PBC

解析合購版

B06302201 陳品儒　B07303050 周鈺淇

B07701236 柯昱丞　B09610020 黃韻文

B09702032 林子昕　P09323028 吳金擇

CONTENTS

01

PART ONE

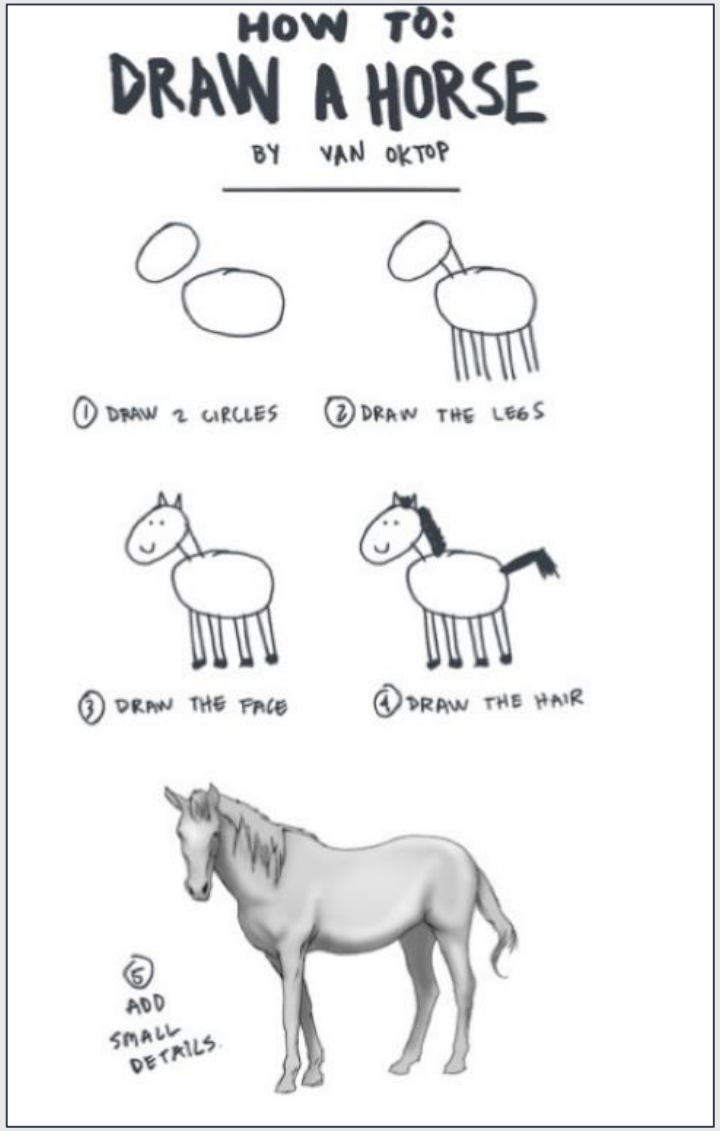
專題理念

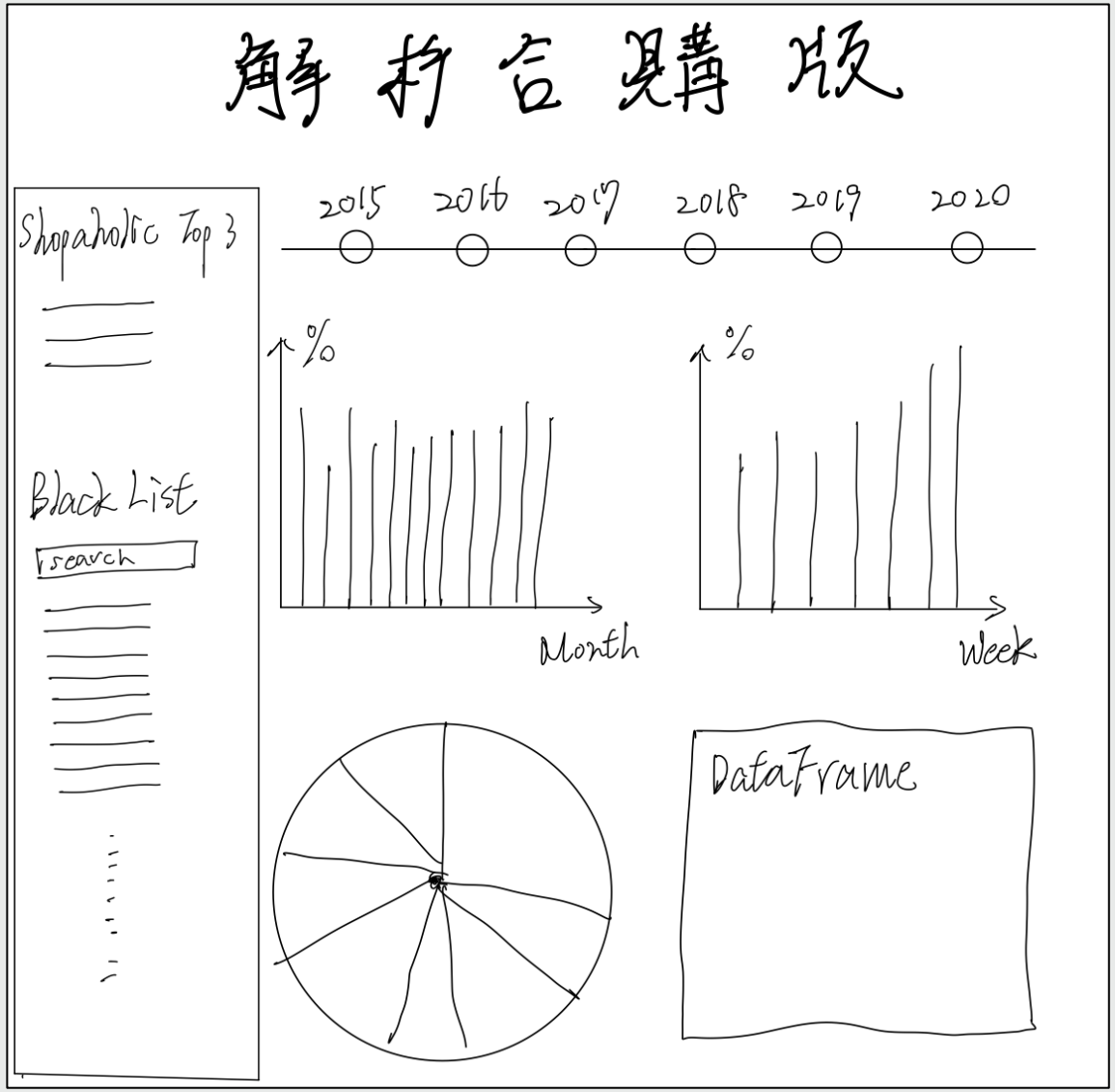
專題理念

PART ONE

專題理念

PART ONE





02

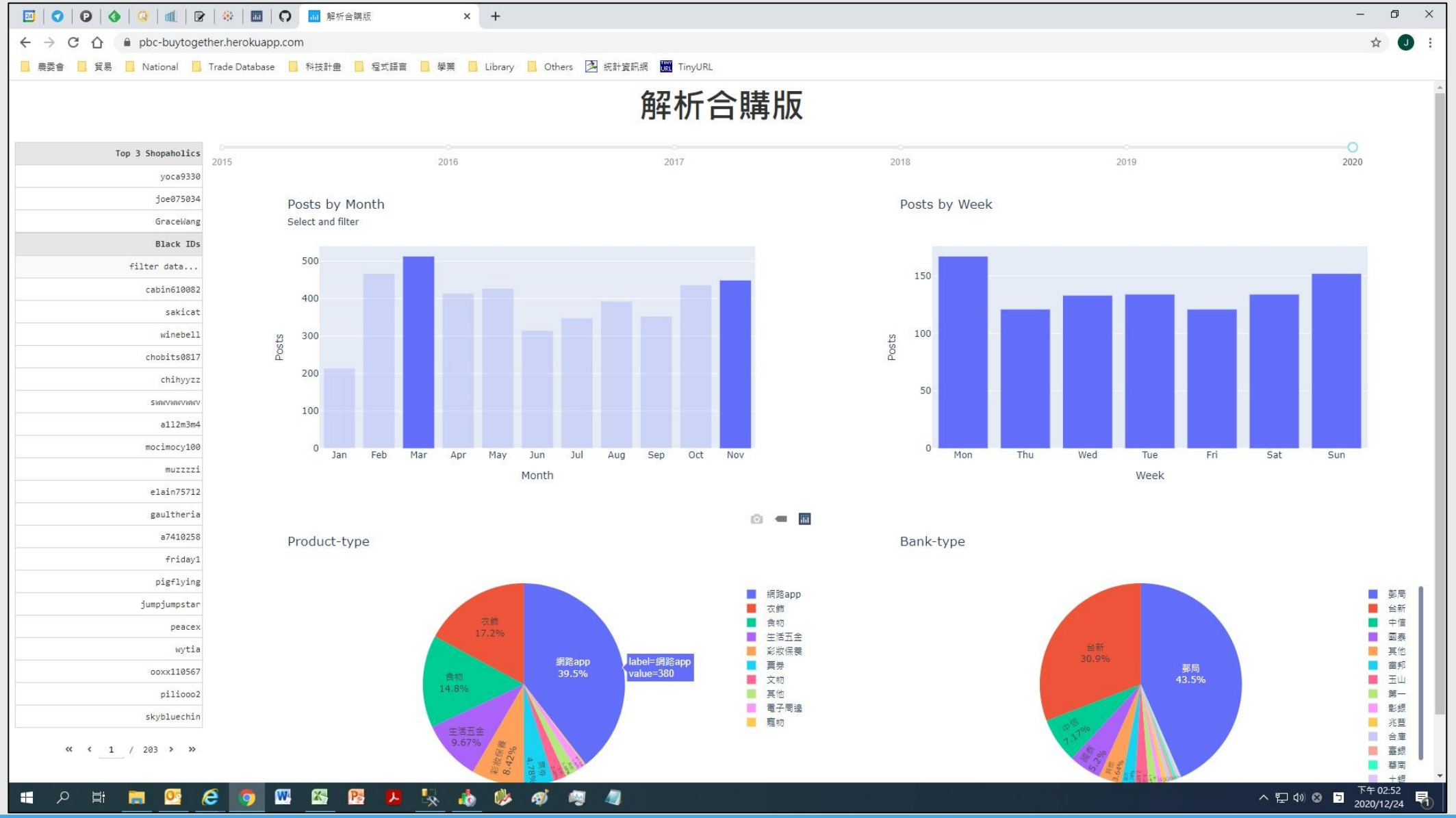
PART TWO

網頁簡介

https://pbc-buytogether.herokuapp.com/

網頁簡介

PART TWO



03

PART THREE

程式簡介

https://github.com/PinJu-Chen/PTT-Analysis

程式簡介

PART THREE

01 爬取資料

02 資料整理

03 製作圖表

04 網頁部署

程式簡介 - 01 爬取資料

PTT crawler.py | data_manipulation.py | app.py

