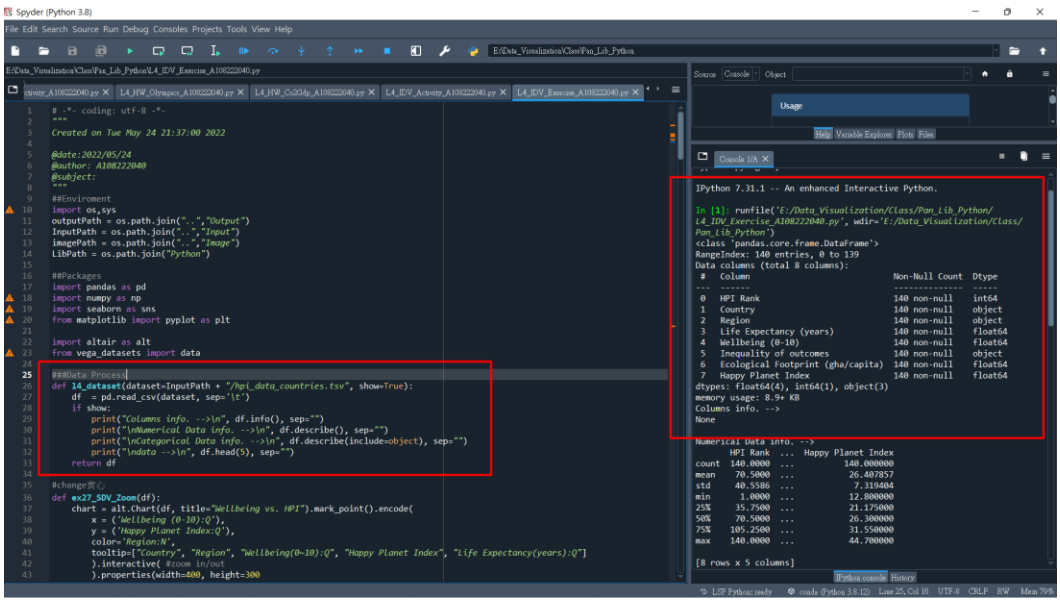
A108222040Data_Visulization_Week14&15&16

Week14

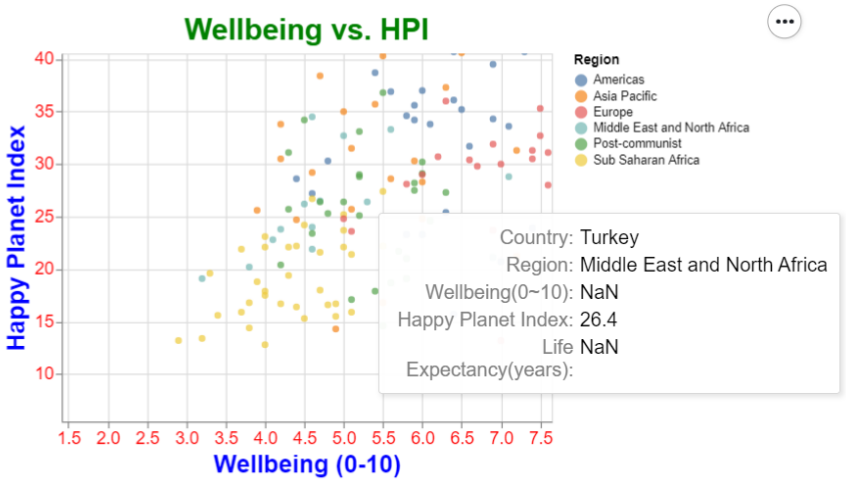
Show dataset(國家快樂指數)



Ex27 circle 是實心 point 是空心

```
    return df

#change實心
def ex27_SDV_Zoom(df):
    chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_circle().encode(
        x = ('Wellbeing (0-10):Q'),
        y = ('Happy Planet Index:Q'),
        color='Region:N',
        tooltip=["Country", "Region", "Wellbeing(0-10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
        ).interactive( #zoom in/out
        ).properties(width=400, height=300
        ).configure_title(color='green', fontSize=24
        ).configure_axis(labelFontSize=14,
                         labelColor="red",
                         titleFontSize=20,
                         titleColor="blue",
        )
    chart.show()
```

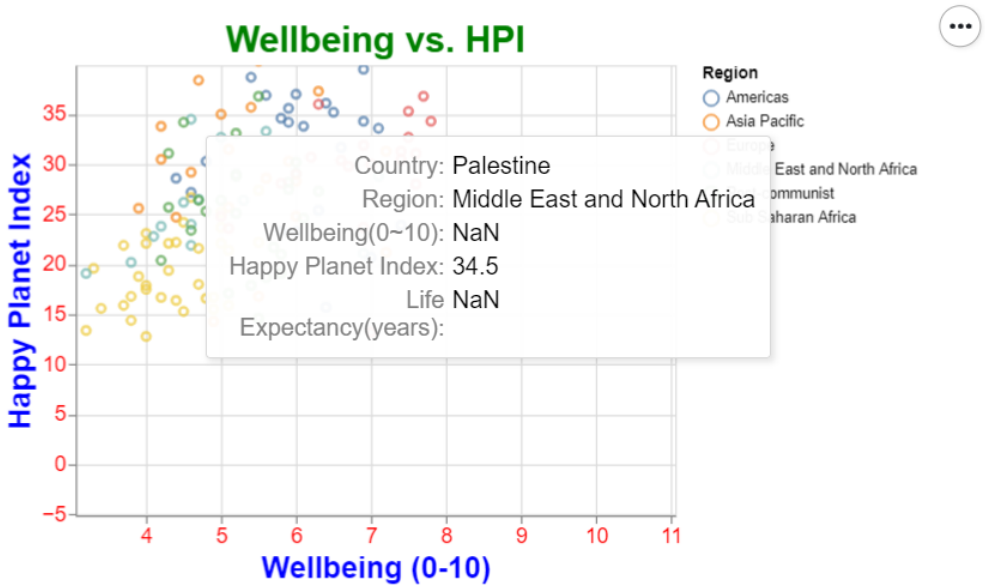
Result



Ex28

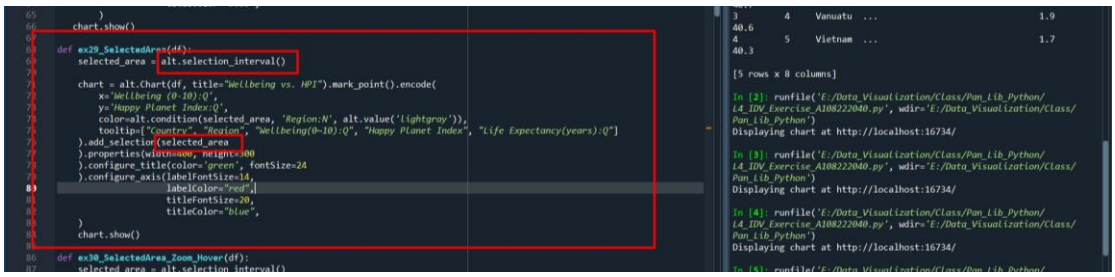
```
    chart.show()

def ex28_SDV_Zoom_Hover(df):
    chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_circle().encode(
        x = 'Wellbeing (0-10):Q',
        y = 'Happy Planet Index:Q',
        color='Region:N',
        tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
    ).interactive( #zoom in/out
    ).properties(width=400, height=300
    ).configure_title(color='green', fontSize=24
    ).configure_axis(labelFontSize=14,
                     labelColor="red",
                     titleFontSize=20,
                     titleColor="blue",
        )
    chart.show()

