|  |  |  |
| --- | --- | --- |
| **Αποτέλεσμα εικόνας για university of patras** | **Πολυτεχνική Σχολή**  **Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής** | **Αρχική** |

**ΘΕΩΡΙΑ ΑΠΟΦΑΣΕΩΝ**

**2020-2021**

**Project#1**

**Machine Learning with Python: COVID-19**

|  |  |  |
| --- | --- | --- |
| **Μέλη Ομάδας** | | |
| **Ρουχωτά** | **Ειρήνη** | **1059654** |
| **Χατζημιχάλης** | **Ιωάννης** | **1059613** |
| **Παπασταύρου** | **Χριστίνα** | **1059621** |

***Εισαγωγή***

Χρησιμοποιώντας το παρακάνω σύνολο δεδομένων (dataset) δόθηκαν απαντήσεις σε μία σειρά από ερωτήμα σχετικά με :

* Διάδοση του COVID (Spread of the novel coronavirus),
* Ανάλυση (Analysis),
* Οπτικοποίηση (Visualization),
* Πρόβλεψη και συγκρίσεις (Prediction & Comparisons)

Πηγή δεδομένων-Data Source:

<https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports>

***Δοκιμή προγράμματος***

>python project.py [list-of-questions-to-be-displayed]

όπου list-of-questions-to-be-displayed:

είναι μια προαιρετική λίστα των ερωτήσεων (χωρισμένες με κόμμα) στις οποίες θέλουμε να εμφανίσουμε της σχετικές απαντήσεις. Αν δεν δώσουμε το συγκεκριμένο όρισμα, τότε το πρόγραμμα θα εμφανίσει τις απαντήσεις σε όλα τα ερωτήματα.

Για παράδειγμα η εντολή

>python project.py 1,12,3

θα εμφανίσει τις απαντήσεις στα ερωτήματα 1, 3, και 12.

***Ανάλυση δεδομένων***

Η μορφή των αρχείων στο σύνολο των δεδομένων είναι τέτοια ώστε κάθε αρχείο επόμενης ημερομηνίας περιέχει τις νέες πληροφορίες καθώς και αυτές των προηγούμενων αρχείων. Έτσι τα περισσότερα ερωτήματα απαντήθηκαν αναλύοντας το πιο πρόσφατο αρχείο. Σε μερικά ερωτήματα (ερ. 5) που απαιτήθηκαν μόνο οι πιο πρόσφατες πληροφορίες, οι απαντήσεις δόθηκαν αναλύοντας τις σχετικές διαφορές (σήμερα-χτες) των δύο πιο πρόσφατων αρχείων. Τέλος κάποια ερωτήματα (ερ. 14) απαιτούσαν τη στατιστική ανάλυση όλων των αρχείων δηλαδή του συνόλου των δεδομένων.

Το πρόγραμμα μας, κάθε φορά που εκτελείται, βρίσκει αυτόματα τα πιο πρόσφατα αρχεία (στην ημερομηνία εκτέλεσης του) και η ανάλυση γίνεται πάνω σε αυτά.

Η αναζήτηση των πιο πρόσφατων αρχείων γίνεται από το παρακάτω κώδικα:

#-----------------------------------------------------------

# Returns a list with the names of the last -n-

# available files

# Most recent files appear at the front

# Hence last\_files[0] contains the name ofe the most recent

# available file e.t.c

def find\_last\_update(n):

last\_files = []

global yyyy

yyyy = datetime.now().year

global mm

mm = datetime.now().month

global dd

dd = datetime.now().day

k=0

while (k<n):

while (1):

try:

fname = get\_filename(dd, mm, yyyy)

pd\_file = pd.read\_csv(file\_prefix + fname)

last\_files.append(file\_prefix + fname)

break;

except IOError as e:

OneDayBack()

OneDayBack()

k+=1

return(last\_files)

***Ερωτήματα και απαντήσεις***

1.  Γράψτε ένα πρόγραμμα σε Python 3.9 για να εμφανίσετε τις πρώτες 5 σειρές από το καταγεγραμμένων σύνολο δεδομένων COVID-19.  Εκτυπώστε επίσης τις πληροφορίες του συνόλου δεδομένων και ελέγξτε τις τιμές που λείπουν.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*0 nan nan nan Afghanistan 2021-01-10 05:22:12 33.9391 67.71 53400 2264 43740 7396 Afghanistan 137.175 4.2397*

