**TB DATA CLEANING, DESCRIPTIVE SATISTICS, CORRELATION**

Introduction This R Markdown document performs data cleaning and exploratory analysis on the TB dataset using Python. It includes loading the dataset, basic exploration, visualization, and data preprocessing.

Load Dataset and Basic Exploration

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# Load dataset  
tb = pd.read\_csv('https://raw.githubusercontent.com/Jobmrtall/MSD/refs/heads/main/TB.csv')  
  
# Display basic dataset information  
print(tb.shape) # Print dataset dimensions (rows, columns)

## (5120, 47)

print(tb.isna().sum()) # Count missing values in each column

## Country or territory name 0  
## ISO 2-character country/territory code 24  
## ISO 3-character country/territory code 0  
## ISO numeric country/territory code 0  
## Region 0  
## Year 0  
## Estimated total population number 0  
## Estimated prevalence of TB (all forms) per 100 000 population 0  
## Estimated prevalence of TB (all forms) per 100 000 population, low bound 20  
## Estimated prevalence of TB (all forms) per 100 000 population, high bound 20  
## Estimated prevalence of TB (all forms) 0  
## Estimated prevalence of TB (all forms), low bound 20  
## Estimated prevalence of TB (all forms), high bound 20  
## Method to derive prevalence estimates 0  
## Estimated mortality of TB cases (all forms, excluding HIV) per 100 000 population 0  
## Estimated mortality of TB cases (all forms, excluding HIV), per 100 000 population, low bound 0  
## Estimated mortality of TB cases (all forms, excluding HIV), per 100 000 population, high bound 0  
## Estimated number of deaths from TB (all forms, excluding HIV) 0  
## Estimated number of deaths from TB (all forms, excluding HIV), low bound 0  
## Estimated number of deaths from TB (all forms, excluding HIV), high bound 0  
## Estimated mortality of TB cases who are HIV-positive, per 100 000 population 0  
## Estimated mortality of TB cases who are HIV-positive, per 100 000 population, low bound 1942  
## Estimated mortality of TB cases who are HIV-positive, per 100 000 population, high bound 1942  
## Estimated number of deaths from TB in people who are HIV-positive 0  
## Estimated number of deaths from TB in people who are HIV-positive, low bound 1942  
## Estimated number of deaths from TB in people who are HIV-positive, high bound 1942  
## Method to derive mortality estimates 0  
## Estimated incidence (all forms) per 100 000 population 0  
## Estimated incidence (all forms) per 100 000 population, low bound 94  
## Estimated incidence (all forms) per 100 000 population, high bound 94  
## Estimated number of incident cases (all forms) 0  
## Estimated number of incident cases (all forms), low bound 94  
## Estimated number of incident cases (all forms), high bound 94  
## Method to derive incidence estimates 2133  
## Estimated HIV in incident TB (percent) 1475  
## Estimated HIV in incident TB (percent), low bound 1478  
## Estimated HIV in incident TB (percent), high bound 1478  
## Estimated incidence of TB cases who are HIV-positive per 100 000 population 1475  
## Estimated incidence of TB cases who are HIV-positive per 100 000 population, low bound 1478  
## Estimated incidence of TB cases who are HIV-positive per 100 000 population, high bound 1478  
## Estimated incidence of TB cases who are HIV-positive 1475  
## Estimated incidence of TB cases who are HIV-positive, low bound 1478  
## Estimated incidence of TB cases who are HIV-positive, high bound 1478  
## Method to derive TBHIV estimates 5120  
## Case detection rate (all forms), percent 449  
## Case detection rate (all forms), percent, low bound 449  
## Case detection rate (all forms), percent, high bound 449  
## dtype: int64

tb.info() # Summary of dataset structure

## <class 'pandas.core.frame.DataFrame'>  
## RangeIndex: 5120 entries, 0 to 5119  
## Data columns (total 47 columns):  
## # Column Non-Null Count Dtype   
## --- ------ -------------- -----   
## 0 Country or territory name 5120 non-null object   
## 1 ISO 2-character country/territory code 5096 non-null object   
## 2 ISO 3-character country/territory code 5120 non-null object   
## 3 ISO numeric country/territory code 5120 non-null int64   
## 4 Region 5120 non-null object   
## 5 Year 5120 non-null int64   
## 6 Estimated total population number 5120 non-null int64   
## 7 Estimated prevalence of TB (all forms) per 100 000 population 5120 non-null float64  
## 8 Estimated prevalence of TB (all forms) per 100 000 population, low bound 5100 non-null float64  
## 9 Estimated prevalence of TB (all forms) per 100 000 population, high bound 5100 non-null float64  
## 10 Estimated prevalence of TB (all forms) 5120 non-null float64  
## 11 Estimated prevalence of TB (all forms), low bound 5100 non-null float64  
## 12 Estimated prevalence of TB (all forms), high bound 5100 non-null float64  
## 13 Method to derive prevalence estimates 5120 non-null object   
## 14 Estimated mortality of TB cases (all forms, excluding HIV) per 100 000 population 5120 non-null float64  
## 15 Estimated mortality of TB cases (all forms, excluding HIV), per 100 000 population, low bound 5120 non-null float64  
## 16 Estimated mortality of TB cases (all forms, excluding HIV), per 100 000 population, high bound 5120 non-null float64  
## 17 Estimated number of deaths from TB (all forms, excluding HIV) 5120 non-null float64  
## 18 Estimated number of deaths from TB (all forms, excluding HIV), low bound 5120 non-null float64  
## 19 Estimated number of deaths from TB (all forms, excluding HIV), high bound 5120 non-null float64  
## 20 Estimated mortality of TB cases who are HIV-positive, per 100 000 population 5120 non-null float64  
## 21 Estimated mortality of TB cases who are HIV-positive, per 100 000 population, low bound 3178 non-null float64  
## 22 Estimated mortality of TB cases who are HIV-positive, per 100 000 population, high bound 3178 non-null float64  
## 23 Estimated number of deaths from TB in people who are HIV-positive 5120 non-null float64  
## 24 Estimated number of deaths from TB in people who are HIV-positive, low bound 3178 non-null float64  
## 25 Estimated number of deaths from TB in people who are HIV-positive, high bound 3178 non-null float64  
## 26 Method to derive mortality estimates 5120 non-null object   
## 27 Estimated incidence (all forms) per 100 000 population 5120 non-null float64  
## 28 Estimated incidence (all forms) per 100 000 population, low bound 5026 non-null float64  
## 29 Estimated incidence (all forms) per 100 000 population, high bound 5026 non-null float64  
## 30 Estimated number of incident cases (all forms) 5120 non-null float64  
## 31 Estimated number of incident cases (all forms), low bound 5026 non-null float64  
## 32 Estimated number of incident cases (all forms), high bound 5026 non-null float64  
## 33 Method to derive incidence estimates 2987 non-null object   
## 34 Estimated HIV in incident TB (percent) 3645 non-null float64  
## 35 Estimated HIV in incident TB (percent), low bound 3642 non-null float64  
## 36 Estimated HIV in incident TB (percent), high bound 3642 non-null float64  
## 37 Estimated incidence of TB cases who are HIV-positive per 100 000 population 3645 non-null float64  
## 38 Estimated incidence of TB cases who are HIV-positive per 100 000 population, low bound 3642 non-null float64  
## 39 Estimated incidence of TB cases who are HIV-positive per 100 000 population, high bound 3642 non-null float64  
## 40 Estimated incidence of TB cases who are HIV-positive 3645 non-null float64  
## 41 Estimated incidence of TB cases who are HIV-positive, low bound 3642 non-null float64  
## 42 Estimated incidence of TB cases who are HIV-positive, high bound 3642 non-null float64  
## 43 Method to derive TBHIV estimates 0 non-null float64  
## 44 Case detection rate (all forms), percent 4671 non-null float64  
## 45 Case detection rate (all forms), percent, low bound 4671 non-null float64  
## 46 Case detection rate (all forms), percent, high bound 4671 non-null float64  
## dtypes: float64(37), int64(3), object(7)  
## memory usage: 1.8+ MB

