UNIVERSITY OF KARACHI

DATA ANALYSIS REPORT
"PAKISTAN CORONA DATA SET"

GROUP MEMBERS

- Sara Tanveer
- Nimra Rasheed
- Sadaf Azhar

SUBMITTED TO : Sir Syed Umaid Ahmed

TABLE OF CONTENT

INTRODUCTION

Data Science:

Data Science is the process of extracting knowledge and insights from data by using scientific methods

Programming + Statistics + Business

In other words :

"Torture the data and it will confess to anything"

(Ronal Coase, Economics, Nobel Prize)

Data Analysis:

Data analysis is defined as a process of cleaning, transforming and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

A simple example of Data Analysis is whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing the particular decision. This is nothing but analyzing our past or future and making decision based on it. So that is nothing but data analysis. Now same thing analyst does for business purposes, it is called Data Analysis.

So why we use Python for Data Science?

It provides great libraries to deals with data science application. One of the main reasons why Python is widely used in the scientific and research communities is because of its ease of use and simple syntax which makes it easy to adapt for people who do not have engineering background.

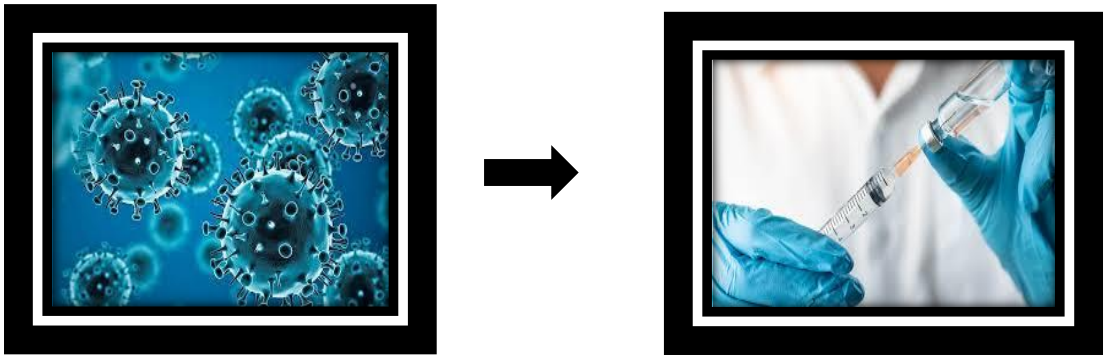
PAKISTAN CORONA VIRUS DATASET

Pakistan witnessed its first Corona virus patient on February 26th 2020. It's a bumpy ride since then. The cases are increasing gradually and we haven't seen the worst yet. While, there are few government resources for cumulative updates, there is no place where you can find the city level patients data. It's also not possible to find the running chronological tally of patients as they test positive. We have decided to create our own dataset for all the researchers out there with such details so we can model the infection spread and forecast the situation in coming days. We hope, by doing so, we will be able to inform policy makers on various intervention models, and healthcare professionals to be ready for the influx of new patients. We certainly hope, that this little contribution will go a long way for saving lives in Pakistan

Content:

The dataset contains seven columns for date, number of cases, number of deaths, number of people recovered, travel history of those cases, and location of the cases (province and city).

The first version has the data from first case of February 26 2020 to April 19, 2020. We intend to publish weekly updates



LIBRARIES

Some important libraries are here which we are using for Pakistan Corona Virus Dataset.

Matplotlib:

This library is responsible for plotting numerical data. And that's why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc.

Pandas:

Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.

Numpy :

The name "Numpy" stands for "Numerical Python". It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally to perform several operations on tensors. Array Interface is one of the key features of this library.

Seaborn:

Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

Plotly :

The plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

STEPS OF ANALYSING DATA

STEP # 01

First of all showing all "csv" files

```
In [1]: import os
        path='.\datasets'
        os.listdir(path)
```

```
Out[1]: ['PK COVID-19-10may.csv',
         'PK COVID-19-22apr.csv',
         'PK COVID-19-30apr.csv',
         'PK COVID-19-3jun.csv',
         'PK COVID-19.csv']
```

STEP# 02

Generating a Data Frame for all csv files

```
In [2]: import pandas as pd
files=[file for file in os.listdir(path) if not file.startswith('.')]
Covid_19_Cases=pd.DataFrame()

for file in files:
    raw_data=pd.read_csv(path+'/'+file,encoding='latin1')
    Covid_19_Cases=pd.concat([Covid_19_Cases,raw_data],ignore_index=True,axis=0)
Covid_19_Cases.to_csv('Covid.csv',index=False)
```

STEP# 03

Importing all important libraries

```
In [4]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import warnings

warnings.filterwarnings('ignore')
```

STEP# 04

Showing first 10 raws

```
In [5]: Covid_19_Cases.head(10)
```

Out[5]:

	Date	Cases	Deaths	Recovered	Travel_history	Province	City
0	2/26/2020	1.0	0.0	0.0	China	Islamabad Capital Territory	Islamabad
1	2/26/2020	2.0	0.0	0.0	Iran/Taftan	Sindh	Karachi
2	2/29/2020	1.0	0.0	0.0	China	Islamabad Capital Territory	Islamabad
3	2/29/2020	1.0	0.0	0.0	Iran/Taftan	Sindh	Karachi
4	3/2/2020	1.0	0.0	0.0	Iran/Taftan	Gilgit-Baltistan	Gilgit
5	3/6/2020	0.0	0.0	1.0	Unknown	Sindh	Karachi
6	3/7/2020	1.0	0.0	0.0	Iran/Taftan	Sindh	Karachi
7	3/9/2020	6.0	0.0	0.0	Syria	Sindh	Karachi
8	3/9/2020	3.0	0.0	0.0	UK	Sindh	Karachi
9	3/10/2020	1.0	0.0	0.0	Iran/Taftan	Sindh	Karachi

