

Global AI Company

## Deploy App to heroku

System: win10

Editor: Ziyu (Stanley) Zhang

08/2020

Some useful websites:

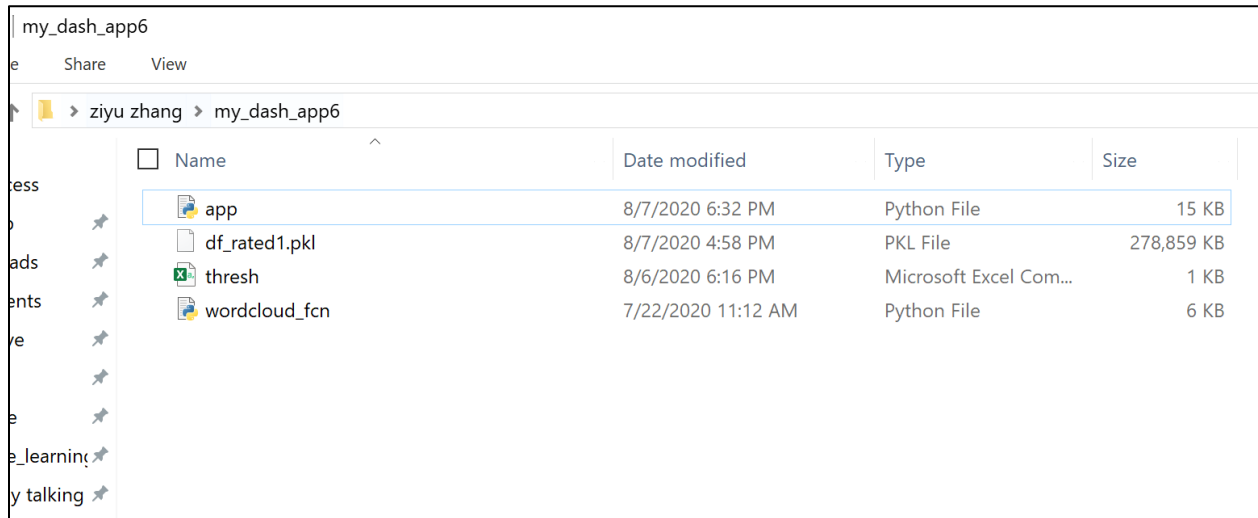
<https://plotly.com/python/reference/>

<https://plotly.com/python/>

<https://community.plotly.com/>

## Deploy an application to heroku, step by step

1. Paste your dashboard codes and all required data files into one folder (we push this folder later to heroku)

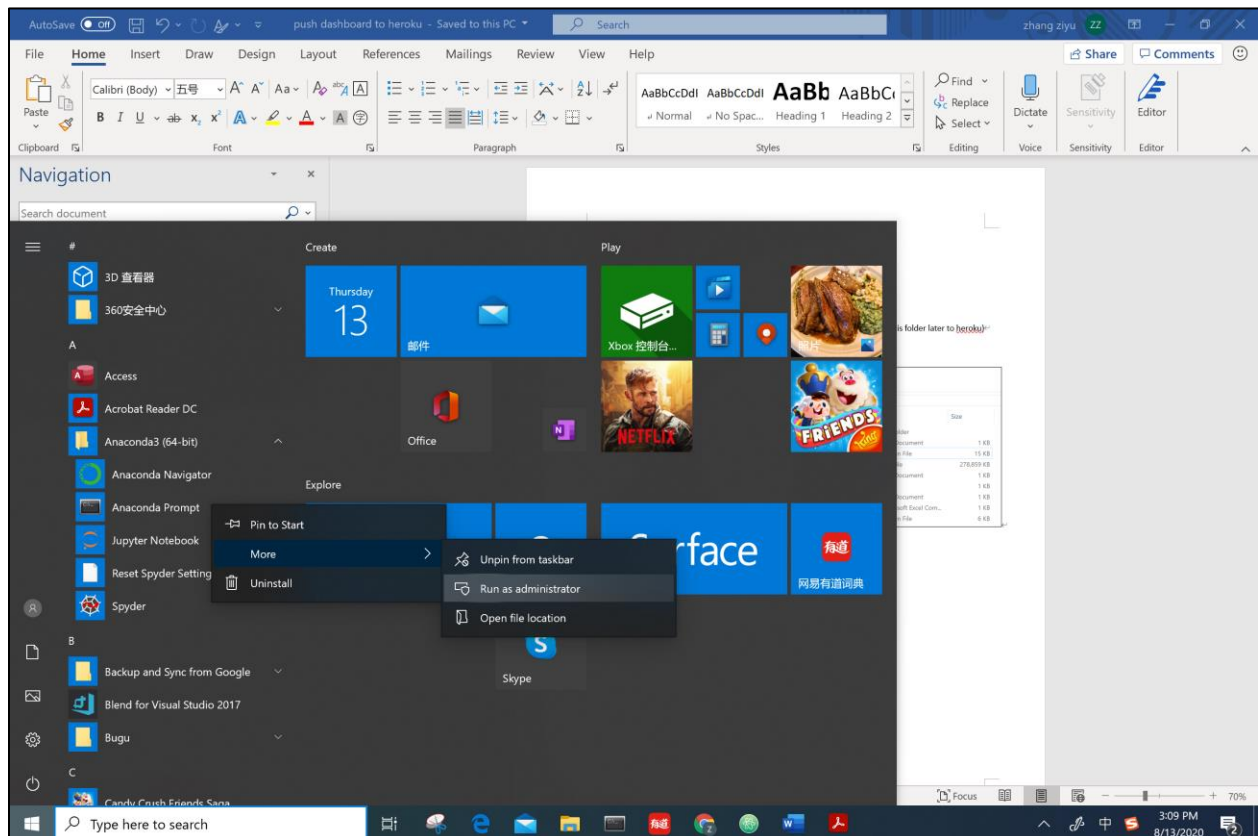


app.py is the dashboard code

wordcloud\_fcn includes some functions we need to import in the app.py (this file is optional, you can also write all functions in the app.py directly)

df Rated1.pkl and thresh.csv are the data files, we will import them in app.py

2. Run you command as administrator and open the folder above



Administrator: Anaconda Prompt

```
(base) C:\WINDOWS\system32>cd C:\Users\zhang\my_dash_app6
```

(I use Anaconda Prompt instead of command here, of course you can use command, procedure is the same)

3. Run the app.py in command

```
(base) C:\Users\zhang\my_dash_app6>python app.py
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\zhang\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]   C:\Users\zhang\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]   C:\Users\zhang\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
127.0.0.1 - - [13/Aug/2020 15:19:57] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Aug/2020 15:19:58] "GET /_dash-layout HTTP/1.1" 200 -
