In [78]:

```
import pandas as pd
import altair as alt
```

In [79]:

```
df = pd.read_csv('data montly.csv')
df['Date'] = pd.to_datetime(df['Date'])
df.head()
```

Out[79]:

	Date	y	All industries GDP	weekly earnings	House price	New orders	2 year bond yields	BC weekly earnings	BC House price
0	2020-08-01	100.91	1901415.0	15478130.0	104.6	49322424.0	0.28	2126667.0	109.7
1	2020-09-01	82.12	1918805.0	15816823.0	106.1	55488507.0	0.26	2179716.0	111.6
2	2020-10-01	95.56	1930273.0	15994443.0	107.3	54024272.0	0.24	2206001.0	112.0
3	2020-11-01	112.16	1943992.0	15944465.0	108.0	52570722.0	0.27	2217884.0	111.9
4	2020-12-01	122.92	1948654.0	15998246.0	108.3	51295764.0	0.25	2227694.0	112.2

In [80]:

```
df = df.iloc[:, :-2]
```

In [123…

```
df = df.rename(columns={'Date': 'date'})
df = df.rename(columns={'All industries GDP': 'GDP'})
df = df.rename(columns={' weekly earnings':'weekly_earning'})
df = df.rename(columns={'House price': 'house_price'})
df = df.rename(columns={'New orders': 'new_orders'})
df = df.rename(columns={'2 year bond yields': 'bond_yields'})
```

In [124…

```
df.head()
```

Out[124]:

	date	y	GDP	weekly_earning	house_price	new_orders	2 year bond yields
0	2020-08-01	100.91	1901415.0	15478130.0	104.6	49322424.0	0.28
1	2020-09-01	82.12	1918805.0	15816823.0	106.1	55488507.0	0.26
2	2020-10-01	95.56	1930273.0	15994443.0	107.3	54024272.0	0.24
3	2020-11-01	112.16	1943992.0	15944465.0	108.0	52570722.0	0.27
4	2020-12-01	122.92	1948654.0	15998246.0	108.3	51295764.0	0.25

In [125…

```
df.corr()
```

Out[125]:

	y	GDP	weekly_earning	house_price	new_orders	2 year bond yields
y	1.000000	0.915526	0.912915	0.911169	0.803119	0.828298
GDP	0.915526	1.000000	0.986552	0.965450	0.838994	0.905063
weekly_earning	0.912915	0.986552	1.000000	0.973632	0.834314	0.910528
house_price	0.911169	0.965450	0.973632	1.000000	0.881472	0.847036
new_orders	0.803119	0.838994	0.834314	0.881472	1.000000	0.740044
2 year bond yields	0.828298	0.905063	0.910528	0.847036	0.740044	1.000000