STEP# 05

Showing last 10 raws

```
In [6]: Covid_19_Cases.tail(10)
```

Out[6]:

	Date	Cases	Deaths	Recovered	Travel_history	Province	City
7036	4/19/2020	7.0	0.0	0.0	Local - Social Contact	Punjab	Nankana
7037	4/19/2020	0.0	0.0	4.0	Local - Social Contact	Punjab	Bahawalpur
7038	4/19/2020	76.0	0.0	0.0	Local - Social Contact	Punjab	Lodhran
7039	4/19/2020	5.0	0.0	0.0	Local - Social Contact	Punjab	Khanewal
7040	4/19/2020	5.0	0.0	0.0	Local - Social Contact	Punjab	Toba Tek Singh
7041	4/19/2020	3.0	0.0	0.0	Local - Social Contact	Punjab	Sheikhupura
7042	4/19/2020	1.0	0.0	0.0	Local - Social Contact	Punjab	Bhakhar
7043	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7044	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7045	Total	8415.0	168.0	2135.0	NaN	NaN	NaN

STEP# 06

Sorting Dates

```
In [7]: Covid_19_Cases=Covid_19_Cases.sort_values('Date')
```

STEP# 07

Checking Travel_history

```
In [9]: Covid_19_Cases['Travel_history'].unique()

Out[9]: array(['China', 'Iran/Taftan', 'Syria', 'Local - Social Contact', 'USA',
               'UK', 'Unknown', 'KSA', 'Local - Covid Relative',
               'International Passenger', 'Dubai', 'Tableeghi Jamaat', 'Jail',
               'Tableegi Jamaat', 'India', 'Afghanistan', nan], dtype=object)
```

STEP# 08

Checking Province Column

```
In [11]: Covid_19_Cases['Province'].unique()
```

```
Out[11]: array(['Islamabad Capital Territory', 'Sindh', 'Baluchistan',
                'Gilgit-Baltistan', 'Punjab', 'Khyber Pakhtunkhwa',
                'khyber Pakhtunkhwa', 'Azad Jummu Kashmir',
                'Federal Administration Tribal Area',
                'islamabad Capital Territory', nan], dtype=object)
```

STEP# 09

Correcting Province name

```
In [12]: Covid_19_Cases['Province'].replace('Khyber Pakhtunkhwa','Khyber Pakhtunkhwa',
                                            inplace=True)
         Covid_19_Cases['Province'].replace('islamabad Capital Territory',
                                            'Islamabad Capital Territory',inplace=True)
```