# Display the frequency of country occurrences and regional distribution  
print(tb.value\_counts('Country or territory name'))

## Country or territory name  
## Afghanistan 24  
## Netherlands 24  
## New Zealand 24  
## Nicaragua 24  
## Niger 24  
## ..  
## Serbia 9  
## Cura?ao 4  
## Bonaire, Saint Eustatius and Saba 4  
## Sint Maarten (Dutch part) 4  
## South Sudan 3  
## Name: count, Length: 219, dtype: int64

print(tb['Region'].value\_counts())

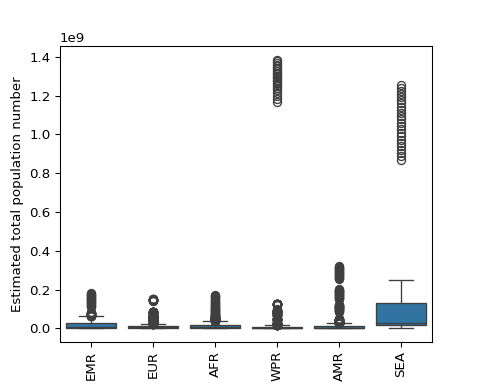
## Region  
## EUR 1281  
## AFR 1107  
## AMR 1088  
## WPR 864  
## EMR 528  
## SEA 252  
## Name: count, dtype: int64

Data Visualization This section generates boxplots and line plots to understand population distribution and TB prevalence trends.

# Boxplot of population distribution by region  
sns.boxplot(x='Region', y='Estimated total population number', data=tb)  
plt.xticks(rotation=90) # Rotate x-axis labels for readability

## ([0, 1, 2, 3, 4, 5], [Text(0, 0, 'EMR'), Text(1, 0, 'EUR'), Text(2, 0, 'AFR'), Text(3, 0, 'WPR'), Text(4, 0, 'AMR'), Text(5, 0, 'SEA')])

plt.show()



Data Cleaning and Imputation This section removes columns with excessive missing data, imputes categorical and numerical missing values, and converts the Year column into the appropriate format.

import warnings  
warnings.filterwarnings('ignore')  
# Drop columns with more than 50% missing values  
tb.dropna(axis=1, thresh=len(tb) \* 0.5, inplace=True)  
  
# Drop rows with less than 5% missing values  
to\_drop = tb.columns[tb.isna().sum() <= len(tb) \* 0.05]  
tb = tb.dropna(subset=to\_drop)  
  
# Convert Year to datetime format  
tb['Year'] = pd.to\_datetime(tb['Year'], format='%Y').dt.year

Imputation of Missing Value Categorical missing values are filled using mode, and numerical missing values are imputed using the median.

# Separate categorical and numerical columns  
cat\_cols = tb.select\_dtypes(include=['object']).columns  
num\_cols = tb.select\_dtypes(include=['float64']).columns  
  
# Impute categorical columns using mode  
t = tb[['Year', 'Country or territory name', 'ISO 2-character country/territory code',  
 'ISO 3-character country/territory code', 'Region',  
 'Method to derive prevalence estimates', 'Method to derive mortality estimates',  
 'Method to derive incidence estimates']]  
t = t.fillna('unknown')  
mode\_df = t.groupby(['Region', 'Year'])['Method to derive incidence estimates'].agg(pd.Series.mode).reset\_index()  
  
# Convert mode values to a dictionary  
mode\_dict = mode\_df.set\_index(['Region', 'Year'])['Method to derive incidence estimates'].to\_dict()  
  
def impute\_mode(row):  
 if pd.isna(row['Method to derive incidence estimates']):  
 return mode\_dict.get((row['Region'], row['Year']), row['Method to derive incidence estimates'])  
 return row['Method to derive incidence estimates']  
  
# Apply mode imputation  
tb['Method to derive incidence estimates'] = tb.apply(impute\_mode, axis=1)  
  
# Function to impute numerical columns using median  
def impute\_median(df, col):  
 median\_dict = df.groupby(['Region', 'Year'])[col].median().to\_dict()  
 df[col] = df.apply(lambda row: median\_dict.get((row['Region'], row['Year']), row[col]) if pd.isna(row[col]) else row[col], axis=1)  
  
# Apply median imputation for numerical columns  
for col in num\_cols:  
 impute\_median(tb, col)  
tb.head()

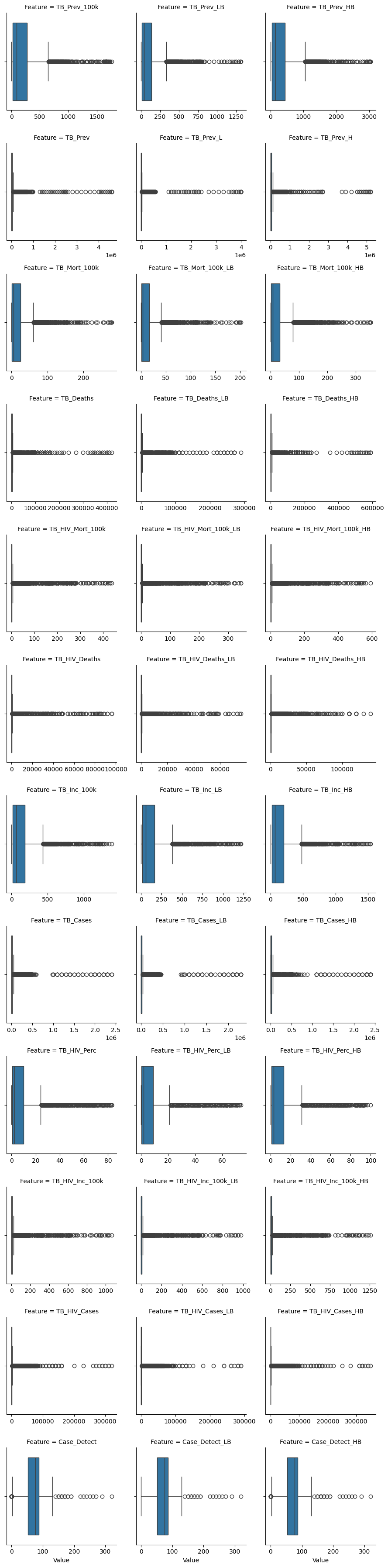
## Country or territory name ... Case detection rate (all forms), percent, high bound  
## 0 Afghanistan ... 24.0   
## 1 Afghanistan ... 110.0   
## 2 Afghanistan ... 89.0   
## 3 Afghanistan ... 96.0   
## 4 Afghanistan ... 91.5   
##   
## [5 rows x 46 columns]