*1 nan nan nan Albania 2021-01-10 05:22:12 41.1533 20.1683 63033 1233 37327 24473 Albania 2190.32 1.95612*

*2 nan nan nan Algeria 2021-01-10 05:22:12 28.0339 1.6596 101913 2803 69011 30099 Algeria 232.407 2.75039*

*3 nan nan nan Andorra 2021-01-10 05:22:12 42.5063 1.5218 8586 85 7724 777 Andorra 11112.4 0.989984*

*4 nan nan nan Angola 2021-01-10 05:22:12 -11.2027 17.8739 18156 416 13205 4535 Angola 55.242 2.29125*

*- --- --- --- ----------- ------------------- -------- ------- ------ ---- ----- ----- ----------- --------- --------*

*Deaths 1926544.0*

*Recovered 49729445.0*

*Confirmed 89630087.0*

*Active 37974297.0*

*dtype: float64*

*FIPS Admin2 ... Incident\_Rate Case\_Fatality\_Ratio*

*0 NaN NaN ... 137.175124 4.239700*

*1 NaN NaN ... 2190.318994 1.956118*

*2 NaN NaN ... 232.407243 2.750385*

*3 NaN NaN ... 11112.405358 0.989984*

*4 NaN NaN ... 55.242049 2.291254*

*... ... ... ... ... ...*

*3972 NaN NaN ... 1.554368 2.313285*

*3973 NaN NaN ... 2875.691769 1.079066*

*3974 NaN NaN ... 7.054256 28.992395*

*3975 NaN NaN ... 144.511878 1.746528*

*3976 NaN NaN ... 137.920344 2.356212*

*Σχετική υπορουτίνα:*

#----------------------------------------------------------

# Subroutine q1: Answer in question #1 of the project

def q1(file\_name):

df=get\_dataframe(file\_name)

print(tabulate(df.iloc[:5,]))

print(df[["Deaths","Recovered","Confirmed","Active"]].sum())

print(df[df.isna().any(axis=1)])

2.  Γράψτε ένα πρόγραμμα Python 3.9 για να λάβετε τον πιο πρόσφατο αριθμό των επιβεβαιωμένων κρουσμάτων (Conﬁrmed), των θανάτων (Deaths), των ιαθέντων (recovered) και ενεργών καταγεγραμμένων περιπτώσεων (active cases) COVID-19.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| | Admin2 | Province\_State | Country\_Region | Confirmed | Deaths | Recovered | Active |*

*|------|-------------------------------------------|------------------------------------------|----------------------------------|-------------|----------|-------------|------------|*

*| 0 | nan | nan | Afghanistan | 53400 | 2264 | 43740 | 7396 |*

*| 1 | nan | nan | Albania | 63033 | 1233 | 37327 | 24473 |*

*| 2 | nan | nan | Algeria | 101913 | 2803 | 69011 | 30099 |*

*| 3 | nan | nan | Andorra | 8586 | 85 | 7724 | 777 |*

*| 4 | nan | nan | Angola | 18156 | 416 | 13205 | 4535 |*

*| 5 | nan | nan | Antigua and Barbuda | 169 | 5 | 152 | 12 |*

*| 6 | nan | nan | Argentina | 1714409 | 44417 | 1504330 | 165662 |*

*| 7 | nan | nan | Armenia | 161794 | 2922 | 148893 | 9979 |*

*| 8 | nan | Australian Capital Territory |*

*Σχετική υπορουτίνα*

# Subroutine q2: Answer in question #2 of the project

def q2(file\_name):

df=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region",

"Confirmed","Deaths","Recovered","Active"]]

tab\_print(df)

3.  Γράψτε ένα πρόγραμμα Python 3.9 για να λάβετε τον πιο πρόσφατο αριθμό επιβεβαιωμένων θανάτων και ιαθέντων ατόμων από καταγεγραμμένων περιπτώσεις COVID-19 των Country/Region - Province/State wise.