```

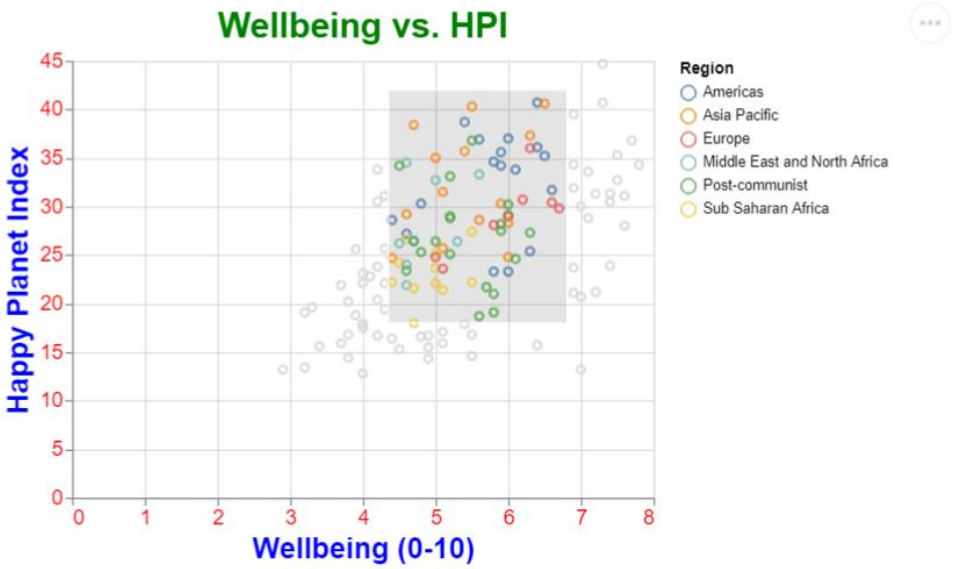
Result



Ex29

可以選擇一塊區域做觀看



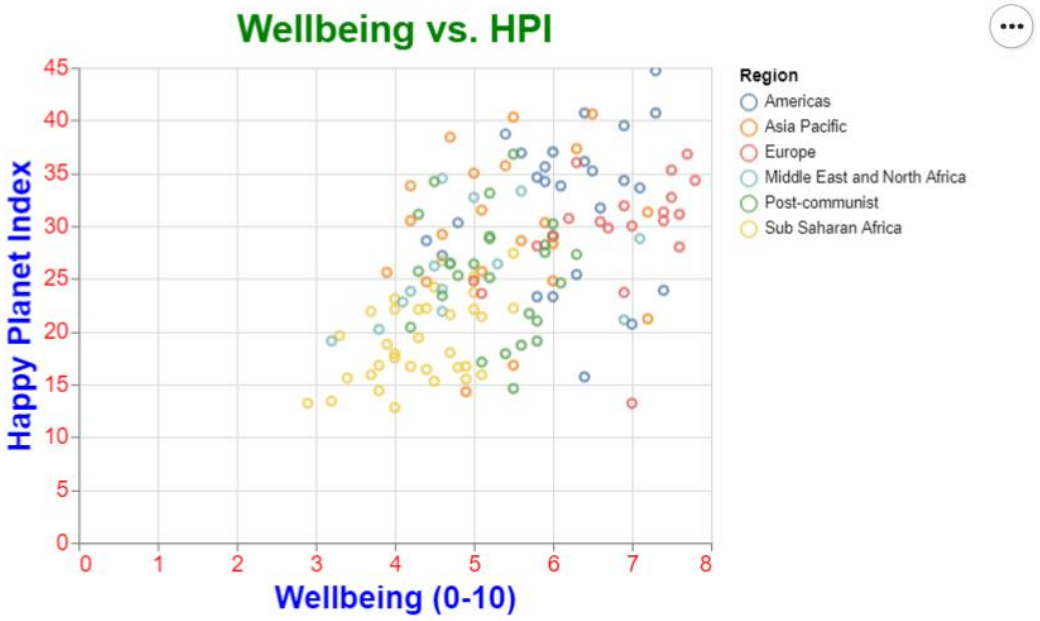


Ex30

可以選擇一塊區域做觀看 Zoom 移動

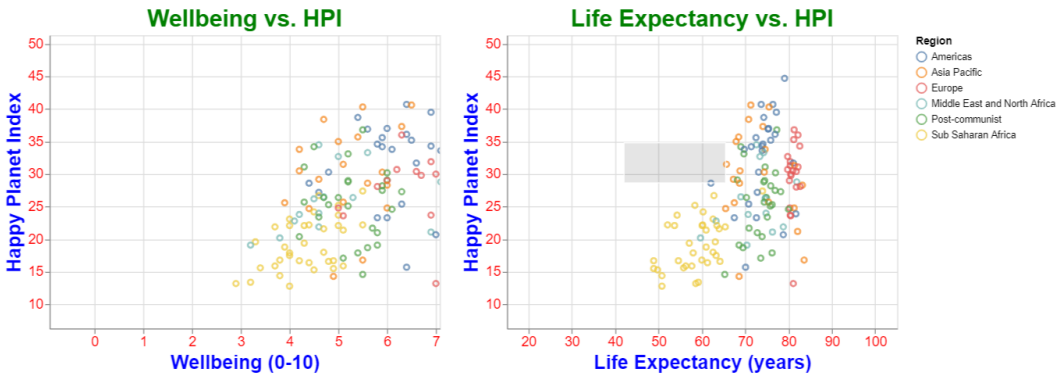
紅框為與 29 不同之處

```
    chart.show()

def ex30_SelectedArea_Zoom_Hover(df):
    selected_area = alt.selection_interval()

    chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_point().encode(
        x='Wellbeing (0-10):Q',
        y='Happy Planet Index:Q',
        color=alt.condition(selected_area, 'Region:N', alt.value('lightgray')),
        tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
    ).interactive(
    ).add_selection(selected_area
    ).properties(width=400, height=300
    ).configure_title(color='green', fontSize=24
    ).configure_axis(labelFontSize=14,
                     labelColor="red",
                     titleFontSize=20,
                     titleColor="blue",
    )
```



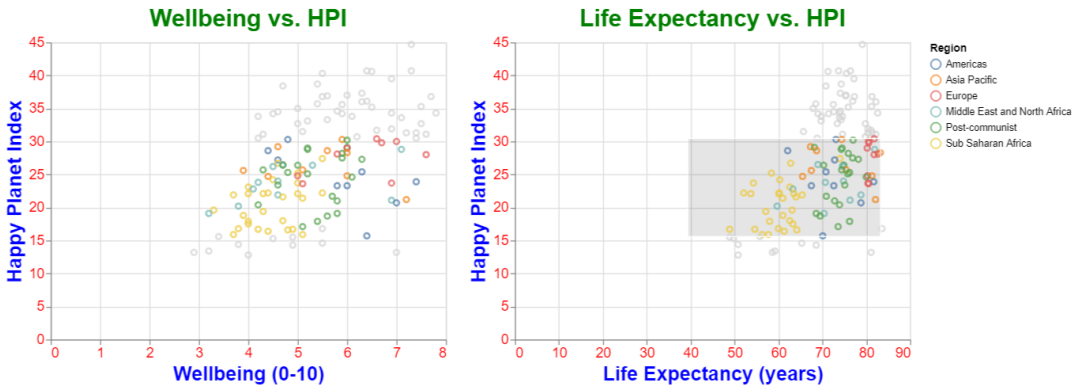
Ex31 左右兩個區域 可以同時移動

```python

def ex31_MultiplePlot(df):
    selectedArea = alt.selection_interval()

    chart = alt.Chart(df, title="HPI").mark_point().encode(
    x='Wellbeing (0-10):Q',
    y='Happy Planet Index:Q',
    color='Region:N',
    tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
    ).interactive(
    ).add_selection(selectedArea
    ).properties(width=400, height=300)
    # .configure_title(color='green', fontSize=24
    # ).configure_axis(labelFontSize=14,
    #                  labelColor="red",
    #                  titleFontSize=20,
    #                  titleColor="blue",
    # )
    chart1 = chart.encode(x='Wellbeing (0-10)').properties(title="Wellbeing vs. HPI")
    chart2 = chart.encode(x='Life Expectancy (years)').properties(title="Life Expectancy vs. HPI")
    chart = alt.hconcat(chart1, chart2
                ).configure_title(color='green', fontSize=24
                ).configure_axis(labelFontSize=14,
                                 labelColor="red",
                                 titleFontSize=20,
                                 titleColor="blue",
                )
    chart.show()
```



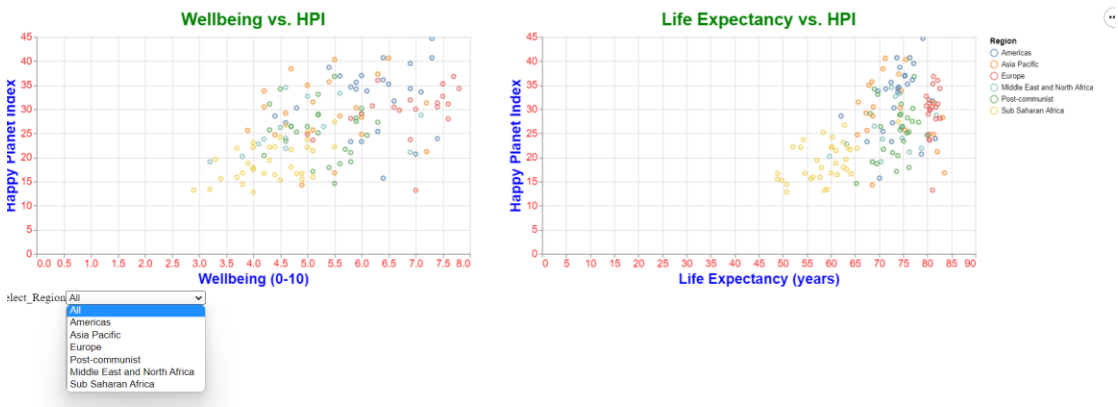
Ex31 右邊選擇區域 左邊就會亮該處

```python

def ex31_Slecetion(df):
    selectedArea = alt.selection_interval()

    chart = alt.Chart(df, title="HPI").mark_point().encode(
    #x='Wellbeing (0-10):Q',
    y='Happy Planet Index:Q',
    color=alt.condition(selectedArea, 'Region', alt.value('lightgray')),
    tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
    #),interactive()
    ).add_selection(selectedArea
    ).properties(width=400, height=300)
    # .configure_title(color='green', fontSize=24
    # ).configure_axis(labelFontSize=14,
    #                  labelColor="red",
    #                  titleFontSize=20,
    #                  titleColor="blue",
    # )
    chart1 = chart.encode(x='Wellbeing (0-10)').properties(title="Wellbeing vs. HPI")
    chart2 = chart.encode(x='Life Expectancy (years)').properties(title="Life Expectancy vs. HPI")
    charts = alt.hconcat(chart1, chart2
               ).configure_title(color='green', fontSize=24
               ).configure_axis(labelFontSize=14,
                                labelColor="red",
                                titleFontSize=20,
                                titleColor="blue",
               )
    charts.show()
```