acessing the cleaned imputed data, then renaming the name of columns to more clear short and precise name

tb=pd.read\_csv('https://raw.githubusercontent.com/Jobmrtall/MSD/refs/heads/main/tb\_cleand.csv')  
rename\_dict = {  
 "Country or territory name": "Country",  
 "ISO 2-character country/territory code": "ISO2",  
 "ISO 3-character country/territory code": "ISO3",  
 "ISO numeric country/territory code": "ISONum",  
 "Region": "Region",  
 "Year": "Year",  
 "Estimated total population number": "Pop",  
 "Estimated prevalence of TB (all forms) per 100 000 population": "TB\_Prev\_100k",  
 "Estimated prevalence of TB (all forms) per 100 000 population, low bound": "TB\_Prev\_LB",  
 "Estimated prevalence of TB (all forms) per 100 000 population, high bound": "TB\_Prev\_HB",  
 "Estimated prevalence of TB (all forms)": "TB\_Prev",  
 "Estimated prevalence of TB (all forms), low bound": "TB\_Prev\_L",  
 "Estimated prevalence of TB (all forms), high bound": "TB\_Prev\_H",  
 "Method to derive prevalence estimates": "Prev\_Method",  
 "Estimated mortality of TB cases (all forms, excluding HIV) per 100 000 population": "TB\_Mort\_100k",  
 "Estimated mortality of TB cases (all forms, excluding HIV), per 100 000 population, low bound": "TB\_Mort\_100k\_LB",  
 "Estimated mortality of TB cases (all forms, excluding HIV), per 100 000 population, high bound": "TB\_Mort\_100k\_HB",  
 "Estimated number of deaths from TB (all forms, excluding HIV)": "TB\_Deaths",  
 "Estimated number of deaths from TB (all forms, excluding HIV), low bound": "TB\_Deaths\_LB",  
 "Estimated number of deaths from TB (all forms, excluding HIV), high bound": "TB\_Deaths\_HB",  
 "Estimated mortality of TB cases who are HIV-positive, per 100 000 population": "TB\_HIV\_Mort\_100k",  
 "Estimated mortality of TB cases who are HIV-positive, per 100 000 population, low bound": "TB\_HIV\_Mort\_100k\_LB",  
 "Estimated mortality of TB cases who are HIV-positive, per 100 000 population, high bound": "TB\_HIV\_Mort\_100k\_HB",  
 "Estimated number of deaths from TB in people who are HIV-positive": "TB\_HIV\_Deaths",  
 "Estimated number of deaths from TB in people who are HIV-positive, low bound": "TB\_HIV\_Deaths\_LB",  
 "Estimated number of deaths from TB in people who are HIV-positive, high bound": "TB\_HIV\_Deaths\_HB",  
 "Method to derive mortality estimates": "Mort\_Method",  
 "Estimated incidence (all forms) per 100 000 population": "TB\_Inc\_100k",  
 "Estimated incidence (all forms) per 100 000 population, low bound": "TB\_Inc\_LB",  
 "Estimated incidence (all forms) per 100 000 population, high bound": "TB\_Inc\_HB",  
 "Estimated number of incident cases (all forms)": "TB\_Cases",  
 "Estimated number of incident cases (all forms), low bound": "TB\_Cases\_LB",  
 "Estimated number of incident cases (all forms), high bound": "TB\_Cases\_HB",  
 "Method to derive incidence estimates": "Inc\_Method",  
 "Estimated HIV in incident TB (percent)": "TB\_HIV\_Perc",  
 "Estimated HIV in incident TB (percent), low bound": "TB\_HIV\_Perc\_LB",  
 "Estimated HIV in incident TB (percent), high bound": "TB\_HIV\_Perc\_HB",  
 "Estimated incidence of TB cases who are HIV-positive per 100 000 population": "TB\_HIV\_Inc\_100k",  
 "Estimated incidence of TB cases who are HIV-positive per 100 000 population, low bound": "TB\_HIV\_Inc\_100k\_LB",  
 "Estimated incidence of TB cases who are HIV-positive per 100 000 population, high bound": "TB\_HIV\_Inc\_100k\_HB",  
 "Estimated incidence of TB cases who are HIV-positive": "TB\_HIV\_Cases",  
 "Estimated incidence of TB cases who are HIV-positive, low bound": "TB\_HIV\_Cases\_LB",  
 "Estimated incidence of TB cases who are HIV-positive, high bound": "TB\_HIV\_Cases\_HB",  
 "Case detection rate (all forms), percent": "Case\_Detect",  
 "Case detection rate (all forms), percent, low bound": "Case\_Detect\_LB",  
 "Case detection rate (all forms), percent, high bound": "Case\_Detect\_HB"  
}  
  
# Apply renaming  
tb.rename(columns=rename\_dict, inplace=True)

Description: visualizing the distribution of each column with float data type to examine the outliers

tf = tb.select\_dtypes(include=['float64']).columns  
  
# Reshape data to long format for FacetGrid  
tb\_melted = tb.melt(value\_vars=tf, var\_name='Feature', value\_name='Value')  
  
# Create the FacetGrid  
g = sns.FacetGrid(tb\_melted, col='Feature', col\_wrap=3, sharex=False, sharey=False)  
g.map(sns.boxplot, 'Value')



# Adjust layout  
plt.xticks(rotation=45)

## (array([-100., 0., 100., 200., 300., 400.]), [Text(-100.0, 0, '−100'), Text(0.0, 0, '0'), Text(100.0, 0, '100'), Text(200.0, 0, '200'), Text(300.0, 0, '300'), Text(400.0, 0, '400')])

plt.show()

Description: by calculating the interquantile range we obtained the lower and upper bound. Then we kept replacing values that goes surppaising the bounds in either ways by the value of upper and lower bounds accordingly.this is done to prevent creation of na values which could have happen if we were to remove the outliers for every specfic feature. As a row having an outlier in some feature would have a value within the interquartile range in another. Sp we used the clip method

