

# HEALTHCARE FINANCING

*Data Analysis*



# MAIN GOAL

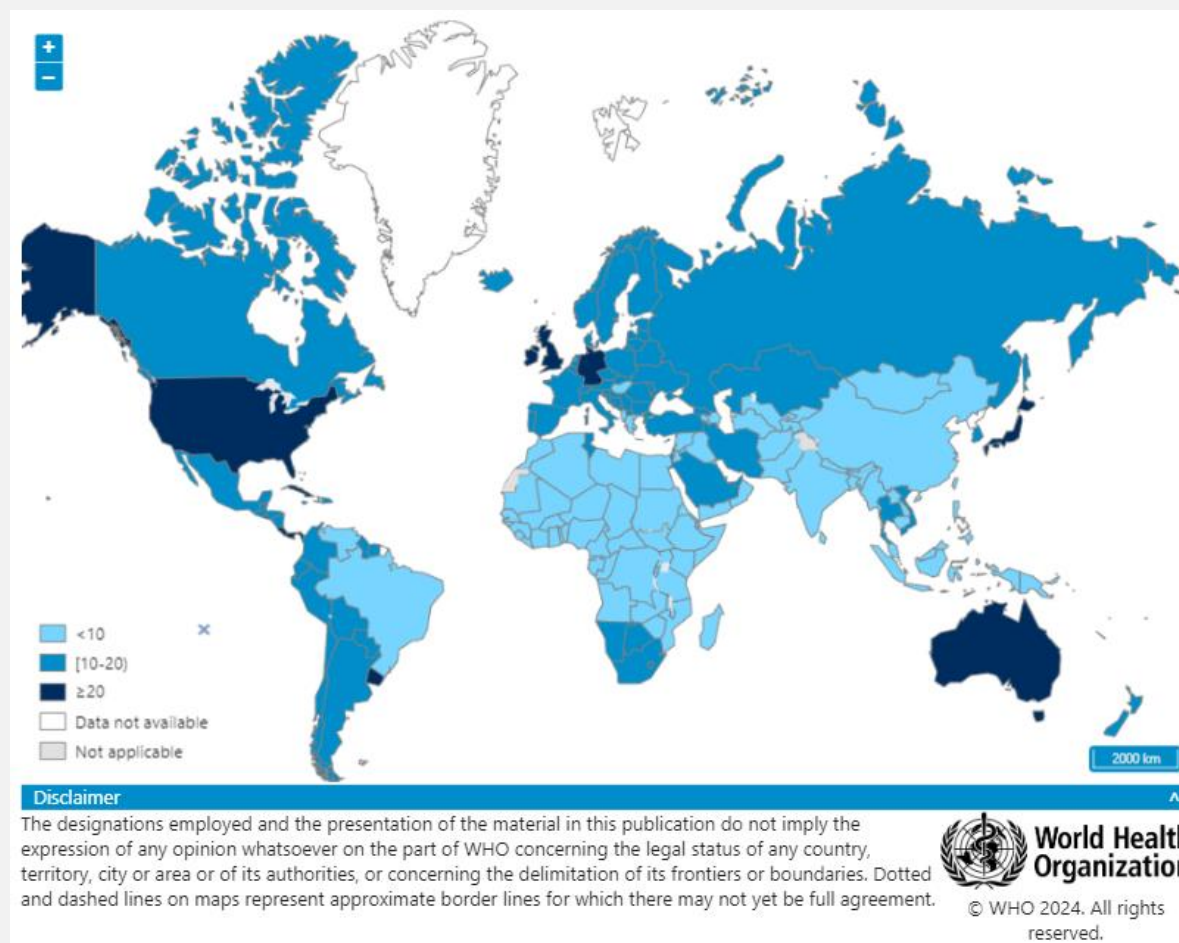
---

Our mission is to get summary of the data and insights that clarifies the expenditure change across years for the same region and the difference between regions in different years using data analysis.

# DATA DESCRIPTION

## Overview

Identify the health expenditure amount per capita across several world regions per year from 2000 till 2022.



# DATA DESCRIPTION

## Exploring Data

### Importing Libraries needed

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
import matplotlib.cm as cm
```

✓ 7.5s

Python

# DATA DESCRIPTION

## Exploring Data

### Loading the dataset and exploring the data

```
# Import Data file
che_per_capita = pd.read_csv("D:\\GHED_CHE_pc_US_SHA2011.csv")

# Explore first 5 rows of the data
che_per_capita.head()
```

✓ 0.5s

Python

```
# Explore last 5 rows of the data
che_per_capita.tail()
```

✓ 0.1s

Python



# DATA DESCRIPTION

## Exploring Data

Showing first 5 rows of the data

	Id	IndicatorCode	SpatialDimension	SpatialDimensionValueCode	ParentLocationCode	ParentLocation	TimeDimension	TimeDim	DisaggregatingDimension1
0	857	GHED_CHE_pc_US_SHA2011	COUNTRY	IRL	EUR	Europe	YEAR	2017	NaN
1	920	GHED_CHE_pc_US_SHA2011	COUNTRY	MYS	WPR	Western Pacific	YEAR	2014	NaN
2	1426	GHED_CHE_pc_US_SHA2011	COUNTRY	DEU	EUR	Europe	YEAR	2018	NaN
3	2061	GHED_CHE_pc_US_SHA2011	COUNTRY	LAO	WPR	Western Pacific	YEAR	2010	NaN
4	4070	GHED_CHE_pc_US_SHA2011	COUNTRY	TUR	EUR	Europe	YEAR	2014	NaN

5 rows × 25 columns

n1ValueCode	...	DataSourceDimensionValueCode	Value	NumericValue	Low	High	Comments	Date	TimeDimensionValue	TimeDimensionBegin	TimeDimensionEnd
NaN	...	NaN	5010.26	5010.25635	NaN	NaN	NaN	2024-12-10T14:55:59Z	2017	2017-01-01	2017-12-31
NaN	...	NaN	414.44	414.43924	NaN	NaN	NaN	2024-12-10T14:55:59Z	2014	2014-01-01	2014-12-31
NaN	...	NaN	5507.68	5507.67676	NaN	NaN	NaN	2024-12-10T14:55:59Z	2018	2018-01-01	2018-12-31
NaN	...	NaN	34.52	34.52461	NaN	NaN	NaN	2024-12-10T14:55:59Z	2010	2010-01-01	2010-12-31
NaN	...	NaN	520.41	520.40753	NaN	NaN	NaN	2024-12-10T14:55:59Z	2014	2014-01-01	2014-12-31

# DATA DESCRIPTION

## Exploring Data

### Exploring the data summary