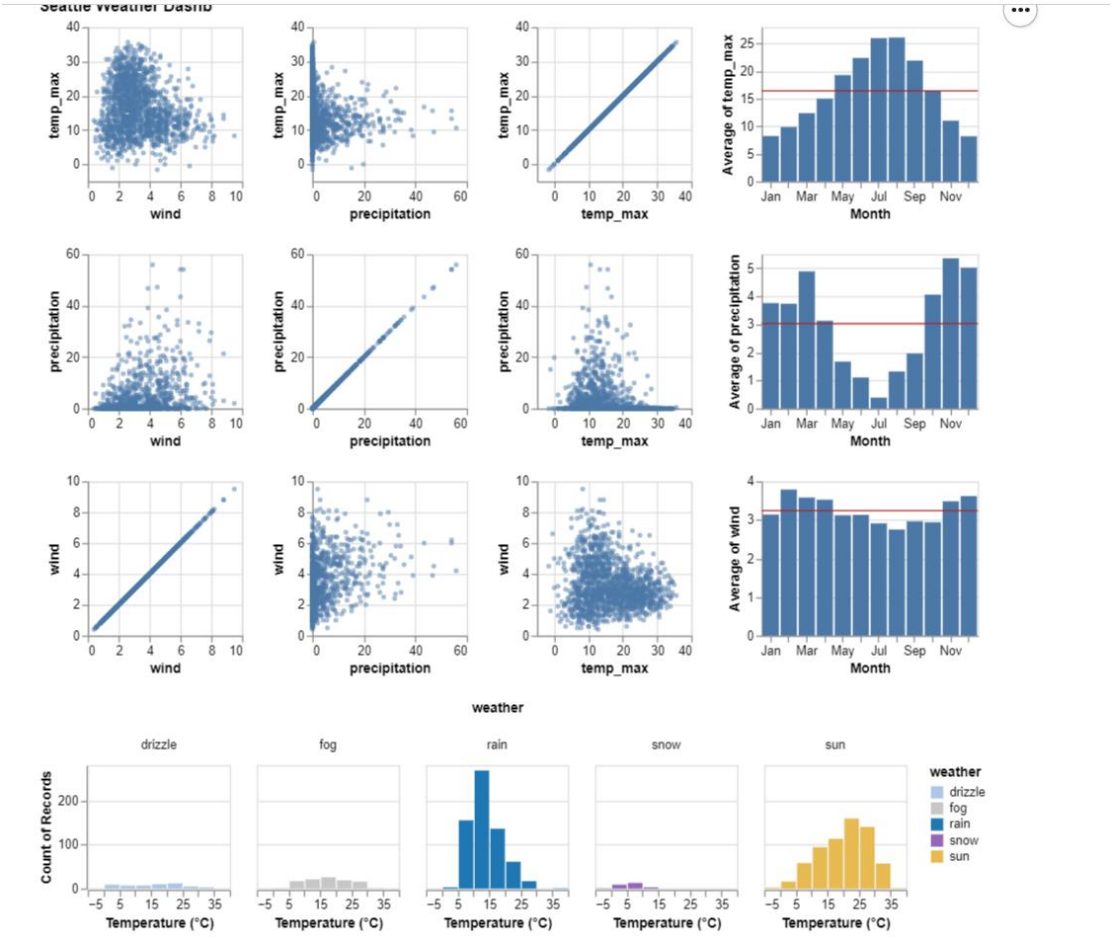
Ex32 有下拉式選單

```
160
161 def ex32_Selection_FeatureValue_MP(df):
162     #input_dropdown = alt.binding_select(options=list(r))
163     input_dropdown = alt.binding_select(
164     options=[None] + list(df.Region.unique()), labels = ['All'] + list(df.Region.unique()))
165     selected_points = alt.selection_single(fields=['Region'],
166                                            bind=input_dropdown, name='Select')
167
168     #chart
169     chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_point().encode(
170     x='Wellbeing (0-10):Q',
171     y='Happy Planet Index:Q',
172     color=alt.condition(selected_points, alt.Color("Region:N"), alt.value("lightgray")),
173     tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
174     ).interactive(
175     ).add_selection(selected_points
176     ).properties(width=600, height=300)
177
178     chart1 = chart.encode(x='Wellbeing (0-10)').properties(title="Wellbeing vs. HPI")
179     chart2 = chart.encode(x='Life Expectancy (years)').properties(title="Life Expectancy vs. HPI")
180     charts = alt.hconcat(chart1, chart2, spacing=50
181               ).configure_title(color='green', fontSize=24
182               ).configure_axis(labelFontSize=14,
183                                labelColor="red",
184                                titleFontSize=20,
185                                titleColor="blue",
186               )
187     charts.show()
```



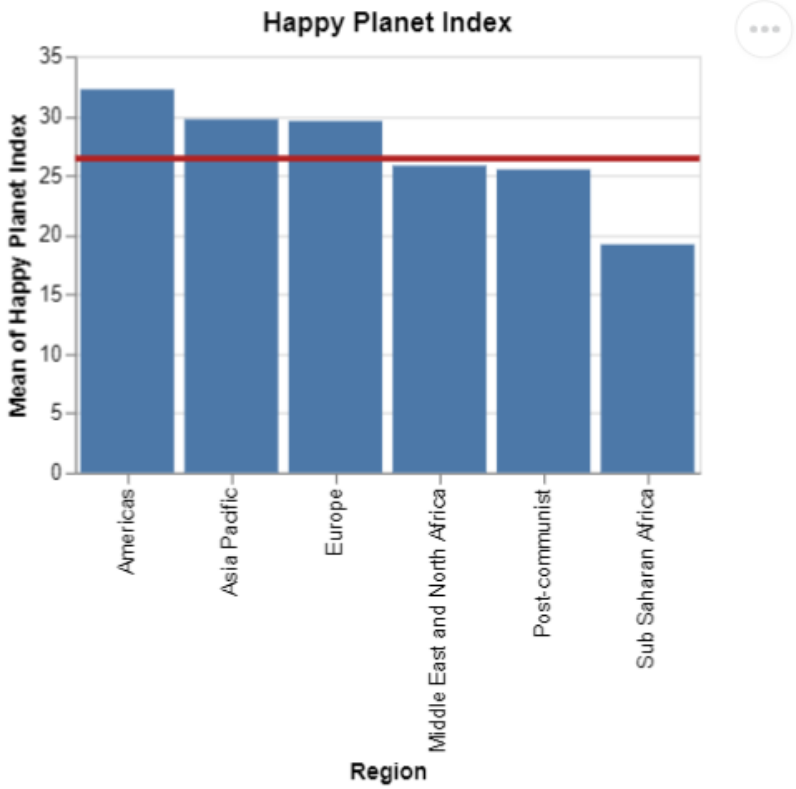
範例 weather

```
    charts.show()

def testAltair_MP_weather():
    #---Load Data
    weather = 'https://vega.github.io/vega-datasets/data/weather.csv'
    df = pd.read_csv(weather)

    #--plot
    splom = alt.Chart().mark_point(filled=True, size=15, opacity=0.5
            ).encode(alt.X(alt.repeat('column'), type='quantitative'),
                     alt.Y(alt.repeat('row'), type='quantitative')
            ).properties(width=125, height=125
            ).repeat(row=['temp_max', 'precipitation', 'wind'],
                     column=['wind', 'precipitation', 'temp_max'])

    dateHist = alt.layer(alt.Chart().mark_bar().encode(
                         alt.X('month(date):O', title='Month'),
                         alt.Y(alt.repeat('row'), aggregate='average', type='quantitative')),
                         alt.Chart().mark_rule(stroke='firebrick').encode(
                         alt.Y(alt.repeat('row'), aggregate='average', type='quantitative'))
            ).properties(width=175, height=125
            ).repeat(row=['temp_max', 'precipitation', 'wind'])

    tempHist = alt.Chart(weather).mark_bar().encode(
                         alt.X('temp_max:Q', bin=True, title='Temperature (°C)'),
                         alt.Y('count():Q'),
                         alt.Color('weather:N', scale=alt.Scale(
                         domain=['drizzle', 'fog', 'rain', 'snow', 'sun'],
                         range=['#aec7e8', '#c7c7c7', '#1f77b4', '#9467bd', '#e7ba52'
            ).properties(width=115, height=100
            ).facet(column='weather:N')

    chart = alt.vconcat(alt.hconcat(splom, dateHist), tempHist, data=weather, title='Seattle Weather Dashb'
                    ).transform_filter('datum.location == "Seattle"'
                    ).resolve_legend(color='independent'
                    ).configure_axis(labelAngle=0)

    chart.show()
```



Ex33 bar 一樣是用快樂指數

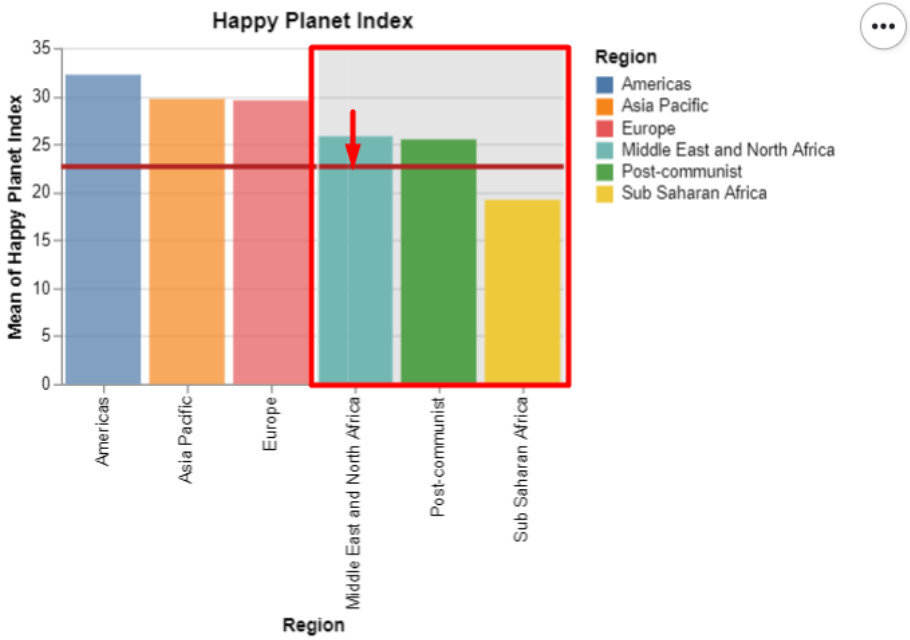
```
225
226 def ex33_Bar(df):
227     columns = df.columns
228     xData = columns[2] + ":N"
229     yData = "mean(" + columns[7] + "):Q"
230     bars = alt.Chart(df, title="Happy Planet Index").mark_bar().encode(
231         x=xData, # x=alt.X('x', axis=alt.Axis(format='%', title='Wellbeing'))
232         y=yData, # y=alt.Y('y', axis=alt.Axis(labels=False))
233         #color='Region:N',
234         #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
235         #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
236     ).properties(width=300, height=200)
237     #).interactive()
238     #).add_selection(selected_points
239
240     line = alt.Chart(df).mark_rule(color='firebrick').encode(
241         y=yData,
242         size=alt.SizeValue(3))
243
244     #-- Concate 1: horizontal two figures
245     #charts = alt.hconcat(chart1, chart2, spacing=50
246     #           ).configure_title(color='green', fontSize=24
247     #           ).configure_axis(labelFontSize=14,
248     #                            labelColor="red",
249     #                            titleFontSize=20,
250     #                            titleColor="blue",
251     #           )
252
253     #--Concate 2 : overlap
254     chart = alt.layer(bars, line, data=df).interactive()
255     chart.show()
```