127.0.0.1 - - [13/Aug/2020 15:19:58] "GET /_dash-dependencies HTTP/1.1" 200 -
127.0.0.1 - - [13/Aug/2020 15:19:58] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [13/Aug/2020 15:20:00] "POST /_dash-update-component HTTP/1.1" 200 -
```

If you get a link as above, you paste it into chrome and find everything goes smoothly, go to next step; Otherwise, there is something wrong in your app.py, debug it locally before trying to push it.

4. Ctrl+C to stop your program in command, add other required files into the folder

my\_dash\_app6

ShareView

> ziyu zhang > my\_dash\_app6

<input type="checkbox"/>	Name	Date modified	Type	Size
	__pycache__	8/13/2020 3:03 PM	File folder	
	.gitignore	5/17/2020 9:06 PM	Text Document	1 KB
	app	8/7/2020 6:32 PM	Python File	15 KB
	df Rated1.pkl	8/7/2020 4:58 PM	PKL File	278,859 KB
	nltk	7/17/2020 10:56 PM	Text Document	1 KB
	Procfile	7/22/2020 6:59 PM	File	1 KB
	requirements	7/18/2020 12:33 AM	Text Document	1 KB
	thresh	8/6/2020 6:16 PM	Microsoft Excel Com...	1 KB
	wordcloud_fcn	7/22/2020 11:12 AM	Python File	6 KB

.gitignore: used by git, identifies files that *won't* be pushed to production

Procfile: used for deployment

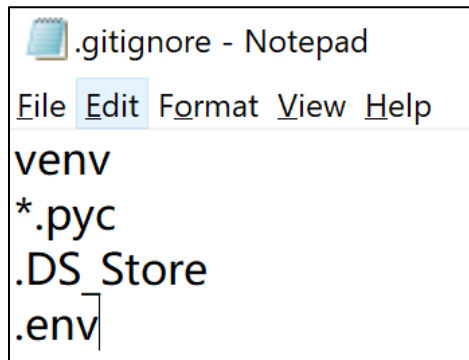
requirements.txt: describes your Python dependencies, can be created automatically

nlk.txt: used in the word cloud project, just skip this if your project doesn't need it (optional)

