Configuration Manual

MSc Research Project
Fintech

Valentina Luciana Díaz Estévez
Student ID: 22243445

School of Computing
National College of Ireland

Supervisor: Faithful Onwuegbuche

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	……. ……………………… Valentina Luciana Díaz Estévez ……………………………
Student ID:	………………………22243445…………….………………………………………………..…….
Programme:	…………………… Fintech …………………… **Year:** ………2023/2024…….
Module:	……………………………………… MSC Research Project ………….……………….….
Lecturer:	………………………………… Faithful Onwuegbuche ……………………….………
Submission Due Date:	……………………12/08/2024………………………………………………….………
Project Title:	Determinants of Financial Inclusion in Argentina and Ireland: A Comparative Analysis
Word Count:	………………834………… **Page Count:** ……………………10……..…..………

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……………………Valentina Diaz Estevez……………………………………

Date: …………12/08/2024…………………….…………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Valentina Luciana Díaz Estévez
Student ID: 22243445

1 Introduction

This user configuration handbook provides a full, sequential description of required elements for the product and method, which are necessary elements of the study project titled "Determinants of Financial Inclusion in Argentina and Ireland: A Comparative Perspective". The procedures given include the hardware and software requirements as well. Furthermore, the handbook includes exemplary code snippets used in various models, as well as their associated results, all with the goal of providing practical instruction.

2 Data Gathering

- This is the first dataset on financial inclusion indicators for Argentina and Ireland. It has 37 indicators combined, each one representing different financial parameters. The data comes from the IMF's FAS and the World Bank's Global Financial Inclusion Database (Findex). This dataset comprised various indicators such as percentage of population having a bank account, and the existence of ATMs and bank branches. Data is first ingested into the system using the CSV format, and during the preprocessing phase, the date information is carefully prepared and missing values are filled by applying linear interpolation.

- **Second Dataset:** This second dataset has data that is taken from such reputed sources as World Development Indicators (WDI), World Happiness Report, and The Heritage Foundation and includes historical socioeconomic data relevant to Argentina and Ireland. Specifically, it carefully documents a number of financial and economic indicators (date, GDP, inflation (GDP deflator), Life Satisfaction from the World Happiness Index Report, and Economic Freedom. The data are submitted in CSV format, and then a preprocessing step was applied to very carefully format the date and attune all variables so that they are comparable.

3 System Configuration

In this section, Hardware and Software specification used in the study will be discussed

3.1 Local machine Hardware Specification

Device specifications	
Device name	Valenluciana
Processor	13th Gen Intel(R) Core(TM) i5-1335U 1.30 GHz
Installed RAM	8.00 GB (7.69 GB usable)
Device ID	606A6635-EA6D-4AA1-85E6-0FA8BFEFDF68
Product ID	00342-43394-10393-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Figure 1: Hardware requirement

Windows specifications	
Edition	Windows 11 Home Single Language
Version	23H2
Installed on	01/09/2023
OS build	22631.3880
Experience	Windows Feature Experience Pack 1000.22700.1020.0

Microsoft Services Agreement
Microsoft Software Licence Terms

Figure 2: Operating System Configurations

3.2 Google Colab Hardware Specification

Offers 12.7 GB of RAM and allocates a GPU, either Tesla K80 or Tesla T4, with 11.4GB memory based on the runtime environment. In addition, the system provides a disk space of 107.7GB to store data.

4 Installation and package required

The following are the steps for the development of both PCA and Fixed Effects Regression (Built-in, 2023; GeeksforGeeks, 2018; Younes et al., 2021).

```
import pandas as pd
import numpy as np
```

Figure 3: Loading and read dataset

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
```

Figure 4: Normalization Process

```
!pip install --upgrade seaborn
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 5: Visualizations

```python
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 6: Application of first stage PCA

```python
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer
```

Figure 7: Application of second stage PCA

```python
import statsmodels.formula.api as smf

# Combine data for fixed effects regression
df_arg['Country'] = 'Argentina'
df_irl['Country'] = 'Ireland'
combined_data = pd.concat([df_arg, df_irl])

# Prepare formula for the regression
formula = 'FII ~ Population_15 + Individuals_using_internet + Life_satisfaction

# Fit the Fixed Effects model
fixed_effects_model = smf.ols(formula, data=combined_data).fit()
print(fixed_effects_model.summary())
```

Figure 8: Application of Fixed Effects Regression

5 Data Preprocessing

5.1 Financial Indicators Datasets Ireland and Argentina