```
# Show data summary  
che_per_capita.info()
```

✓ 0.1s

Python

# DATA DESCRIPTION

## Exploring Data

### Data Summary

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4637 entries, 0 to 4636
Data columns (total 25 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Id                                           4637 non-null   int64
1   IndicatorCode                               4637 non-null   object
2   SpatialDimension                           4637 non-null   object
3   SpatialDimensionValueCode                  4637 non-null   object
4   ParentLocationCode                         4384 non-null   object
5   ParentLocation                             4384 non-null   object
6   TimeDimension                              4637 non-null   object
7   TimeDim                                    4637 non-null   int64
8   DisaggregatingDimension1                   0 non-null      float64
9   DisaggregatingDimension1ValueCode           0 non-null      float64
10  DisaggregatingDimension2                    0 non-null      float64
11  DisaggregatingDimension2ValueCode            0 non-null      float64
12  DisaggregatingDimension3                    0 non-null      float64
13  DisaggregatingDimension3ValueCode            0 non-null      float64
14  DataSourceDimension                         0 non-null      float64
15  DataSourceDimensionValueCode                 0 non-null      float64
16  Value                                        4635 non-null   float64
17  NumericValue                                4635 non-null   float64
18  Low                                           0 non-null      float64
19  High                                           0 non-null      float64
20  Comments                                     0 non-null      float64
21  Date                                          4637 non-null   object
22  TimeDimensionValue                          4637 non-null   int64
23  TimeDimensionBegin                          4637 non-null   object
24  TimeDimensionEnd                            4637 non-null   object
dtypes: float64(13), int64(3), object(9)
memory usage: 905.8+ KB
```

Total number of values for each column

Total number of valid data types for each column

Columns with some empty values

Complete empty columns



# DATA PRE-PROCESSING

## Data Cleaning

### Checking number of empty columns and deleting them

```
# Check number of empty columns
che_per_capita.isna().all().sum()
```

Python

11

```
# Delete complete empty columns
che_per_capita_U1 = che_per_capita.dropna(axis=1, how='all')
che_per_capita_U1.shape
```

Python

(4637, 14)

Number of columns after  
deleting empty ones

Columns' quantity changed from 25 to 14 after cleaning empty columns.

# DATA PRE-PROCESSING

## Data Cleaning

### Cleaning columns with unique values

```
# Checking columns with only one unique value
che_per_capita_U1.loc[:,che_per_capita_U1.apply(pd.Series.nunique, axis=0) == 1]
```

✓ 0.1s

Python

	IndicatorCode	TimeDimension	Date
0	GHED_CHE_pc_US_SHA2011	YEAR	2024-12-10T14:55:59Z
1	GHED_CHE_pc_US_SHA2011	YEAR	2024-12-10T14:55:59Z
2	GHED_CHE_pc_US_SHA2011	YEAR	2024-12-10T14:55:59Z
3	GHED_CHE_pc_US_SHA2011	YEAR	2024-12-10T14:55:59Z
4	GHED_CHE_pc_US_SHA2011	YEAR	2024-12-10T14:55:59Z

Deleting these 3  
columns



```
# Delete columns with only one unique value
che_per_capita_U1 = che_per_capita_U1.loc[:,che_per_capita_U1.apply(pd.Series.nunique, axis=0) != 1]
```

✓ 0.0s

Python

# DATA PRE-PROCESSING

## Data Cleaning

### Cleaning columns with same values (Duplicate columns)

```
# Check if there are any duplicate columns

# Calculate number of duplicate columns
duplicate_columns = che_per_capita_U1.T.duplicated(keep=False)

# Print the duplicate columns
print("Duplicate columns:")
for col in che_per_capita_U1.columns[duplicate_columns]:
    print(col)
```

✓ 1.5s

Python

```
Duplicate columns:
TimeDim
TimeDimensionValue
```

```
#Removing duplicate columns
che_per_capita_U1 = che_per_capita_U1.loc[:, ~che_per_capita_U1.T.duplicated(keep='first')]
che_per_capita_U1.shape
```

✓ 1.0s

Python

```
(4637, 10)
```

# DATA PRE-PROCESSING

## Data Cleaning

Cleaning un-needed columns (columns DO NOT affect the data)

```
# Delete un-needed columns
che_per_capita_U2 = che_per_capita_U1.drop(['Id', 'NumericValue', 'TimeDimensionBegin', 'TimeDimensionEnd'], axis=1)
che_per_capita_U2
```

Python

	SpatialDimension	SpatialDimensionValueCode	ParentLocationCode	ParentLocation	TimeDimension	TimeDim	Value
0	COUNTRY	IRL	EUR	Europe	YEAR	2017	5010.26
1	COUNTRY	MYS	WPR	Western Pacific	YEAR	2014	414.44
2	COUNTRY	DEU	EUR	Europe	YEAR	2018	5507.68
3	COUNTRY	LAO	WPR	Western Pacific	YEAR	2010	34.52
4	COUNTRY	TUR	EUR	Europe	YEAR	2014	520.41
...	...	...	...	...	...	...	...
4632	COUNTRY	QAT	EMR	Eastern Mediterranean	YEAR	2010	1309.79
4633	COUNTRY	CUB	AMR	Americas	YEAR	2018	981.90
4634	COUNTRY	HTI	AMR	Americas	YEAR	2016	60.07
4635	COUNTRY	TUV	WPR	Western Pacific	YEAR	2022	1084.67
4636	COUNTRY	POL	EUR	Europe	YEAR	2010	809.20

4637 rows × 7 columns

# DATA PRE-PROCESSING

## Data Cleaning

### Checking duplicate or empty rows

```
# Check duplicate rows  
che_per_capita.duplicated().sum()
```

✓ 0.1s

Python

0

```
# Check number of empty rows  
len(che_per_capita[che_per_capita.isna().all(axis=1)])
```

✓ 0.0s

Python

0

# DATA PRE-PROCESSING

## Data Cleaning

### Cleaning un-needed rows (rows with inhomogeneous values)

```
che_per_capita_U4 = che_per_capita_U3[(che_per_capita_U3['SpatialDimension']=='COUNTRY')]  
che_per_capita_U4.info()
```

Python

```
<class 'pandas.core.frame.DataFrame'>  
Index: 4384 entries, 0 to 4636  
Data columns (total 6 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   SpatialDimension       4384 non-null   object  
1   SpatialDimensionValueCode 4384 non-null   object  
2   ParentLocationCode      4384 non-null   object  
3   TimeDimension           4384 non-null   object  
4   TimeDim                 4384 non-null   int64  
5   Value                   4384 non-null   float64  
dtypes: float64(1), int64(1), object(4)  
memory usage: 239.8+ KB
```



# DATA PRE-PROCESSING

## Data Cleaning

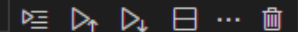
### Fill missing data in the dependent column

```
# Show the rows with empty cell in "Value" column
che_per_capita_U2[che_per_capita_U2['Value'].isna()]
```

✓ 0.0s

Python

	SpatialDimension	SpatialDimensionValueCode	ParentLocationCode	ParentLocation	TimeDim	Value
2176	COUNTRY	CUB	AMR	Americas	2022	NaN
4227	COUNTRY	CUB	AMR	Americas	2021	NaN



```
# Calculating the mean value for all observations at Cuba "CUB" country
mean_value_cub = che_per_capita_U2[che_per_capita_U2['SpatialDimensionValueCode']=='CUB'][['Value']].mean()
mean_value_cub
```