STEP# 10

Checking City Column

```
In [13]: Covid_19_Cases['City'].unique()
```

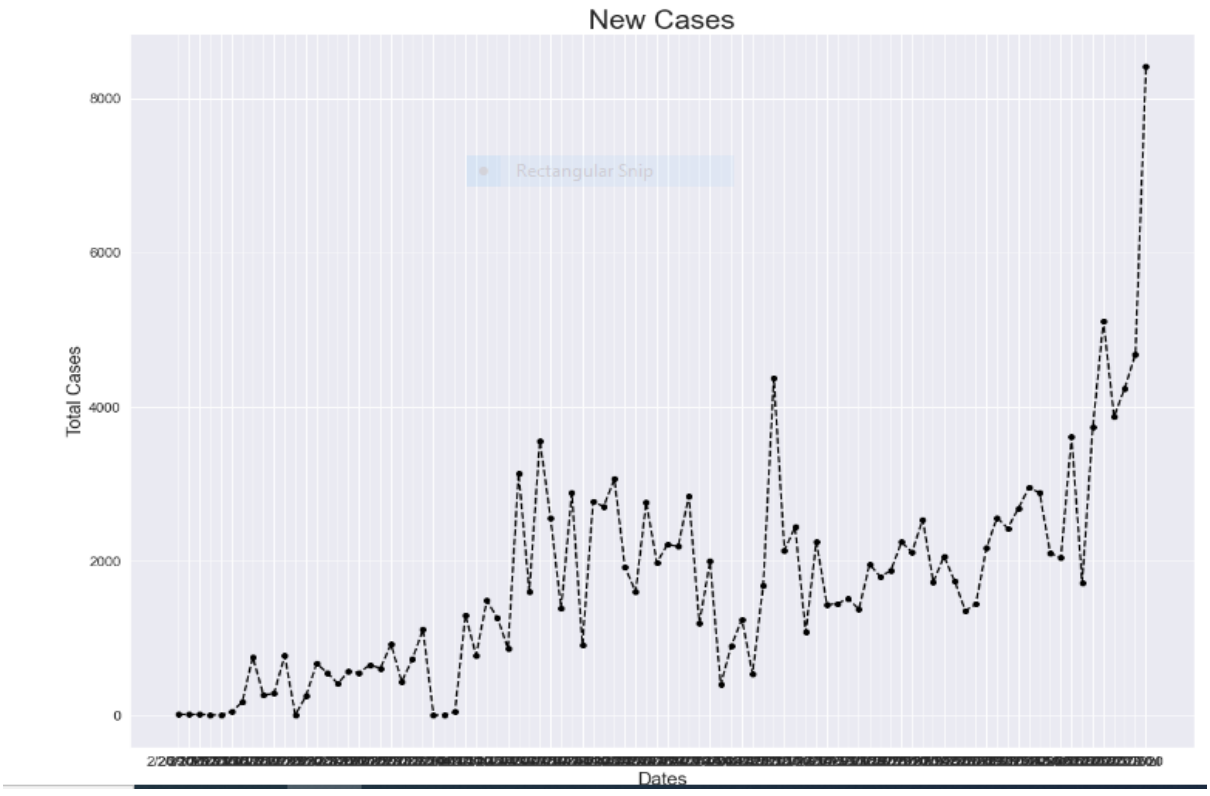
```
Out[13]: array(['Islamabad', 'Karachi', 'Hyderabad', 'Quetta', 'Skardu', 'Gilgit',
                'Taftan', 'Sukkur', 'Lahore', 'Dera Ismail Khan', 'Mardan',
                'Mansehra', 'Peshawar', 'Charsadda', 'Dera Ghazi Khan', 'Buner',
                'Mirpur', 'Hangu ', 'Mardan ', 'Nagar', 'Multan', 'Gujrat ',
                'Jehlum', 'Rawalpindi', 'Khyber', 'Gujranwala', 'Karak', 'Dadu',
                'Sargodha', 'Faisalabad', 'Mandi Bahauddin', 'Rahim yar khan',
                'South Waziristan', 'Gujrat', 'Swabi', 'Kharmang', 'Astore',
                'Narowal', 'Mianwali', 'Attock', 'Dir Upper', 'Swat', 'Nankana',
                'Bahawal Nagar', 'Malakand', 'Orakzai', 'Nowshera', 'Shangla',
                'Kohat', 'Shigar', 'Larkana', 'Khushab', 'Raiwind', 'Vehari',
                'Dir Lower', 'Abbottabad', 'Ghanche', 'Muzaffarabad', 'Bahawalpur',
                'Rahim Yar Khan', 'Bannu', 'Bajaur', 'Laiya', 'Jacobabad',
                'Haripur', 'Kurrum', 'Kasur', 'Hafizabad', 'Sialkot', 'Tank',
                'Lodhran', 'Bhambore ', 'Trarkhel', 'Shaheed Benazirabad',
                'Sheikhupura', 'Hangu', 'Bhakhar', 'Khairpur', 'North Waziristan',
                'Sujawal', 'Sanghar', 'Lakki Marwat', 'Mohmand', 'Pakpatan',
                'Khanewal', 'Sahewal', 'Rajanpur', 'Muzaffargar', 'Naushero Feroz',
                'Diamer', 'Chakwal', 'Jhang', 'Shikarpur', 'Ghotki',
                'Tando Alahyar', 'Tando M. Khan', 'Jamshoro', 'Badin', 'Umerkot',
                'Tharparkar', 'Mirpurkhas', 'Kashmore', 'Kambar Shahdadkot',
                'Toba Tek Singh', 'Battagram', 'Thatta', 'Torghar',
                'Chitral Upper', 'Chitral Lower', 'Kohistan upper',
                'Kohistan Lower', 'Kolai Pallas', 'Tor Ghar', 'Bohmand', 'Matiari',
                'Kurram', 'ISlamabad', 'Chiniot', 'Shujawal', 'Okara', 'NSB',
                'islamabad', 'Pishin', 'Kohistan Upper', 'Chitral', 'Mohamand',
                'Hunza', 'Kolai Pllas', 'Ghizer', 'Mirpur Khas', 'Other',
                'Tando Allahyar', 'Ziarat', 'Dira Ismail Khan', 'Tor  Ghar', nan],
               dtype=object)
```

STEP# 11

Plotting New Cases

In [16]:
```
sns.set(font_scale=2.7)
plt.figure(figsize=(20,15))
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel("Dates",fontsize=20)
plt.ylabel('Total Cases',fontsize=20)
plt.title("New Cases",fontsize=30)

a=Covid_19_Cases.groupby('Date')['Cases'].sum().index
b=Covid_19_Cases.groupby('Date')['Cases'].sum().values
plt.plot(a,b,marker='o',ls="--",lw=2,color='black')
plt.show()
```

