

Corona Cases In Pakistan

Final Project



May 5, 2025

23F-0572 23F-0672 23F-0675 23F-0706

**Corona Virus in Pakistan**

**Group Members:**

**Group Leader: Hamza Saleem**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 23F-0572 | Hamza Saleem | BSCS-4F |
| 2 | 23F-0672 | Waheed Zafar | BSCS-4F |
| 3 | 23F-0675 | M. Sumair | BSCS-4F |
| 4 | 23F-0706 | Usman Ashfaq | BSCS-4F |

**1. Problem Statement**

The dataset contains information on the total number of COVID-19 cases, total deaths, and recovery cases for every province of Pakistan. This data provides insights into the spread and impact of the virus at a local level. By analyzing the COVID-19 situation in different cities within each province, we can identify areas with high case counts, assess the severity of the outbreak, and monitor the recovery progress. This dataset serves as a valuable resource for understanding the regional dynamics of the pandemic and guiding targeted interventions to control its spread.

**2. Objective**

The objective of this work is to analyze the COVID-19 dataset for cities in each province of Pakistan, focusing on total cases, deaths, and recovery rates. This analysis aims to provide a comprehensive understanding of the regional impact of the virus, identify areas of concern, and assess the effectiveness of control measures. By examining city-level data, we can gain insights into the spread and severity of the pandemic within different provinces, aiding in targeted interventions, resource allocation, and decision-making for effective management of COVID-19 in Pakistan.

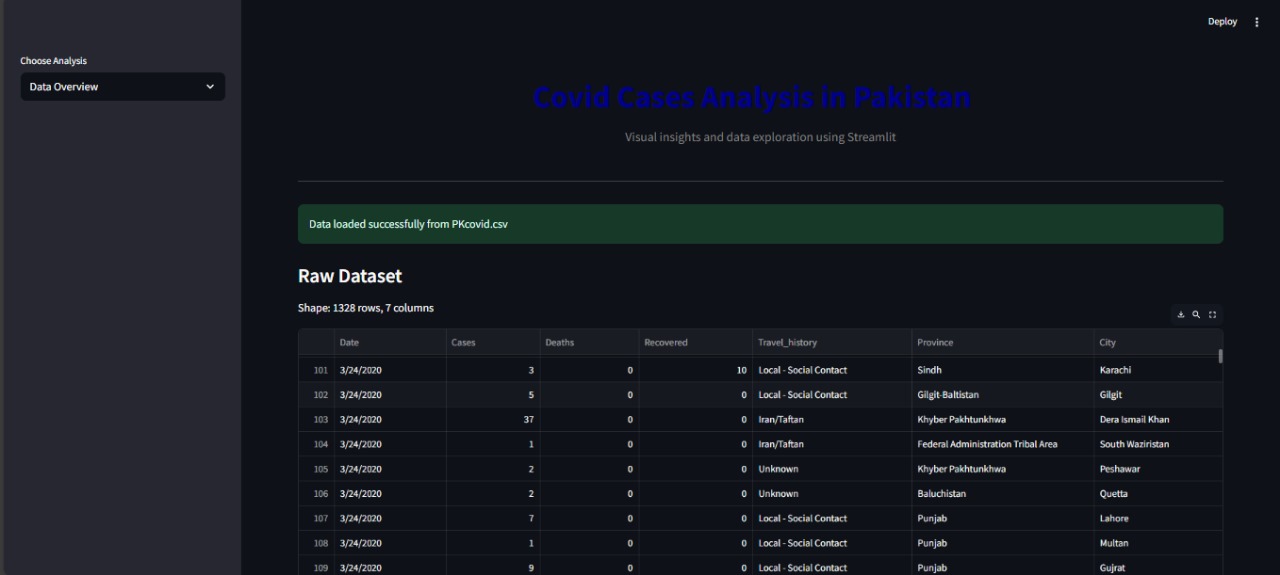
**3. Data Description**

[**https://www.kaggle.com/datasets/zusmani/pakistan-corona-virus-citywise-data**](https://www.kaggle.com/datasets/zusmani/pakistan-corona-virus-citywise-data)

The data set is about corona cases in Pakistan from February 26 ,2020 to May 10, 2020.