# Compute IQR-based lower and upper bounds  
Q1 = tb[tf].quantile(0.25)  
Q3 = tb[tf].quantile(0.75)  
IQR = Q3 - Q1  
  
lower\_bound = Q1 - 1.5 \* IQR  
upper\_bound = Q3 + 1.5 \* IQR  
  
# Clip values to IQR range  
tb[tf] = tb[tf].clip(lower=lower\_bound, upper=upper\_bound, axis=1)  
  
  
tb.isna().sum()

## Country 0  
## ISO2 0  
## ISO3 0  
## ISONum 0  
## Region 0  
## Year 0  
## Pop 0  
## TB\_Prev\_100k 0  
## TB\_Prev\_LB 0  
## TB\_Prev\_HB 0  
## TB\_Prev 0  
## TB\_Prev\_L 0  
## TB\_Prev\_H 0  
## Prev\_Method 0  
## TB\_Mort\_100k 0  
## TB\_Mort\_100k\_LB 0  
## TB\_Mort\_100k\_HB 0  
## TB\_Deaths 0  
## TB\_Deaths\_LB 0  
## TB\_Deaths\_HB 0  
## TB\_HIV\_Mort\_100k 0  
## TB\_HIV\_Mort\_100k\_LB 0  
## TB\_HIV\_Mort\_100k\_HB 0  
## TB\_HIV\_Deaths 0  
## TB\_HIV\_Deaths\_LB 0  
## TB\_HIV\_Deaths\_HB 0  
## Mort\_Method 0  
## TB\_Inc\_100k 0  
## TB\_Inc\_LB 0  
## TB\_Inc\_HB 0  
## TB\_Cases 0  
## TB\_Cases\_LB 0  
## TB\_Cases\_HB 0  
## Inc\_Method 0  
## TB\_HIV\_Perc 0  
## TB\_HIV\_Perc\_LB 0  
## TB\_HIV\_Perc\_HB 0  
## TB\_HIV\_Inc\_100k 0  
## TB\_HIV\_Inc\_100k\_LB 0  
## TB\_HIV\_Inc\_100k\_HB 0  
## TB\_HIV\_Cases 0  
## TB\_HIV\_Cases\_LB 0  
## TB\_HIV\_Cases\_HB 0  
## Case\_Detect 0  
## Case\_Detect\_LB 0  
## Case\_Detect\_HB 0  
## dtype: int64

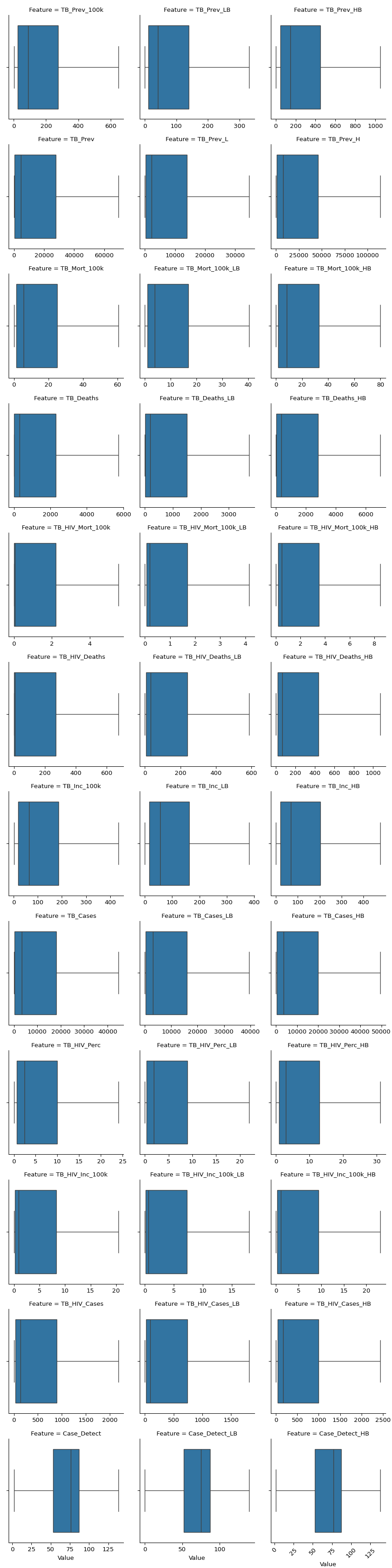
visualizing the data after managing the outliers

tf = tb.select\_dtypes(include=['float64']).columns  
  
# Reshape data to long format for FacetGrid  
tb\_melted = tb.melt(value\_vars=tf, var\_name='Feature', value\_name='Value')  
  
# Create the FacetGrid  
g = sns.FacetGrid(tb\_melted, col='Feature', col\_wrap=3, sharex=False, sharey=False)  
g.map(sns.boxplot, 'Value')

# Adjust layout  
plt.xticks(rotation=45)

## (array([-25., 0., 25., 50., 75., 100., 125., 150.]), [Text(-25.0, 0, '−25'), Text(0.0, 0, '0'), Text(25.0, 0, '25'), Text(50.0, 0, '50'), Text(75.0, 0, '75'), Text(100.0, 0, '100'), Text(125.0, 0, '125'), Text(150.0, 0, '150')])

plt.show()



DESCRIPTIVE STATISTICS OF TB DATA SET The descriptive statistics were done for specfic grouos of the data set which were grouped by means of 1 region 2 countries 3 year 4 year and region

tb\_r=tb.groupby(['Region']).describe(include=[float])  
print(tb\_r.head())

## TB\_Prev\_100k ... Case\_Detect\_HB   
## count mean std min ... 25% 50% 75% max  
## Region ...   
## AFR 1083.0 331.695753 191.750355 15.00 ... 38.00 50.0 63.5 138.0  
## AMR 1045.0 64.719445 80.208763 0.95 ... 66.00 82.0 87.0 138.0  
## EMR 528.0 143.515909 190.869760 2.70 ... 58.00 84.0 87.0 138.0  
## EUR 1269.0 89.991292 130.188088 0.91 ... 74.00 85.0 88.5 138.0  
## SEA 252.0 368.882937 187.565652 51.00 ... 39.75 56.5 71.0 92.0  
##   
## [5 rows x 288 columns]

tb\_c=tb.groupby('Country').describe(include=[float])  
print(tb\_c.head())

## TB\_Prev\_100k ... Case\_Detect\_HB   
## count mean std ... 50% 75% max  
## Country ...   
## Afghanistan 24.0 376.416667 46.469173 ... 51.0 65.50 96.0  
## Albania 24.0 29.333333 5.345999 ... 81.0 81.00 91.0  
## Algeria 24.0 124.375000 14.451756 ... 67.0 69.00 71.0  
## American Samoa 24.0 14.566667 9.240349 ... 76.0 84.75 96.0  
## Andorra 24.0 29.920833 18.991874 ... 87.0 87.00 88.0  
##   
## [5 rows x 288 columns]

tb\_y=tb.groupby('Year').describe(include=['float'])  
print(tb\_y.head())