✓ 0.0s

Python

```
Value      632.68
dtype: float64
```

```
# Fill missing data in 'Value' column for Cuba country with mean value for Cuba country across years
che_per_capita_U2['Value'] = che_per_capita_U2['Value'].fillna(float(mean_value_cub))
print(che_per_capita_U2['Value'].isna().sum())
```

✓ 0.0s

Python

0

# DESCRIPTIVE ANALYSIS

## Exploratory Data Analysis (EDA)

### Summary Statistics of Numerical Columns

```
che_per_capita_U4.describe()
```

✓ 0.0s

Python

	TimeDim	Value
count	4384.000000	4384.000000
mean	2011.082573	948.149466
std	6.630224	1668.315155
min	2000.000000	4.180000
25%	2005.000000	64.210000
50%	2011.000000	264.850000
75%	2017.000000	886.402500
max	2022.000000	12434.430000

# DESCRIPTIVE ANALYSIS

## Exploratory Data Analysis (EDA)

### Summary Statistics of Dependent Variable (“Value” Column) across Regions

```
che_per_capita_U4.groupby('ParentLocationCode')[['Value']].describe()
```

✓ 0.2s

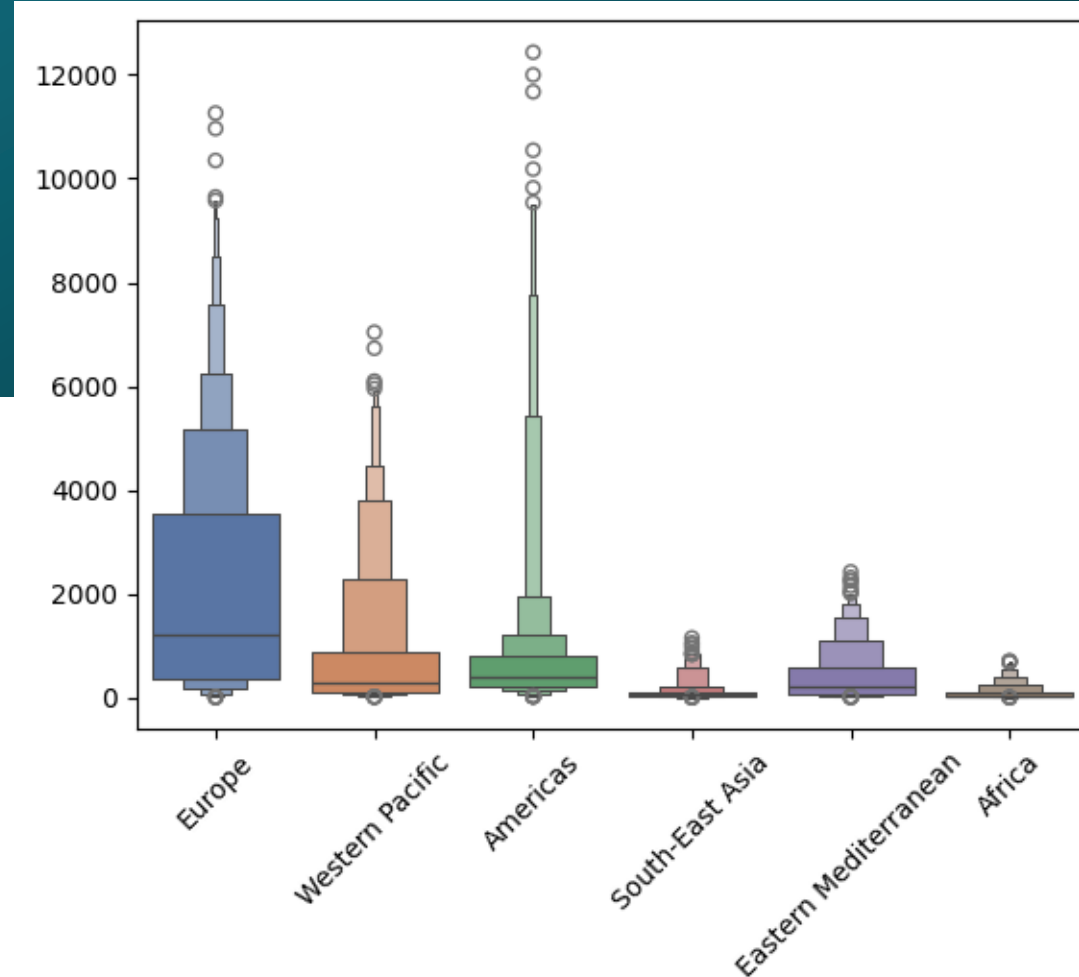
Python

ParentLocationCode	Value							
	count	mean	std	min	25%	50%	75%	max
Africa	1054.0	102.647059	139.531775	4.18	25.9100	47.83	92.855	708.70
Americas	787.0	839.983266	1552.294926	18.61	223.4850	389.35	814.795	12434.43
Eastern Mediterranean	488.0	427.491086	517.375713	11.72	64.6725	212.27	567.940	2422.64
Europe	1207.0	2168.517407	2275.790802	5.91	349.0700	1194.86	3554.560	11262.23
South-East Asia	227.0	130.494141	202.997187	4.47	33.0750	59.75	118.355	1150.69
Western Pacific	621.0	856.347987	1300.908099	12.91	99.5100	285.64	880.840	7039.58