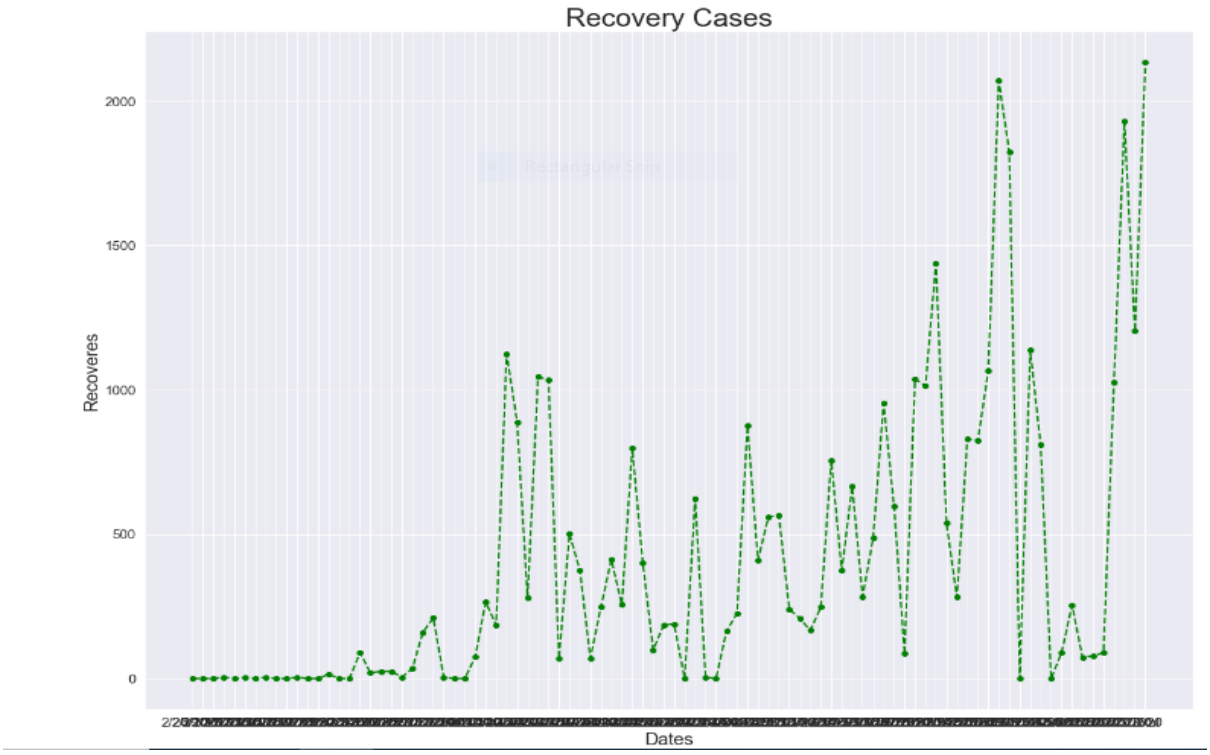


STEP# 12

Plotting Recovered

```
In [17]: sns.set(font_scale=2.7)
         plt.figure(figsize=(20,15))
         plt.xticks(fontsize=15)
         plt.yticks(fontsize=15)
         plt.xlabel("Dates",fontsize=20)
         plt.ylabel('Recoveres',fontsize=20)
         plt.title("Recovery Cases",fontsize=30)

         a=Covid_19_Cases.groupby('Date')['Recovered'].sum().index
         b=Covid_19_Cases.groupby('Date')['Recovered'].sum().values
         plt.plot(a,b,marker='o',ls="--",lw=2,color='Green')
         plt.show()
```

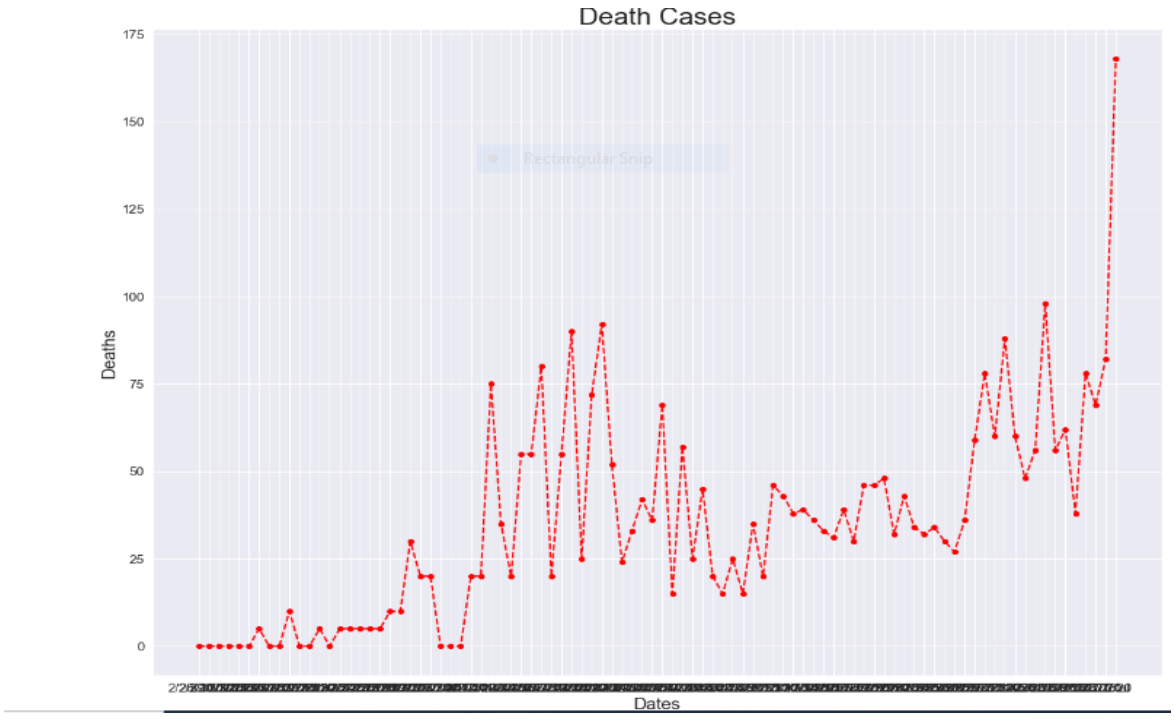


STEP# 13

Plotting Deaths

In [19]:
```
sns.set(font_scale=2.7)
plt.figure(figsize=(20,15))
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel("Dates",fontsize=20)
plt.ylabel('Deaths',fontsize=20)
plt.title("Death Cases",fontsize=30)

a=Covid_19_Cases.groupby('Date')['Deaths'].sum().index
b=Covid_19_Cases.groupby('Date')['Deaths'].sum().values
plt.plot(a,b,marker='o',ls="--",lw=2,color='red')
plt.show()
```