```python
#coding=utf-8
import re
import sys
import json
import requests
import time
from datetime import datetime
from bs4 import BeautifulSoup
```

```python
def store(data):
    with open(fileName, 'a', encoding="utf-8") as f:
        f.write(data.encode(sys.stdin.encoding, "replace").decode(sys.stdin.encoding))

if __name__ == "__main__":
    PttName, ParsingPage = 'BuyTogether', int(3999)
    start_time = time.time()
    print('Start parsing ' + PttName + '....')
    fileName = 'D:\PCB\data-' + PttName + '-' + datetime.now().strftime('%Y%m%d%H%M%S') + '.json'
    # 檢查看板是否為18禁,有些看板為18禁
    soup = over18(PttName)
    ALLpageURL = soup.select('.btn.wide')[1]['href']
    # 得到本看板全部的index數量
    ALLpage = int(getPageNumber(ALLpageURL)) + 1
    index_list = []
    for index in range(ALLpage, ALLpage - int(ParsingPage), -1):
        page_url = 'https://www.ptt.cc/bbs/' + PttName + '/index' + str(index) + '.html'
        index_list.append(page_url)

    store('[\n')
    crawler(index_list)

    # 移除最後一個 "," 號
    with open(fileName, 'r', encoding="utf-8") as f:
        content = f.read()
```