# DESCRIPTIVE ANALYSIS

## Exploratory Data Analysis (EDA)

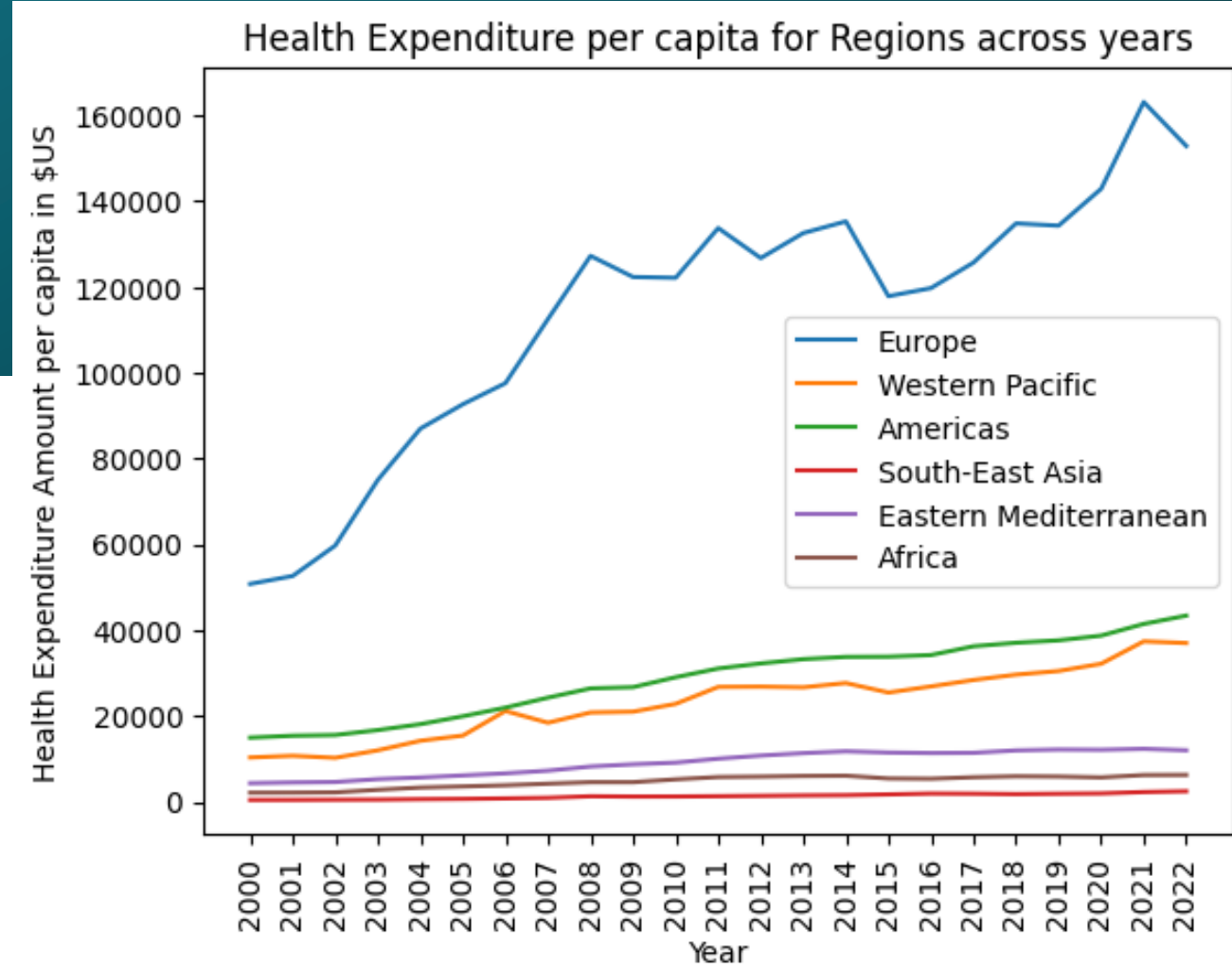
Plotting Summary Statistics of Dependent Variable (“Value” Column) across different Regions