Ex34 用 bar 呈現 從上面選擇區域呈現 bar

```python
def ex34_Bar_Selection(df):
    selectedBars = alt.selection(type="interval", encodings=["x"])

    columns = df.columns
    xData = columns[2] + ":N"
    yData = "mean(" + columns[7] + "):Q"
    bars = alt.Chart(df, title="Happy Planet Index").mark_bar().encode(
        x=xData, # x=alt.X('x', axis=alt.Axis(format='%', title='Wellbeing'))
        y=yData, # y=alt.Y('y', axis=alt.Axis(labels=False))
        opacity=alt.condition(selectedBars, alt.OpacityValue(1), alt.OpacityValue(0.7)),
        color='Region:N',
        #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
        #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
    ).properties(width=300, height=200
    ).add_selection(selectedBars
    ).interactive()


    line = alt.Chart(df).mark_rule(color='firebrick').encode(
        y=yData,
        size=alt.SizeValue(3)
    ).transform_filter(selectedBars)

    chart = alt.layer(bars, line, data=df)
    chart.show()
```

選擇區域數值較低 線也較往下



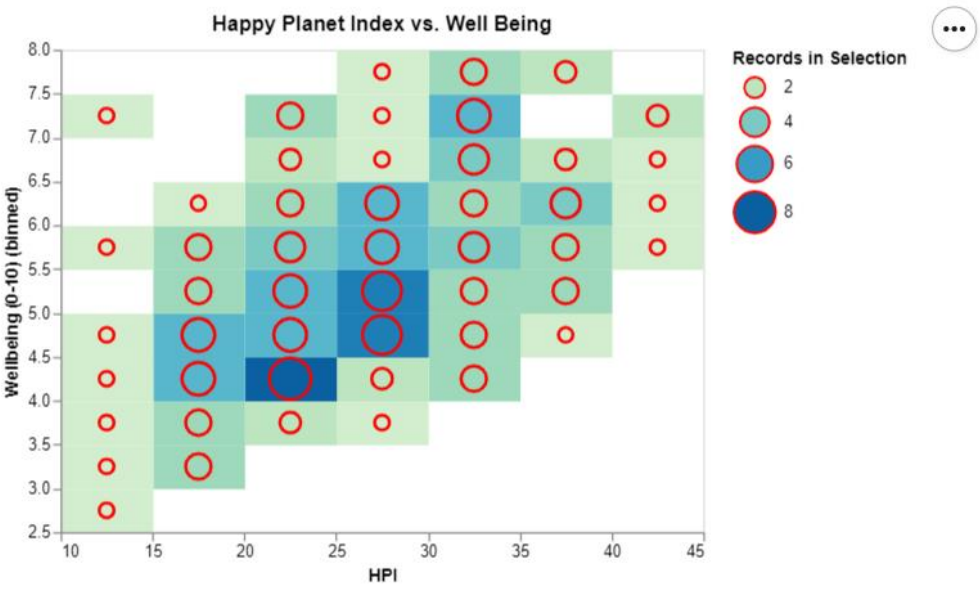
Ex35 heatmap

```
282
283 def ex35_Heatmap(df):
284     columns = df.columns
285     xData = columns[7] + ":Q"
286     yData = columns[4] + ":Q"
287     chart = alt.Chart(df, title="Happy Planet Index vs. Well Being").mark_rect().encode(
288         x=alt.X(xData, title="HPI", bin=True),
289         y=alt.Y(yData, bin=True),
290         color=alt.Color('count()', scale=alt.Scale(scheme="greenblue"),
291                                    legend=alt.Legend(title="Total Countries")),
292         #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
293         #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
294     ).properties(width=400, height=300
295     ).interactive()
296
297     chart.show()
298
```

可以看到熱度相關

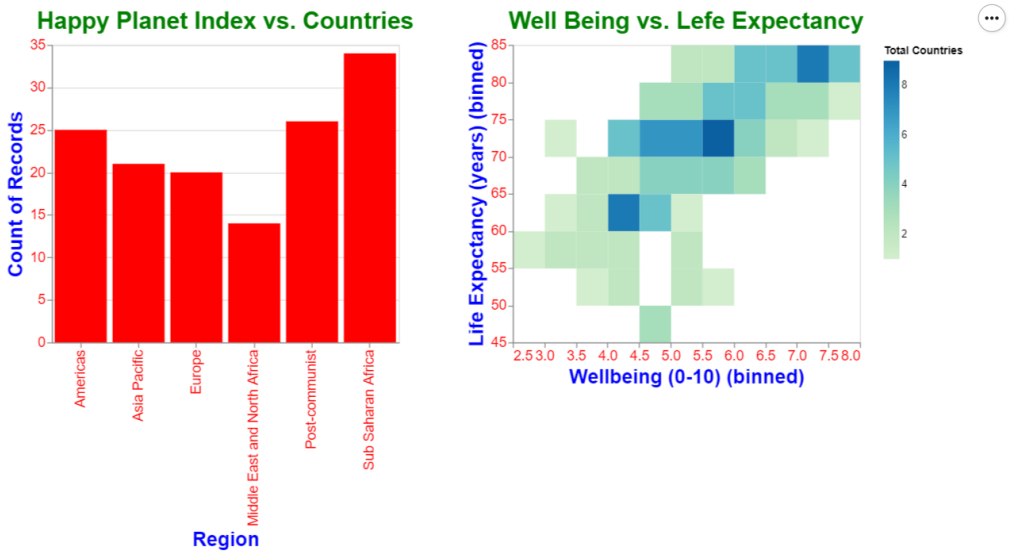
Ex35 熱度圖上面有圈圈代表

```python

def ex35_Heatmap_Circle(df):
    #-- chart 1
    columns = df.columns
    xData = columns[7] + ":Q"
    yData = columns[4] + ":Q"
    heatmap = alt.Chart(df, title="Happy Planet Index vs. Well Being").mark_rect().encode(
        x=alt.X(xData, title="HPI", bin=True),
        y=alt.Y(yData, bin=True),
        color=alt.Color('count()', scale=alt.Scale(scheme="greenblue")),
        #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
        #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
    ).properties(width=400, height=300)

    #-- chart2
    circles = heatmap.mark_point().encode(
        alt.ColorValue("red"),
        alt.Size("count()", legend=alt.Legend(title="Records in Selection"))
        ).interactive()

    #--overlap
    chart = alt.layer(heatmap, circles)
    chart.show()

```



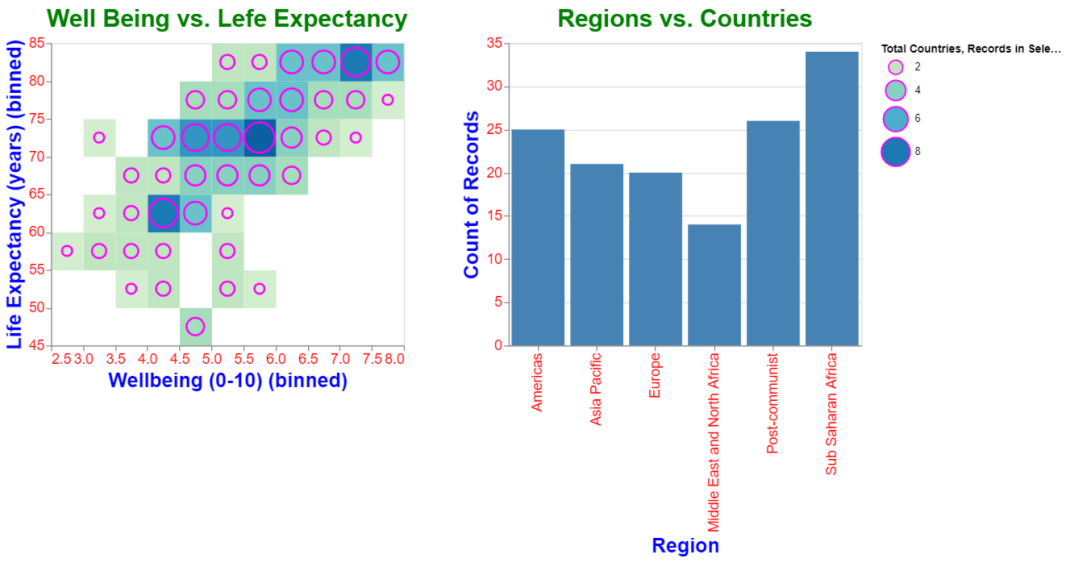
Ex36 bar & heatmap

```
321
322 def ex36_Bar_Heatmap(df):
323     #-- chart 1
324     bars = alt.Chart(df, title="Happy Planet Index vs. Countries").mark_bar().encode(
325         x="Region:N",
326         y="count():Q",
327         color=alt.ColorValue("red")
328     ).properties(width=350, height=300).interactive()
329
330     #-- chart2
331     columns = df.columns
332     xData = columns[4] + ":Q"
333     yData = columns[3] + ":Q"
334     heatmap = alt.Chart(df, title="Well Being vs. Lefe Expectancy").mark_rect().encode(
335         x=alt.X(xData, bin=True),
336         y=alt.Y(yData, bin=True),
337         color=alt.Color('count()', scale=alt.Scale(scheme="greenblue"),
338                         legend=alt.Legend(title="Total Countries")),
339         #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
340         #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
341     ).properties(width=350, height=300).interactive()
342
343     #-- overlap
344     chart = alt.hconcat(bars, heatmap, spacing=50, title="",
345               ).configure_title(color='green', fontSize=24, align="center", anchor="middle"
346               ).configure_axis(labelFontSize=14,
347                                labelColor="red",
348                                titleFontSize=20,
349                                titleColor="blue",
350                                )
351     chart.show()
```