Check [\*PYTHON DASHBOARDS heroku 手册.pdf\*](#) in the new\_intern\_guide for details about the above files

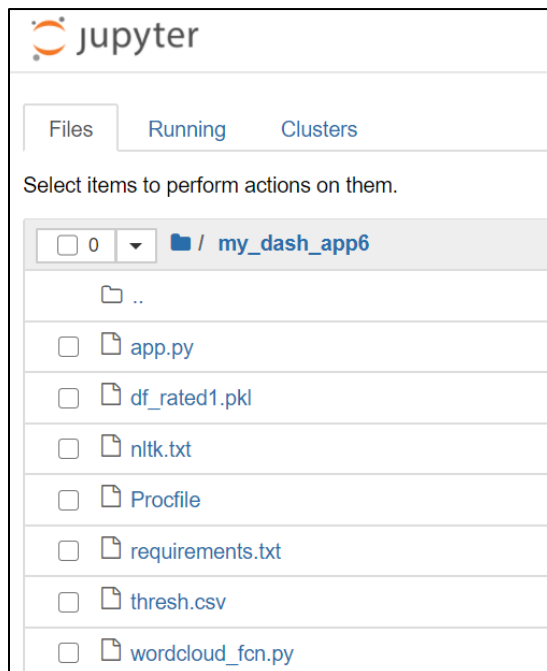
Note:

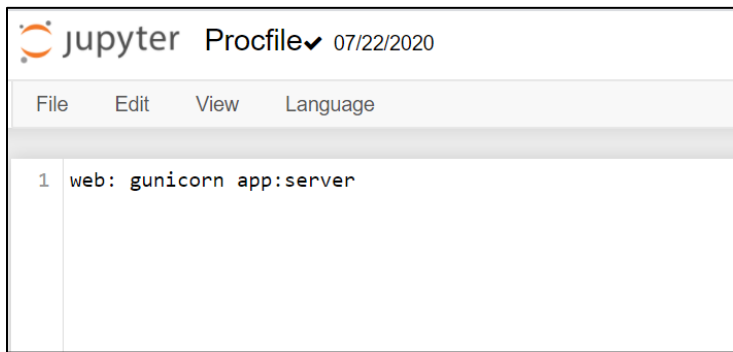
For the .gitignore file, create a new txt file, and then rename it to .gitignore, details as below



For the requirements.txt, it can be generated by running ***pip freeze > requirements.txt*** at command. (if failed, I paste a sample requirements.txt at appendix, you can just copy and paste the content)

For the Procfile, generate a new text file in jupyter notebook (under the desired folder repository), rename it to Procfile (with no suffix) and paste ***web: gunicorn app:server*** into it, as below





5. Create a virtual environment in command (***python -m virtualenv venv***)  
venv is the name of the virtualenv, you can replace it by any other name

```
(base) C:\Users\zhang\my_dash_app6>python -m virtualenv venv
created virtual environment CPython3.6.9.final.0-64 in 1488ms
creator CPython3Windows(dest=C:\Users\zhang\my_dash_app6\venv, clear=
seeder FromAppData(download=False, pip=latest, setuptools=latest,
AppData\Local\pypa\virtualenv\seed-app-data\v1.0.1)
activators BashActivator, BatchActivator, FishActivator, PowerShellA
```

6. Activate the virtual environment (***.\venv\Scripts\activate***)

```
(base) C:\Users\zhang\my_dash_app6>.\venv\Scripts\activate
(venv) (base) C:\Users\zhang\my_dash_app6>
```

If you succeed, there will exist a (venv) before your command

7. Install all required packages and then run app.py in the virtual environment (***python app.py***)  
e.g.: If you missed some required packages, you will get an error as below:

```
(venv) (base) C:\Users\zhang\my_dash_app6>python app.py
Traceback (most recent call last):
  File "app.py", line 3, in <module>
    import pandas as pd
ModuleNotFoundError: No module named 'pandas'
```

Now you should use ***pip install pandas*** to install it in virtual environment

```
(venv) (base) C:\Users\zhang\my_dash_app6>pip install pandas
Collecting pandas
  Using cached pandas-1.1.0-cp36-cp36m-win_amd64.whl (9.4 MB)
Collecting pytz>=2017.2
  Using cached pytz-2020.1-py2.py3-none-any.whl (510 kB)
Collecting numpy>=1.15.4
  Using cached numpy-1.19.1-cp36-cp36m-win_amd64.whl (12.9 MB)
Collecting python-dateutil>=2.7.3
  Using cached python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Collecting six>=1.5
  Using cached six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: pytz, numpy, six, python-dateutil, pandas
Successfully installed numpy-1.19.1 pandas-1.1.0 python-dateutil-2.8.1 pytz-2020.1
WARNING: You are using pip version 20.1; however, version 20.2.2 is available.
You should consider upgrading via the 'C:\Users\zhang\my_dash_app6\venv\Scripts\python.exe -m pip install --upgrade pip' command.
```

If you get an error said 'ModuleNotFoundError: No module named 'dash'', then just do ***pip install dash***, similarly for any other package

When you have installed all required packages in the virtual environment, you can run you app.py in the virtual environment, and it will give you a link, which you can run smoothly (as below)

```
(venv) (base) C:\Users\zhang\my_dash_app6>python app.py
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\zhang\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]   C:\Users\zhang\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]   C:\Users\zhang\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
Dash is running on http://127.0.0.1:8050/

Warning: This is a development server. Do not use app.run_server
in production, use a production WSGI server like gunicorn instead.

* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

8. Login to your heroku account in command (heroku login)

```
(venv) (base) C:\Users\zhang\my_dash_app6>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/7ffea50d-
3QAAAACZAAEZGFOYW0AAAAMNzMuMTk3LjE5NS4xZAAGc2lnbmVkbGZA0X-X6XMB.cn1lVqkAl
Logging in... done
Logged in as richard.v.rothenberg@gmail.com
```

If you succeed logging in, you will see your account here

9. Initialize git repository, create heroku app as below

```
(venv) (base) C:\Users\zhang\my_dash_app6>git init
Reinitialized existing Git repository in C:/Users/zhang/my_dash_app6/.git/

(venv) (base) C:\Users\zhang\my_dash_app6>heroku create gai-wordcloud-app
Creating • gai-wordcloud-app... done
https://gai-wordcloud-app.herokuapp.com/ | https://git.heroku.com/gai-wordcloud-app.git
```

Note: your app name may already be taken, in such circumstance, you need to change to another name

Then, add and commit as below ('q' can be anything, just a comment)

```
(venv) (base) C:\Users\zhang\my_dash_app6>git add .

(venv) (base) C:\Users\zhang\my_dash_app6>git commit -m 'q'
[master (root-commit) 8e368fd] 'q'
 8 files changed, 636 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Procfile
 create mode 100644 app.py
 create mode 100644 dfRated1.pkl
 create mode 100644 nltk.txt
 create mode 100644 requirements.txt
 create mode 100644 thresh.csv
 create mode 100644 wordcloud_fcnn.py
```

Then, push as below