**4. Results**

* **Main Interface**



A screenshot of a computer

AI-generated content may be incorrect.

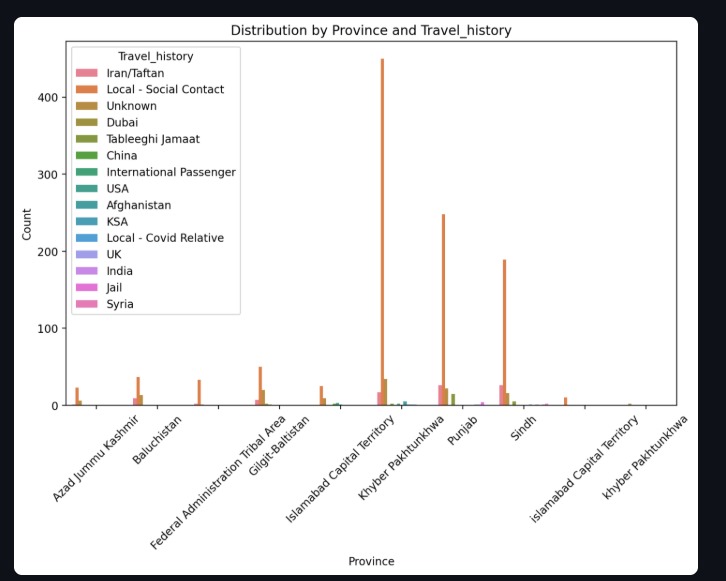
* Summary Description of Data

**Single Bar Chart:**

A graph of distribution of province

AI-generated content may be incorrect.

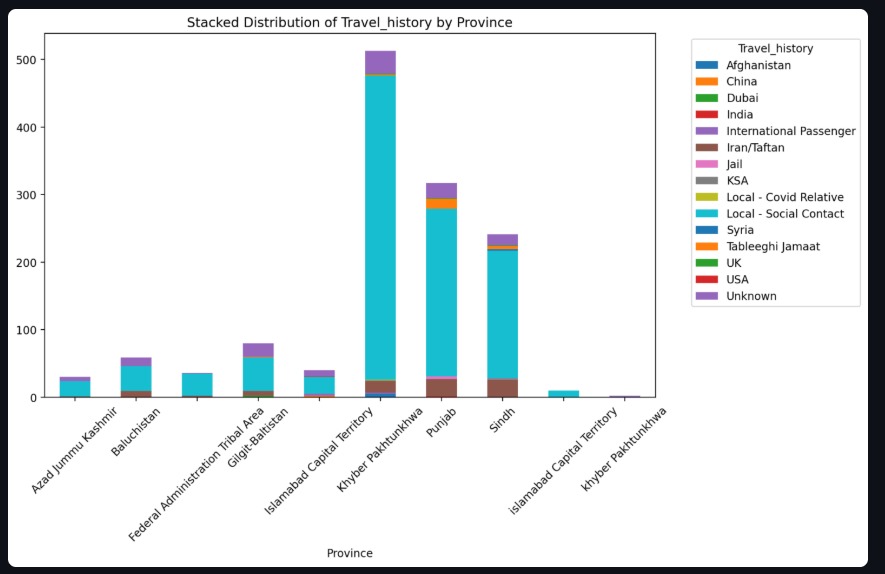
**Multiple Bar Chart:**



**Component Bar Chart:**

A screenshot of a computer

AI-generated content may be incorrect.



**Histogram:**

A graph showing the distribution of cases

AI-generated content may be incorrect.

A graph showing the distribution of death

AI-generated content may be incorrect.

A graph showing a distribution of recovery

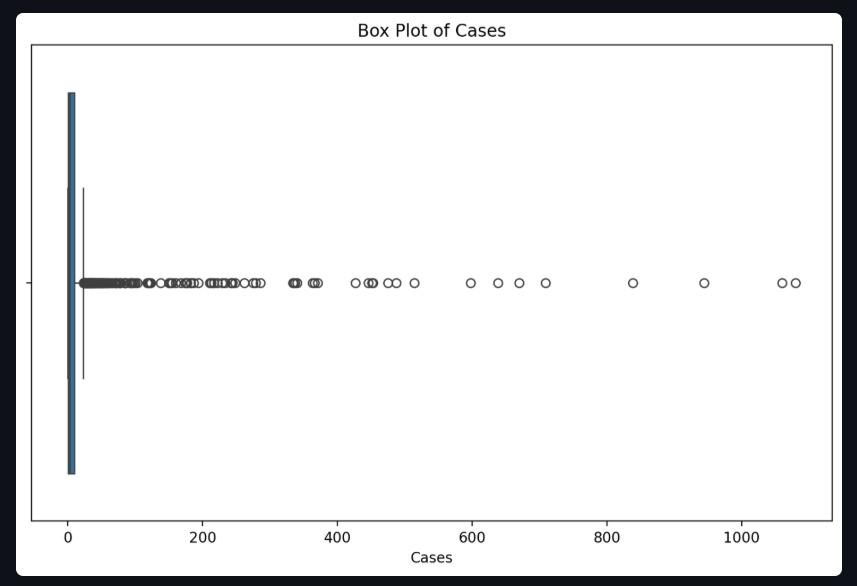
AI-generated content may be incorrect.

**Pie Chart:**

A pie chart with different colored circles

AI-generated content may be incorrect.

**Box Plot:**



**Probability Distribution:**

A graph showing the distribution of cases

AI-generated content may be incorrect.

A graph showing a number of deaths

AI-generated content may be incorrect.

A graph showing a number of recovery data

AI-generated content may be incorrect.

**Regression Analysis:**

A graph showing the growth of cases

AI-generated content may be incorrect.

A graph showing the results of a recovery

AI-generated content may be incorrect.

A graph showing the loss of death

AI-generated content may be incorrect.

**Regression Result:**

OLS Regression Results

==============================================================================

Dep. Variable: Recovered R-squared: 0.039

Model: OLS Adj. R-squared: 0.038

Method: Least Squares F-statistic: 53.70

Date: Mon, 05 May 2025 Prob (F-statistic): 4.04e-13

Time: 13:37:23 Log-Likelihood: -5142.0

No. Observations: 1328 AIC: 1.029e+04

Df Residuals: 1326 BIC: 1.030e+04

Df Model: 1

Covariance Type: nonrobust

==============================================================================

coef std err t P>|t| [0.025 0.975]

------------------------------------------------------------------------------

const 1.8289 0.331 5.527 0.000 1.180 2.478

Cases 0.0282 0.004 7.328 0.000 0.021 0.036

==============================================================================

Omnibus: 2094.458 Durbin-Watson: 1.944

Prob(Omnibus): 0.000 Jarque-Bera (JB): 914328.894

Skew: 9.751 Prob(JB): 0.00

Kurtosis: 130.058 Cond. No. 89.1

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

**5. Python Code:**

import streamlit as st

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import os

import statsmodels.api as sm

from sklearn.linear\_model import LinearRegression

def loadData():

filePath = "PKcovid.csv"

if os.path.exists(filePath):

try:

data = pd.read\_csv(filePath)

st.success("Data loaded successfully from PKcovid.csv")

return data

except Exception as e:

st.error(f"Error reading file: {e}")

return None

else:

st.error("PKcovid.csv not found in the same directory as app.py")

st.error("Please ensure:")

st.error("1. The CSV file is in the same folder as your Python script")

st.error("2. The filename is exactly 'PKcovid.csv' (case-sensitive)")

return None

def showData(data):

st.subheader("Raw Dataset")

st.write(f"Shape: {data.shape[0]} rows, {data.shape[1]} columns")

st.dataframe(data)

st.subheader("Descriptive Statistics")

st.write(data.describe())

def plotSingleBarChart(data):

st.subheader("Single Bar Chart")

categoricalCols = data.select\_dtypes(include=['object', 'category']).columns.tolist()

if not categoricalCols:

st.warning("No categorical columns found in the data.")

return

column = st.selectbox("Select a column for Bar Chart", categoricalCols, key='singleBar')

fig, ax = plt.subplots(figsize=(10, 6))

sns.countplot(x=data[column], ax=ax, order=data[column].value\_counts().index)

plt.xticks(rotation=45)

ax.set\_title(f"Distribution of {column}")

st.pyplot(fig)

def plotMultipleBarChart(data):

st.subheader("Multiple Bar Chart")

categoricalCols = data.select\_dtypes(include=['object', 'category']).columns.tolist()

if len(categoricalCols) < 2:

st.warning("Need at least 2 categorical columns for multiple bar chart.")

return

cols = st.multiselect("Select 2+ categorical columns", categoricalCols, key='multiBar')

if len(cols) >= 2:

grouped = data.groupby(cols).size().reset\_index(name='Count')

fig, ax = plt.subplots(figsize=(10, 6))

sns.barplot(x=cols[0], y='Count', hue=cols[1], data=grouped, ax=ax)

plt.xticks(rotation=45)

ax.set\_title(f"Distribution by {cols[0]} and {cols[1]}")

st.pyplot(fig)

def plotComponentBarChart(data):

st.subheader("Component (Stacked) Bar Chart")

categoricalCols = data.select\_dtypes(include='object').columns.tolist()

if len(categoricalCols) < 2:

st.warning("Need at least 2 categorical columns for stacked chart.")

return

catCol1 = st.selectbox("X-axis category", categoricalCols, key='stackedX')

remainingCols = [c for c in categoricalCols if c != catCol1]

catCol2 = st.selectbox("Stack by", remainingCols, key='stackedY')

stacked = data.groupby([catCol1, catCol2]).size().unstack(fill\_value=0)

fig, ax = plt.subplots(figsize=(10, 6))

stacked.plot(kind='bar', stacked=True, ax=ax)

plt.xticks(rotation=45)

ax.set\_title(f"Stacked Distribution of {catCol2} by {catCol1}")

ax.legend(title=catCol2, bbox\_to\_anchor=(1.05, 1), loc='upper left')

st.pyplot(fig)

def plotHistogram(data):

st.subheader("Histogram")

numericCols = data.select\_dtypes(include='number').columns.tolist()

if not numericCols:

st.warning("No numeric columns found.")

return

numCol = st.selectbox("Select numeric column", numericCols, key='hist')

fig, ax = plt.subplots(figsize=(10, 6))

sns.histplot(data[numCol], bins=20, kde=True, ax=ax)

ax.set\_title(f"Distribution of {numCol}")

ax.set\_xlabel(numCol)

ax.set\_ylabel("Frequency")

st.pyplot(fig)

def plotPieChart(data):

st.subheader("Pie Chart")

categoricalCols = data.select\_dtypes(include='object').columns.tolist()

if not categoricalCols:

st.warning("No categorical columns found.")

return

catCol = st.selectbox("Select category", categoricalCols, key='pie')

counts = data[catCol].value\_counts()

fig, ax = plt.subplots(figsize=(8, 8))

ax.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=90)

ax.set\_title(f"Distribution of {catCol}")

st.pyplot(fig)

def plotBoxPlot(data):

st.subheader("Box Plot")

numericCols = data.select\_dtypes(include='number').columns.tolist()

if not numericCols:

st.warning("No numeric columns found.")

return

numCol = st.selectbox("Select numeric column", numericCols, key='box')

fig, ax = plt.subplots(figsize=(10, 6))

sns.boxplot(x=data[numCol], ax=ax)

ax.set\_title(f"Box Plot of {numCol}")

st.pyplot(fig)

def plotProbabilityDistribution(data):

st.subheader("Probability Distribution")

numericCols = data.select\_dtypes(include='number').columns.tolist()

if not numericCols:

st.warning("No numeric columns found in the data.")

return

numCol = st.selectbox("Select numeric column", numericCols, key='probDist')

fig, ax = plt.subplots(figsize=(10, 6))

sns.histplot(data[numCol], kde=True, stat='density', ax=ax)

ax.set\_title(f"Probability Distribution of {numCol}")

ax.set\_xlabel(numCol)

ax.set\_ylabel("Density")

st.pyplot(fig)

def plotRegression(data):

st.subheader("Regression Analysis")

numericCols = data.select\_dtypes(include='number').columns.tolist()

if len(numericCols) < 2:

st.warning("Need at least 2 numeric columns for regression.")

return

col1 = st.selectbox("Independent Variable (X)", numericCols, key='regX')

col2 = st.selectbox("Dependent Variable (Y)", [c for c in numericCols if c != col1], key='regY')

X = data[[col1]]

y = data[col2]

model = LinearRegression()

model.fit(X, y)

yPred = model.predict(X)

fig, ax = plt.subplots(figsize=(10, 6))

sns.regplot(x=data[col1], y=data[col2], ax=ax)

ax.set\_title(f"Regression: {col2} ~ {col1}")

ax.set\_xlabel(col1)

ax.set\_ylabel(col2)

st.pyplot(fig)

st.subheader("Regression Results")

XSm = sm.add\_constant(X)

modelSm = sm.OLS(y, XSm).fit()

st.text(str(modelSm.summary()))

def main():

st.set\_page\_config(layout="wide")

st.markdown("<h1 style='text-align: center; color: navy;'>Covid Cases Analysis in Pakistan</h1>", unsafe\_allow\_html=True)

st.markdown("<hr>", unsafe\_allow\_html=True)

data = loadData()

if data is None:

st.stop()

analysisOptions = [

"Data Overview",

"Single Bar Chart",

"Multiple Bar Chart",

"Component Bar Chart",

"Histogram",

"Pie Chart",

"Box Plot",

"Probability Distribution",

"Regression Analysis"

]

analysis = st.sidebar.selectbox("Choose Analysis", analysisOptions)

if analysis == "Data Overview":

showData(data)

elif analysis == "Single Bar Chart":

plotSingleBarChart(data)

elif analysis == "Multiple Bar Chart":

plotMultipleBarChart(data)

elif analysis == "Component Bar Chart":

plotComponentBarChart(data)

elif analysis == "Histogram":

plotHistogram(data)

elif analysis == "Pie Chart":

plotPieChart(data)

elif analysis == "Box Plot":

plotBoxPlot(data)

elif analysis == "Probability Distribution":

plotProbabilityDistribution(data)

elif analysis == "Regression Analysis":

plotRegression(data)

if \_name\_ == "\_main\_":

main()

**6. Conclusion**

In this project, we analyzed the cases of COVID-19 in Pakistan and explored various aspects of the data. Here are the main findings:

* **Case Summary:**

The total number of reported cases is 1080 along with recoveries and deaths. The Standard deviation of total cases is 82.97734 which display significant variation in the number of cases in different regions of Pakistan.

* **Probabilities:**

We calculated the probability of death and recovery by province. The probability of death differs in each province according to the number of cases.

* **Graphical Analysis:**

We have created several graphs to visualize the data. A scatter plot of total deaths by province provides an overview of the distribution of deaths in different regions.

Pie charts show the distribution of cases, deaths and recoveries in each province, highlighting the proportion of each category.

* **Predictions:**

We used linear regression to predict the number of deaths and recoveries based on the number of cases. The distribution of cases, deaths and recoveries varies from province to province, highlighting the need for targeted response and resources. Predictions based on linear regression can help predict the potential number of deaths and recoveries based on the number of cases. These findings can help to understand the current situation and plan appropriate strategies to effectively fight the pandemic.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*