STEP# 13

Making file for Comulative Cases

```
In [20]: Covid_19_Cases['Cum_Cases']=Covid_19_Cases['Cases'].cumsum()
         Covid_19_Cases['Cum_Recovered']=Covid_19_Cases['Recovered'].cumsum()
         Covid_19_Cases['Cum_Deaths']=Covid_19_Cases['Deaths'].cumsum()
```

STEP# 14

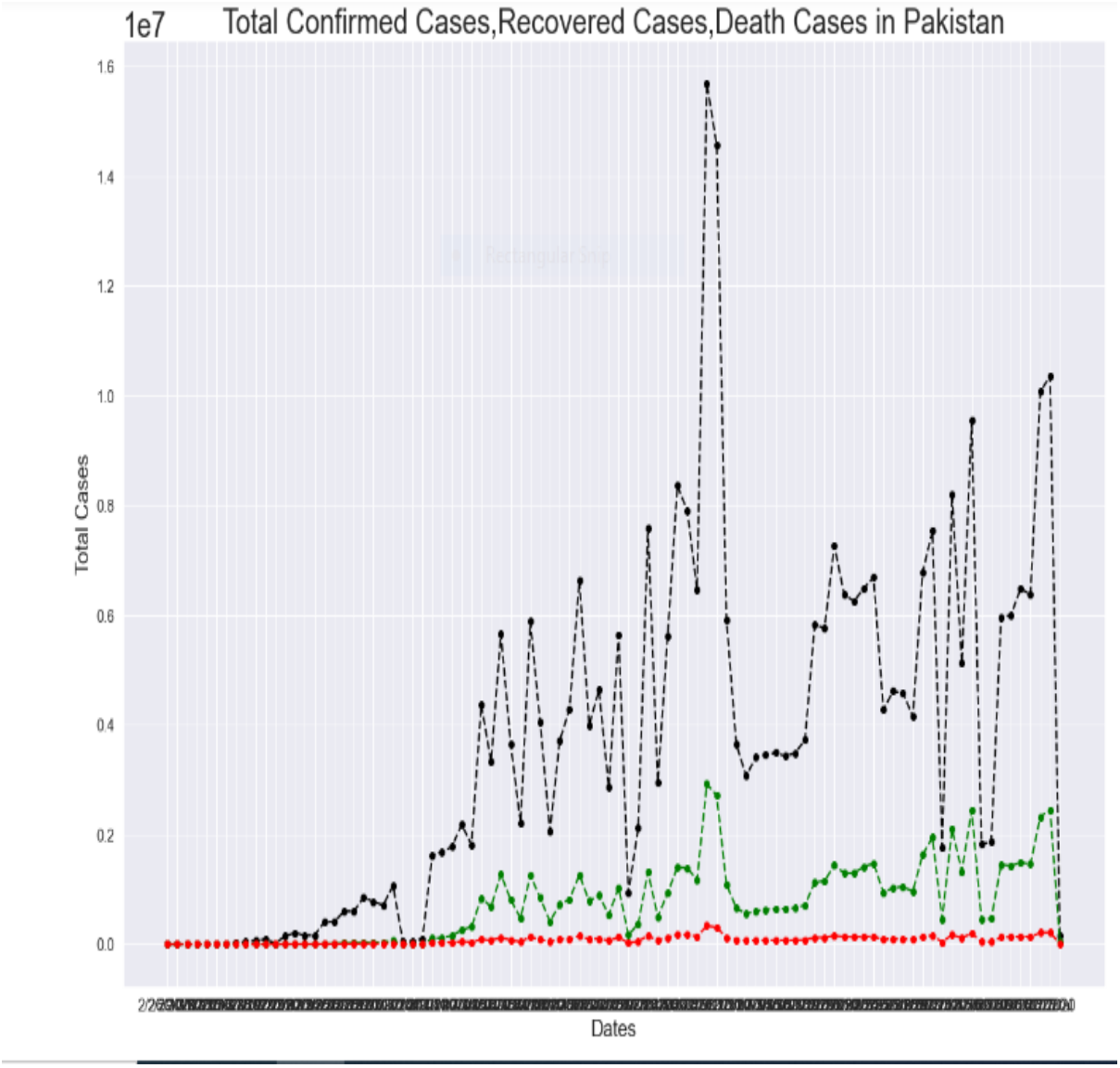
Plotting Comulative Cases

In [22]:
```
plt.figure(figsize=(20,15))
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel("Dates",fontsize=20)
plt.ylabel('Total Cases',fontsize=20)
plt.title("Total Confirmed Cases,Recovered Cases,Death Cases in Pakistan",fontsize=30)

a=Covid_19_Cases.groupby('Date')['Cum_Cases'].sum().index
b=Covid_19_Cases.groupby('Date')['Cum_Cases'].sum().values
plt.plot(a,b,marker='o',ls='--',lw=2,color='black')

c=Covid_19_Cases.groupby('Date')['Cum_Recovered'].sum().index
d=Covid_19_Cases.groupby('Date')['Cum_Recovered'].sum().values
plt.plot(c,d,marker='o',ls='--',lw=2,color='Green')

e=Covid_19_Cases.groupby('Date')['Cum_Deaths'].sum().index
f=Covid_19_Cases.groupby('Date')['Cum_Deaths'].sum().values
plt.plot(e,f,marker='o',ls='--',lw=2,color='red')
plt.show()
```