*Σημειώσεις: Διαβάζει τα δεδομένα από τα 2 πιο πρόσφατα αρχεία για να υπολογίσει τις διαφορές (δηλαδή τη διαφορά των αριθμών στα δύο αρχεία)*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| | Recent\_Deaths | Recent\_Recovered |*

*| ('Brazil', 'Pernambuco') | 35 | 1131 |*

*| ('Brazil', 'Piaui') | 6 | 994 |*

*| ('Brazil', 'Rio Grande do Norte') | 24 | 9815 |*

*| ('Brazil', 'Rio Grande do Sul') | 87 | 4243 |*

*| ('Brazil', 'Rio de Janeiro') | 122 | 3750 |*

*| ('Brazil', 'Rondonia') | 18 | 2176 |*

*| ('Brazil', 'Roraima') | 9 | 25 |*

*| ('Brazil', 'Santa Catarina') | 38 | 6011 |*

*| ('Brazil', 'Sao Paulo') | 244 | 14053 |*

*| ('Brazil', 'Sergipe') | 5 | 1406 |*

*| ('Brazil', 'Tocantins') | 6 | 1029 |*

*| ('Canada', 'Alberta') | 6 | 411 |*

*…....*

*| ('Russia', 'Khakassia Republic') | 3 | 51 |*

*| ('Russia', 'Khanty-Mansi Autonomous Okrug') | 3 | 143 |*

*| ('Russia', 'Kirov Oblast') | 1 | 141 |*

*| ('Russia', 'Komi Republic') | 2 | 116 |*

*| ('Russia', 'Kostroma Oblast') | 2 | 61 |*

*| ('Russia', 'Krasnodar Krai') | 16 | 366 |*

*| ('Russia', 'Krasnoyarsk Krai') | 11 | 180 |*

*| ('Russia', 'Kurgan Oblast') | 7 | 85 |*

*| ('Russia', 'Kursk Oblast') | 3 | 155 |*

*| ('Russia', 'Leningrad Oblast') | 13 | 168 |*

*Σχετική υπορουτίνα*

# Subroutine q3: Answer in question #3 of the project

def q3(file\_name, file\_name\_prev):

df1=get\_dataframe(file\_name)

[["Province\_State","Country\_Region","Deaths","Recovered"]]

df2=df1.groupby(['Country\_Region','Province\_State']).sum()

df1prev=get\_dataframe(file\_name\_prev)

[["Province\_State","Country\_Region","Deaths","Recovered"]]

df2prev=df1prev.groupby(['Country\_Region','Province\_State']).sum()

df=pd.merge(df2,df2prev,on=['Country\_Region','Province\_State'],

how='left')

df['Recent\_Deaths']=df['Deaths\_x']-df['Deaths\_y']

df['Recent\_Recovered']=df['Recovered\_x']-df['Recovered\_y']

tab\_print(df[['Recent\_Deaths', 'Recent\_Recovered']])

4.  Γράψτε ένα πρόγραμμα Python 3.9 για να μάθετε για τις περιπτώσεις των κινεζικών επαρχιών (Chinese province wise) των επιβεβαιωμένων, των θανάτων και των ιαθέντων καταγεγραμμένων περιπτώσεων COVID-19

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| | Province\_State | Country\_Region | Confirmed | Deaths | Recovered |*

*|-----|------------------|------------------|-------------|----------|-------------|*

*| 111 | Anhui | China | 993 | 6 | 986 |*

*| 112 | Beijing | China | 996 | 9 | 950 |*

*| 113 | Chongqing | China | 590 | 6 | 584 |*

*| 114 | Fujian | China | 526 | 1 | 497 |*

*| 115 | Gansu | China | 182 | 2 | 180 |*

*| 116 | Guangdong | China | 2068 | 8 | 2022 |*

*| 117 | Guangxi | China | 264 | 2 | 261 |*

*Σχετική υπορουτίνα*

# Subroutine q4: Answer in question #4 of the project

def q4(file\_name):

df=get\_dataframe(file\_name)

[["Province\_State","Country\_Region","Confirmed",

"Deaths","Recovered"]]

Chinadf = df.loc[(df.Country\_Region == "China")]

tab\_print(Chinadf)

5.  Γράψτε ένα πρόγραμμα Python 3.9 για να λάβετε τις πιο πρόσφατες περιπτώσεις θανάτων από COVID-19

*-Σημειώσεις: Διαβάζει τα δεδομένα από τα 2 πιο πρόσφατα αρχεία για να υπολογίσει τις διαφορές (δηλαδή τη διαφορά των αριθμών στα δύο αρχεία)*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| | Admin2 | Province\_State | Country\_Region | Recent\_Deaths |*

*|------|-------------------------------------------|------------------------------------------|----------------------------------|-----------------|*