Ex37 bar heatmap bar 在右

```python
def ex37_Bar_Link_Heatmap(df):
    selectedRegion = alt.selection(type="interval", encodings=["x"])

    #-- chart 1
    bars = alt.Chart(df, title="Regions vs. Countries").mark_bar().encode(
        x="Region:N",
        y="count():Q",
        color=alt.condition(selectedRegion,
                            alt.ColorValue("steelblue"), alt.ColorValue("grey"))
    ).properties(width=350, height=300
    ).add_selection(selectedRegion
    ).interactive()

    #-- chart 2
    columns = df.columns
    xData = columns[4] + ":Q"
    yData = columns[3] + ":Q"
    heatmap = alt.Chart(df, title="Well Being vs. Lefe Expectancy").mark_rect().encode(
        x=alt.X(xData, bin=True),
        y=alt.Y(yData, bin=True),
        color=alt.Color('count()', scale=alt.Scale(scheme="greenblue"),
                        legend=alt.Legend(title="Total Countries")),
        #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
        #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
    ).properties(width=350, height=300)

    circles = heatmap.mark_point().encode(
        alt.ColorValue('magenta'),
        alt.Size("count()", legend=alt.Legend(title="Records in Selection"))
        ).interactive()

    heatmapCircle = alt.layer(heatmap, circles)

    #--overlap
    chart = alt.hconcat(heatmapCircle, bars, spacing=50, title="",
              ).configure_title(color='green', fontSize=24, align="center", anchor="middle"
              ).configure_axis(labelFontSize=14,
                               labelColor="red",
                               titleFontSize=20,
                               titleColor="blue",
                               )
    chart.show()
```

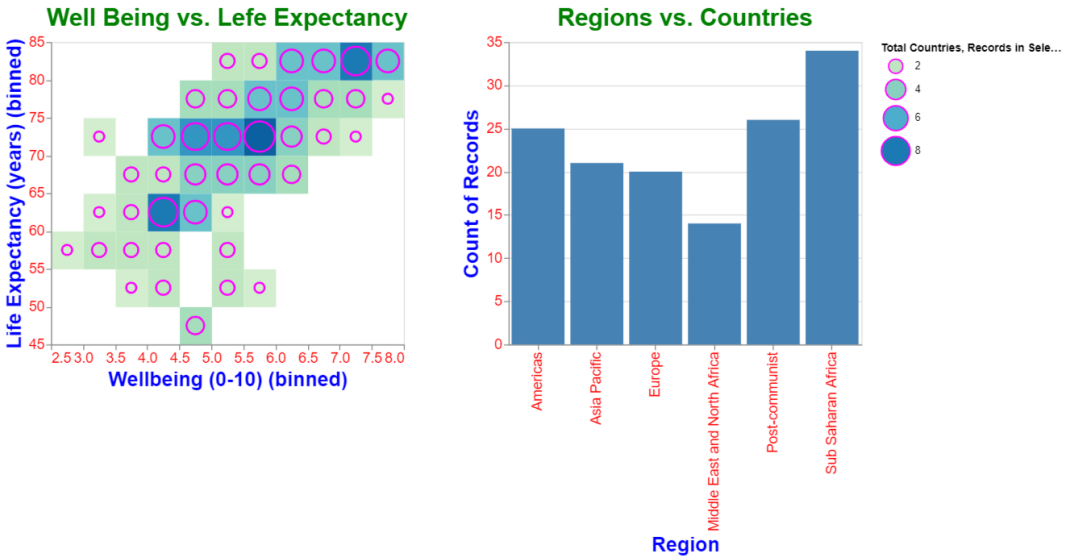


L4_Activity Google Play Store Rating

右邊是年紀評價 app 數量的 bar

左邊是幾個 app 與評價的熱度圖

```python
if (__name__ == "__main__"):
    df = l4_dataset(InputPath + "/googleplaystore.csv")

    gps_apps_df = df.dropna()

    selectedCategory = alt.selection(type="single", encodings=['x'])

    alt.data_transformers.enable('default', max_rows=None)
    bars = alt.Chart(gps_apps_df, title="Content Rating").mark_bar().encode(
        x='Content Rating:N',
        y='count():Q',
        color=alt.condition(selectedCategory, alt.ColorValue("steelblue"), alt.ColorValue("grey"))
    ).properties(width=100, height=100
    ).add_selection(selectedCategory)

    #-- Heatmap indicating number of apps across app Category and Rating ranges.
    heatmap = alt.Chart(gps_apps_df, title="Rating of Category").mark_rect().encode(
        alt.X('Category:N'),
        alt.Y('Rating:Q', bin=True),
        alt.Color('count()',
            scale=alt.Scale(scheme='greenblue'),
            legend=alt.Legend(title='Total Apps')
        )
    ).properties(width=400, height=100)

    #-- Circle stick on heatmap
    circles = heatmap.mark_point().encode(
        alt.ColorValue('magenta'),
        alt.Size('count()', scale=alt.Scale(domain=(1, 600), range=(1, 200)),
                            legend=alt.Legend(title='Apps in Selection'))
    ).transform_filter(selectedCategory)

    heatmapCircles = alt.layer(heatmap, circles)

    chart = alt.hconcat(heatmapCircles, bars, spacing=50, title="Google Play Store Apps Rating"
                ).configure_title(color='green', fontSize=22, align="center", anchor="middle"
                ).configure_axis(labelFontSize=14,
                                 labelColor="red",
                                 titleFontSize=20,
                                 titleColor="blue",
                                 )
    chart.show()
```



HW Co2 & gdp

紅色框為與 Google app 不同之處

Spyder (Python 3.8)

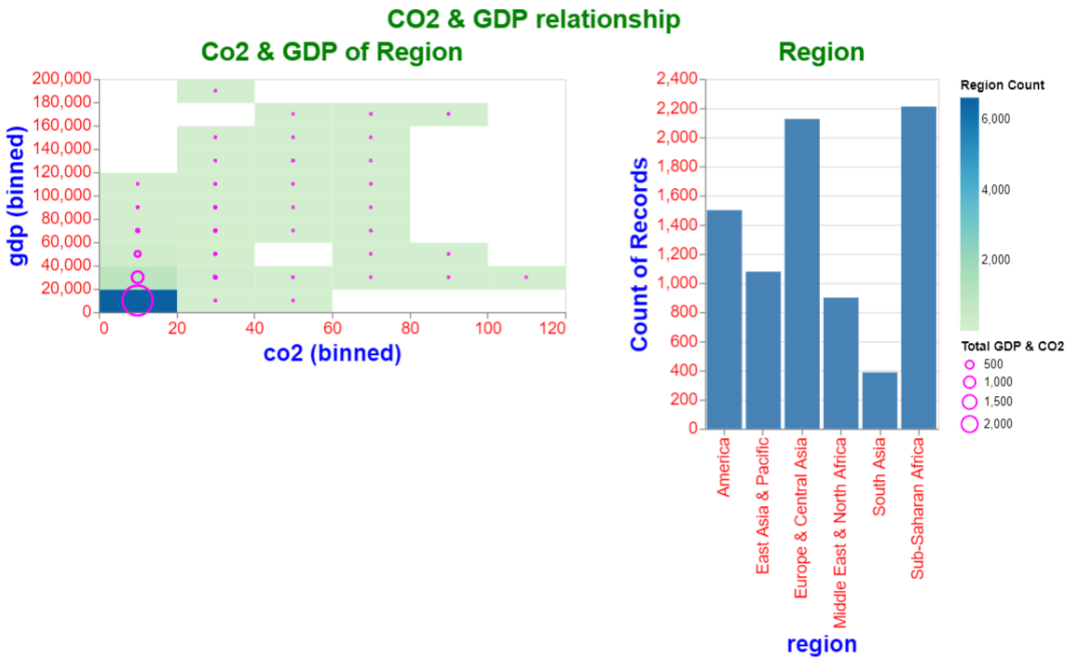
File Edit Search Source Run Debug Consoles Projects Tools View Help

E:\Data_Visualization\Class\Pan_Lib_Python

E:\Data_Visualization\Class\Pan_Lib_Python\L4_HW_Co2Gdp_A108222040.py

L5_IDV_Activity_A108222040.py | APP_IDV_Weather.py | L4_HW_Olympics_A108222040.py | L4_HW_Co2Gdp_A108222040.py

```
33
34  if (__name__ == "__main__"):
35      df = l4_dataset(InputPath + "/new_co2_gdp.csv")
36
37      gdp_aco2_df = df.dropna()
38
39      selectedRegion = alt.selection(type="single", encodings=['x'])
40
41      alt.data_transformers.enable('default', max_rows=None)
42      bars = alt.Chart(gdp_aco2_df, title="Region").mark_bar().encode(
43          x='region:N',
44          y='count():Q',
45          color=alt.condition(selectedRegion, alt.ColorValue("steelblue"), alt.ColorValue("grey"))
46      ).properties(width=200
47      ).add_selection(selectedRegion)
48
49      #-- Heatmap indicating number of apps across app Category and Rating ranges.
50      heatmap = alt.Chart(gdp_aco2_df, title="Co2 & GDP of Region").mark_rect().encode(
51          alt.X('co2', bin=True),
52          alt.Y('gdp', bin=True),
53          alt.Color('count()',
54              scale=alt.Scale(scheme='greenblue'),
55              legend=alt.Legend(title='Region Count')
56          )
57      ).properties(width=400, height=200)
58
59      #-- Circle stick on heatmap
60      circles = heatmap.mark_point().encode(
61          alt.ColorValue('magenta'),
62          alt.Size('count()', scale=alt.Scale(domain=(1, 2000), range=(1, 200)),
63                   legend=alt.Legend(title='Total GDP & CO2'))
64      ).transform_filter(selectedRegion)
65
66      heatmapCircles = alt.layer(heatmap, circles)
67
68      chart = alt.hconcat(heatmapCircles, bars, spacing=50, title="CO2 & GDP relationship"
69                ).configure_title(color='green', fontSize=22, align="center", anchor="middle"
70                ).configure_axis(labelFontSize=14,
71                                 labelColor="red",
72                                 titleFontSize=20,
73                                 titleColor="blue",
74                                 )
75      chart.show()
```