Total Confirmed Cases,Recovered Cases,Death Cases in Pakistan
1e7
Total Cases
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Dates

STEP# 15

Shorting the Name with help of "if" , "else" libraries

In [24]:
```
def short(x):
    if x=='Islamabad Capital Territort':
        return'ISB'
    elif x=='Sindh':
        return'SD'
    elif x=='Gilgit-Baltistan':
        return'GB'
    elif x=='Baluchistan':
        return'BL'
    elif x=='Punjab':
        return'PJ'
    elif x=='Khyber Pakhtunkhwa':
        return'KPK'
    elif x=='Azad Jammu Kashmir':
        return'AJK'
    else:
        return'FATA'
Covid_19_Cases['Province_Acr']=Covid_19_Cases['Province'].apply(short)
```

STEP# 16

Plotting Province vise Cases

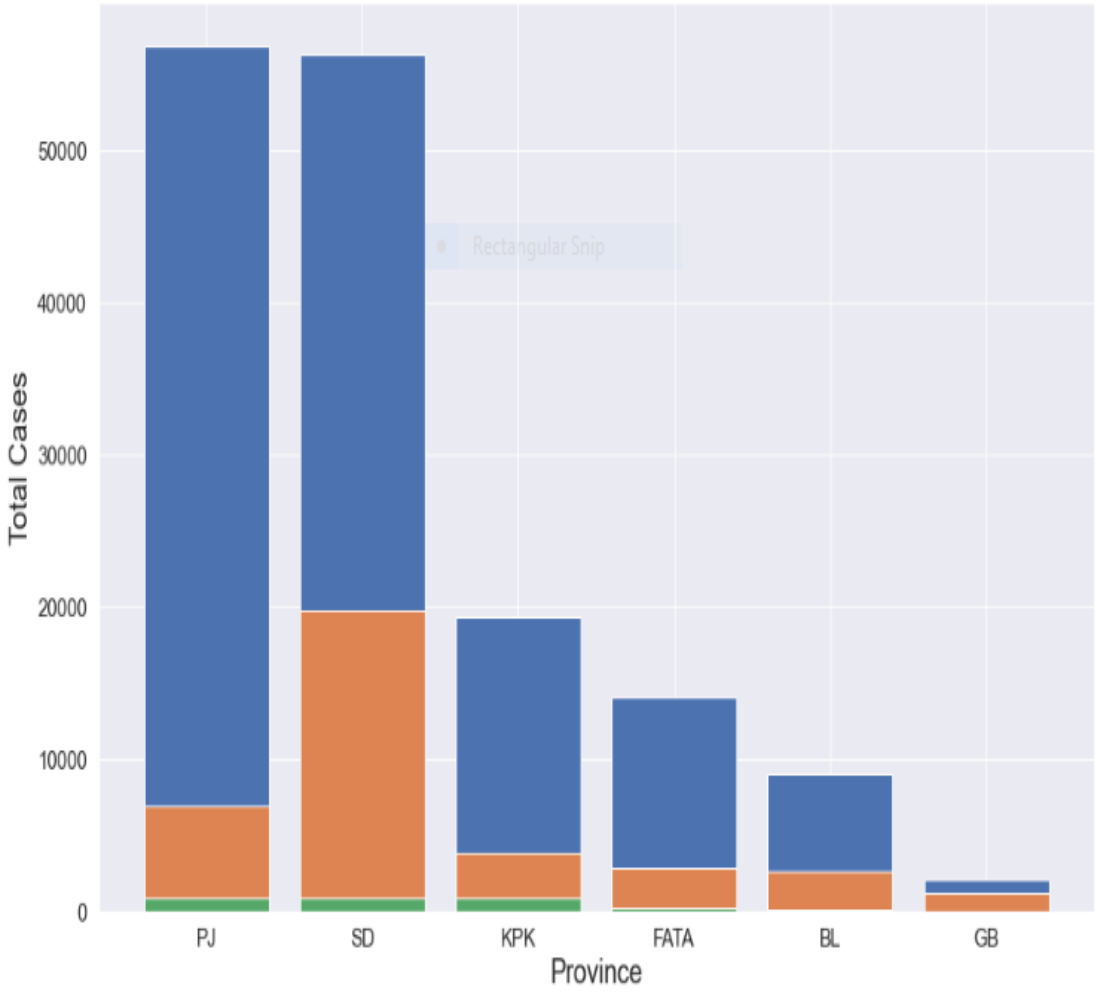
In [25]:
```
plt.figure(figsize=(15,10))
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel("Province",fontsize=20)
plt.ylabel('Total Cases',fontsize=20)

a=Covid_19_Cases.groupby('Province_Acr')['Cases'].sum().sort_values(ascending=False).index
b=Covid_19_Cases.groupby('Province_Acr')['Cases'].sum().sort_values(ascending=False).values
plt.bar(a,b)

c=Covid_19_Cases.groupby('Province_Acr')['Recovered'].sum().sort_values(ascending=False).index
d=Covid_19_Cases.groupby('Province_Acr')['Recovered'].sum().sort_values(ascending=False).values
plt.bar(c,d)

e=Covid_19_Cases.groupby('Province_Acr')['Deaths'].sum().sort_values(ascending=False).index
f=Covid_19_Cases.groupby('Province_Acr')['Deaths'].sum().sort_values(ascending=False).values
plt.bar(e,f)

plt.show()
```