*| 0 | nan | nan | Afghanistan | 7 |*

*| 1 | nan | nan | Albania | 3 |*

*| 2 | nan | nan | Algeria | 5 |*

*| 3 | nan | nan | Andorra | 1 |*

*| 4 | nan | nan | Angola | 1 |*

*| 5 | nan | nan | Antigua and Barbuda | 0 |*

*| 6 | nan | nan | Argentina | 144 |*

*| 7 | nan | nan | Armenia | 14 |*

*| 8 | nan | Australian Capital Territory | Australia | 0 |*

*| 9 | nan | New South Wales | Australia | 0 |*

*| 10 | nan | Northern Territory | Australia | 0 |*

*| 11 | nan | Queensland | Australia | 0 |*

*| 12 | nan | South Australia | Australia | 0 |*

*| 13 | nan | Tasmania | Australia | 0 |*

*| 14 | nan | Victoria | Australia | 0 |*

*| 15 | nan | Western Australia | Australia | 0 |*

*Σχετική υπορουτίνα*

# Subroutine q5: Answer in question #5 of the project

def q5(file\_name,file\_name\_prev):

df=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region","Deaths"]]

dfprev=get\_dataframe(file\_name\_prev)

[["Admin2","Province\_State","Country\_Region","Deaths"]]

d=df["Deaths"].sum()

d1=dfprev["Deaths"].sum()

df=pd.merge(df,dfprev,on=['Country\_Region','Province\_State',

'Admin2'], how='left')

df['Recent\_Deaths']=df['Deaths\_x']-df['Deaths\_y']

tab\_print(df[["Admin2","Province\_State","Country\_Region",

"Recent\_Deaths"]])

print(d-d1)

6. Γράψτε ένα πρόγραμμα Python 3.9 για να απαριθμήσετε χώρες που δεν έχουν καταγραφεί κρούσματα COVID-19.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο. Φορτώνει τα ονόματα όλων των χωρών από σχετική ιστοσελίδα. Γίνεται left join στα δύο σχετικά dataframes για να συσχετιστεί το όνομα της χώρας με τα δεδομένα COVID. Εμφανίζονται τα ονόματα των χωρών που δεν υπάρχουν καταγεγραμμένα κρούσματα*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| | Name |*

*|----|----------------------------------------------|*

*| 1 | Åland Islands |*

*| 2 | American Samoa |*

*| 3 | Anguilla |*

*| 4 | Antarctica |*

*| 5 | Aruba |*

*| 6 | Bermuda |*

*| 7 | Bolivia, Plurinational State of |*

*| 8 | Bonaire, Sint Eustatius and Saba |*

*| 9 | Bouvet Island |*

*| 10 | British Indian Ocean Territory*

*Σχετική υπορουτίνα*

#------------------------------------------------------------------

# Subroutine q6: Answer in question #6 of the project

def q6(file\_name):

df=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region","Confirmed"]]

.groupby(['Country\_Region']).sum()

df.rename(index={'US':'United States'},inplace=True)

dfallcountries=pd.read\_csv('https://raw.githubusercontent.com/

datasets/country-list/master/data.csv')

df2 = pd.merge(dfallcountries, df, left\_on = 'Name',

right\_index=True, how = 'left')

df3 = df2[df2['Confirmed'].isna()][['Name']]

df3.index = range(1, len(df3.index)+1)

tab\_print(df3)

7. Γράψτε ένα πρόγραμμα Python 3.9 για να απαριθμήσετε χώρες με όλες τις καταγεγραμμένες περιπτώσεις COVID-19 που πέθαναν.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| Country\_Region | Deaths |*

*|--------------------------|------------|*

*| Afghanistan | 2264 |*

*| Albania | 1233 |*

*| Algeria | 2803 |*

*| Andorra | 85 |*

*| Angola | 416 |*

*| Antigua and Barbuda | 5 |*

*| Argentina | 44417 |*

*| Armenia | 2922 |*

*| Australia | 909 |*

*| Austria | 6687 |*

*| Azerbaijan | 2890 |*

*| Bahamas | 175 |*

*| Bahrain | 355 |*

*| Bangladesh | 7756 |*

*| Barbados | 7 |*

*| Belarus | 1507 |*

*Σχετική υπορουτίνα*

#----------------------------------------------------------------

# Subroutine q7: Answer in question #7 of the project

def q7(file\_name):

df=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region","Deaths"]].

groupby(['Country\_Region']).sum()

df1 = df.loc[(df.Deaths != 0)]

tab\_print(df1)

8. Γράψτε ένα πρόγραμμα Python 3.9 για να απαριθμήσετε χώρες με όλες τις καταγεγραμμένες περιπτώσεις του COVID-19 που έχουν ιαθεί.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| Country\_Region | Recovered |*