程式簡介 - 02 資料整理

PTT crawler.py | data_manipulation.py | app.py

```python
# -*- coding: utf-8 -*-
import json
import pandas as pd
# import re

# 讀取json檔
with open('rawdata.json', 'r', encoding="utf-8") as f:
    data = json.loads(f.read())
```

```python
# 以list建立datafram，以方便後續繪圖
buy2df = pd.DataFrame(all_list, columns=['id','author','title','date','ip','likes','content'])
```

```python
# 全改為小寫，以方便對照
df['content'] = df['content'].str.lower()
for i in b.keys():
    for j in range(len(b[i])):
        b[i][j] = b[i][j].lower()

# 建立bank清單，並以長度做排序
bank = []
for i in b.keys():
    for j in range(len(b[i])):
        bank.append((i, b[i][j]))

# 定義銀行函數
def banktype(content):
    for i in range(len(bank)):
            if content.find(bank[i][1]) > 0:
                return bank[i][0]
    return '其他'

# 將分類函數套用至df
df['bank'] = df['content'].apply(lambda x: banktype(x))

df.to_csv(r'rawdata.csv', index=False)
```

程式簡介 - 03 製作圖表

PTT crawler.py | data_manipulation.py | app.py

```python
# -*- coding: utf-8 -*-
import pandas as pd
import plotly.express as px
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import dash_table
```

```python
# 黑名單
blackdf = df[(df.title.str.contains('黑人')|
             df.title.str.contains('灰人')|
             df.title.str.contains('黑名單')|
             df.title.str.contains('判決'))&
             ((df.year == 2020)|
             (df.year == 2019)|
             (df.year == 2018)|
             (df.year == 2017))&
             (~df.title.str.contains('RE:'))&
             (~df.title.str.contains('Re:'))]
blackdf['black_ID'] = blackdf.title.str.replace('^.+?([a-zA-Z0-9]+).+$', r'\1')


'''-------------------main------------------'''
# css
external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

# 建立app
app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
# server = app.server

# 主標
header = html.H1(children="解析合購版",style={'font-weight': 'bold'})
```

程式簡介 - 03 製作圖表


```
57   # Top 3 table
58   top3 = dash_table.DataTable(
68
69   # Blacklist table
70   blacklist = dash_table.DataTable(
83
84   # 時間軸
85   year_slider = dcc.RangeSlider(id='year_slider',
97
98   # 以dcc.Graph建立instance，存放月份圖
99   month_bar = dcc.Graph(id='month_bar',
100                         selectedData=None,
101                         className="five columns")
102
103  # 以dcc.Graph建立instance，存放星期圖
104  week_bar = dcc.Graph(id='week_bar', className="five columns")
105
106  # 以dcc.Graph建立instance，存放商品圓餅圖
107  product_pie = dcc.Graph(id='product_pie', className="five columns")
108
109  # 以dcc.Graph建立instance，存放支付銀行圓餅圖
110  bank_pie = dcc.Graph(id='bank_pie', className="five columns")
```