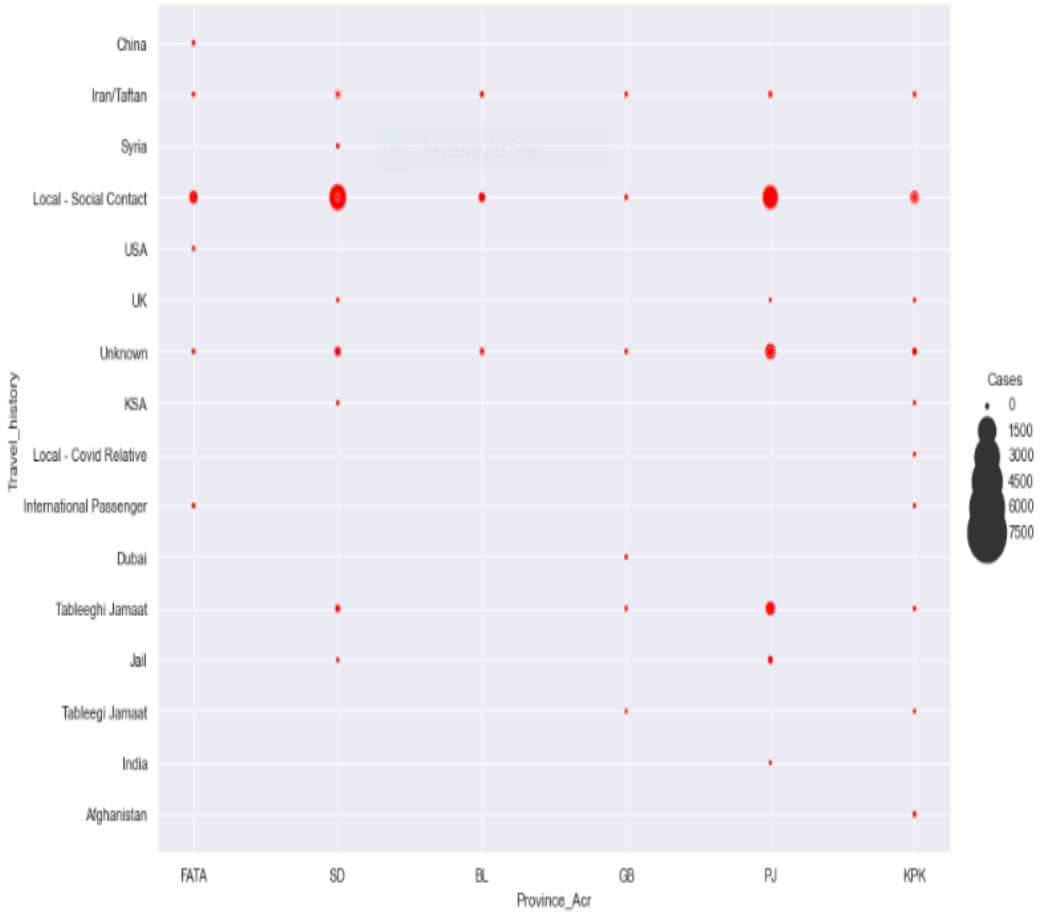

Total Cases
50000
40000
30000
20000
10000
0
PJ
SD
KPK
FATA
BL
GB
Province

STEP# 17

Plotting graph between Province and Travel_history

```
In [26]: plt.figure(figsize=(15,9))
         sns.set(font_scale=1)
         sns.relplot(x="Province_Acr",y="Travel_history",size="Cases",sizes=(10,1500),alpha=.5,color='red',
                     height=8,aspect=1.6,data=Covid_19_Cases).tight_layout()
         plt.show()
```

```
<Figure size 1080x648 with 0 Axes>
```