HW Olympics

紅色框為與 Google app 不同之處

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

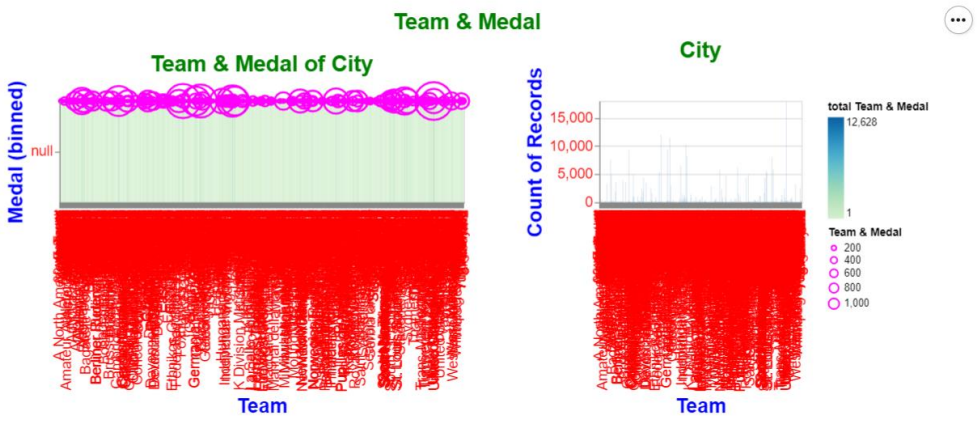
E:\Data_Visualization\Class\Pan_Lib_Python

E:\Data_Visualization\Class\Pan_Lib_Python\L4_HW_Olympics_A108222040.py

L5_IDV_Activity_A108222040.py | APP_IDV_Weather.py | L4_HW_Olympics_A108222040.py

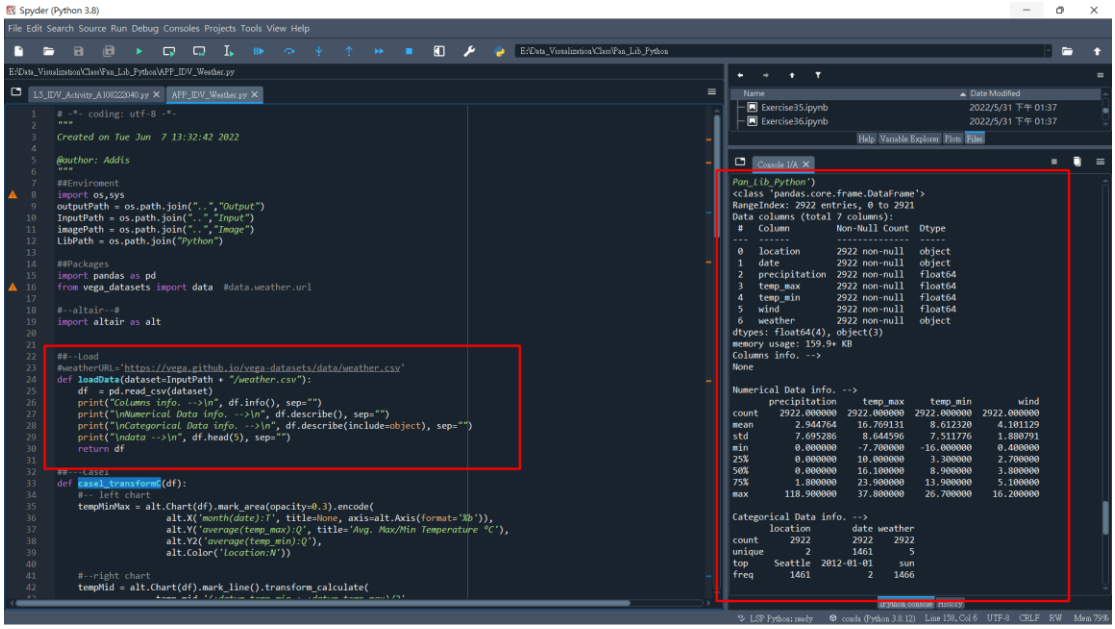
```
33
34  if (__name__ == "__main__"):
35      df = l4_dataset(InputPath + "/athlete_events.csv")
36
37      medal_df = df.dropna(how='all')
38
39      selectedCategory = alt.selection(type="single", encodings=['x'])
40
41      alt.data_transformers.enable('default', max_rows=None)
42      bars = alt.Chart(medal_df, title="City").mark_bar().encode(
43          x='Team:N',
44          y='count():Q',
45          color=alt.condition(selectedCategory, alt.ColorValue("steelblue"), alt.ColorValue("grey"))
46      ).properties(width=200, height=100
47      ).add_selection(selectedCategory)
48
49      #-- Heatmap indicating number of apps across app Category and Rating ranges.
50      heatmap = alt.Chart(medal_df, title="Team & Medal of City").mark_rect().encode(
51          alt.X('Team:N'),
52          alt.Y('Medal', bin=True),
53          alt.Color('count()',
54              scale=alt.Scale(scheme='greenblue'),
55              legend=alt.Legend(title='total Team & Medal')
56          )
57      ).properties(width=400, height=100)
58
59      #-- Circle stick on heatmap
60      circles = heatmap.mark_point().encode(
61          alt.ColorValue('magenta'),
62          alt.Size('count()', scale=alt.Scale(domain=(1, 1000), range=(1, 110)),
63                           legend=alt.Legend(title='Team & Medal'))
64      ).transform_filter(selectedCategory)
65
66      heatmapCircles = alt.layer(heatmap, circles)
67
68      chart = alt.hconcat(heatmapCircles, bars, spacing=50, title="Team & Medal"
69                  ).configure_title(color='green', fontSize=22, align="center", anchor="middle"
70                  ).configure_axis(labelFontSize=14,
71                                   labelColor="red",
72                                   titleFontSize=20,
73                                   titleColor="blue",
74                                   )
75      chart.show()
```