## TB\_Prev\_100k ... Case\_Detect\_HB   
## count mean std min ... 25% 50% 75% max  
## Year ...   
## 1990 206.0 196.323786 209.441174 2.5 ... 41.25 74.0 87.0 138.0  
## 1991 208.0 195.505288 207.803219 2.8 ... 42.00 75.0 87.0 138.0  
## 1992 209.0 195.398565 206.295280 2.9 ... 42.00 71.0 87.0 138.0  
## 1993 209.0 195.160287 207.744287 2.9 ... 45.00 73.0 87.0 138.0  
## 1994 209.0 199.293780 211.309112 2.8 ... 45.50 76.0 87.0 138.0  
##   
## [5 rows x 288 columns]

tb\_r\_y=tb.groupby(['Year','Region']).describe(include=[float])  
print(tb\_r\_y.head())

## TB\_Prev\_100k ... Case\_Detect\_HB   
## count mean std ... 50% 75% max  
## Year Region ...   
## 1990 AFR 45.0 362.488889 197.088308 ... 41.0 48.000 110.0  
## AMR 42.0 90.328571 107.029420 ... 80.5 86.750 93.0  
## EMR 22.0 165.818182 205.154155 ... 83.5 88.875 138.0  
## EUR 53.0 81.822642 127.729049 ... 86.0 90.000 100.0  
## SEA 10.0 414.200000 176.054632 ... 38.0 55.500 81.0  
##   
## [5 rows x 288 columns]

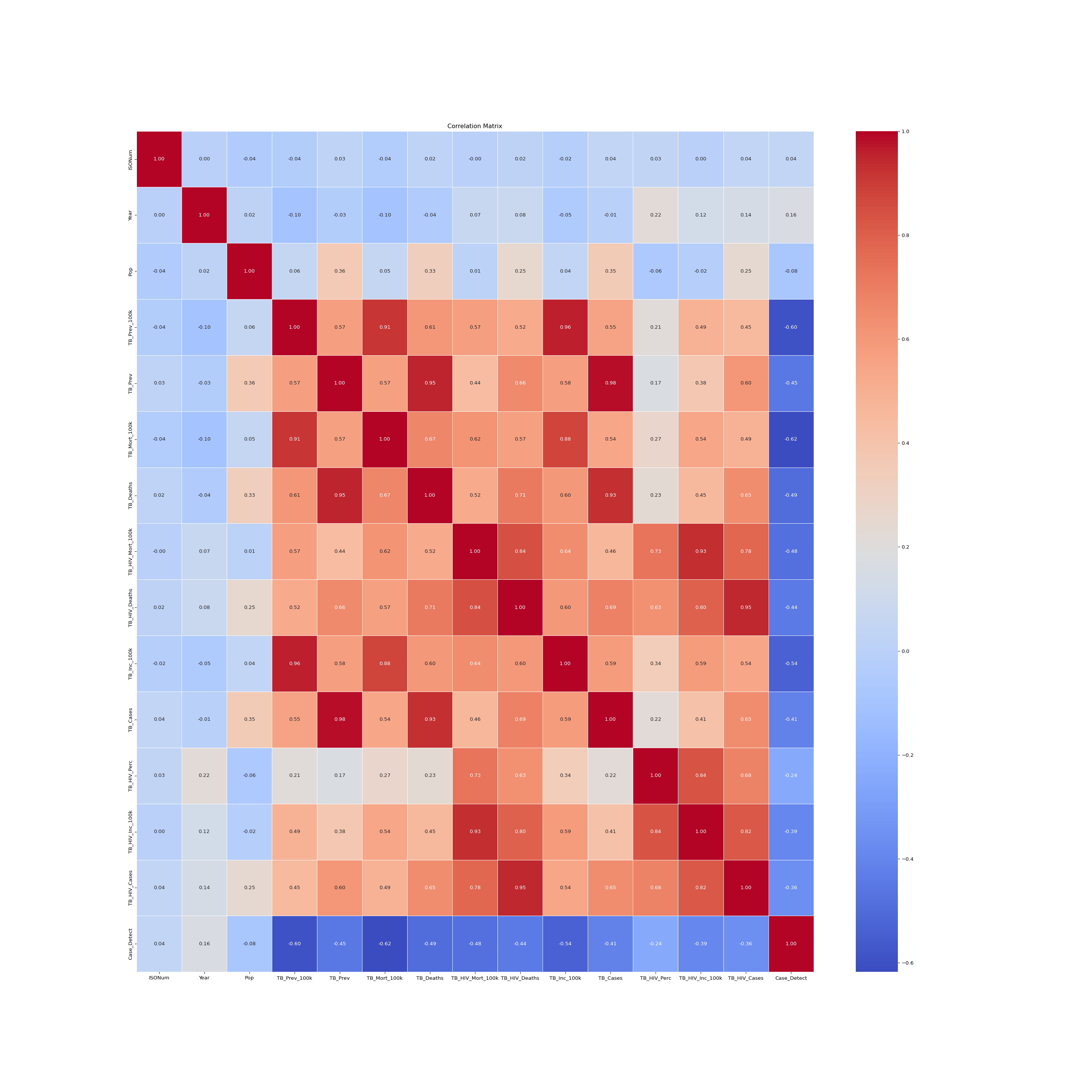
CORRELATION To do the correlation we filterd a data set containg only the cloumns with the numeric values. Additionally we dropped the columns of lower and upper bounds of the a specfic attribute using only the center or actual value for correlating.

tb\_num=tb.select\_dtypes(['int64','float64'])  
tb\_num\_fltr=tb\_num[['ISONum', 'Year', 'Pop', 'TB\_Prev\_100k', 'TB\_Prev', 'TB\_Mort\_100k',  
 'TB\_Deaths', 'TB\_HIV\_Mort\_100k', 'TB\_HIV\_Deaths', 'TB\_Inc\_100k',  
 'TB\_Cases', 'TB\_HIV\_Perc', 'TB\_HIV\_Inc\_100k', 'TB\_HIV\_Cases', 'Case\_Detect']]  
corr=tb\_num\_fltr.corr()  
corr

## ISONum Year ... TB\_HIV\_Cases Case\_Detect  
## ISONum 1.000000 0.001431 ... 0.040039 0.042394  
## Year 0.001431 1.000000 ... 0.142483 0.162380  
## Pop -0.043889 0.020399 ... 0.245424 -0.082731  
## TB\_Prev\_100k -0.035716 -0.103872 ... 0.449611 -0.597419  
## TB\_Prev 0.025750 -0.030544 ... 0.603292 -0.450347  
## TB\_Mort\_100k -0.039392 -0.100422 ... 0.490746 -0.617514  
## TB\_Deaths 0.023879 -0.043000 ... 0.645259 -0.494501  
## TB\_HIV\_Mort\_100k -0.004266 0.065506 ... 0.777133 -0.481307  
## TB\_HIV\_Deaths 0.018532 0.083466 ... 0.948343 -0.442043  
## TB\_Inc\_100k -0.019741 -0.053347 ... 0.540247 -0.535787  
## TB\_Cases 0.040389 -0.005193 ... 0.646063 -0.409702  
## TB\_HIV\_Perc 0.034980 0.222211 ... 0.682762 -0.240075  
## TB\_HIV\_Inc\_100k 0.000968 0.122915 ... 0.822568 -0.392417  
## TB\_HIV\_Cases 0.040039 0.142483 ... 1.000000 -0.361757  
## Case\_Detect 0.042394 0.162380 ... -0.361757 1.000000  
##   
## [15 rows x 15 columns]

visualizing the correlation by means of heatmap

plt.figure(figsize=(30, 30)) # Set figure size  
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5) # Heatmap  
plt.title("Correlation Matrix") # Add title  
plt.show()

sorting the correlation values pulling out rank of correlaation only on there magnitude no sign used in this section

corr = tb\_num\_fltr.corr()  
  