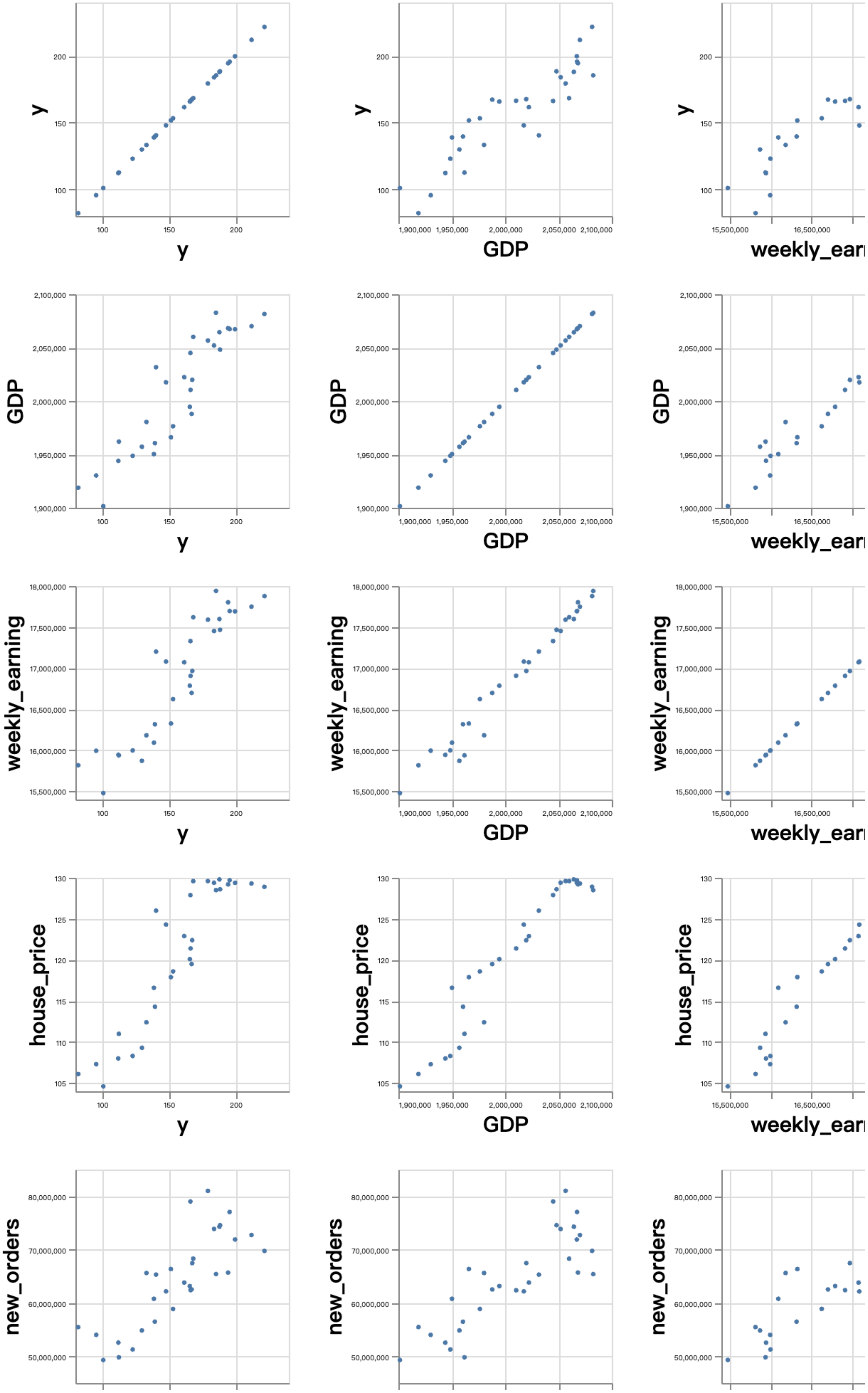
In [127…

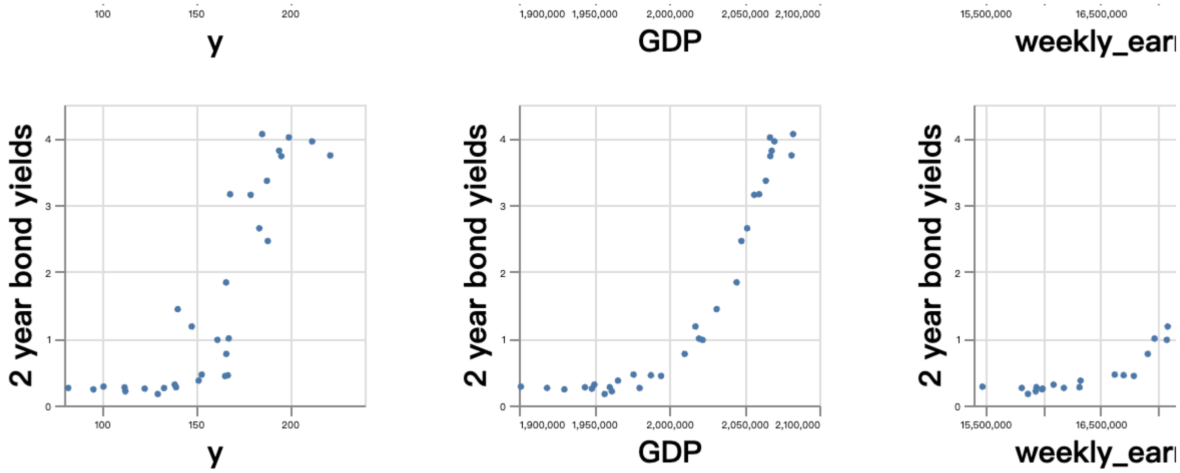
```
numeric_cols=df.select_dtypes('number')
numeric_cols=numeric_cols.columns.tolist()
```