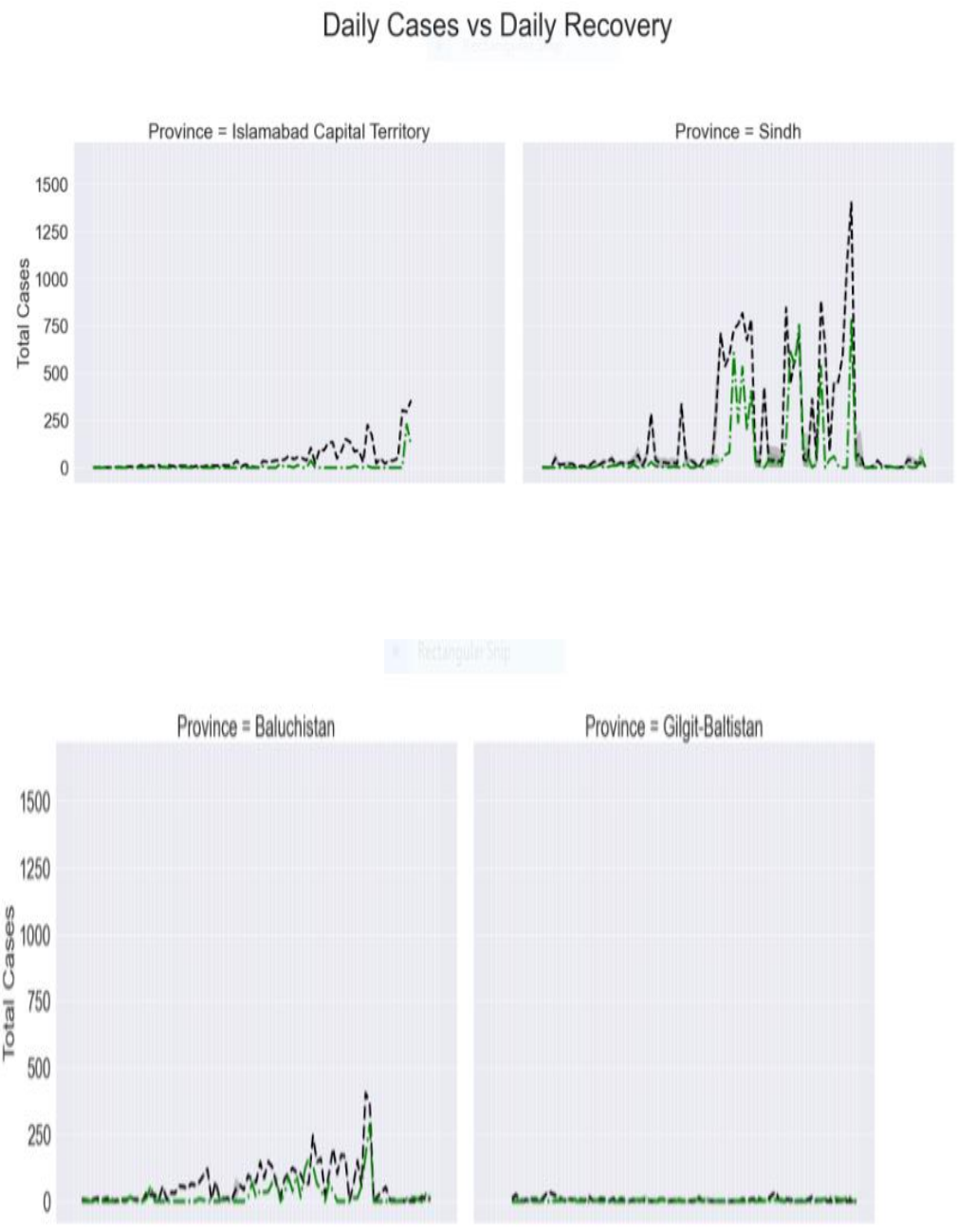
STEP# 18

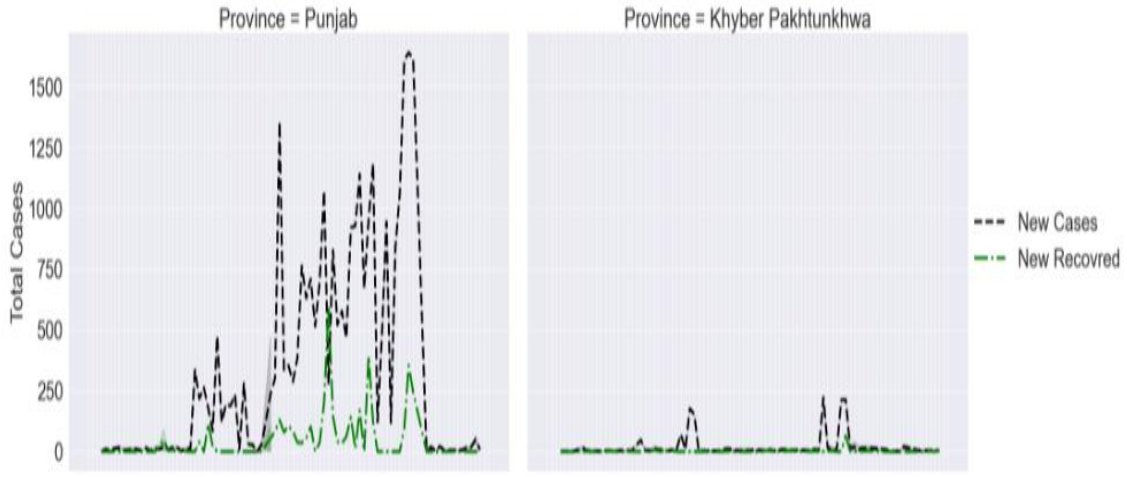
Plotting graph between Daily Cases and Daily Recovery

In [31]:
```
sns.set(font_scale=2.7)
g=sns.FacetGrid(Covid_19_Cases,col="Province",height=11,aspect=1.2,col_wrap=2,margin_titles=True)
g.map(sns.lineplot,'Date','Cases',color='black',label='New Cases', ls='--', lw=4)
g.map(sns.lineplot,'Date','Recovered',color='green',label='New Recovred', ls='-.', lw=4)
g.set_xticklabels(rotation=90)
g.tight_layout()
g.set_ylabels('Total Cases')
g.add_legend()

g.fig.subplots_adjust(top=0.9)
g.fig.suptitle('Daily Cases vs Daily Recovery',x=0.45,y=0.95,fontsize=50)

plt.show()
```


Daily Cases vs Daily Recovery
Province = Islamabad Capital Territory
Province = Sindh
Province = Baluchistan
Province = Gilgit-Baltistan
Total Cases
0
250
500
750
1000
1250
1500



Reference :

https://www.kaggle.com/muhammaddaniyal214/prediction-corona-virus-covid19