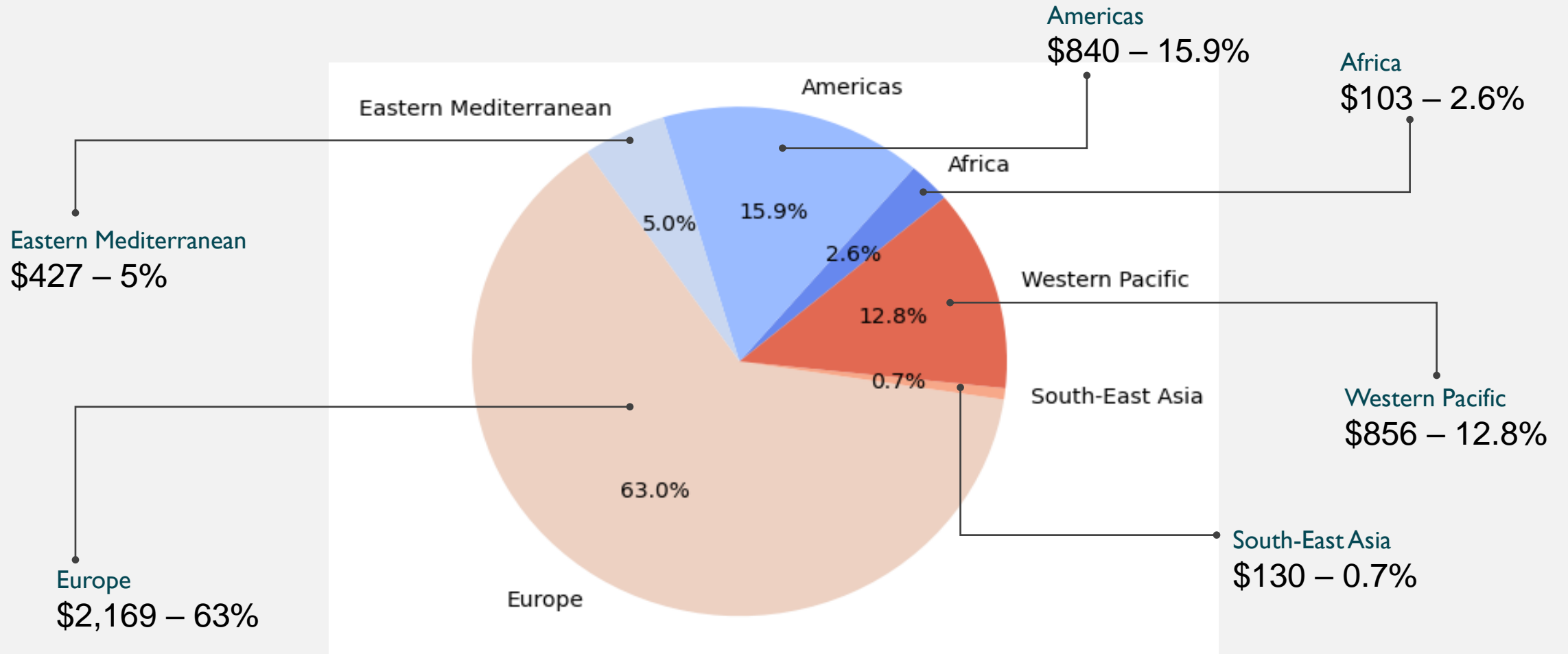
# DESCRIPTIVE ANALYSIS

## Exploratory Data Analysis (EDA)

Health Expenditure per capita Change across years for each region



# AVERAGE OF HEALTH EXPENDITURE PER CAPITA % BY REGIONS PER YEAR



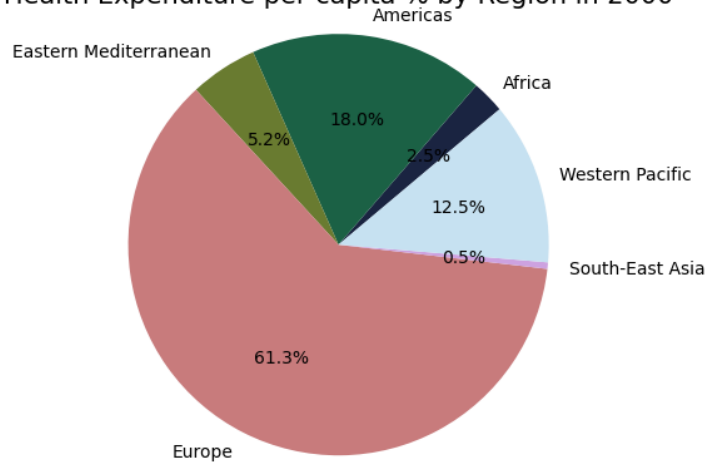


# DESCRIPTIVE ANALYSIS

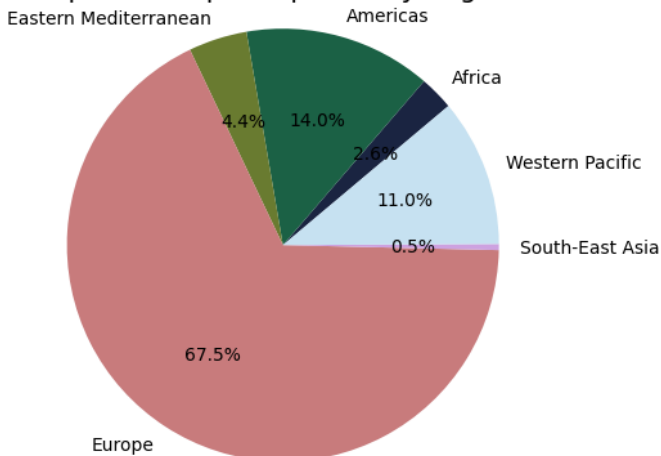
## Exploratory Data Analysis (EDA)

### Health Expenditure per capita Percentage by Regions in different Years

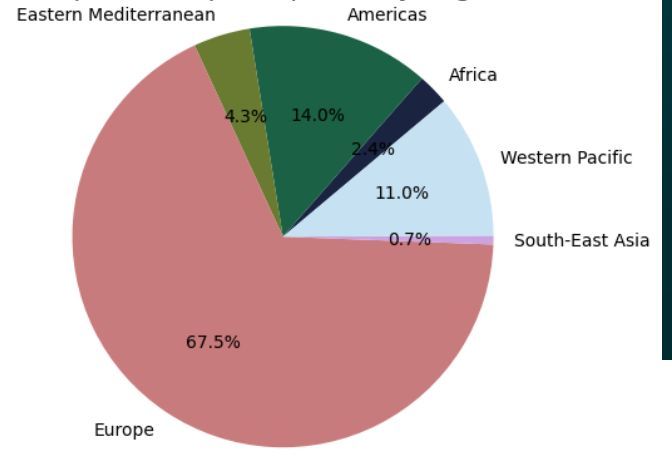
Health Expenditure per capita % by Region in 2000



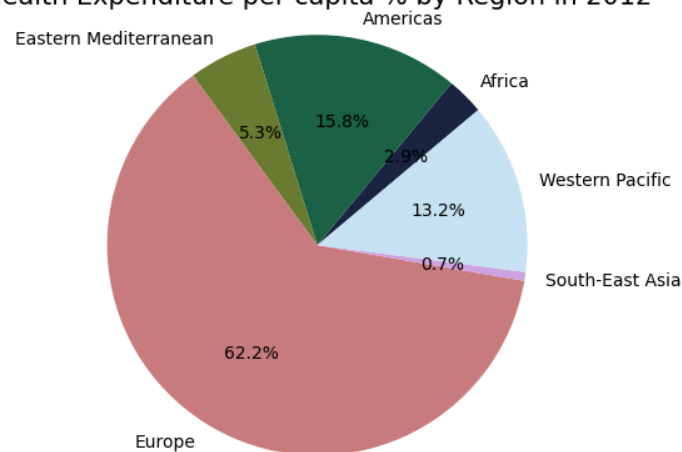
Health Expenditure per capita % by Region in 2004



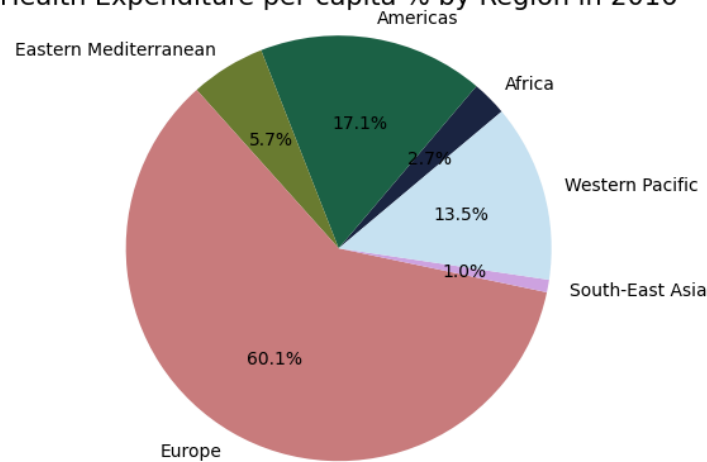
Health Expenditure per capita % by Region in 2008



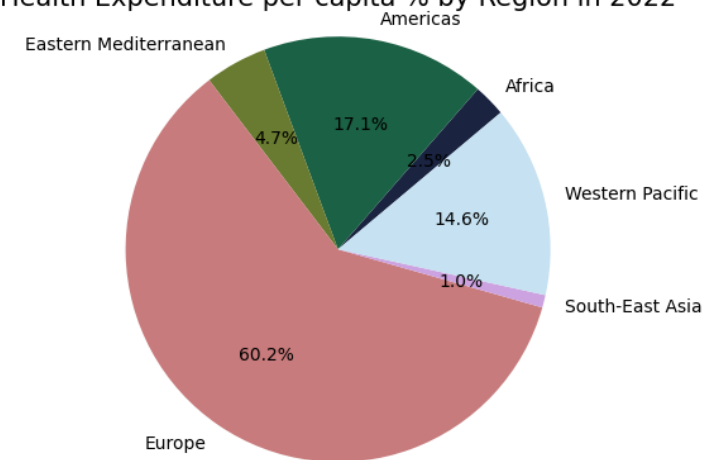
Health Expenditure per capita % by Region in 2012



Health Expenditure per capita % by Region in 2016



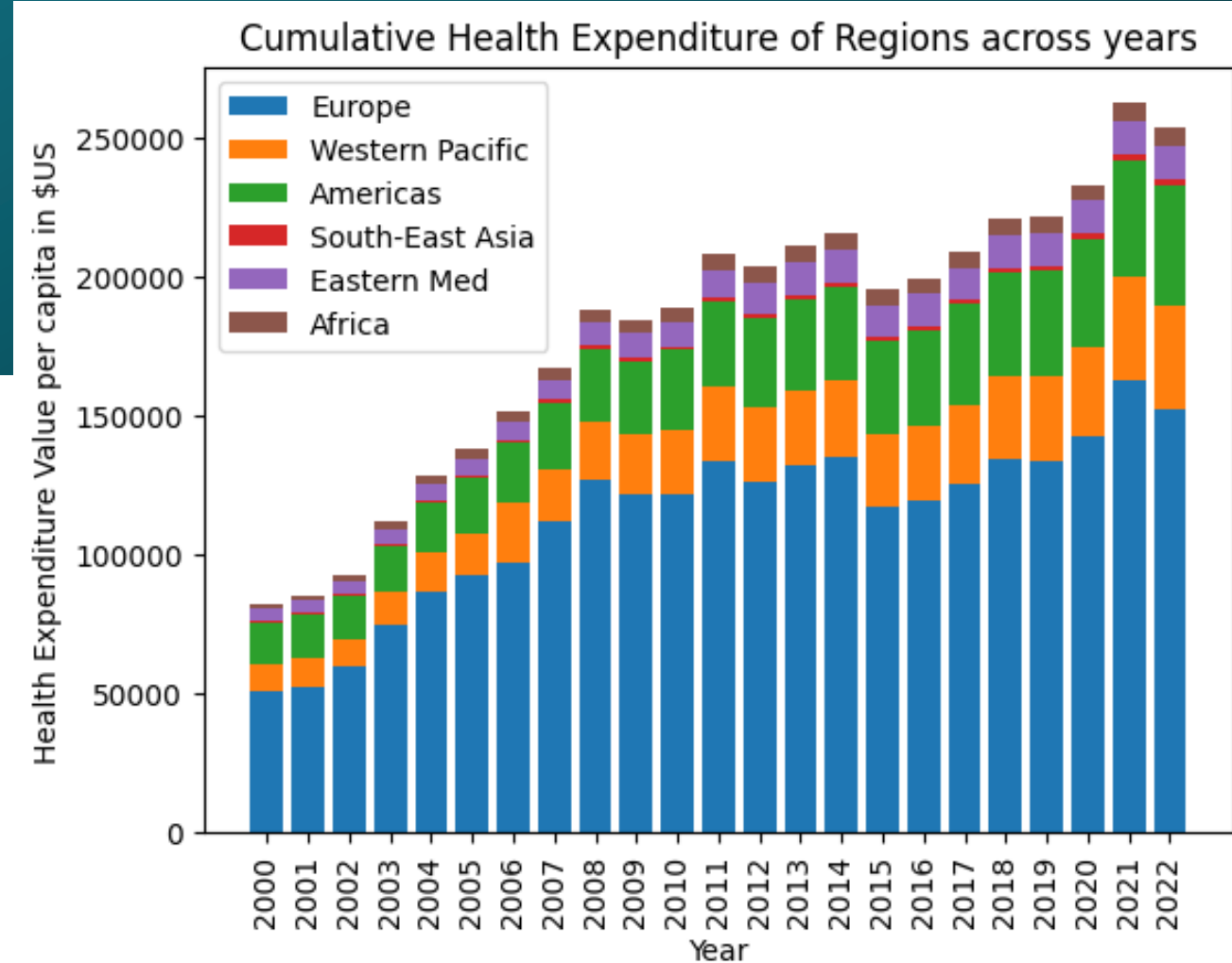
Health Expenditure per capita % by Region in 2022



# DESCRIPTIVE ANALYSIS

## Exploratory Data Analysis (EDA)

### Cumulative Health Expenditure per capita of Regions across Years

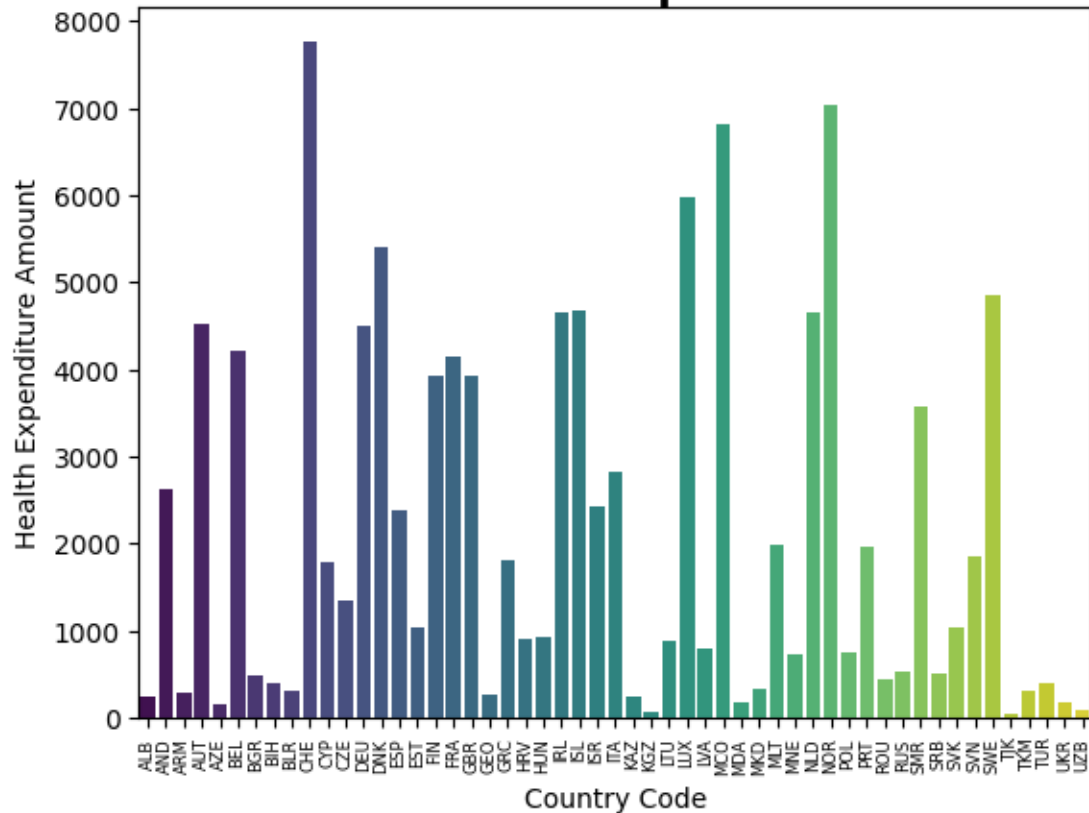


# DESCRIPTIVE ANALYSIS

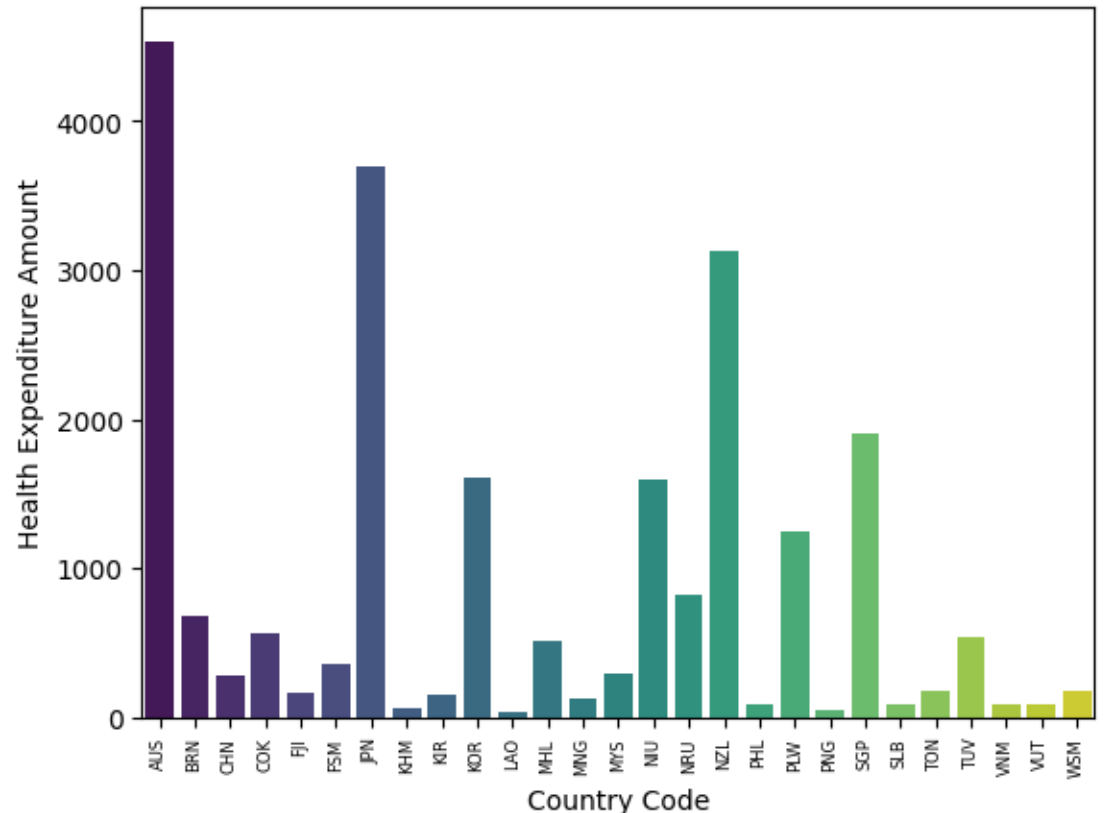
## Exploratory Data Analysis (EDA)

Average Health Expenditure Amount per Capita for Regions' Countries per year

### Europe



### Western Pacific

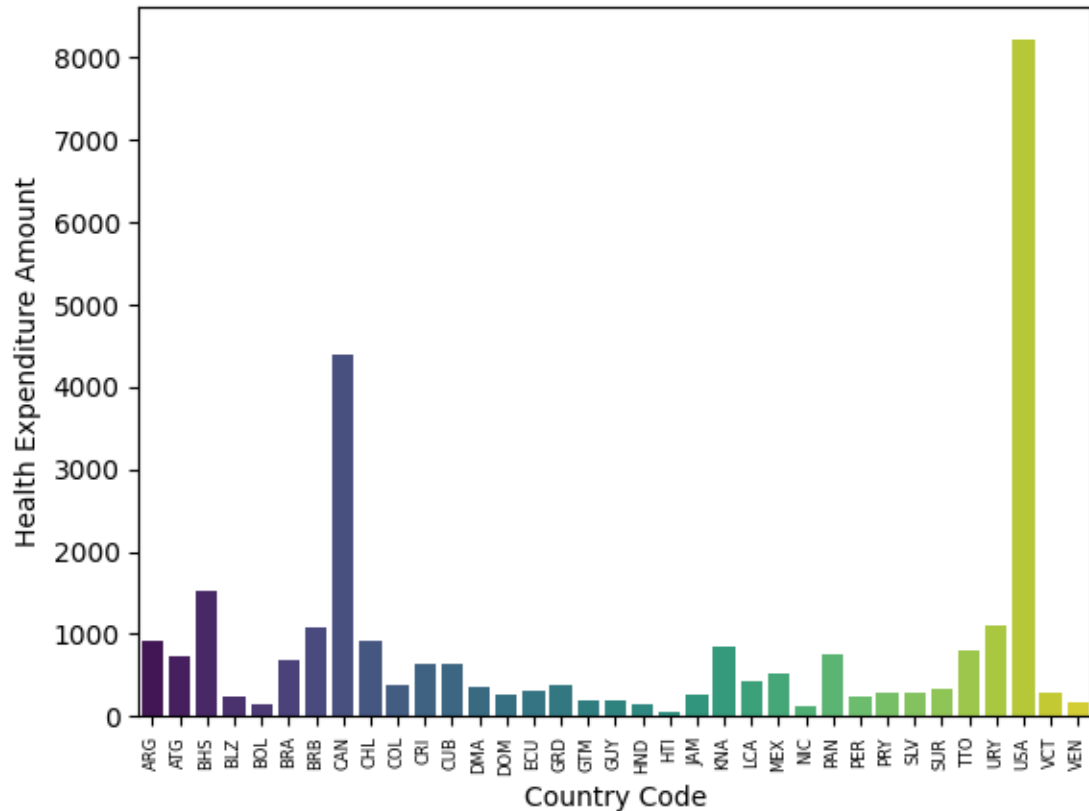


# DESCRIPTIVE ANALYSIS

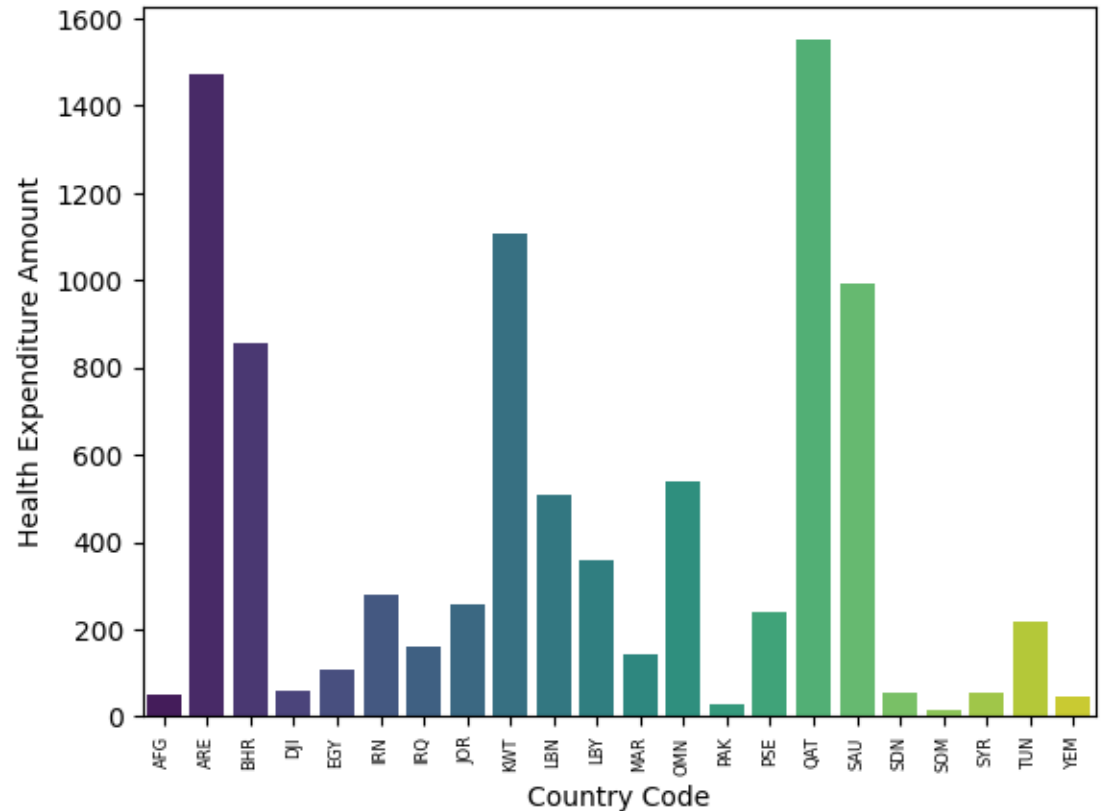
## Exploratory Data Analysis (EDA)

Average Health Expenditure Amount per Capita for Regions' Countries per year - Continued

### Americas



### Eastern Mediterranean

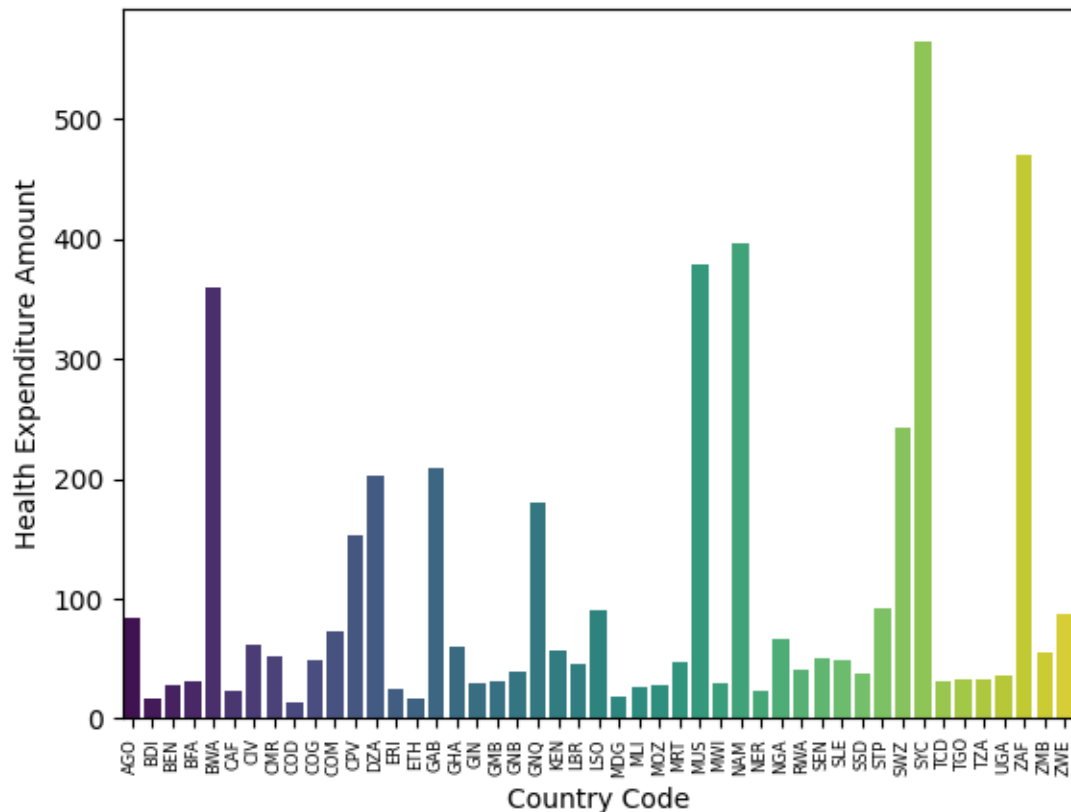


# DESCRIPTIVE ANALYSIS

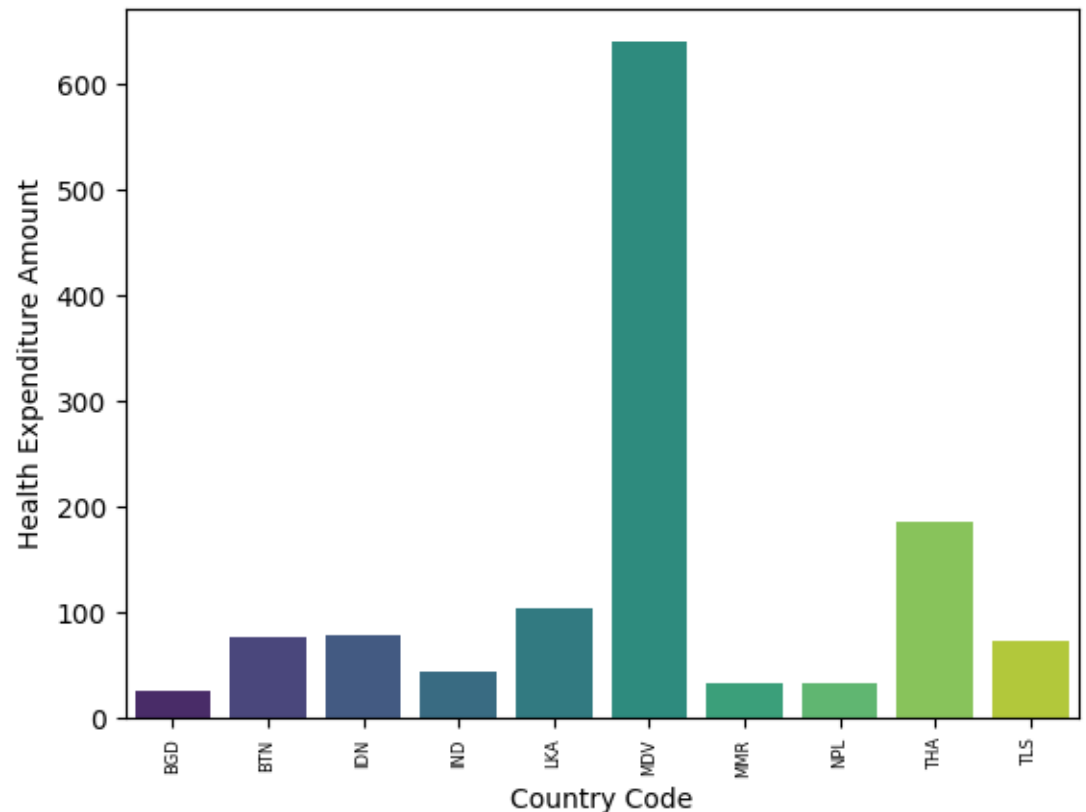
## Exploratory Data Analysis (EDA)

### Average Health Expenditure Amount per Capita for Regions' Countries per year - Continued

#### Africa



#### South-East Asia



# THANK YOU



Marwa Aboelenin



marwamostafa92@hotmail.com



0106-066-4560