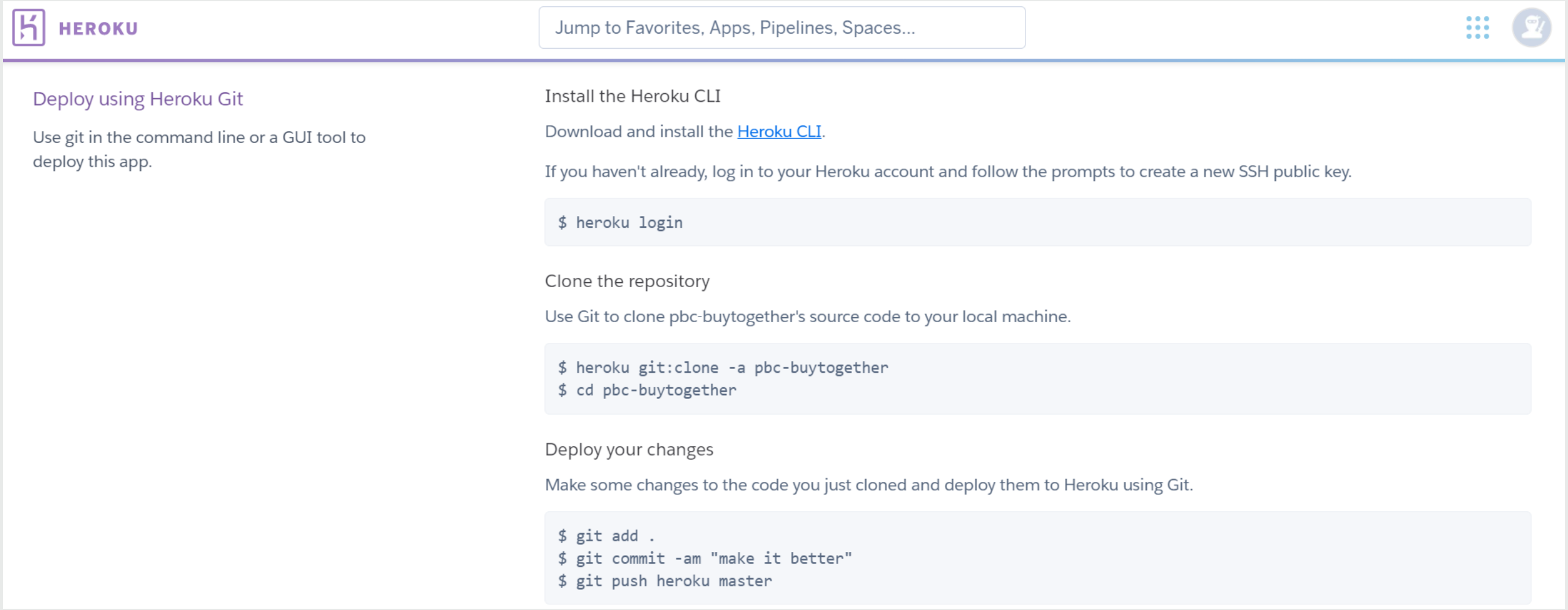
程式簡介 - 03 製作圖表


```
'''-------------------Month callback------------------'''
@app.callback(
    Output('month_bar', 'figure'),
    Output('month_bar', 'selectedData'),
    Input('year_slider', 'value'))
def update_month_bar(selected_year):
    # 年度
    dfm = dff[(dff['year'] <= max(selected_year)) & (dff['year'] >= min(selected_year))]

    # 更新 month_bar 月份圖
    chartm = px.bar(x=dfm.groupby('month').size().index,
                    y=dfm.groupby('month').size(),
                    title="Posts by Month" + "<br>" + '<span style="font-size: 12px;">Select and filter</span>',
                    labels={"x": "Month",
                            "y": "Posts"},
                    category_orders={"x":
                                     ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                                      'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']}
                    )
    # 設定clickmode，作為後續的input
    chartm.update_layout(clickmode='event+select')
    return chartm, None
```

程式簡介 - 04 網頁部署

PART THREE



04

PART FOUR

心得展望

心得展望

PART FOUR

複習
基礎觀念

學習
相關套件

BI
商業智慧

心得展望

PART FOUR

短期

- 修正版面小bug
- 即時抓取資料、即時更新圖表

中期

- 使用NLTK及wordcloud製作文字雲

長期

- 應用於其他資料，製作更多商業智慧分析

THANKS

謝　謝　聆　聽