*|----------------------------------|-------------|*

*| Afghanistan | 43740 |*

*| Albania | 37327 |*

*| Algeria | 69011 |*

*| Andorra | 7724 |*

*| Angola | 13205 |*

*| Antigua and Barbuda | 152 |*

*| Argentina | 1504330 |*

*| Armenia | 148893 |*

*| Australia | 25827 |*

*| Austria | 351382 |*

*| Azerbaijan | 207088 |*

*Σχετική υπορουτίνα*

#---------------------------------------------------------------

# Subroutine q8: Answer in question #8 of the project

def q8(file\_name):

df=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region","Recovered"]].

groupby(['Country\_Region']).sum()

df1 = df.loc[(df.Recovered != 0)]

tab\_print(df1)

9. Γράψτε ένα πρόγραμμα Python για να λάβετε τα top 10 countries data (Last Update, Country/Region, Conﬁrmed, Deaths, Recovered) of Novel Coronavirus (COVID-19).

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο*

*Ενδεικτικά αποτελέσματα εκτέλεσης*

*| Country\_Region | Recovered | Confirmed | Deaths |*

*|------------------|-------------|-------------|------------|*

*| US | 0 | 22132396 | 372428 |*

*| India | 10075950 | 10450284 | 150999 |*

*| Brazil | 7205410 | 8075998 | 202631 |*

*| Russia | 2726336 | 3344175 | 60523 |*

*| United Kingdom | 6851 | 3026342 | 81000 |*

*| France | 208090 | 2824920 | 67734 |*

*| Turkey | 2190047 | 2317118 | 22631 |*

*| Italy | 1606630 | 2257866 | 78394 |*

*| Spain | 150376 | 2050360 | 51874 |*

*Σχετική υπορουτίνα*

#-----------------------------------------------------------------

# Subroutine q9: Answer in question #9 of the project

def q9(file\_name):

df=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region","Recovered",

"Confirmed","Deaths"]].groupby(['Country\_Region']).sum()

df.sort\_values(by='Confirmed', ascending=False, inplace=True)

tab\_print(df.iloc[:10,])

10. Γράψτε ένα πρόγραμμα Python 3.9 για να δημιουργήσετε ένα plot (lines) του συνόλου των total deaths, conﬁrmed, recovered and active cases Country wise, όπου οι θάνατοι είναι περισσότεροι από 150.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο. Αποθηκεύει στο αρχείο q10.png το σχετικό γράφημα*

*Σχετική υπορουτίνα*

#----------------------------------------------------------------

# Subroutine q10: Answer in question #10 of the project

def q10(file\_name,n):

df1=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region","Confirmed",

"Recovered", "Deaths", "Active"]].groupby(['Country\_Region'])

.sum()

df0 = df1.loc[(df1.Deaths>150)]

df = df0.head(n)

plot\_title = 'Top '+ str(n) + ' Cases)'

x = np.arange(n)

fig, ax = plt.subplots()

df.plot.line(y='Confirmed', color='red', label='Cd',ax=ax)

df.plot.line(y='Recovered', color='green', label='Rd', ax=ax)

df.plot.line(y='Deaths', color='blue', label='Dts',ax=ax)

df.plot.line(y='Active', color='cyan', label='Act',ax=ax)

# Formatting plot

ax.set\_xticks(x)

ax.set\_xticklabels(df.index)

ax.legend()

formatter = FuncFormatter(millions)

ax.yaxis.set\_major\_formatter(formatter)

plt.xlabel('X Label')

plt.ylabel('Y Label')

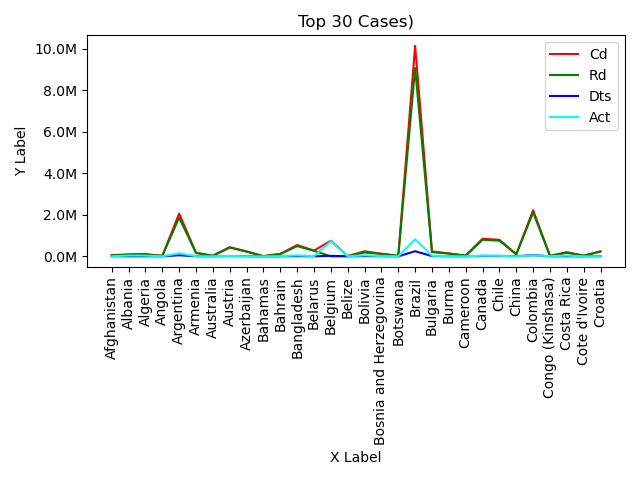
plt.title(plot\_title)

plt.xticks(rotation=90)

fig.tight\_layout()

plt.show()

plt.savefig('q10.png')