很多國家導致都聚在一起



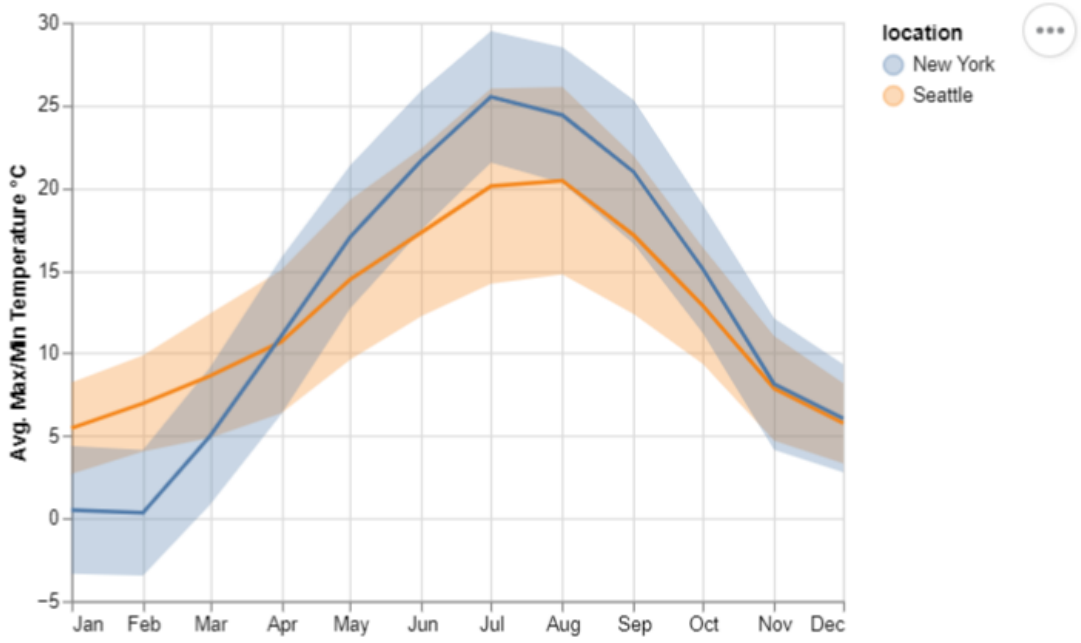
Weather Week16

Dataset -> weather



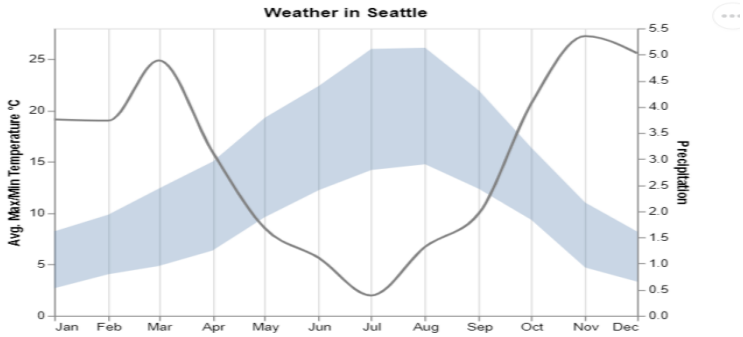
Case1 New York Seattle 平均高低溫

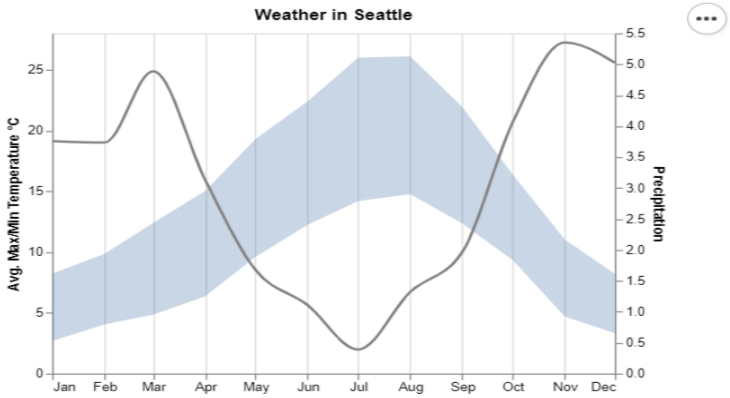
```python
##---Case1
def case1_transformC(df):
    #-- left chart
    tempMinMax = alt.Chart(df).mark_area(opacity=0.3).encode(
                    alt.X('month(date):T', title=None, axis=alt.Axis(format='%b')),
                    alt.Y('average(temp_max):Q', title='Avg. Max/Min Temperature °C'),
                    alt.Y2('average(temp_min):Q'),
                    alt.Color('location:N'))

    #--right chart
    tempMid = alt.Chart(df).mark_line().transform_calculate(
                    temp_mid='(+datum.temp_min + +datum.temp_max)/2'
                ).encode(alt.X('month(date):T'),
                         alt.Y('average(temp_mid):Q'),
                         alt.Color('location:N'))

    #-- tempMinMax + tempMid
    chart = alt.layer(tempMinMax, tempMid)
    charts = alt.hconcat(tempMinMax, tempMid, spacing=50, title="Weather", center=True,
              ).configure_title(color='green', fontSize=24, anchor="middle", align="right",
              ).configure_axis(labelFontSize=14,
                               labelColor="red",
                               titleFontSize=20,
                               titleColor="blue",
                               )
    chart.show()
```



Case2 Seattle

```
58
59  def case2_transformF(df):
60      #--method 1
61      precip = alt.Chart(df).transform_filter('datum.location == "Seattle"'
62                ).mark_line(interpolate='monotone', stroke='grey'
63                ).encode(alt.X('month(date):T', title=None, axis=alt.Axis(format='%b')),
64                         alt.Y('average(precipitation):Q', title='Precipitation'))
65
66      tempMinMax = alt.Chart(df).transform_filter('datum.location == "Seattle"'
67                ).mark_area(opacity=0.3).encode(
68                         alt.X('month(date):T', title=None, axis=alt.Axis(format='%b')),
69                         alt.Y('average(temp_max):Q', title='Avg. Max/Min Temperature °C'),
70                         alt.Y2('average(temp_min):Q'))
71
72      chart = alt.layer(tempMinMax, precip, title="Weather in Seattle").resolve_scale(y='independent')
73      chart.show()
74
75      #--method 2
76      precip = alt.Chart(df).mark_line(interpolate='monotone', stroke='grey'
77                ).encode(alt.X('month(date):T', title=None),
78                         alt.Y('average(precipitation):Q', title='Precipitation'))
79
80      tempMinMax = alt.Chart().mark_area(opacity=0.3).encode(
81                         alt.X('month(date):T', title=None, axis=alt.Axis(format='%b')),
82                         alt.Y('average(temp_max):Q', title='Avg. Max/Min Temperature °C'),
83                         alt.Y2('average(temp_min):Q'))
84
85      chart = alt.layer(tempMinMax, precip, data=df, title="Weather in Seattle"
86                      ).transform_filter('datum.location == "Seattle"'
87                      ).resolve_scale(y='independent')
88      chart.show()
```

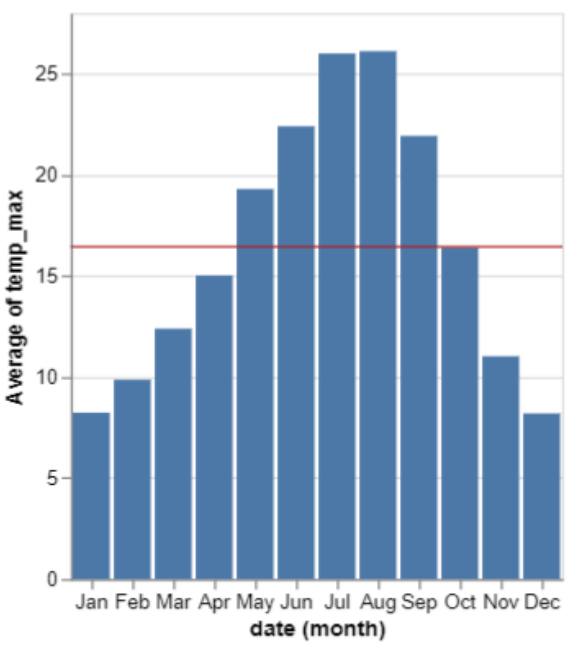


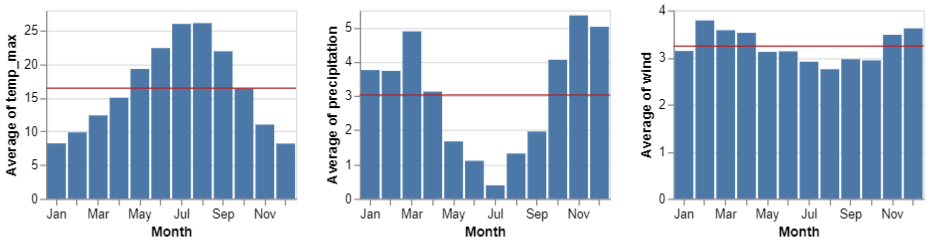


Case6 綜合

```python
#Q1
def case6_composition(df):
    #-- method 1
    basic1 = alt.Chart(df).transform_filter('datum.location == "Seattle"'
              ).mark_bar(
              ).encode(alt.X('month(date):O'), alt.Y('average(temp_max):Q'))
    basic2 = alt.Chart(df).transform_filter('datum.location == "Seattle"'
              ).mark_rule(stroke='firebrick'
              ).encode(alt.Y('average(temp_max):Q'))

    chart = alt.layer(basic1, basic2)
    chart.show()

    #--method 2
    chart = alt.layer(alt.Chart().mark_bar().encode(
                        alt.X('month(date):O', title='Month'),
                        alt.Y(alt.repeat('column'), aggregate='average', type='quantitative')),
                    alt.Chart().mark_rule(stroke='firebrick').encode(
                        alt.Y(alt.repeat('column'), aggregate='average', type='quantitative'))
                    ).properties(width=200, height=150
                    ).repeat(data=df, column=['temp_max', 'precipitation', 'wind']
                    ).transform_filter('datum.location == "Seattle"')
    chart.show()

    #--method 3
    splom = alt.Chart().mark_point(filled=True, size=15, opacity=0.5
              ).encode(alt.X(alt.repeat('column'), type='quantitative'),
                       alt.Y(alt.repeat('row'), type='quantitative')
              ).properties(width=125, height=125
              ).repeat(row=['temp_max', 'precipitation', 'wind'],
                       column=['wind', 'precipitation', 'temp_max']
              )

    dateHist = alt.layer(alt.Chart(df).mark_bar().encode(
                            alt.X('month(date):O', title='Month'),
                            alt.Y(alt.repeat('row'), aggregate='average', type='quantitative')),
                            #alt.Color('month(date):O', scale = alt.Scale(range=['#aec7e8', 'c7c7c7', '#1f77b4', '#9467bd',
                        alt.Chart().mark_rule(stroke='firebrick').encode(
                            alt.Y(alt.repeat('row'), aggregate='average', type='quantitative'))
                           ).properties(width=175, height=125
                           ).repeat(row=['temp_max', 'precipitation', 'wind'])

    tempHist =
```

```python
                       alt.Y(alt.repeat('row'), type='quantitative')
              ).properties(width=125, height=125
              ).repeat(row=['temp_max', 'precipitation', 'wind'],
                       column=['wind', 'precipitation', 'temp_max']
              )

    dateHist = alt.layer(alt.Chart(df).mark_bar().encode(
                            alt.X('month(date):O', title='Month'),
                            alt.Y(alt.repeat('row'), aggregate='average', type='quantitative')),
                            #alt.Color('month(date):O', scale = alt.Scale(range=['#aec7e8', 'c7c7c7', '#1f77b4', '#9467bd',
                        alt.Chart().mark_rule(stroke='firebrick').encode(
                            alt.Y(alt.repeat('row'), aggregate='average', type='quantitative'))
                           ).properties(width=175, height=125
                           ).repeat(row=['temp_max', 'precipitation', 'wind'])

    tempHist =  alt.Chart(df).mark_bar(
                  ).encode(alt.X('temp_max:Q', bin=True, title='Temperature (°C)'),
                           alt.Y('count():Q'),
                           alt.Color('weather:N', scale=alt.Scale(
                                     domain=['drizzle', 'fog', 'rain', 'snow', 'sun'],
                                     range=['blue', 'black', 'yellow', 'pink', 'red']))
                  ).properties(width=115, height=100
                  ).facet(column="weather:N")

    chart = alt.vconcat(alt.hconcat(splom, dateHist),
                        tempHist, data=df, title='Seattle Weather Dashboard', center=True
              ).transform_filter('datum.location == "Seattle"'
              ).resolve_legend(color='independent'
              ).configure_axis(
                              labelAngle=0,
                              labelFontSize=10,
                              labelColor="orange",
                              titleFontSize=16,
                              titleColor="green",)
    chart.show()

if (__name__ == "__main__"):
    df = loadData()
    #case1_transformC(df)
    #case2_transformP(df)
    case6_composition(df)
```

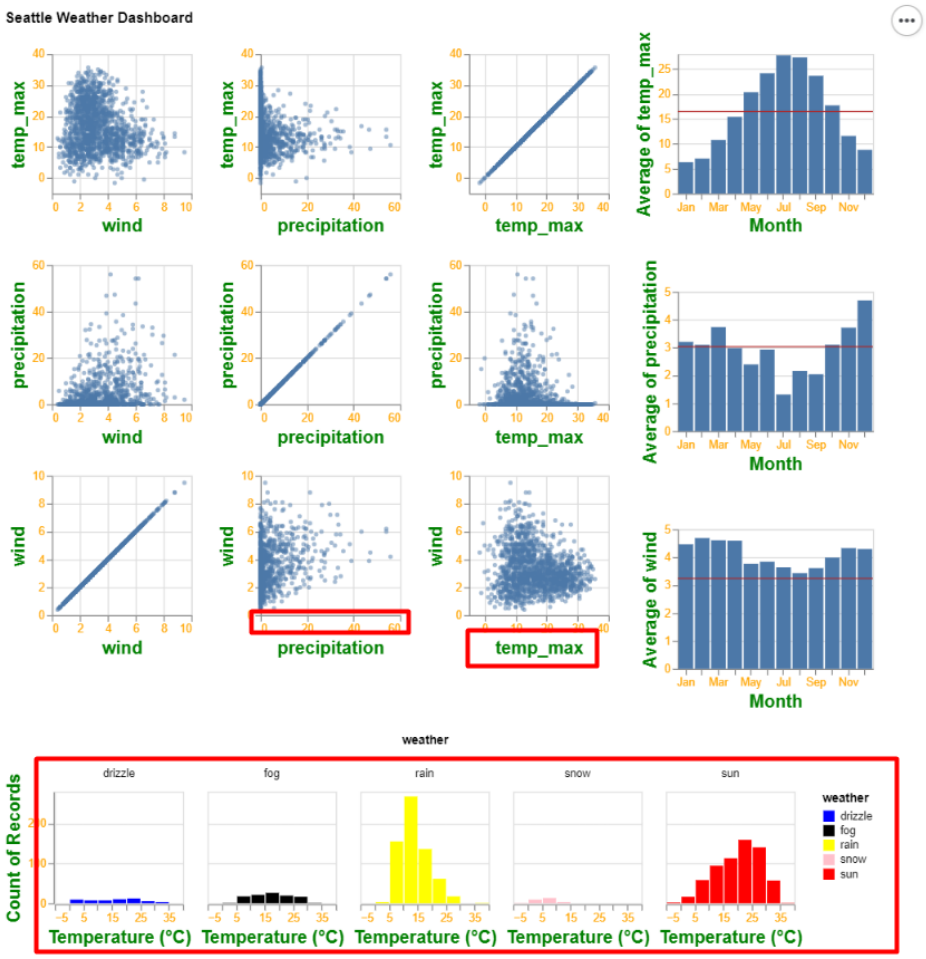

Average of temp_max
25
20
15
10
5
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
date (month)


Average of temp_max
Jan Mar May Jul Sep Nov
Month
Average of precipitation
Jan Mar May Jul Sep Nov
Month
Average of wind
Jan Mar May Jul Sep Nov
Month

有改變 x y 軸的字顏色

刻度也有改變

最下面每個圖的 bar 都有改變

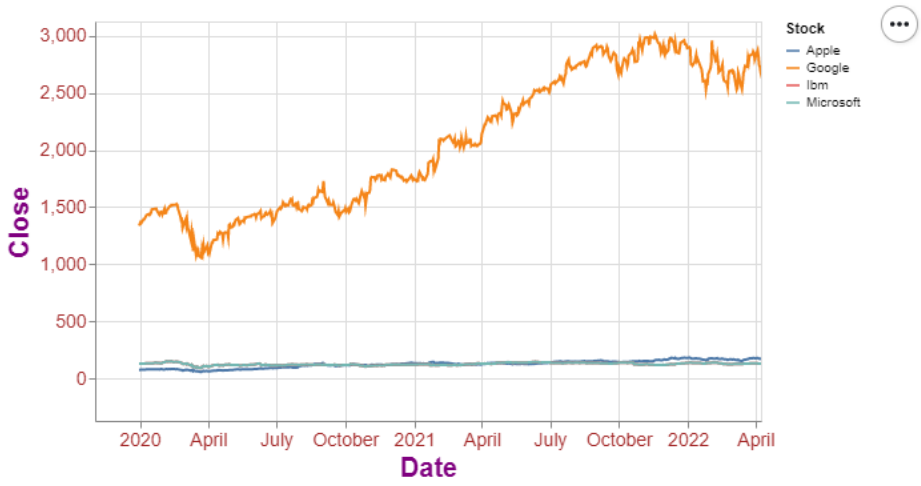


Stock

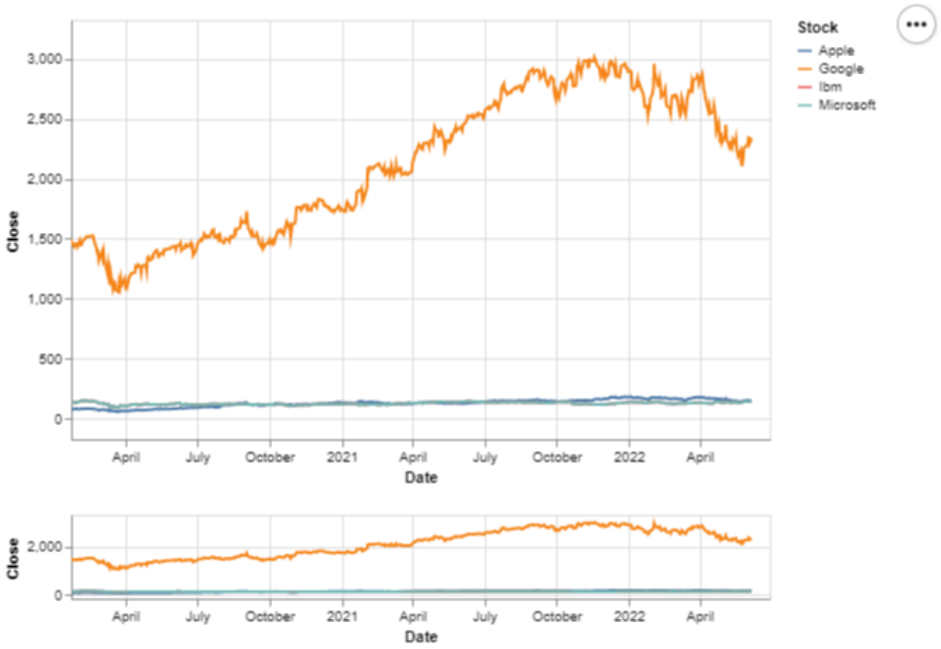
L5_IDV_Activity_A108222040.py

```python
def getStocks():
    #-- get data
    startDate = "2020-1-1"
    endDate = "2022-6-6"
    source = "yahoo"
    #-- get data
    appleStock = pdData.DataReader("AAPL", start=startDate, end=endDate, data_source=source
                                   ).reset_index()[["Date", "Close", "High"]]
    ibmStock = pdData.DataReader("IBM", start=startDate, end=endDate, data_source=source
                                 ).reset_index()[["Date", "Close", "High"]]
    googleStock = pdData.DataReader("GOOG", start=startDate, end=endDate, data_source=source
                                    ).reset_index()[["Date", "Close", "High"]]
    microsoftStock = pdData.DataReader("IBM", start=startDate, end=endDate, data_source=source
                                       ).reset_index()[["Date", "Close", "High"]]

    #--save data
    appleStock.to_csv(InputPath + "/AppleStock_20200101_20220606.csv")
    ibmStock.to_csv(InputPath + "/IbmStock_20200101_20220606.csv")
    googleStock.to_csv(InputPath + "/GoogleStock_20200101_20220606.csv")
    microsoftStock.to_csv(InputPath + "/MicrosfotStock_20200101_20220606.csv")

    #--process data
    appleStock["Stock"] = "Apple"
    ibmStock["Stock"] = "Ibm"
    googleStock["Stock"] = "Google"
    microsoftStock["Stock"] = "Microsoft"

    stocks = pd.concat([appleStock, ibmStock, googleStock, microsoftStock])
    stocks["Month"] = stocks.Date.dt.month
    stocks["Year"] = stocks.Date.dt.year
    stocks["Day"] = stocks.Date.dt.day
    stocks.to_csv(InputPath + "/stocks_20200101_20220606.csv")

    return stocks
```

```python
def IDV_Altair_Stocks(stocks):
    stockSelection1 = alt.selection_single(fields=["Stock"], bind="legend")

    #--rightChart
    rightChart = alt.Chart(stocks).mark_line().encode(
                        x="Date", y="Close", color="Stock",
                        opacity=alt.condition(stockSelection1, alt.value(1), alt.value(0.1)),
                        tooltip=["Date", "Close"] #hover
        ).properties(height=300, width=500
        ).configure_title(color="green", fontSize=24
        ).configure_axis(labelFontSize=14,
                         labelColor="brown",
                         titleFontSize=20,
                         titleColor="purple"
        ).add_selection(stockSelection1
        ).interactive()

    rightChart.show()


    #-- left chart
    stockSelection1 = alt.selection_single(fields=["Stock"], bind="legend")

    rightChart = alt.Chart(stocks).mark_line().encode(
                        x="Date", y="Close", color="Stock",
                        opacity=alt.condition(stockSelection1, alt.value(1), alt.value(0.1)),
                        tooltip=["Date", "Close"] #hover
        ).properties(height=300, width=500
        ).add_selection(stockSelection1
        ).interactive()

    upper = rightChart.encode(alt.X('Date:T', scale=alt.Scale(domain=stockSelection1)))

    lower = rightChart.properties(height=60).add_selection(stockSelection1)

    leftChart = upper & lower
    leftChart.show()
```



下面與上面一樣 可以連動



只抓 microsoft 的股票

```
#-- left chart
brush = alt.selection(type='interval', encodings=['x'])

base = alt.Chart(stocks.loc[stocks["Stock"] == "Microsoft"]
           ).mark_line().encode(x = 'Date:T', y = 'Close:Q'
           ).properties(width=600, height=200)

upper = base.encode(alt.X('Date:T', scale=alt.Scale(domain=brush)))

lower = base.properties(height=60).add_selection(brush)

leftChart = upper & lower
```