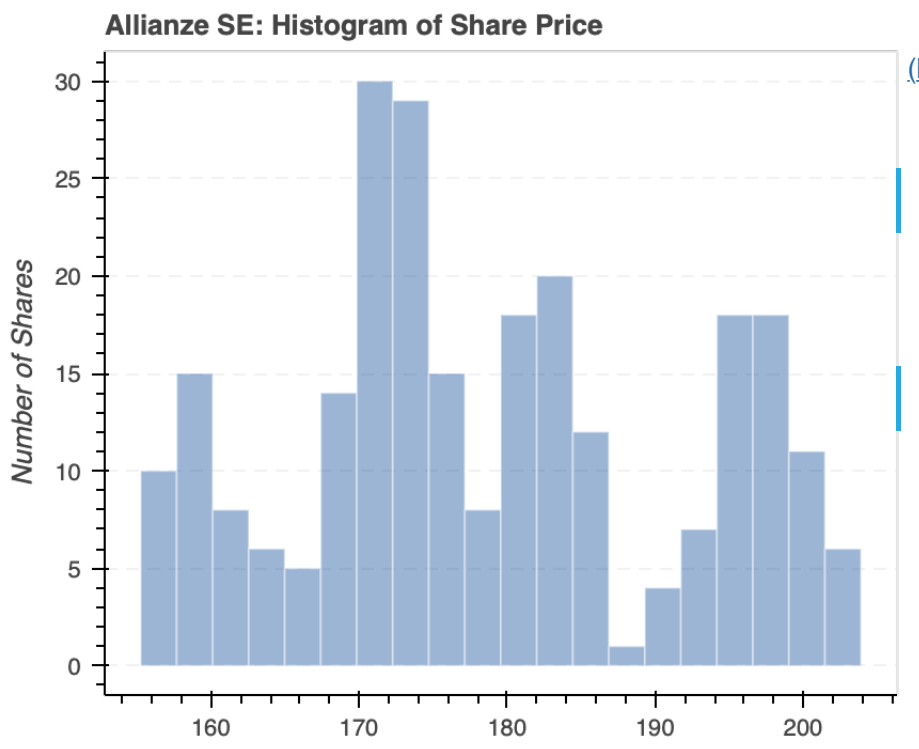
Layouts

In [18]:
```
widgets = column(ticker, data_table)
main_row = row(p,widgets)
series = column(ts1)
layout = column(main_row, series)
```

Showing the last results making real Python processes

In [19]:
```
def modify_doc(doc):
  doc.add_root(layout)

handler = FunctionHandler(modify_doc)
app = Application(handler)
curdoc().clear()
show(app)
```



(http

Year

2017

index	Open	High	Low	Close	Adj Close	Volume
count	255	255	255	255	255	255
mean	178.78	179.85	177.73	178.79	145.77	1234866.9
std	13.02	13.08	13.01	12.97	12.84	485343.49
min	155.8	157.2	154.25	155.25	122.76	0
25%	170.4	171.7	169.38	170.25	134.97	966588.5
50%	176.55	177.5	175.1	176.35	145.6	1162888
75%	187.2	188.82	186.53	188.53	155.78	1418429.5
max	204.3	204.5	202.5	203.9	168.49	3922199