```
(venv) (base) C:\Users\zhang\my_dash_app6>git push heroku master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 50% (5/10), 2.09 MiB | 561.00 KiB/s
```

If everything goes well, you will push successfully and get something like this:

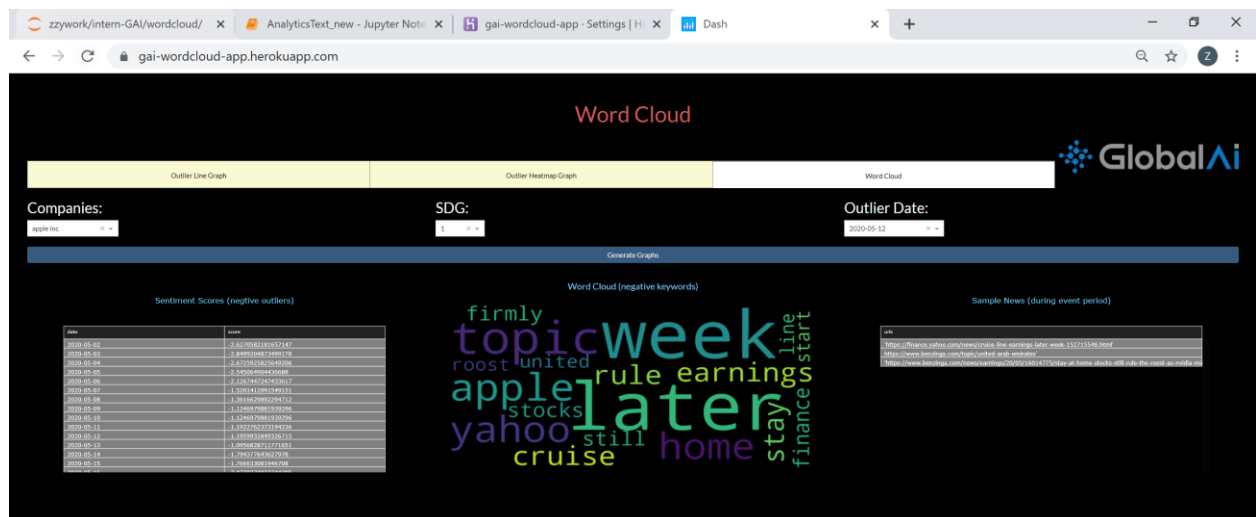
```
remote:
remote: -----> Compressing...
remote: Done: 270.8M
remote: -----> Launching...
remote: Released v3
remote: https://gai-wordcloud-app.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/gai-wordcloud-app.git
 * [new branch]      master -> master
```

(note, here you can see the slug size of your app, it's 270.8M, this number can't be larger than 500MB, otherwise you have to reduce the size of your data files)

Then open your app at command (***heroku open***)

```
(venv) (base) C:\Users\zhang\my_dash_app6>heroku open
(venv) (base) C:\Users\zhang\my_dash_app6>
```

Then check your application at your browser:



Note: when you use mouse to make any change to your application, the web page will first shows 'Updating' and then 'Dash' when it finishes updating. If your graph doesn't show well when the web page finished updating, you need to check errors in command (see below)

### Troubleshooting:

1. If your app doesn't open successfully, you can use command line to check the error (***heroku logs --tail***)



```
(venv) (base) C:\Users\zhang\my_dash_app6>heroku logs --tail
2020-08-13T21:08:07.171714+00:00 app[web.1]: 10.43.250.98 - - [13/Aug/2020
dash_core_components/async-datepicker.v1_10_lm1592438751.js HTTP/1.1" 200
m/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
Edge/18.18362"
2020-08-13T21:08:07.175817+00:00 app[web.1]: 10.93.223.38 - - [13/Aug/2020
dash_core_components/async-graph.v1_10_lm1592438751.js HTTP/1.1" 200 5072
zilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
8.18362"
2020-08-13T21:08:07.178578+00:00 heroku[router]: at=info method=GET path="
```

And check what error you get:

```
sync-plotlyjs.v1_10_lm1592438751.js" host=gai-wordcloud-app.herokuapp.com request_id=86f0b0f1-61dd-42f4
0 fwd="73.197.195.1" dyno=web.1 connect=183ms service=2935ms status=200 bytes=1015036 protocol=https
2020-08-13T21:08:17.376972+00:00 app[web.1]: 10.69.232.242 - - [13/Aug/2020:21:08:17 +0000] "GET / HTTP
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.125
2020-08-13T21:08:17.378929+00:00 heroku[router]: at=info method=GET path="/" host=gai-wordcloud-app.her
t_id=0034a32a-1f06-40b2-8dd2-b12ee0783563 fwd="73.197.195.1" dyno=web.1 connect=148ms service=97ms stat
protocol=https
2020-08-13T21:08:17.788712+00:00 heroku[web.1]: Process running mem=924M(180.5%)
2020-08-13T21:08:17.792397+00:00 heroku[web.1]: Error R14 (Memory quota exceeded)
2020-08-13T21:08:17.831971+00:00 app[web.1]: 10.69.232.242 - - [13/Aug/2020:21:08:17 +0000] "GET /_dash
/dash_renderer/polyfill@7.v1_5_lm1597352740.8.7.min.js HTTP/1.1" 200 34243 "https://gai-wordcloud-app.h
ozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.125 Sa
2020-08-13T21:08:17.871137+00:00 app[web.1]: 10.13.235.87 - - [13/Aug/2020:21:08:17 +0000] "GET /_dash-
dash_renderer/react@16.v1_5_lm1597352740.13.0.min.js HTTP/1.1" 200 4898 "https://gai-wordcloud-app.hero
lla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.125 Safar
2020-08-13T21:08:17.873417+00:00 heroku[router]: at=info method=GET path="/_dash-component-suites/dash-
.v1_5_lm1597352740.13.0.min.js" host=gai-wordcloud-app.herokuapp.com request_id=7b0d27b4-b82b-401c-b79d
```

For example, if your codes require too much running memory, your app will crash and you get an error above. (Error 14 memory quota exceeded)

This may because the size of your data file is too large or your algorithm isn't good enough

Solution: You should reduce the size of your data file or modify your code

2. Your app imports png file to show graph, you put the file into desired repository but got an error 'no such file or repository: XXX.png'

Solution: try changing the name of the png file from XXX.png to XXX.PNG in your **code**.

3. Your dash table encounters a problem of 'columns not align' (as below)

index	Year	Country	Continent	Emission
0	2008	Aruba	South America	24.75
1	2009	Aruba	South America	24.88
2	2010	Aruba	South America	24.18
3	2011	Aruba	South America	23.92
4	2008	Andorra	Europe	6.30
5	2009	Andorra	Europe	6.05
6	2010	Andorra	Europe	6.12
7	2011	Andorra	Europe	5.97
8	2008	Afghanistan	Asia	0.16
9	2009	Afghanistan	Asia	0.25
10	2010	Afghanistan	Asia	0.30
11	2011	Afghanistan	Asia	0.43
12	2008	Angola	Africa	1.37
13	2009	Angola	Africa	1.43

Solution: In requirements.txt, change the version of dash to 1.13.3 and the version of dash-table to 4.8.1, then dash will fix this problem automatically.

- The dashboard code works well on your computer, but not on your colleague's

Solution: try adding the following line in your code

```
app.config['suppress_callback_exceptions']=True
```

- Failed to find application object 'server' in 'app'

Solution: Add **server = app.server** under the line of `app = dash.Dash()`

- If you encounter the following problem:

- Following error exists when pushing the app

fatal: 'heroku' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights and the repository exists.

Solution: add a line of code in command: **heroku git:remote -a yourappname**

- RuntimeError: main thread is not in main loop with Matplotlib and Flask

Solution: add the following lines in your code

```
import matplotlib
```

```
matplotlib.use('Agg')
```

```
import matplotlib.pyplot as plt
```

- bash: gunicorn: command not found

Solution: make sure gunicorn is in your requirements.txt

- pip install dns failed in the command line

Solution: try **pip install dnspython** in the command line and **import dns** in your code

## Appendix

- requirements.txt (a sample one, if you failed to generate this file automatically, you can create a new txt

---

file, rename it to requirements.txt and then copy and paste the following into your requirements.txt)

```
blis==0.4.1
Brotli==1.0.7
catalogue==1.0.0
certifi==2020.6.20
chardet==3.0.4
click==7.1.2
cycler==0.10.0
cymem==2.0.3
dash==1.13.4
dash-bootstrap-components==0.10.3
dash-core-components==1.10.1
dash-html-components==1.0.3
dash-renderer==1.5.1
dash-table==4.8.1
Flask==1.1.2
Flask-Compress==1.5.0
future==0.18.2
gunicorn==20.0.4
idna==2.10
importlib-metadata==1.7.0
itsdangerous==1.1.0
Jinja2==2.11.2
joblib==0.16.0
kiwisolver==1.2.0
MarkupSafe==1.1.1
matplotlib==3.3.0
murmurhash==1.0.2
nltk==3.5
numpy==1.19.0
pandas==1.0.5
Pillow==7.2.0
plac==1.1.3
plotly==4.9.0
preshed==3.0.2
pyparsing==2.4.7
python-dateutil==2.8.1
pytz==2020.1
regex==2020.7.14
requests==2.24.0
retrying==1.3.3
```

```
scipy==1.5.1
seaborn==0.10.1
six==1.15.0
spacy==2.3.2
srsly==1.0.2
thinc==7.4.1
tqdm==4.48.0
urllib3==1.25.9
wasabi==0.7.1
Werkzeug==1.0.1
wordcloud==1.7.0
zipp==3.1.0
```

2. nltk.txt (a sample one, including the nltk packages we need to download in our code)