In [128…

```
alt.Chart(df).mark_point(opacity=1,size=2).encode(
   alt.X(alt.repeat('column'),type="quantitative",scale=alt.Scale(zero=False)),
   alt.Y(alt.repeat('row'),type="quantitative",scale=alt.Scale(zero=False))
).properties(height=150,width=150).repeat(row=numeric_cols,column=numeric_cols)
```

Out[128]:

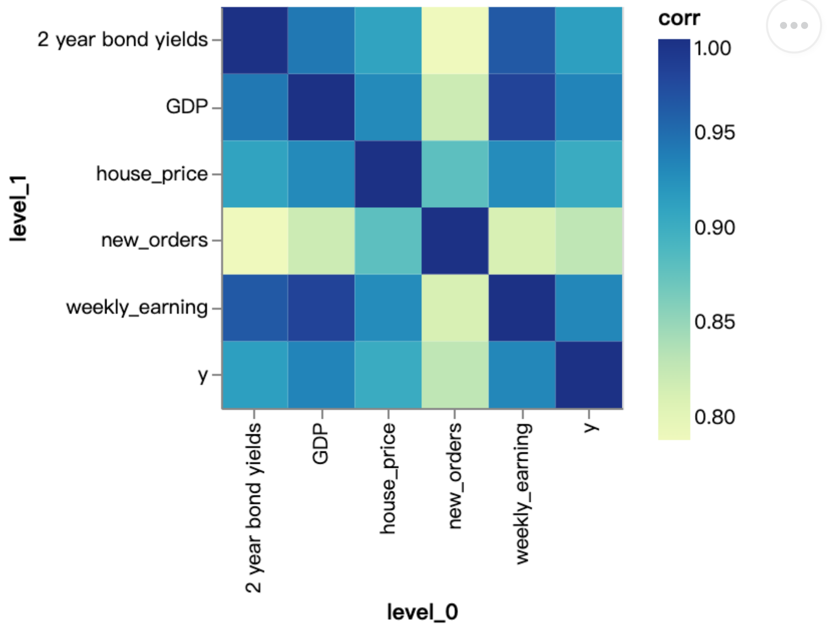

y
GDP
weekly_earning
house_price
new_orders
y
GDP
weekly_earr
100
150
200
1,900,000
1,950,000
2,000,000
2,050,000
2,100,000
15,500,000
16,500,000
15,500,000
16,000,000
16,500,000
17,000,000
17,500,000
18,000,000
105
110
115
120
125
130
50,000,000
60,000,000
70,000,000
80,000,000



In [129…

```
corr_df = df[numeric_cols].corr("spearman").stack().reset_index(name='corr')
corr_df

alt.Chart(corr_df).mark_rect().encode(
    x='level_0:N',
    y='level_1:N',
    color='corr:Q'
).properties(width=200,height=200)
```

Out[129]:



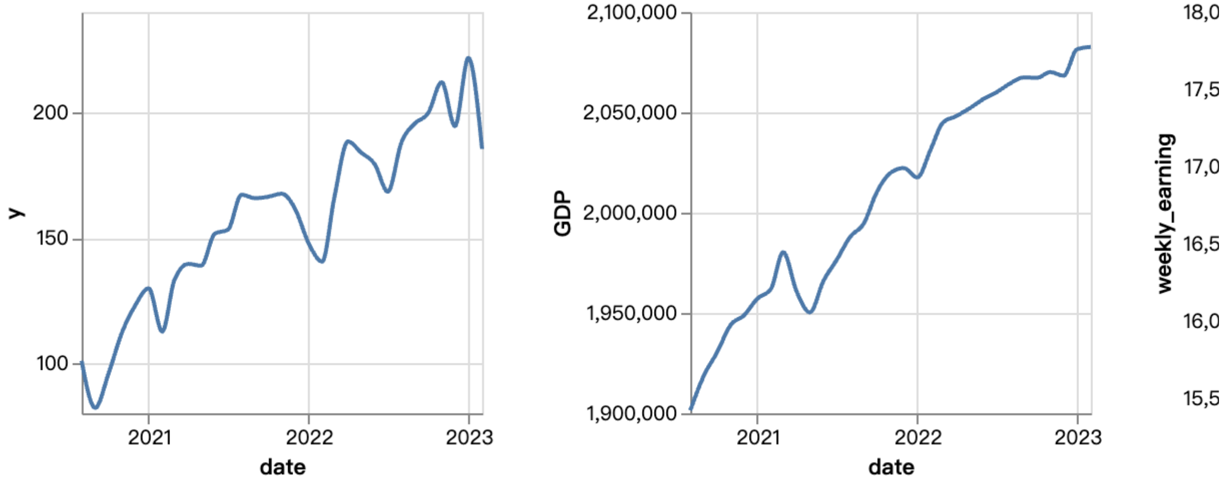
In [153…

```
def time_series_trend(df, lst, start_date=None, end_date=None):
    """
    Parameters:
        df: input dataframe
        lst: list of column name in df
        start_date (optional): if want to check trend of a range of time, speci
        end_date (optional): if want to check trend of a range of time, specify

    Returen:
        time trend plot (altair)
    """
    if start_date and end_date:
        df = df[(df['date']>=start_date) & (df['date']<=end_date)]
    chart=alt.Chart(df).mark_line(interpolate='monotone').encode(
        alt.X('date'),
        alt.Y(alt.repeat(),type='quantitative',scale=alt.Scale(zero=False))
    ).repeat(repeat=lst).configure_view(width=200, height=200)
    return chart
```