# Convert correlation matrix to a long format (stacked form)  
corr\_pairs = corr.unstack()  
  
# Remove self-correlations (correlation of a variable with itself is always 1)  
corr\_pairs = corr\_pairs[corr\_pairs < 1].drop\_duplicates()  
  
# Sort correlations from highest to lowest  
sorted\_corr\_mag = corr\_pairs.abs().sort\_values(ascending=False)  
  
sorted\_corr\_mag

## TB\_Prev TB\_Cases 0.983871  
## TB\_Prev\_100k TB\_Inc\_100k 0.957268  
## TB\_Prev TB\_Deaths 0.950370  
## TB\_HIV\_Deaths TB\_HIV\_Cases 0.948343  
## TB\_HIV\_Mort\_100k TB\_HIV\_Inc\_100k 0.931728  
## ...   
## Pop TB\_HIV\_Mort\_100k 0.010135  
## Year TB\_Cases 0.005193  
## ISONum TB\_HIV\_Mort\_100k 0.004266  
## Year 0.001431  
## TB\_HIV\_Inc\_100k 0.000968  
## Length: 105, dtype: float64

based on the there signs now its ranked

sorted\_corr\_sign = corr\_pairs.sort\_values(ascending=False)  
sorted\_corr\_sign

## TB\_Prev TB\_Cases 0.983871  
## TB\_Prev\_100k TB\_Inc\_100k 0.957268  
## TB\_Prev TB\_Deaths 0.950370  
## TB\_HIV\_Deaths TB\_HIV\_Cases 0.948343  
## TB\_HIV\_Mort\_100k TB\_HIV\_Inc\_100k 0.931728  
## ...   
## Case\_Detect -0.481307  
## TB\_Deaths Case\_Detect -0.494501  
## TB\_Inc\_100k Case\_Detect -0.535787  
## TB\_Prev\_100k Case\_Detect -0.597419  
## TB\_Mort\_100k Case\_Detect -0.617514  
## Length: 105, dtype: float64

filtering out those with high correlation as high was defined for those scoring above 0.8

high\_corr=sorted\_corr\_sign[sorted\_corr\_sign>0.8]  
high\_corr

## TB\_Prev TB\_Cases 0.983871  
## TB\_Prev\_100k TB\_Inc\_100k 0.957268  
## TB\_Prev TB\_Deaths 0.950370  
## TB\_HIV\_Deaths TB\_HIV\_Cases 0.948343  
## TB\_HIV\_Mort\_100k TB\_HIV\_Inc\_100k 0.931728  
## TB\_Deaths TB\_Cases 0.928760  
## TB\_Prev\_100k TB\_Mort\_100k 0.912788  
## TB\_Mort\_100k TB\_Inc\_100k 0.875446  
## TB\_HIV\_Mort\_100k TB\_HIV\_Deaths 0.843326  
## TB\_HIV\_Perc TB\_HIV\_Inc\_100k 0.835217  
## TB\_HIV\_Inc\_100k TB\_HIV\_Cases 0.822568  
## dtype: float64

DESCRIPTION ON HIGHLY CORRELATED FEATURES The following correlations indicate strong positive relationships between tuberculosis-related variables.

1. **TB Prevalence vs. TB Cases (0.98)**

Meaning: Nearly a perfect correlation. The total number of TB cases strongly determines the overall prevalence.

Implication: TB control efforts must focus on reducing new cases to bring down prevalence.

1. **TB Prevalence per 100k vs. TB Incidence per 100k (0.96)**

Meaning: Countries with a high TB burden also experience high rates of new infections.

Implication: To reduce TB prevalence, new infections must be prevented through better screening and treatment.

1. **TB Prevalence vs. TB Deaths (0.95)**

Meaning: More TB cases lead to higher mortality.

Implication: Countries with high TB prevalence need better healthcare infrastructure to lower TB-related deaths.

1. **TB-HIV Deaths vs. TB-HIV Cases (0.95)**

Meaning: The number of HIV-positive TB cases strongly predicts the number of deaths.

Implication: HIV patients with TB need early treatment to prevent mortality.

1. **TB-HIV Mortality per 100k vs. TB-HIV Incidence per 100k (0.93)**

Meaning: High TB-HIV incidence rates lead to more TB-HIV deaths.

Implication: Integrated TB and HIV treatment is crucial to reducing fatalities.

1. **TB Deaths vs. TB Cases (0.93)**

Meaning: Higher TB case numbers are linked to more deaths.

Implication: Early detection and treatment programs can significantly reduce TB mortality.

1. **TB Prevalence per 100k vs. TB Mortality per 100k (0.91)**

Meaning: Countries with high TB prevalence also have high mortality rates.

Implication: This correlation highlights the need for stronger TB treatment and prevention strategies.

1. **TB Mortality per 100k vs. TB Incidence per 100k (0.87)**

Meaning: Countries with high TB incidence often have high mortality rates.

Implication: This suggests a need for better case management and access to treatment.

1. **TB-HIV Mortality per 100k vs. TB-HIV Deaths (0.84)**

Meaning: More TB-HIV deaths increase the overall TB-HIV mortality rate.

Implication: Strengthening TB-HIV co-treatment programs is critical.

1. **TB-HIV Percentage vs. TB-HIV Incidence per 100k (0.83)**

Meaning: Higher HIV-TB incidence leads to a higher percentage of TB cases being HIV-positive.

Implication: Countries with high TB-HIV burden should prioritize ART (antiretroviral therapy) and TB prevention.

1. **TB-HIV Incidence per 100k vs. TB-HIV Cases (0.82)**

Meaning: More HIV-positive TB cases indicate a higher incidence rate.

Implication: This supports the need for improved HIV testing among TB patients.

Key Takeaways

TB Prevalence is closely tied to new cases, deaths, and mortality.

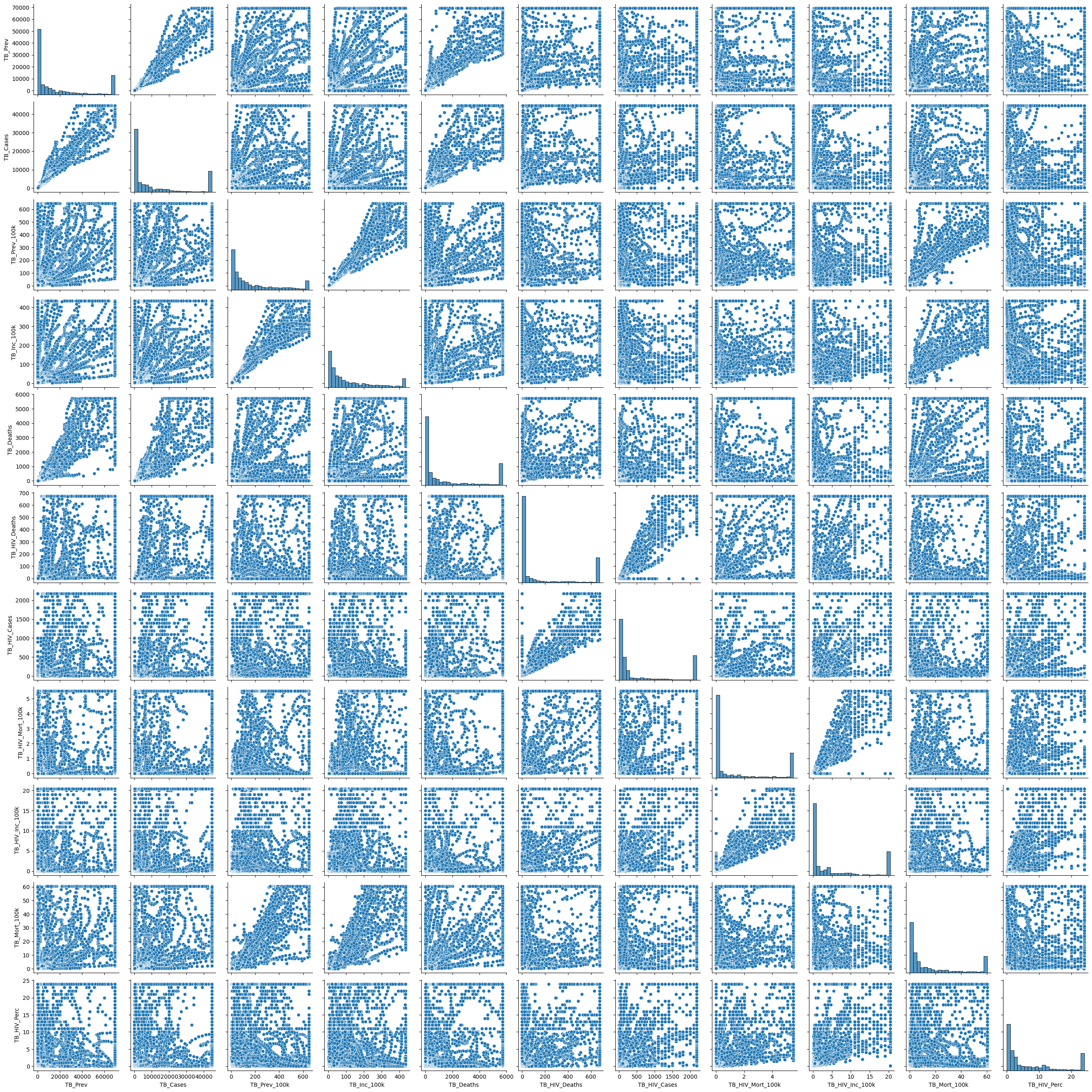
HIV-associated TB significantly increases TB mortality rates.

Reducing TB incidence is critical for lowering overall TB burden and deaths.

Integrated TB-HIV programs are necessary to address co-infection risks.

VISUALIZING PAIRPLOTS OF CORRELATIONS.

sns.pairplot(data=tb\_num\_fltr,vars=["TB\_Prev", "TB\_Cases", "TB\_Prev\_100k", "TB\_Inc\_100k", "TB\_Deaths","TB\_HIV\_Deaths", "TB\_HIV\_Cases", "TB\_HIV\_Mort\_100k", "TB\_HIV\_Inc\_100k","TB\_Mort\_100k", "TB\_HIV\_Perc"])



plt.show()

**ANNOVA AND REGRESSION**

This section of our assignment was done by using r codes and the following is the description.

tb=read.csv('https://raw.githubusercontent.com/Jobmrtall/MSD/refs/heads/main/tb\_renamed.csv')

## One-Way ANOVA

We perform an ANOVA test to determine if the mean TB prevalence differs across regions.

# Perform one-way ANOVA  
aov\_model <- aov(TB\_Prev\_100k ~ Region, data = tb)  
  
# Display ANOVA summary  
summary(aov\_model)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Region 5 59751395 11950279 443.3 <2e-16 \*\*\*  
## Residuals 4996 134686737 26959   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Assumption Checks for ANOVA

### 1. Normality of Residuals

We check if the residuals follow a normal distribution using the Shapiro-Wilk test and a Q-Q plot.

# Check sample size and choose appropriate normality test  
res <- residuals(aov\_model)  
  
if (length(res) >= 3 & length(res) <= 5000) {  
 shapiro.test(res) # Shapiro-Wilk test  
} else if (length(res) > 5000) {  
 library(nortest)  
 ad.test(res) # Anderson-Darling test for large datasets  
} else {  
 print("Sample size too small for normality test.")  
}

##   
## Anderson-Darling normality test  
##   
## data: res  
## A = 167.59, p-value < 2.2e-16

1. Normality Check

A. **Shapiro-Wilk test**

Shapiro-Wilk normality test showed data: residuals

W = 0.956, p-value = 2.2e-16

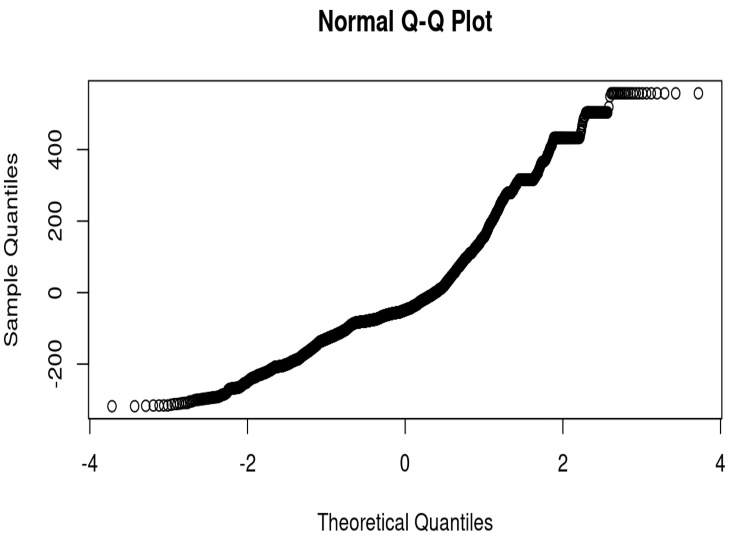
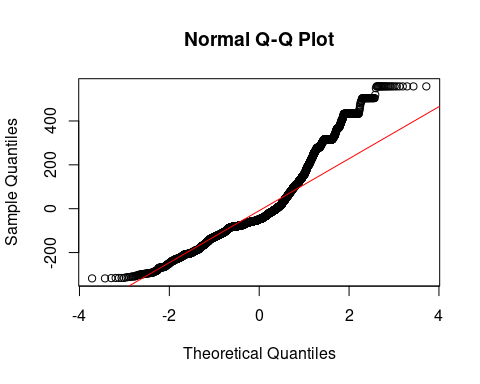
Conclusion\*: Non-normal residuals (`p < 0.05`).

If `p > 0.05`, residuals are normally distributed.

If `p < 0.05`, reject normality (use non-parametric tests like Kruskal-Wallis).

The Shapiro-Wilk test helps determine if the residuals are normally distributed. A p-value greater than 0.05 suggests normality.

# Q-Q plot for residuals  
qqnorm(residuals(aov\_model))  
qqline(residuals(aov\_model), col = "red")

B) **Q-Q plot** -is used to visually assess normality. If the points lie along the red reference line, normality is assumed.

### 

### 2. Homogeneity of Variances

We use Levene’s test to check if the variance across groups is equal.

It test for Homogeneity of Variance. As showen the result the P< 0.001.

Concnclusion: Unequal variances (`p < 0.05`).

library(car) # Load the 'car' package for Levene's test

## Loading required package: carData

leveneTest(TB\_Prev\_100k ~ Region, data = tb)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.

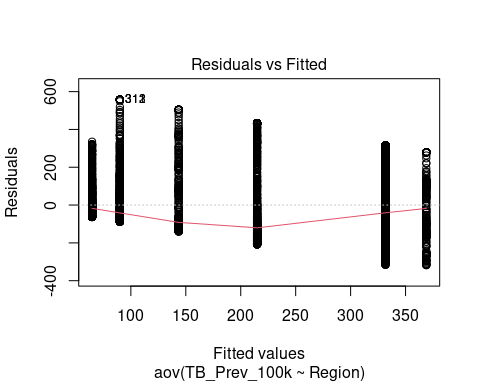
## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 5 144.4 < 2.2e-16 \*\*\*  
## 4996   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Levene’s test assesses the equality of variances. A p-value greater than 0.05 indicates homogeneity of variance.

### 3. Residuals vs. Fitted Values Plot

We plot the residuals against fitted values to check for homoscedasticity.

plot(aov\_model, which = 1)



This plot helps assess whether residuals are randomly scattered around zero, which is an assumption of ANOVA.

## Multiple Linear Regression

We perform a multiple linear regression to predict TB prevalence based on multiple independent variables.

# Multiple linear regression model  
multi\_model <- lm(  
 TB\_Prev\_100k ~ Year + Region + TB\_Inc\_100k +   
 TB\_HIV\_Inc\_100k + TB\_Mort\_100k,  
 data = tb  
)

The linear model includes predictors such as Year, Region, TB Incidence, HIV percentage in incident TB cases, and TB Mortality to predict TB prevalence.

# Display summary of the regression model  
summary(multi\_model)

##   
## Call:  
## lm(formula = TB\_Prev\_100k ~ Year + Region + TB\_Inc\_100k + TB\_HIV\_Inc\_100k +   
## TB\_Mort\_100k, data = tb)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -202.834 -14.212 -4.991 14.029 294.508   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1101.94546 184.13229 5.985 2.32e-09 \*\*\*  
## Year -0.55762 0.09211 -6.054 1.52e-09 \*\*\*  
## RegionAMR 24.62711 2.47705 9.942 < 2e-16 \*\*\*  
## RegionEMR 16.49458 2.95429 5.583 2.49e-08 \*\*\*  
## RegionEUR 24.41967 2.68840 9.083 < 2e-16 \*\*\*  
## RegionSEA 15.07385 3.43128 4.393 1.14e-05 \*\*\*  
## RegionWPR 27.55162 2.74128 10.051 < 2e-16 \*\*\*  
## TB\_Inc\_100k 1.10330 0.01046 105.461 < 2e-16 \*\*\*  
## TB\_HIV\_Inc\_100k -2.23186 0.14353 -15.549 < 2e-16 \*\*\*  
## TB\_Mort\_100k 3.32796 0.06524 51.009 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 43.41 on 4992 degrees of freedom  
## Multiple R-squared: 0.9516, Adjusted R-squared: 0.9515   
## F-statistic: 1.091e+04 on 9 and 4992 DF, p-value: < 2.2e-16

The summary provides estimates, standard errors, t-values, p-values, and R-squared values to evaluate model performance.

When we see the implications

The prevalence of tuberculosis is predicated by region(European Region (EUR), the Eastern Mediterranean Region (EMR), and the Western Pacific Region (WPR)) , HIV in incident cases , incidence of Tb, and Mortality of Tb(P<0.001).

**Strong positive predictors**

* European Region (EUR)
* Eastern Mediterranean Region (EMR)
* Western Pacific Region (WPR)
* Incidence of Tb, and
* Mortality of Tb

**Negative strong predictor**

* HIV in incident cases
* Year(Trend over time]

**Result Interpretation**

The regression model examines factors influencing the Prevalence of TB per 100,000 population. The model demonstrates strong explanatory power, with an R-squared value of 0.954, indicating that 95.4% of the variance in TB prevalence is explained by the predictors.

1. Temporal Trend (Year)

- A significant negative relationship exists between Year and TB prevalence (β = -0.23, p = 0.0115). This suggests a slight annual decline in TB prevalence over time.

2. Regional Differences

- The Region variable (likely categorical) shows significant variations compared to the reference region. For example: Region such as the Region of the Americas (AMR), the European Region (EUR), the Eastern Mediterranean Region (EMR), and the Western Pacific Region (WPR) has coefficients ranging from 6.97 to 28.87 (all p < 0.05), indicating substantially higher TB prevalence in these regions. Regional SEA (e.g., Southeast Asia) has a coefficient of 6.72 (p = 0.0424), also reflecting elevated prevalence.

3. Incidence of TB

- A strong positive relationship exists between TB incidence and prevalence (β = 1.11, p < 2e-16). A 1-unit increase in incidence corresponds to a 1.11-unit rise in prevalence, highlighting its critical role.

4. HIV in Incident TB Cases

- Unexpectedly, higher HIV co-infection percentages correlate with lower TB prevalence (β = -2.61, p < 2e-16). This may reflect higher mortality rates among HIV-TB co-infected individuals, reducing the pool of prevalent cases.

5. Mortality of TB Cases

- Mortality shows a strong positive association with prevalence (β = 3.15, p < 2e-16). This could indicate regions with high prevalence also face challenges in TB management, leading to elevated mortality.