11.  Γράψτε ένα πρόγραμμα Python 3.9 για να απεικονίσετε τις περιπτώσεις θανάτου από state/province wise του COVID-19 στις ΗΠΑ.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο. Αποθηκεύει στο αρχείο q11.png το σχετικό γράφημα*

*Αποτελέσματα εκτέλεσης*

*Σχετική υπορουτίνα*

#-----------------------------------------------------------------

# Subroutine q11: Answer in question #11 of the project

def q11(file\_name, n):

df=get\_dataframe(file\_name)

[["Province\_State","Country\_Region","Deaths"]]

USdf = df.loc[(df.Country\_Region == "US")].

groupby(['Province\_State']).sum()

df = USdf.sort\_values(by="Deaths"], ascending=False).head(n)

plot\_title = 'Top '+ str(n) + ' States (Deaths) in US'

x = np.arange(n)

width = 0.30

fig, ax = plt.subplots()

ax.bar(x + width, df.Deaths, width, color='blue',

label='Deaths')

# Formatting plot

ax.set\_xticks(x)

ax.set\_xticklabels(df.index)

ax.legend()

plt.xlabel('States')

plt.ylabel('Cases in millions')

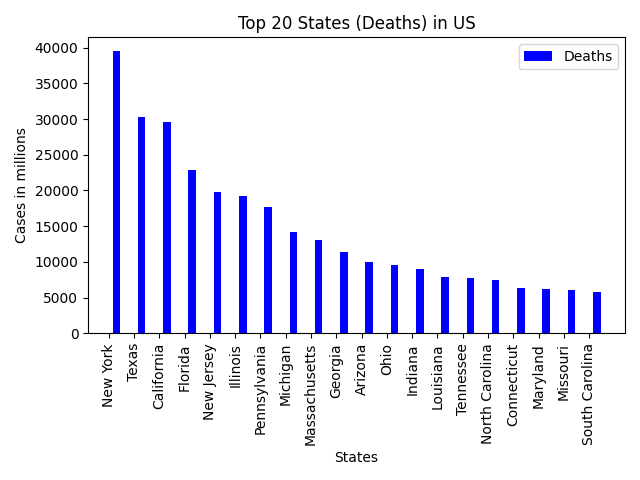
plt.title(plot\_title)

plt.xticks(rotation=90)

fig.tight\_layout()

plt.show()

plt.savefig('q11.png')



12. Γράψτε ένα πρόγραμμα Python για να απεικονίσετε τις ενεργές περιπτώσεις από COVID-19 για state/province wise.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο. Αποθηκεύει ένα αρχείο png για κάθε μία από τις τρεις χώρες (US, China. India) το σχετικό γράφημα. Για τις 3 αυτές χώρες διατίθενται δεδομένα ανά state/province*

*Σχετική υπορουτίνα*

#-----------------------------------------------------------------

# Subroutine q12: Answer in question #12 of the project

def q12(file\_name,n):

q12Country(file\_name,n, "US")

q12Country(file\_name,n, "China")

q12Country(file\_name,n, "India")

def q12Country(file\_name, n, country\_name):

df0=get\_dataframe(file\_name

[["Province\_State","Country\_Region","Active"]]

df1 = df0.loc[(df0.Country\_Region == country\_name)].

groupby(['Province\_State']).sum()

df = df1.sort\_values(by=["Active"], ascending=False).head(n)

plot\_title='Top '+str(n)+' States (Active) '+'for '+

country\_name

x = np.arange(n)

width = 0.30

fig, ax = plt.subplots()

ax.bar(x , df.Active, width, color='green', label='Active')

# Formatting plot

ax.set\_xticks(x)

ax.set\_xticklabels(df.index)

ax.legend()

formatter = FuncFormatter(millions)

ax.yaxis.set\_major\_formatter(formatter)

plt.xlabel('States')

plt.ylabel('Cases in millions')

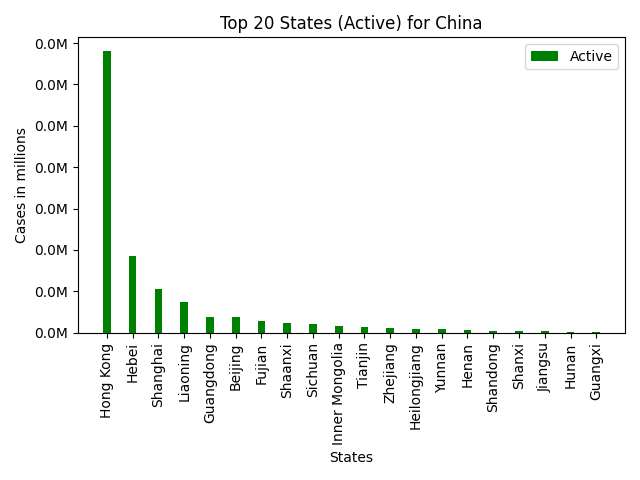
plt.title(plot\_title)

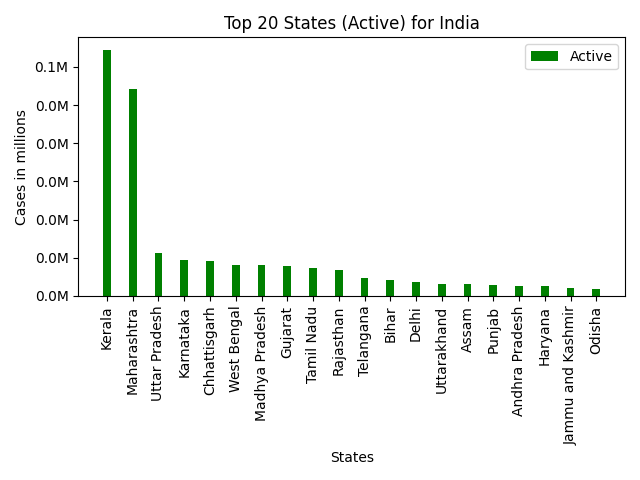
plt.xticks(rotation=90)

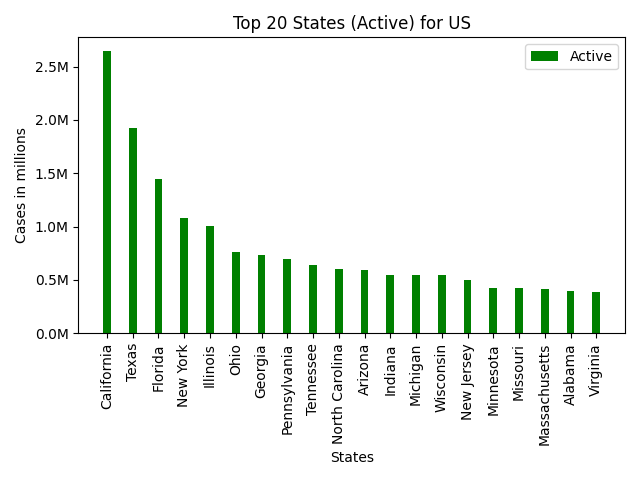
fig.tight\_layout()

plt.show()

plt.savefig('q12\_' + country\_name + '.png')







13. Γράψτε ένα πρόγραμμα Python για να απεικονίσετε την state/province wise, συνδυάζοντας αριθμό επιβεβαιωμένων κρουσμάτων, θανάτων, ιαθέντων και ενεργών κρουσμάτων του ιού COVID-19 στις ΗΠΑ.

*Σημειώσεις: Διαβάζει τα δεδομένα από το πιο πρόσφατο αρχείο. Αποθηκεύει στο αρχείο f13.png το σχετικό γράφημα*

*Σχετική υπορουτίνα*

#----------------------------------------------------------------

# Subroutine q13: Answer in question #13 of the project

def q13(file\_name, n, sorted\_by\_col):

df0=get\_dataframe(file\_name)

[["Admin2","Province\_State","Country\_Region","Deaths",

"Recovered","Confirmed","Active"]]

df1 = df0.loc[(df0.Country\_Region == "US")].

groupby(['Province\_State']).sum()

df =df1.sort\_values(by=[sorted\_by\_col],ascending=False).head(n)

plot\_title = 'Top '+ str(n) + ' States ('+sorted\_by\_col+')'

df['aux'] = df.index

x = np.arange(n)

width = 0.30

fig, ax = plt.subplots()

ax.bar(x - width, df.Confirmed, width, color='red',

label='Confirmed')

ax.bar(x , df.Active, width, color='green',

label='Active')

ax.bar(x + width, df.Deaths, width, color='blue',

label='Deaths')

# Formatting plot

ax.set\_xticks(x)

ax.set\_xticklabels(df.aux)

ax.legend()

formatter = FuncFormatter(millions)

ax.yaxis.set\_major\_formatter(formatter)

plt.xlabel('States')

plt.ylabel('Cases in millions')

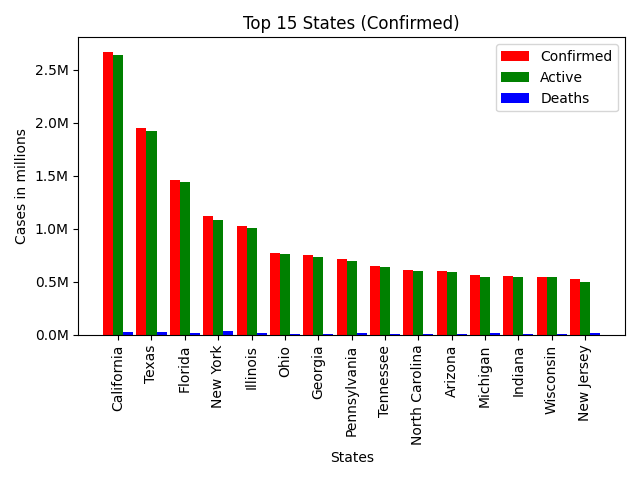
plt.title(plot\_title)

plt.xticks(rotation=90)

fig.tight\_layout()

plt.show()

plt.savefig('q13.png')



14. Γράψτε ένα πρόγραμμα Python για να απεικονίσετε τις περιπτώσεις των παγκόσμια επιβεβαιωμένων κρουσμάτων του ιού COVID-19 (Worldwide Conﬁrmed) με την πάροδο του χρόνου.

*Σημειώσεις: Διαβάζει όλα τα διαθέσιμα αρχεία για να αποτυπώσει τον αριθμό των κρουσμάτων με την πάροδο του χρόνου ξεκινώντας από τις 22/01/2020. Αποθηκεύει στο αρχείο q10.png το σχετικό γράφημα*

*Ενδεικιτικά αποτελέσματα εκτέλεσης*

*Last update = https://github.com/CSSEGISandData/COVID-19/raw/master/csse\_covid\_19\_data/csse\_covid\_19\_daily\_reports/02-15-2021.csv*

*(https://github.com/CSSEGISandData/COVID-19/raw/master/csse\_covid\_19\_data/csse\_covid\_19\_daily\_reports/02-14-2021.csv)*

*Q14...*

*Q14...*

*Reading file 01-22-2020.csv ...ok*

*Q14, Reading file: 01-22-2020.csv*

*Reading file 01-23-2020.csv ...ok*

*Q14, Reading file: 01-23-2020.csv*

*Reading file 01-24-2020.csv ...ok*

*Q14, Reading file: 01-24-2020.csv*

*Reading file 01-25-2020.csv ...ok*

*Q14, Reading file: 01-25-2020.csv*

*.....*

*Reading file 02-12-2021.csv ...ok*

*Q14, Reading file: 02-12-2021.csv*

*Reading file 02-13-2021.csv ...ok*

*Q14, Reading file: 02-13-2021.csv*

*Reading file 02-14-2021.csv ...ok*

*Q14, Reading file: 02-14-2021.csv*

*Reading file 02-15-2021.csv ...ok*

*Q14, Reading file: 02-15-2021.csv*

*Reading file 02-16-2021.csv HTTP Error 404: Not Found*

*Q14, Reading file: 02-16-2021.csv*

*TotalFiles read : 391*

*Σχετική υπορουτίνα*

#-----------------------------------------------------------------

# Subroutine q14: Answer in question #14 of the project

def q14(file\_name):

df=iterate\_all\_files()

plot\_title = 'Confirmed cases worldwide'

x = np.arange(len(df.index))

fig, ax = plt.subplots()

df.plot.line(y='Confirmed',color='red', label='Confirmed',ax=ax)# Formatting plot

ax.set\_xticks(x)

#ax.set\_xticklabels(df.index)

ax.legend()

formatter = FuncFormatter(millions)

ax.yaxis.set\_major\_formatter(formatter)

plt.xlabel('Time')

plt.ylabel('Confirmed Cases Worldwide')

plt.title(plot\_title)

plt.xticks(rotation=90)

fig.tight\_layout()

plt.show()

plt.savefig('q14.png')