In [154…

```
time_series_trend(df,numeric_cols, start_date=None, end_date=None)
```

Out[154]:



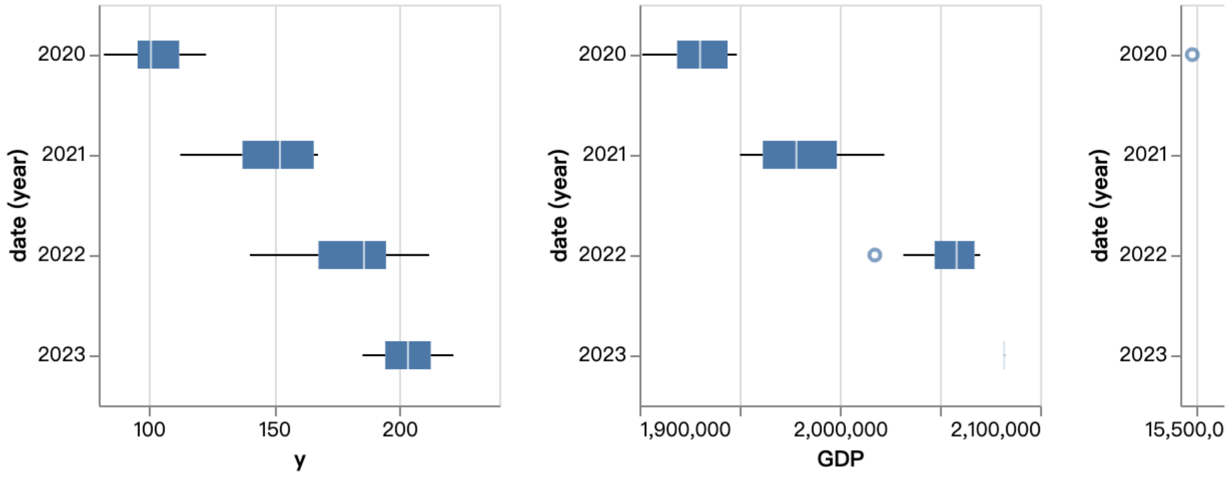
In [155…

```
def boxplot_year(df, lst, years=None):
    """
    Parameters:
        df: input dataframe
        lst: list of column name in df
        years (optional): if want to check boxplot of some years, specify the l

    Returen:
        boxplot (altair)
    """
    if years:
        df = df[df['date'].dt.year.isin(years)]
    chart=alt.Chart(df).mark_boxplot().encode(
        alt.Y("year(date):N"),
        alt.X(alt.repeat(),type="quantitative",scale=alt.Scale(zero=False)),
        alt.Tooltip('Title:N')
        ).repeat(repeat=lst).configure_view(width=200, height=200)

    return chart
```

In [156…

```
boxplot_year(df, numeric_cols, years=None)
```

Out[156]:



While three predictors(GDP, weekly earning and bond yields) continue to increase in 2023, there is a decline in both the new order and housing price. Notably, the new order experiences a more pronounced decrease compared to housing price in 2023.

In [ ]: