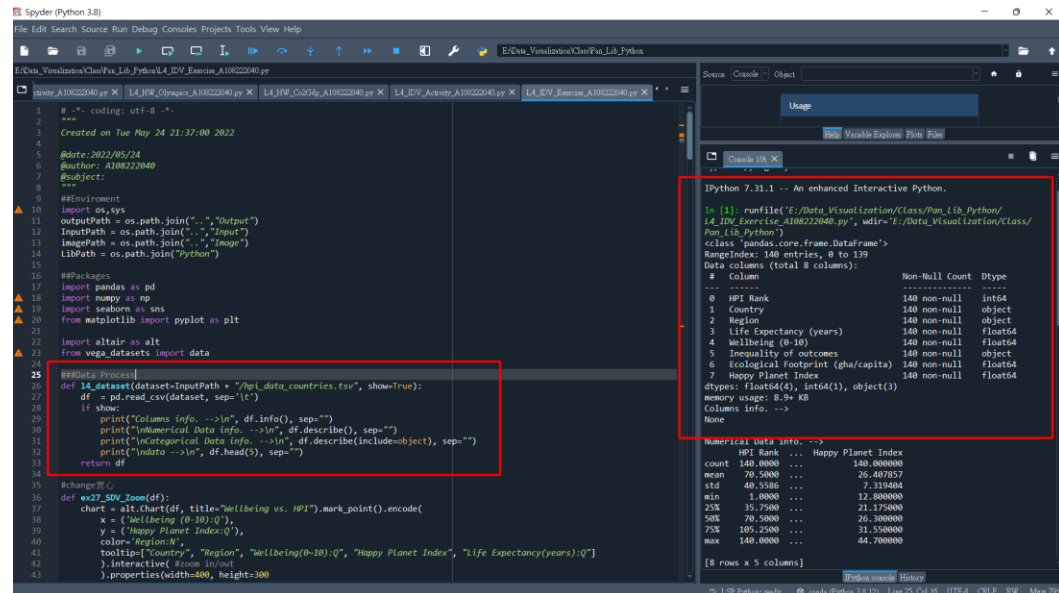


A108222040Data_Visulization_Week14&15&16

Week14

Show dataset(國家快樂指數)



```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue May 24 21:37:00 2022
4
5 @date: 2022/05/24
6 @author: A108222040
7 @subject:
8 """
9 ##Environment
10 import os, sys
11 outputPath = os.path.join(".", "Output")
12 inputPath = os.path.join(".", "Input")
13 imagePath = os.path.join(".", "Image")
14 libPath = os.path.join("Python")
15
16 ##Packages
17 import pandas as pd
18 import numpy as np
19 import seaborn as sns
20 from matplotlib import pyplot as plt
21
22 import altair as alt
23 from vega_datasets import data
24
25 ##Data Process
26 def ex27_SDV_Zoom(df):
27     chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_circle().encode(
28         x=('Wellbeing (0-10):Q'),
29         y=('Happy Planet Index:Q'),
30         color='Region:N',
31         tooltip=['Country', 'Region', 'Wellbeing(0-10):Q', 'Happy Planet Index', 'Life Expectancy(years):Q']
32     ).interactive( #zoom in/out
33     ).properties(width=400, height=300)
34     chart.show()
```

IPython 7.31.1 -- An enhanced Interactive Python.

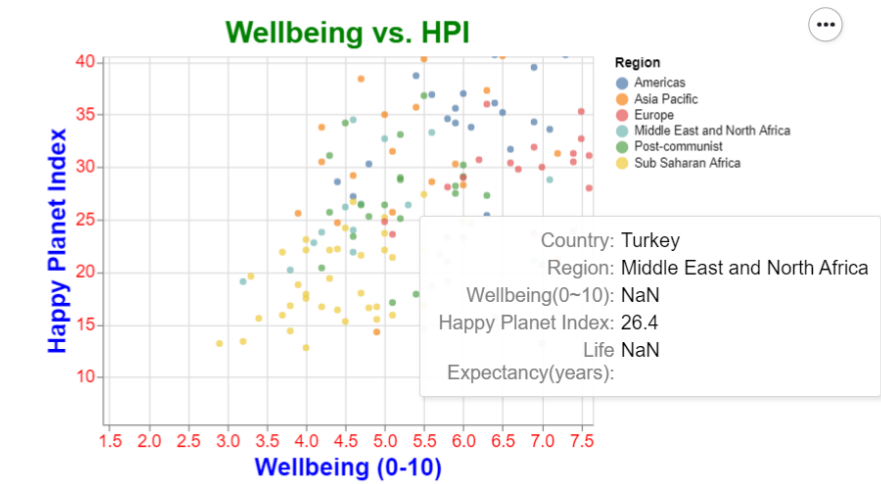
```
In [1]: runfile('E:/Data_Visualization/Class/Pan_Lib_Python/
A108222040.py', wdir='E:/Data_Visualization/Class/
Pan_Lib_Python')
Out[1]: pandas.core.frame.DataFrame
RangeIndex: 140 entries, 0 to 139
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   HPI Rank             140 non-null    int64
 1   Country              140 non-null    object
 2   Region               140 non-null    object
 3   Life Expectancy (years) 140 non-null    float64
 4   Wellbeing (0-10)      140 non-null    float64
 5   Inequality of outcomes 140 non-null    object
 6   Ecological footprint (gha/capita) 140 non-null    float64
 7   Happy Planet Index     140 non-null    float64
dtypes: float64(4), int64(1), object(3)
memory usage: 8.9+ KB
Columns Info: -->
None

Numerical data info: -->
HPI Rank ... Happy Planet Index
count 140.00000 ... 140.000000
mean 70.50000 ... 26.407857
std 40.5556 ... 7.311804
min 1.00000 ... 12.800000
25% 35.75000 ... 21.175000
50% 70.50000 ... 26.300000
75% 105.25000 ... 31.550000
max 140.00000 ... 44.700000
[8 rows x 5 columns]
```

Ex27 circle 是實心 point 是空心

```
33     return df
34
35 #change 實心
36 def ex27_SDV_Zoom(df):
37     chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_circle().encode(
38         x=('Wellbeing (0-10):Q'),
39         y=('Happy Planet Index:Q'),
40         color='Region:N',
41         tooltip=['Country', 'Region', 'Wellbeing(0-10):Q', 'Happy Planet Index', 'Life Expectancy(years):Q']
42     ).interactive( #zoom in/out
43     ).properties(width=400, height=300)
44     chart.show()
```

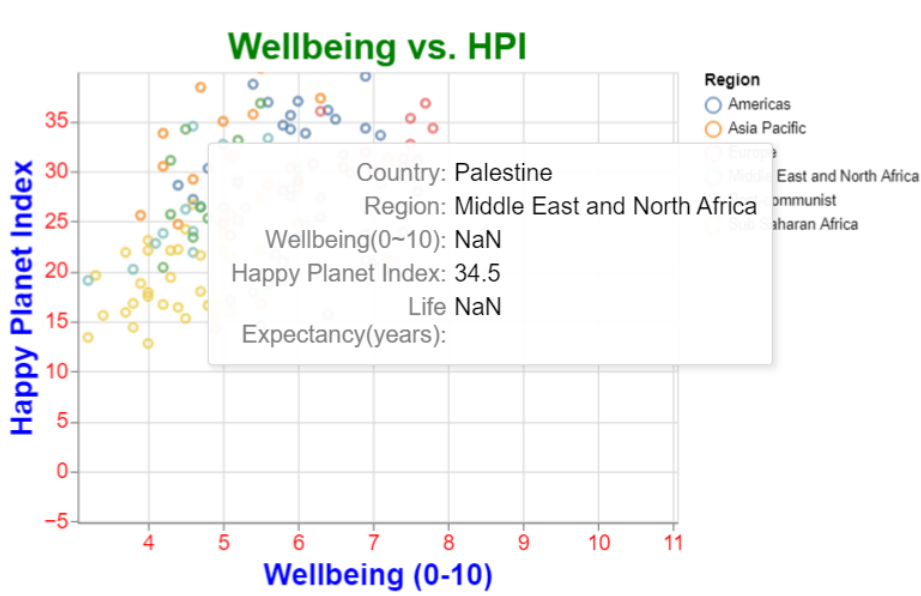
Result



Ex28

```
50 chart.show()
51
52 def ex28_SDV_Zoom_Hover(df):
53     chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_circle().encode(
54         x = 'Wellbeing (0-10):Q',
55         y = 'Happy Planet Index:Q',
56         color='Region:N',
57         tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
58     ).interactive( #zoom in/out
59     ).properties(width=400, height=300)
60     .configure_title(color='green', fontSize=24)
61     .configure_axis(labelFontSize=14,
62                     labelColor="red",
63                     titleFontSize=20,
64                     titleColor="blue",
65                     )
66     chart.show()
67
```

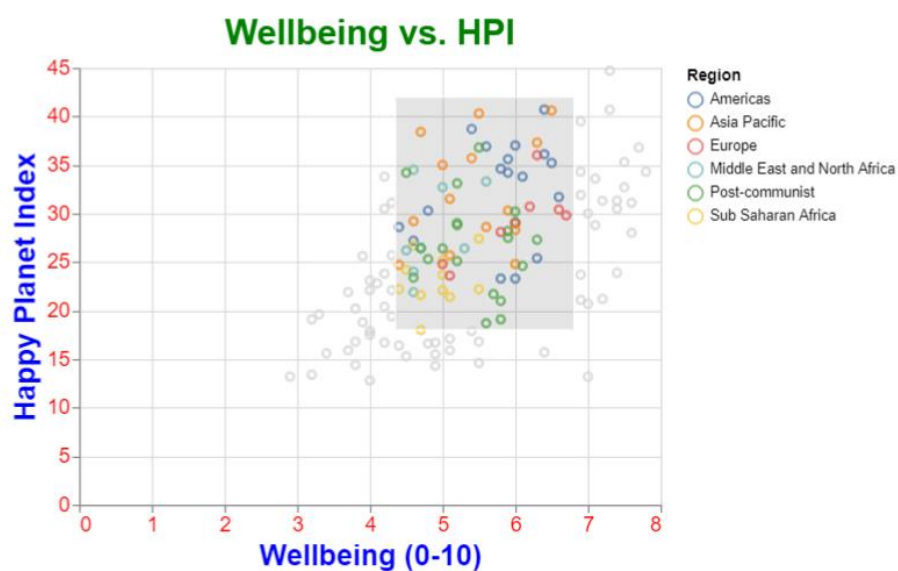
Result



Ex29

可以選擇一塊區域做觀看

```
65 }
66 chart.show()
67
68 def ex29_SelectedArea=(df):
69     selected_area = alt.selection_interval()
70
71     chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_point().encode(
72         x=Wellbeing (0-10):Q,
73         y=Happy Planet Index:Q,
74         color=alt.condition(selected_area, 'Region:N', alt.value('lightgray')),
75         tooltip=["Country:", "Region:", "Wellbeing(0-10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
76     ).add_selection(selected_area)
77     .properties(strokeDash=[5, 5])
78     .configure_title(color='green', fontSize=24)
79     .configure_axis(labelFontSize=14,
80                     labelColor='red',
81                     titleFontSize=20,
82                     titleColor='blue',
83                     )
84     chart.show()
85
86 def ex30_SelectedArea_Zoom_Hover(df):
87     selected_area = alt.selection_interval()
```

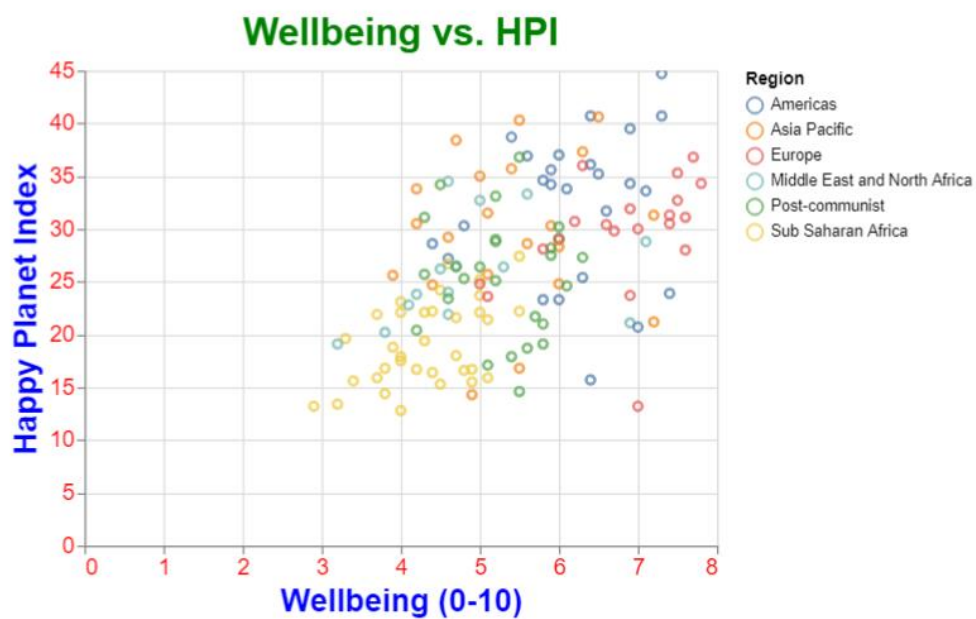


Ex30

可以選擇一塊區域做觀看 Zoom 移動

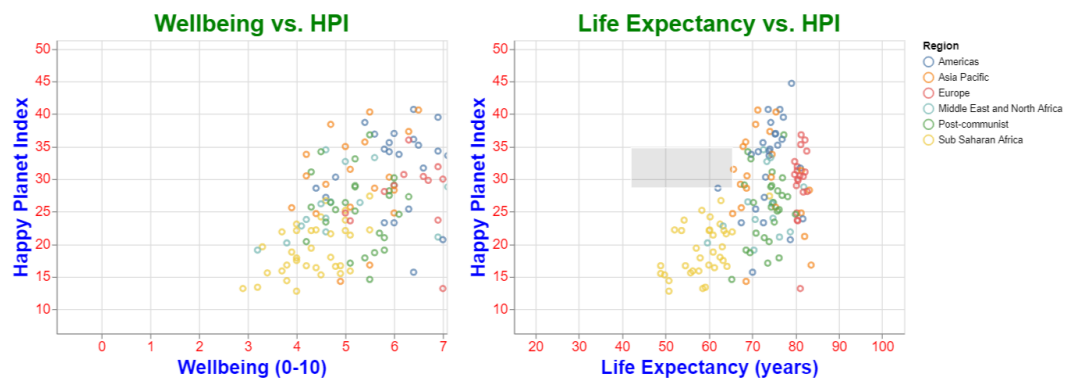
紅框為與 29 不同之處

```
84 chart.show()
85
86 def ex30_SelectedArea_Zoom_Hover(df):
87     selected_area = alt.selection_interval()
88
89     chart = alt.Chart(df, title="Wellbeing vs. HPI").mark_point().encode(
90         x='Wellbeing (0-10):Q',
91         y='Happy Planet Index:Q',
92         color=alt.condition(selected_area, 'Region:N', alt.value('lightgray')),
93         tooltip=['Country', 'Region', 'Wellbeing(0~10):Q', 'Happy Planet Index', 'Life Expectancy(years):Q']
94     ).interactive(
95     ).add_selection(selected_area
96     ).properties(width=400, height=300
97     ).configure_title(color='green', fontSize=24
98     ).configure_axis(labelFontSize=14,
99                     labelColor='red',
100                     titleFontSize=20,
101                     titleColor='blue',
102 )
```



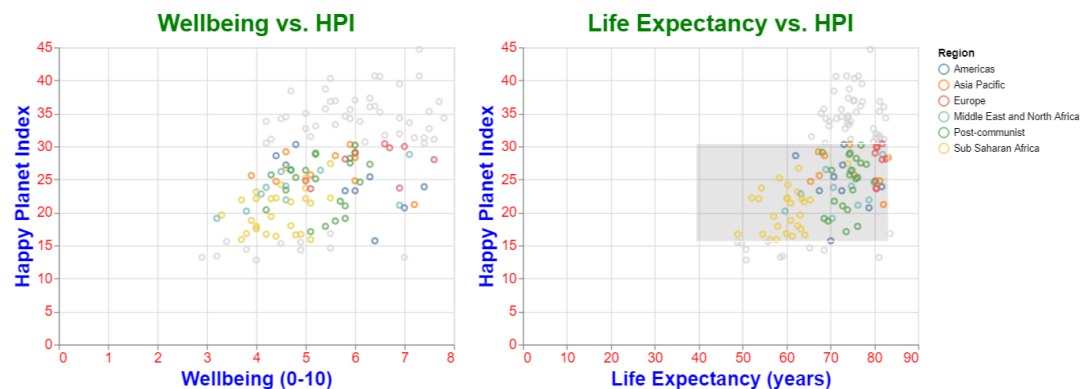
Ex31 左右兩個區域 可以同時移動

```
104
105 def ex31_MultiplePlot(df):
106     selectedArea = alt.selection_interval()
107
108     chart = alt.Chart(df, title="HPI").mark_point().encode(
109         x='Wellbeing (0-10):Q',
110         y='Happy Planet Index:Q',
111         color='Region:N',
112         tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
113     ).interactive(
114     ).add_selection(selectedArea
115     ).properties(width=400, height=300)
116     # .configure_title(color='green', fontSize=24
117     # ).configure_axis(labelFontSize=14,
118     #                   labelColor="red",
119     #                   titleFontSize=20,
120     #                   titleColor="blue",
121     # )
122     chart1 = chart.encode(x='Wellbeing (0-10)').properties(title="Wellbeing vs. HPI")
123     chart2 = chart.encode(x='Life Expectancy (years)').properties(title="Life Expectancy vs. HPI")
124     chart = alt.hconcat(chart1, chart2
125     ).configure_title(color='green', fontSize=24
126     ).configure_axis(labelFontSize=14,
127                     labelColor="red",
128                     titleFontSize=20,
129                     titleColor="blue",
130     )
131     chart.show()
```



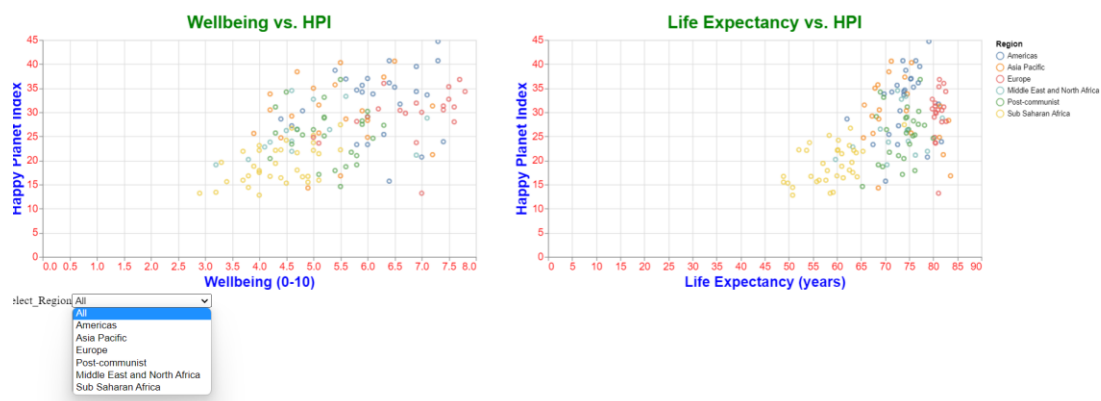
Ex31 右邊選擇區域 左邊就會亮該處

```
132
133 def ex31_Selection(df):
134     selectedArea = alt.selection_interval()
135
136     chart = alt.Chart(df, title="HPI").mark_point().encode(
137         #x='Wellbeing (0-10):Q',
138         y='Happy Planet Index:Q',
139         color=alt.condition(selectedArea, 'Region', alt.value('lightgray')),
140         tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
141     ), interactive()
142     ).add_selection(selectedArea
143     ).properties(width=400, height=300)
144     # .configure_title(color='green', fontSize=24
145     # ).configure_axis(labelFontSize=14,
146     #                  labelColor="red",
147     #                  titleFontSize=20,
148     #                  titleColor="blue",
149     # )
150     chart1 = chart.encode(x='Wellbeing (0-10)').properties(title="Wellbeing vs. HPI")
151     chart2 = chart.encode(x='Life Expectancy (years)').properties(title="Life Expectancy vs. HPI")
152     charts = alt.hconcat(chart1, chart2
153         ).configure_title(color='green', fontSize=24
154         ).configure_axis(labelFontSize=14,
155                        labelColor="red",
156                        titleFontSize=20,
157                        titleColor="blue",
158        )
159     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()
```



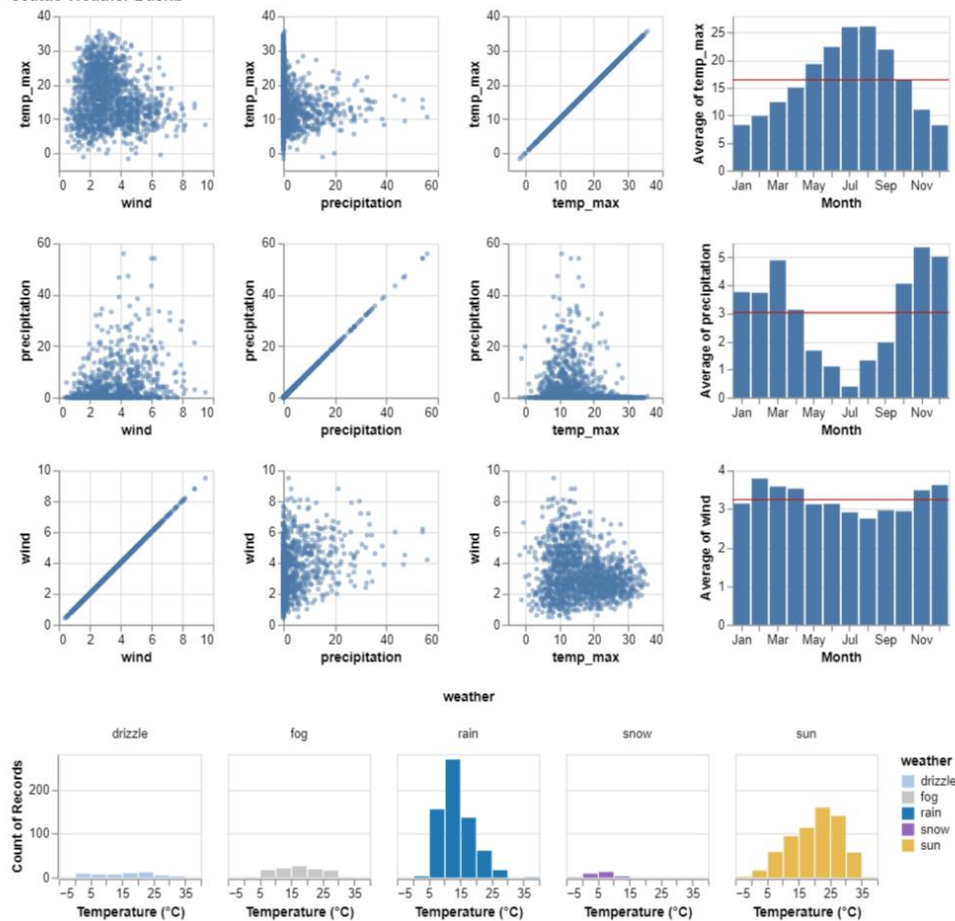
範例 weather

```

187 charts.show()
188
189 def testAltair_MP_weather():
190     # ---Load Data
191     weather = 'https://vega.github.io/vega-datasets/data/weather.csv'
192     df = pd.read_csv(weather)
193
194     # --plot
195     splom = alt.Chart().mark_point(filled=True, size=15, opacity=0.5)
196     .encode(alt.X(alt.repeat('column'), type='quantitative'),
197            alt.Y(alt.repeat('row'), type='quantitative'))
198            .properties(width=125, height=125)
199            .repeat(row=['temp_max', 'precipitation', 'wind'],
200                  column=['wind', 'precipitation', 'temp_max'])
201
202     dateHist = alt.layer(alt.Chart().mark_bar().encode(
203         alt.X('month(date):0', title='Month'),
204         alt.Y(alt.repeat('row'), aggregate='average', type='quantitative')),
205                    alt.Chart().mark_rule(stroke='firebrick').encode(
206                        alt.Y(alt.repeat('row'), aggregate='average', type='quantitative'))
207                    .properties(width=175, height=125)
208                    .repeat(row=['temp_max', 'precipitation', 'wind']))
209
210     tempHist = alt.Chart(weather).mark_bar().encode(
211         alt.X('temp_max:Q', bin=True, title='Temperature (°C)'),
212         alt.Y('count():Q'),
213         alt.Color('weather:N', scale=alt.Scale(
214             domain=['drizzle', 'fog', 'rain', 'snow', 'sun'],
215             range=['#aec7e8', '#c7c7c7', '#1f77b4', '#9467bd', '#e7ba52']))
216         .properties(width=115, height=100)
217         .facet(column='weather:N')
218
219     chart = alt.vconcat(alt.hconcat(splom, dateHist), tempHist, data=weather, title='Seattle Weather Dashb'
220                        .transform_filter('datum.location == "Seattle"')
221                        .resolve_legend(color='independent')
222                        .configure_axis(labelAngle=0))
223
224     chart.show()
225

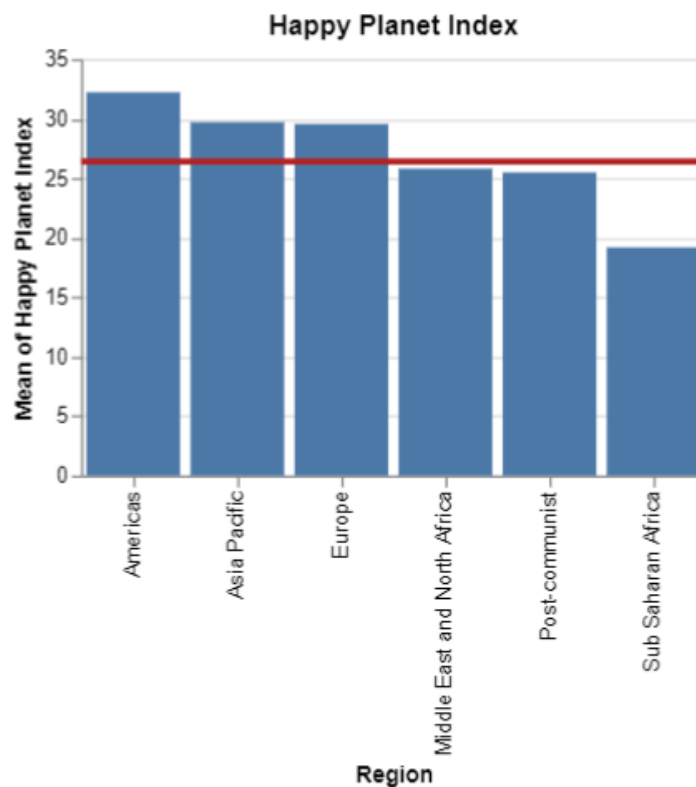
```

Seattle Weather Dashb



Ex33 bar 一樣是用快樂指數

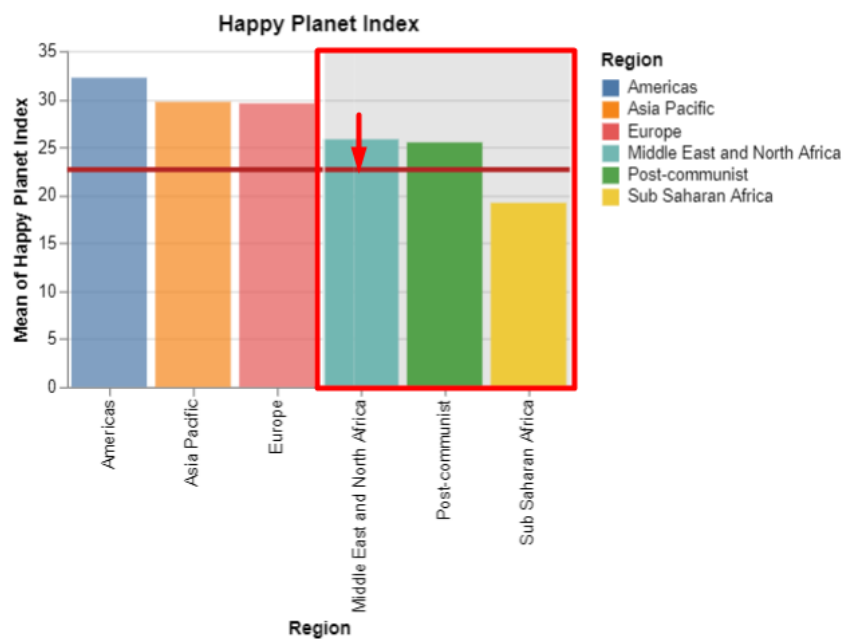
```
225
226 def ex33_Bar(df):
227     columns = df.columns
228     xData = columns[2] + ":N"
229     yData = "mean(" + columns[7] + "):Q"
230     bars = alt.Chart(df, title="Happy Planet Index").mark_bar().encode(
231         x=xData, # x=alt.X('x', axis=alt.Axis(format='% ', title='Wellbeing'))
232         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     #-- Concat 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     #--Concat 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

```
256
257 def ex34_Bar_Selection(df):
258     selectedBars = alt.selection(type="interval", encodings=["x"])
259
260     columns = df.columns
261     xData = columns[2] + ":N"
262     yData = "mean(" + columns[7] + "):Q"
263     bars = alt.Chart(df, title="Happy Planet Index").mark_bar().encode(
264         x=xData, # x=alt.X('x', axis=alt.Axis(format='%', title='Wellbeing'))
265         y=yData, # y=alt.Y('y', axis=alt.Axis(labels=False))
266         opacity=alt.condition(selectedBars, alt.OpacityValue(1), alt.OpacityValue(0.7)),
267         color='Region:N',
268         #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
269         #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
270     ).properties(width=300, height=200)
271     .add_selection(selectedBars)
272     .interactive()
273
274
275     line = alt.Chart(df).mark_rule(color='firebrick').encode(
276         y=yData,
277         size=alt.SizeValue(3)
278     ).transform_filter(selectedBars)
279
280     chart = alt.layer(bars, line, data=df)
281     chart.show()
282
```

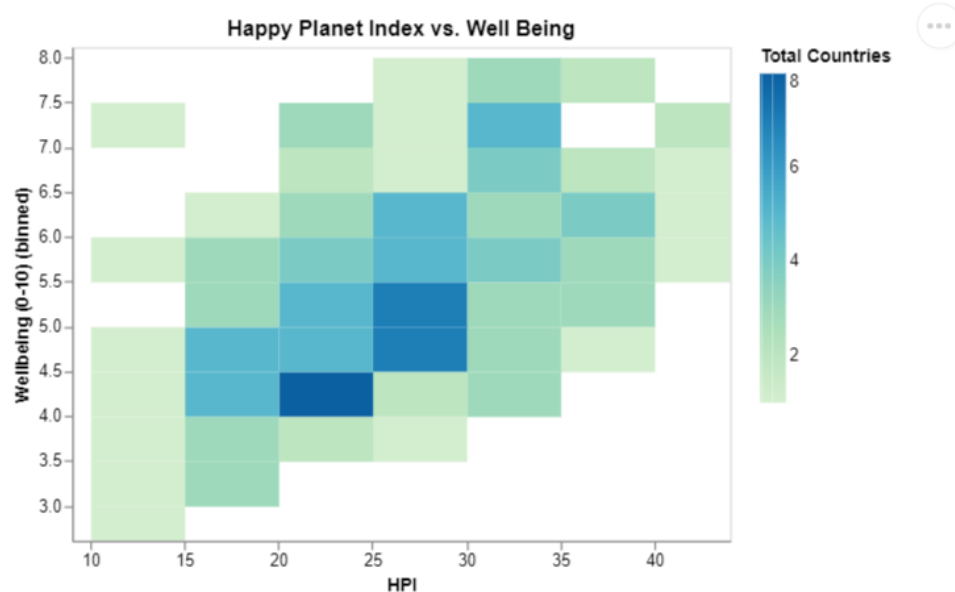
選擇區域數值較低 線也較往下



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

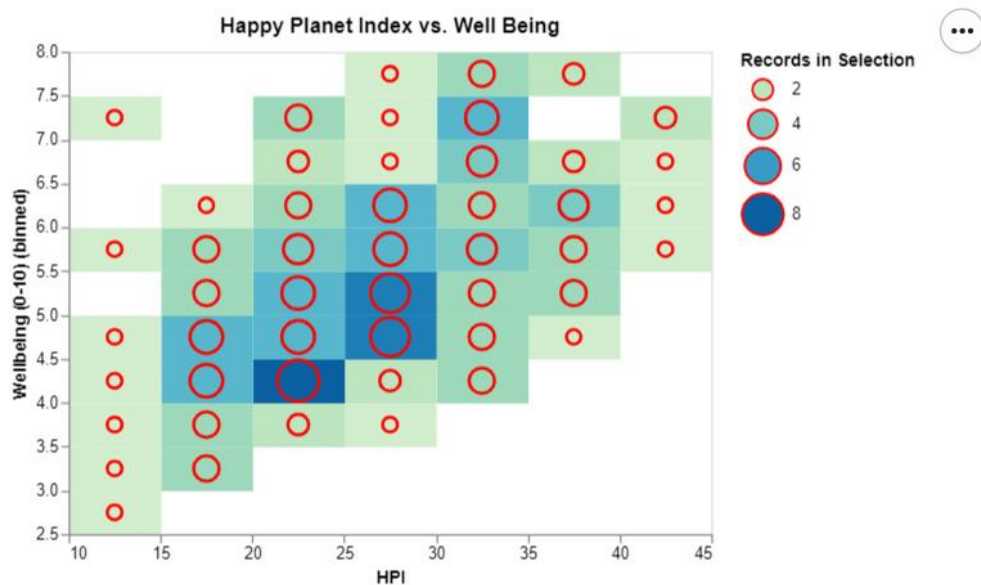


Ex35 熱度圖上面有圈圈代表

```

298
299 def ex35_Heatmap_Circle(df):
300     #-- chart 1
301     columns = df.columns
302     xData = columns[7] + ":Q"
303     yData = columns[4] + ":Q"
304     heatmap = alt.Chart(df, title="Happy Planet Index vs. Well Being").mark_rect().encode(
305         x=alt.X(xData, title="HPI", bin=True),
306         y=alt.Y(yData, bin=True),
307         color=alt.Color('count()', scale=alt.Scale(scheme="greenblue")),
308         #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
309         #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
310     ).properties(width=400, height=300)
311
312     #-- chart2
313     circles = heatmap.mark_point().encode(
314         alt.ColorValue("red"),
315         alt.Size("count()", legend=alt.Legend(title="Records in Selection"))
316     ).interactive()
317
318     #--overlap
319     chart = alt.layer(heatmap, circles)
320     chart.show()
321

```

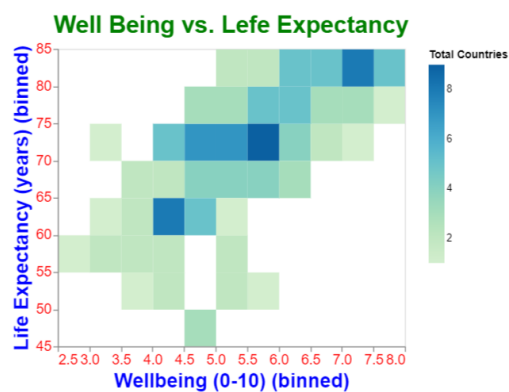
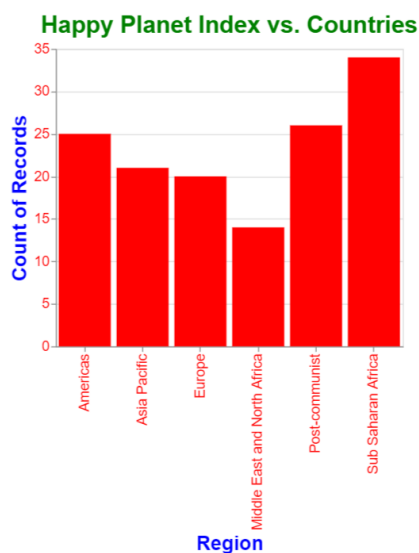


Ex36 bar & heatmap

```

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()

```

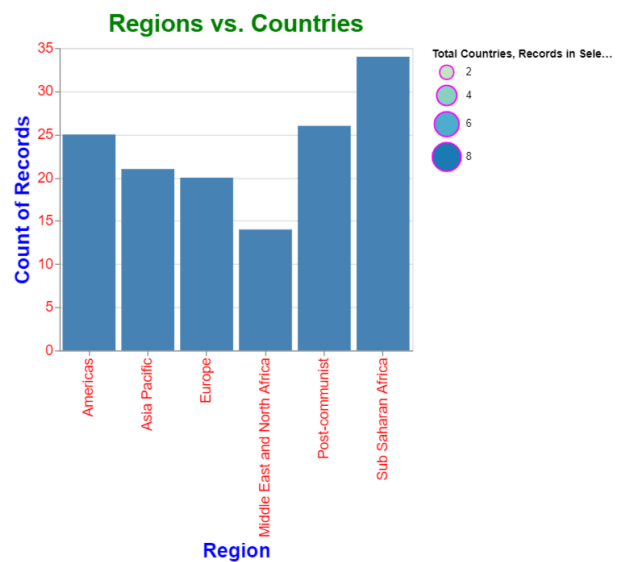
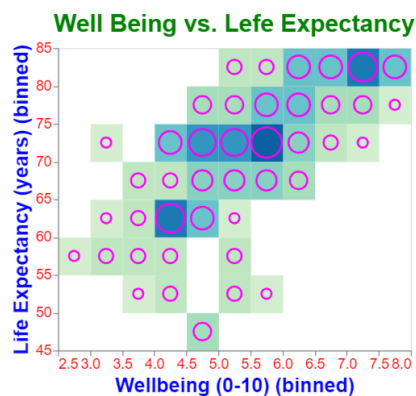


Ex37 bar heatmap bar 在右

```

352
353 def ex37_Bar_Link_Heatmap(df):
354     selectedRegion = alt.selection(type="interval", encodings=["x"])
355
356     #-- chart 1
357     bars = alt.Chart(df, title="Regions vs. Countries").mark_bar().encode(
358         x="Region:N",
359         y="count():Q",
360         color=alt.condition(selectedRegion,
361                             alt.ColorValue("steelblue"), alt.ColorValue("grey"))
362     ).properties(width=350, height=300)
363     .add_selection(selectedRegion)
364     .interactive()
365
366     #-- chart 2
367     columns = df.columns
368     xData = columns[4] + ":Q"
369     yData = columns[3] + ":Q"
370     heatmap = alt.Chart(df, title="Well Being vs. Life Expectancy").mark_rect().encode(
371         x=alt.X(xData, bin=True),
372         y=alt.Y(yData, bin=True),
373         color=alt.Color('count()', scale=alt.Scale(scheme="greenblue"),
374                       legend=alt.Legend(title="Total Countries")),
375         #color=alt.condition(selectedArea, "Region:N", alt.value("lightgray")),
376         #tooltip=["Country", "Region", "Wellbeing(0~10):Q", "Happy Planet Index", "Life Expectancy(years):Q"]
377     ).properties(width=350, height=300)
378
379     circles = heatmap.mark_point().encode(
380         alt.ColorValue('magenta'),
381         alt.Size('count()', legend=alt.Legend(title="Records in Selection"))
382     ).interactive()
383
384     heatmapCircle = alt.layer(heatmap, circles)
385
386     #-- overlap
387     chart = alt.hconcat(heatmapCircle, bars, spacing=50, title="",
388                        ).configure_title(color='green', fontSize=24, align="center", anchor="middle")
389     .configure_axis(labelFontSize=14,
390                    labelColor="red",
391                    titleFontSize=20,
392                    titleColor="blue",
393                    )
394     chart.show()

```

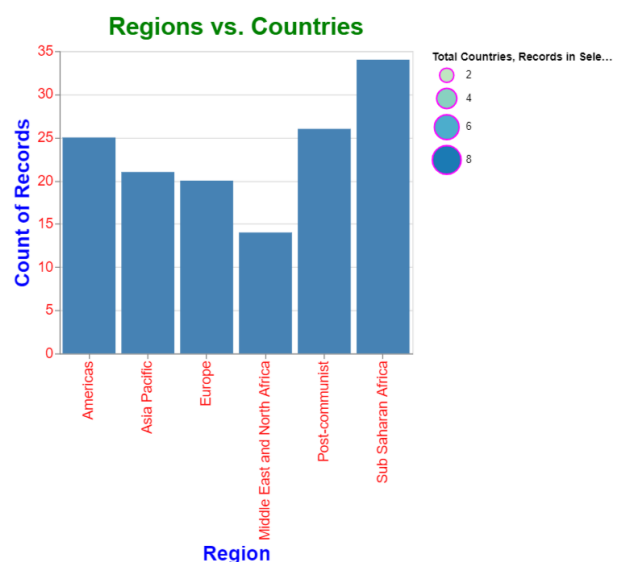
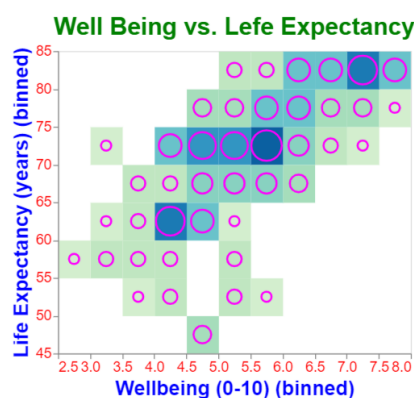


L4_Activity Google Play Store Rating

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

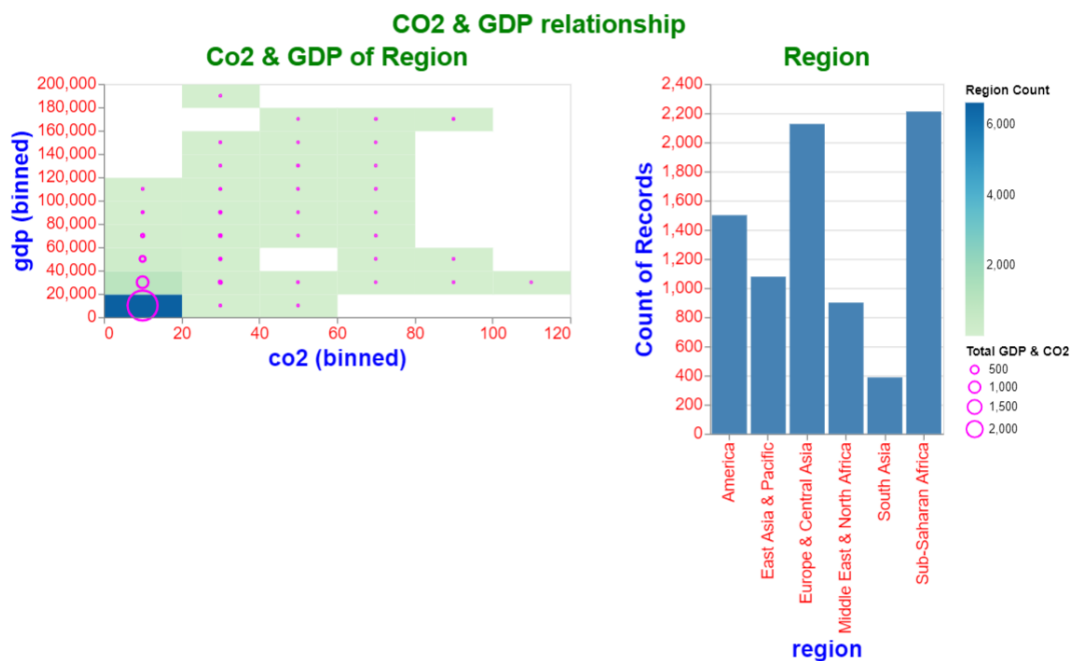
```
34 if (__name__ == "__main__"):
35     df = l4_dataset(InputPath + "/googleplaystore.csv")
36
37     gps_apps_df = df.dropna()
38
39     selectedCategory = alt.selection(type="single", encodings=['x'])
40
41     alt.data_transformers.enable('default', max_rows=None)
42     bars = alt.Chart(gps_apps_df, title="Content Rating").mark_bar().encode(
43         x='Content_Rating:N',
44         y='count():Q',
45         color=alt.condition(selectedCategory, alt.ColorValue("steelblue"), alt.ColorValue("grey"))
46     ).properties(width=100, height=100)
47     .add_selection(selectedCategory)
48
49     #-- Heatmap indicating number of apps across app Category and Rating ranges.
50     heatmap = alt.Chart(gps_apps_df, title="Rating of Category").mark_rect().encode(
51         alt.X('Category:N'),
52         alt.Y('Rating:Q', bin=True),
53         alt.Color('count()'),
54         scale=alt.Scale(scheme='greenblue'),
55         legend=alt.Legend(title='Total Apps')
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, 600), range=(1, 200))),
63         legend=alt.Legend(title='Apps in Selection')
64     ).transform_filter(selectedCategory)
65
66     heatmapCircles = alt.layer(heatmap, circles)
67
68     chart = alt.hconcat(heatmapCircles, bars, spacing=50, title="Google Play Store Apps Rating")
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()
76
```



HW Co2 & gdp

紅色框為與 Google app 不同之處

```
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()
```

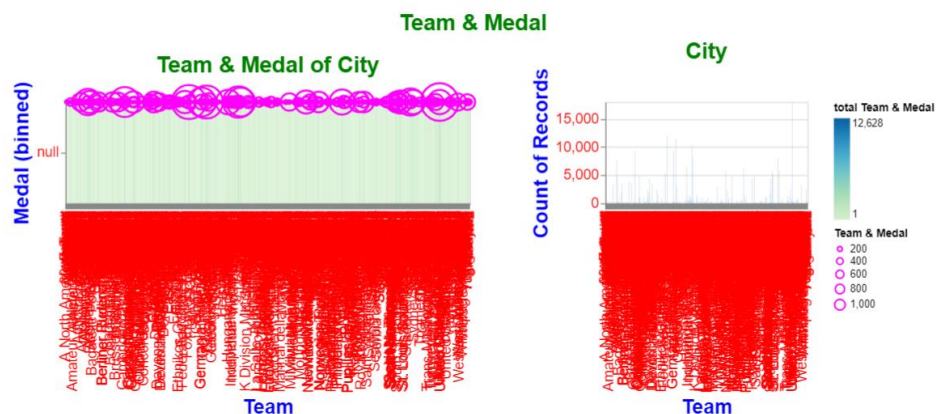


HW Olympics

紅色框為與 Google app 不同之處

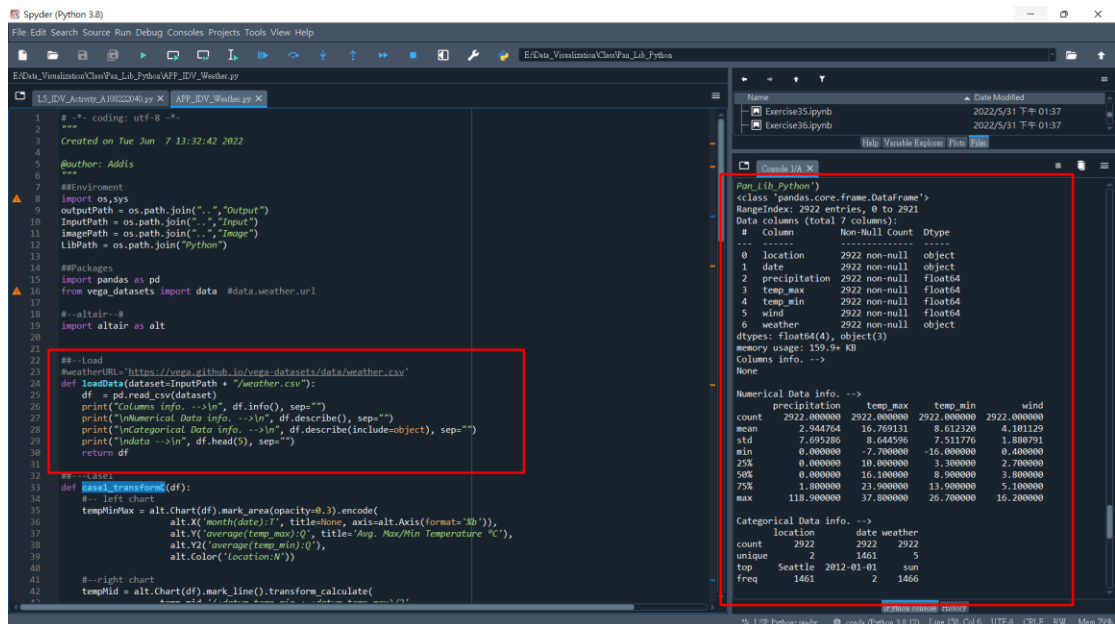
```
33
34 if (__name__ == "__main__"):
35     df = 14_dataset(InputPath + "/athlete_events.csv")
36
37     medal_df = df.dropna(how='all')
38
39     selectedCategory = alt.selection(type="single", encodings=['x'])
40
41     alt.data_transformers.enable('default', max_rows=None)
42     bars = alt.Chart(medal_df, 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



The screenshot shows the Spyder Python IDE interface. The main editor displays a Python script that loads a weather dataset from a CSV file and creates two charts. A red box highlights the data loading and initial processing code.

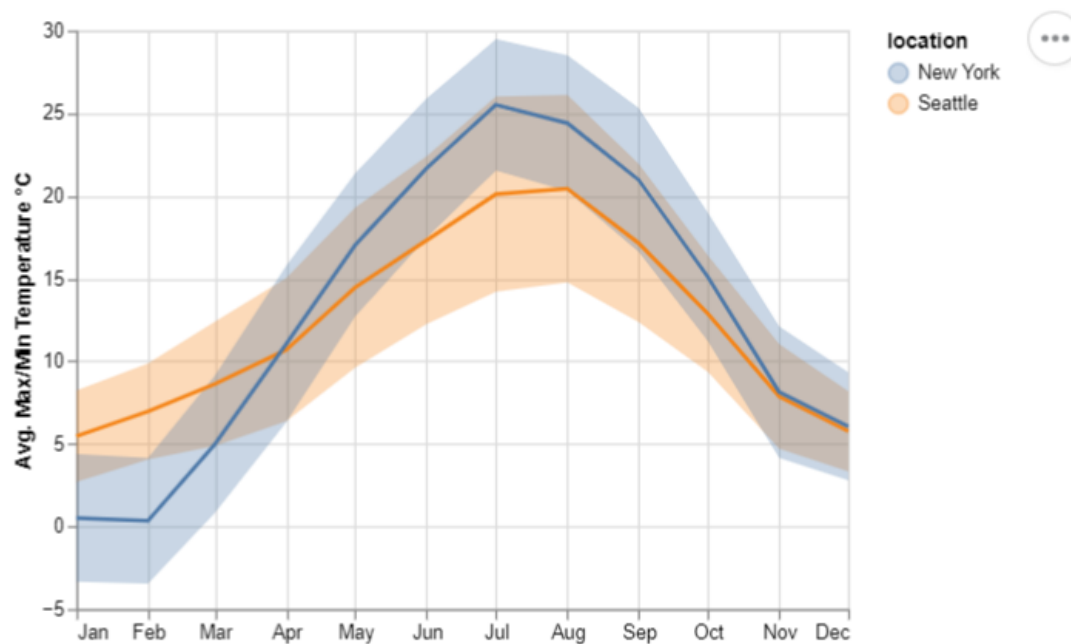
```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Jun 7 13:32:42 2022
4
5 @author: Addis
6 """
7 #Environment
8 import os,sys
9 outputPath = os.path.join(".", "Output")
10 InputPath = os.path.join(".", "Input")
11 ImagePath = os.path.join(".", "Image")
12 LibPath = os.path.join("Python")
13
14 #Packages
15 import pandas as pd
16 from vega_datasets import data #data.weather.url
17
18 #-altair-#
19 import altair as alt
20
21
22
23 #--Load
24 #weatherURL="https://vega.github.io/vega-datasets/data/weather.csv"
25 def loadData(dataset=InputPath + "/weather.csv"):
26     df = pd.read_csv(dataset)
27     print("Columns info. -->|n", df.info(), sep="")
28     print("Numerical Data info. -->|n", df.describe(), sep="")
29     print("Categorical Data info. -->|n", df.describe(include=object), sep="")
30     print("Index info. -->|n", df.head(5), sep="")
31     return df
32
33 #--Case1
34 def basicTransform(df):
35     #-- left chart
36     tempMinMax = alt.Chart(df).mark_area(opacity=0.3).encode(
37         alt.X('month(date):T', title=None, axis=alt.Axis(format='%b')),
38         alt.Y('average(temp_max):Q', title='Avg. Max/Min Temperature °C'),
39         alt.Y2('average(temp_min):Q'),
40         alt.Color('location:N')
41     )
42     #--right chart
43     tempMid = alt.Chart(df).mark_line().transform_calculate(
44         temp_mid='(temp_max+temp_min)/2'
45     ).encode(
46         alt.X('month(date):T', title=None, axis=alt.Axis(format='%b')),
47         alt.Y('temp_mid:Q', title='Avg. Temperature °C'),
48         alt.Color('location:N')
49     )
50     return tempMinMax, tempMid
```

The right sidebar shows the variable explorer and the console output. The console output displays the data structure and summary statistics for the loaded dataset.

```
Run (ipython)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2922 entries, 0 to 2921
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  --
0   location    2922 non-null   object
1   date        2922 non-null   object
2   precipitation 2922 non-null   float64
3   temp_max    2922 non-null   float64
4   temp_min    2922 non-null   float64
5   wind        2922 non-null   float64
6   weather     2922 non-null   object
dtypes: float64(4), object(3)
memory usage: 159.9+ KB
Columns info. -->
None
Numerical Data info. -->
precipitation    temp_max    temp_min    wind
count  2922.000000  2922.000000  2922.000000  2922.000000
mean     2.944764    16.769131    8.612320    4.101129
std       7.695286     8.644596     7.511776     1.880791
min        0.000000    -7.700000   -16.000000    0.400000
25%        0.000000    10.000000    3.300000     2.700000
50%        0.000000    16.100000    8.300000     3.800000
75%        1.800000    23.800000   13.900000     5.100000
max       118.900000   37.800000   26.700000    16.200000
Categorical Data info. -->
location    date    weather
count      2922    2922    2922
unique        2    1461        5
top    Seattle  2012-01-01    sun
freq       1461        2    1466
```

Case1 New York Seattle 平均高低溫

```
32  ##---Case1
33  def case1_transformC(df):
34      #-- left chart
35      tempMinMax = alt.Chart(df).mark_area(opacity=0.3).encode(
36          alt.X('month(date):T', title=None, axis=alt.Axis(format='%b')),
37          alt.Y('average(temp_max):Q', title='Avg. Max/Min Temperature °C'),
38          alt.Y2('average(temp_min):Q'),
39          alt.Color('Location:N'))
40
41      #--right chart
42      tempMid = alt.Chart(df).mark_line().transform_calculate(
43          temp_mid='(+datum.temp_min + +datum.temp_max)/2'
44      ).encode(alt.X('month(date):T'),
45          alt.Y('average(temp_mid):Q'),
46          alt.Color('Location:N'))
47
48      #-- tempMinMax + tempMid
49      chart = alt.layer(tempMinMax, tempMid)
50      charts = alt.hconcat(tempMinMax, tempMid, spacing=50, title="Weather", center=True,
51          ).configure_title(color='green', fontSize=24, anchor="middle", align="right",
52          ).configure_axis(labelFontSize=14,
53          labelColor="red",
54          titleFontSize=20,
55          titleColor="blue",
56          )
57      chart.show()
58
```

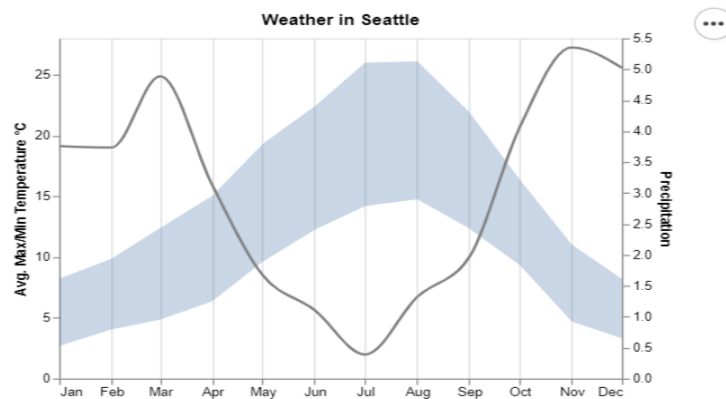
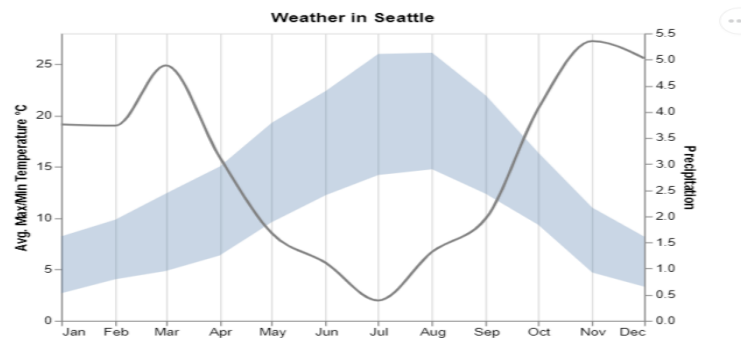


Case2 Seattle

```

57 chart.show()
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()
89

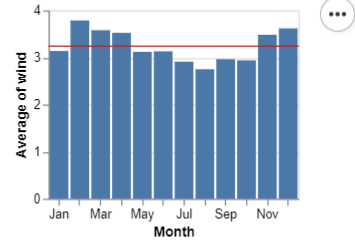
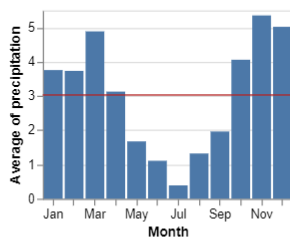
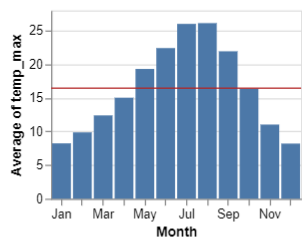
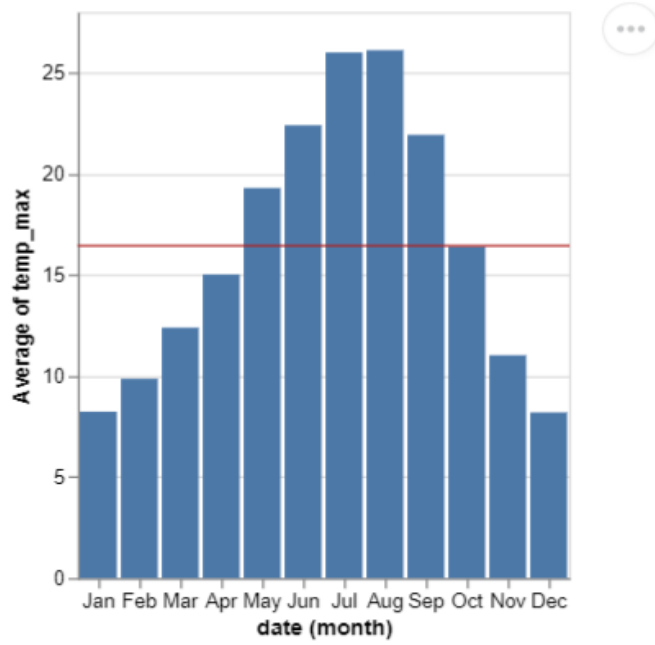
```



Case6 綜合

```
L5_IDV_Activity_A108222040.py X APP_IDV_Weather.py X
93 #Q1
94 def case6_composition(df):
95     #-- method 1
96     basic1 = alt.Chart(df).transform_filter('datum.Location == "Seattle"')
97     basic1.mark_bar()
98     basic1.encode(alt.X('month(date):0', title='Month'), alt.Y('average(temp_max):Q'))
99     basic2 = alt.Chart(df).transform_filter('datum.Location == "Seattle"')
100     basic2.mark_rule(stroke='firebrick')
101     basic2.encode(alt.Y('average(temp_max):Q'))
102
103     chart = alt.layer(basic1, basic2)
104     chart.show()
105
106     #-- method 2
107     chart = alt.layer(alt.Chart().mark_bar().encode(
108         alt.X('month(date):0', title='Month'),
109         alt.Y(alt.repeat('column'), aggregate='average', type='quantitative')),
110         alt.Chart().mark_rule(stroke='firebrick').encode(
111             alt.Y(alt.repeat('column'), aggregate='average', type='quantitative'))
112         ).properties(width=200, height=150)
113         ).repeat(data=df, column=['temp_max', 'precipitation', 'wind'])
114         ).transform_filter('datum.Location == "Seattle"')
115
116     chart.show()
117
118     #-- method 3
119     splom = alt.Chart().mark_point(filled=True, size=15, opacity=0.5)
120     splom.encode(alt.X(alt.repeat('column'), type='quantitative'),
121                 alt.Y(alt.repeat('row'), type='quantitative'))
122     splom.properties(width=125, height=125)
123     splom.repeat(row=['temp_max', 'precipitation', 'wind'],
124                 column=['wind', 'precipitation', 'temp_max'])
125
126     dateHist = alt.layer(alt.Chart(df).mark_bar().encode(
127         alt.X('month(date):0', title='Month'),
128         alt.Y(alt.repeat('row'), aggregate='average', type='quantitative')),
129         #alt.Color('month(date):0', scale = alt.Scale(range=['#aec7e8', 'c7c7c7', '#1f77b4', '#9467bd',
130         alt.Chart().mark_rule(stroke='firebrick').encode(
131             alt.Y(alt.repeat('row'), aggregate='average', type='quantitative'))
132             ).properties(width=175, height=125)
133             ).repeat(row=['temp_max', 'precipitation', 'wind'])
134
135     tempHist = alt.Chart(df).mark_bar(
```

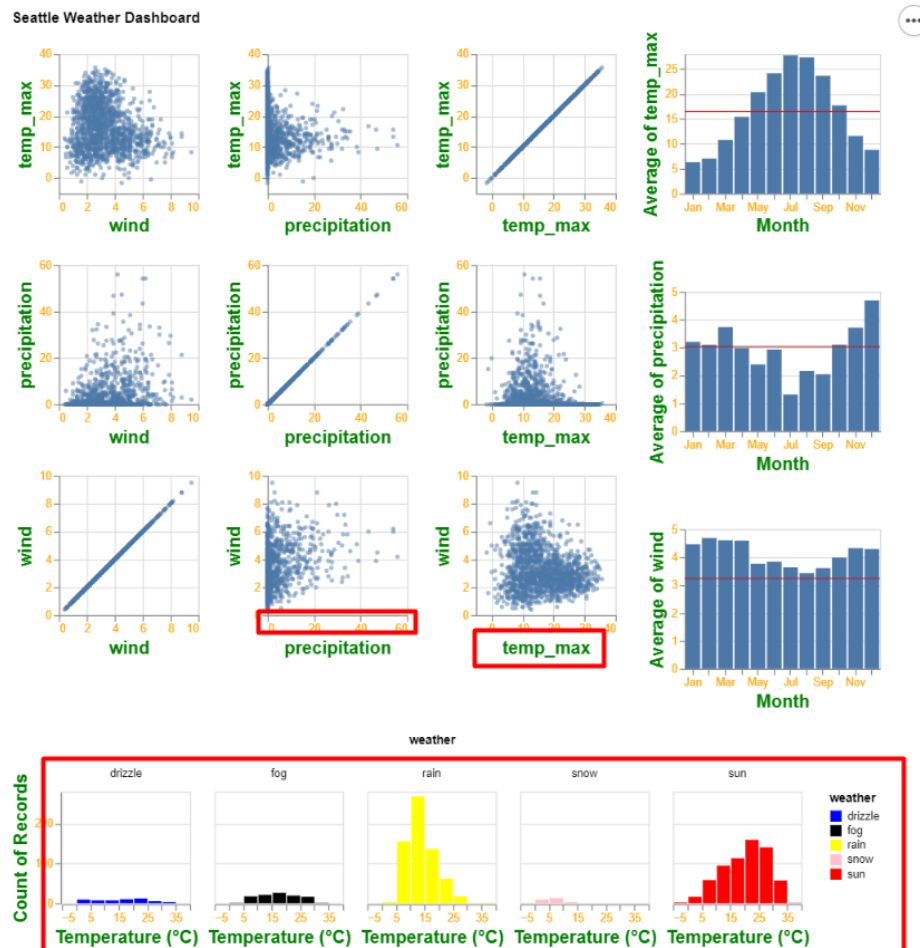
```
L5_IDV_Activity_A108222040.py X APP_IDV_Weather.py X
120         alt.Y(alt.repeat('row'), type='quantitative')
121         ).properties(width=125, height=125)
122         ).repeat(row=['temp_max', 'precipitation', 'wind'],
123                 column=['wind', 'precipitation', 'temp_max'])
124
125
126     dateHist = alt.layer(alt.Chart(df).mark_bar().encode(
127         alt.X('month(date):0', title='Month'),
128         alt.Y(alt.repeat('row'), aggregate='average', type='quantitative')),
129         #alt.Color('month(date):0', scale = alt.Scale(range=['#aec7e8', 'c7c7c7', '#1f77b4', '#9467bd',
130         alt.Chart().mark_rule(stroke='firebrick').encode(
131             alt.Y(alt.repeat('row'), aggregate='average', type='quantitative'))
132             ).properties(width=175, height=125)
133             ).repeat(row=['temp_max', 'precipitation', 'wind'])
134
135     tempHist = alt.Chart(df).mark_bar(
136         ).encode(alt.X('temp_max:Q', bin=True, title='Temperature (°C)'),
137                 alt.Y('count():Q'),
138                 alt.Color('weather:N', scale=alt.Scale(
139                     domain=['drizzle', 'fog', 'rain', 'snow', 'sun'],
140                     range=['blue', 'black', 'yellow', 'pink', 'red'])))
141     tempHist.properties(width=115, height=100)
142     tempHist.facet(column='weather:N')
143
144     chart = alt.vconcat(alt.hconcat(splom, dateHist),
145                         tempHist, data=df, title='Seattle Weather Dashboard', center=True)
146     chart.transform_filter('datum.Location == "Seattle"')
147     chart.resolve_legend(color='independent')
148     chart.configure_axis(
149         labelAngle=0,
150         labelFontSize=10,
151         labelColor='orange',
152         titleFontSize=16,
153         titleColor='green',)
154
155     chart.show()
156
157 if (__name__ == "__main__"):
158     df = loadData()
159     #case1_transformC(df)
160     #case2_transformF(df)
161     case6_composition(df)
```



有改變 x y 軸的字顏色

刻度也有改變

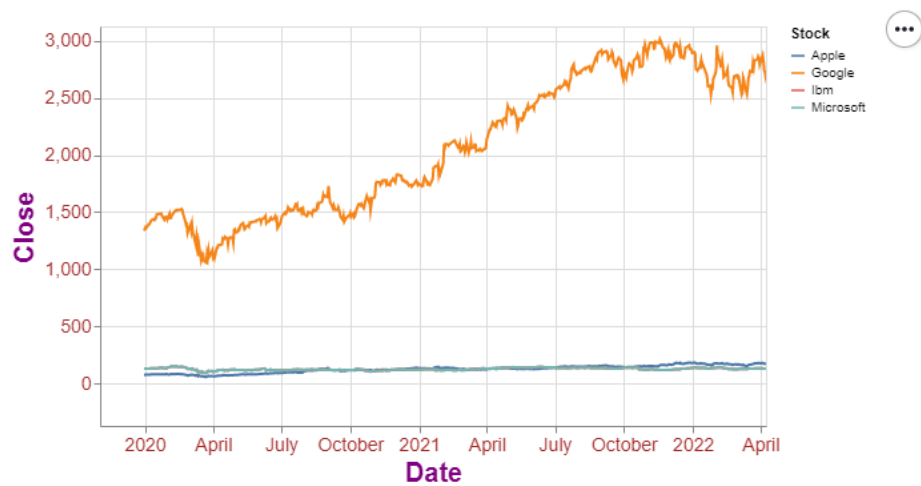
最下面每個圖的 bar 都有改變



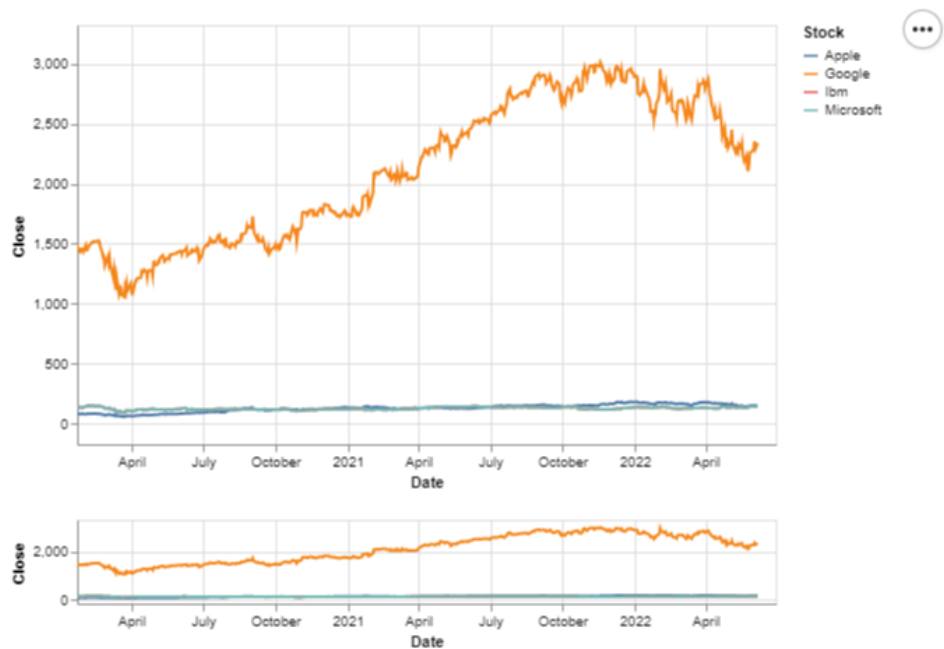
Stock

```
L3_IDV_Activity_A108222040.py X
37 def getStocks():
38     #-- get data
39     startDate = "2020-1-1"
40     endDate = "2022-6-6"
41     source = "yahoo"
42     #-- get data
43     appleStock = pdData.DataReader("AAPL", start=startDate, end=endDate, data_source=source
44                                     ).reset_index()[["Date", "Close", "High"]]
45     ibmStock = pdData.DataReader("IBM", start=startDate, end=endDate, data_source=source
46                                 ).reset_index()[["Date", "Close", "High"]]
47     googleStock = pdData.DataReader("GOOG", start=startDate, end=endDate, data_source=source
48                                     ).reset_index()[["Date", "Close", "High"]]
49     microsoftStock = pdData.DataReader("IBM", start=startDate, end=endDate, data_source=source
50                                     ).reset_index()[["Date", "Close", "High"]]
51
52     #--save data
53     appleStock.to_csv(InputPath + "/AppleStock_20200101_20220606.csv")
54     ibmStock.to_csv(InputPath + "/IbmStock_20200101_20220606.csv")
55     googleStock.to_csv(InputPath + "/GoogleStock_20200101_20220606.csv")
56     microsoftStock.to_csv(InputPath + "/MicrosfotStock_20200101_20220606.csv")
57
58     #--process data
59     appleStock["Stock"] = "Apple"
60     ibmStock["Stock"] = "Ibm"
61     googleStock["Stock"] = "Google"
62     microsoftStock["Stock"] = "Microsoft"
63
64     stocks = pd.concat([appleStock, ibmStock, googleStock, microsoftStock])
65     stocks["Month"] = stocks.Date.dt.month
66     stocks["Year"] = stocks.Date.dt.year
67     stocks["Day"] = stocks.Date.dt.day
68     stocks.to_csv(InputPath + "/stocks_20200101_20220606.csv")
69
70     return stocks
71
```

```
70     return stocks
71
72 def IDV_Altair_Stocks(stocks):
73     stockSelection1 = alt.selection_single(fields=["Stock"], bind="Legend")
74
75     #--rightChart
76     rightChart = alt.Chart(stocks).mark_line().encode(
77         x="Date", y="Close", color="Stock",
78         opacity=alt.condition(stockSelection1, alt.value(1), alt.value(0.1)),
79         tooltip=["Date", "Close"] #hover
80     ).properties(height=300, width=500)
81     .configure_title(color="green", fontSize=24)
82     .configure_axis(labelFontSize=14,
83                     labelColor="brown",
84                     titleFontSize=20,
85                     titleColor="purple")
86     .add_selection(stockSelection1)
87     .interactive()
88
89     rightChart.show()
90
91
92     #-- left chart
93     stockSelection1 = alt.selection_single(fields=["Stock"], bind="Legend")
94
95     rightChart = alt.Chart(stocks).mark_line().encode(
96         x="Date", y="Close", color="Stock",
97         opacity=alt.condition(stockSelection1, alt.value(1), alt.value(0.1)),
98         tooltip=["Date", "Close"] #hover
99     ).properties(height=300, width=500)
100     .add_selection(stockSelection1)
101     .interactive()
102
103     upper = rightChart.encode(alt.X('Date:T', scale=alt.Scale(domain=stockSelection1)))
104
105     lower = rightChart.properties(height=60).add_selection(stockSelection1)
106
107     leftChart = upper & lower
108     leftChart.show()
109
```

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
leftChart.display()
```