```python
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

# Load the datasets
#df_arg = pd.read_csv('/content/sample_data/interpolated_FAS_arg_1_nofintech.csv')
df_arg = pd.read_csv('/content/sample_data/interpolated_FAS_arg_1_nofintech.csv')

# Normalize the data using Min-Max Scaling
scaler = MinMaxScaler()

# Select relevant columns for normalization
columns_to_normalize = ['atm_100k', 'bank_branches_100k', 'insurance_100k', 'dep_account_cbank_1000',
                        'out_dep_cbank_GDP','out_dep_other_GDP','out_loan_cbank_GDP', 'out_loan_cbank_GDP',
                        'number_deposit_cbank_account','Account_15+', 'Fin_Inst_Acc_15+', 'Saved_Fin_Inst_15+',
                        'Owns_CC_15+','Owns_DC_15+','Borrowed_15+', 'Saved_15+', 'Saved_Fin_Inst2_15+',
                        'Borrowed_Fin_Inst_15+','Utility_Pay_Acc_15+']

#df_arg[columns_to_normalize] = scaler.fit_transform(df_arg[columns_to_normalize])
df_arg[columns_to_normalize] = scaler.fit_transform(df_arg[columns_to_normalize])

# Save the interpolated data to a new CSV file
output_file_path = '/content/sample_data/normalized_FAS_arg_1_nofintech.csv'
df_arg[columns_to_normalize].to_csv(output_file_path, index=False)
```

Figure 9: Data Normalization for each country

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Set the style of the visualization
sns.set(style="whitegrid")

# List of variables to plot
variables = [
    'atm_100k',
    'bank_branches_100k',
    'insurance_100k',
    'dep_account_cbank_1000',
    'out_dep_cbank_GDP',
    'out_dep_other_GDP',
    'out_loan_cbank_GDP',
    'number_deposit_cbank_account',
    'Account_15+',
    'Fin_Inst_Acc_15+',
    'Saved_Fin_Inst_15+',
    'Owns_CC_15+',
    'Owns_DC_15+',
    'Borrowed_15+',
    'Saved_15+',
    'Saved_Fin_Inst2_15+',
    'Borrowed_Fin_Inst_15+',
    'Utility_Pay_Acc_15+'
]

# Plot each variable over time for both countries
plt.figure(figsize=(15, 20))

for i, var in enumerate(variables, 1):
    plt.subplot(6, 3, i)
    sns.lineplot(x='Year', y=var, data=df_arg, label='Argentina')
    sns.lineplot(x='Year', y=var, data=df_irl, label='Ireland')
    plt.title(var)
    plt.xlabel('Year')
    plt.ylabel(var)
```

Figure 10: Visualizations

```python
# Calculate statistics for Argentina
calculate_summary_stats(arg_data, 'Argentina')

# Calculate statistics for Ireland
calculate_summary_stats(irl_data, 'Ireland')

# Convert the dictionary to a DataFrame
summary_df = pd.DataFrame(summary_stats)

# Pivot the table for better readability
pivot_table = summary_df.pivot(index=['Country', 'Variable'], columns=[], values=['Mean', 'SD', 'Min', 'Max'])

# Display the pivot table
print(pivot_table)
```

Figure 11: Calculation of Statistics

These processes have been applied also for Demographic Dataset

```python
# Function to apply PCA
def apply_pca(data, variables, n_components=2):
    # Standardize the data
    scaler = StandardScaler()
    scaled_data = scaler.fit_transform(data[variables])

    # Apply PCA
    pca = PCA(n_components=n_components)
    principal_components = pca.fit_transform(scaled_data)

    # Create a DataFrame with the principal components
    pca_df = pd.DataFrame(data=principal_components, columns=[f'PC{i+1}' for i in range(n_components)])

    return pca, pca_df, scaler

# Apply PCA to each sub-index
ownership_pca, ownership_pca_df, ownership_scaler = apply_pca(df, ownership_variables)
availability_pca, availability_pca_df, availability_scaler = apply_pca(df, availability_variables)
usage_pca, usage_pca_df, usage_scaler = apply_pca(df, usage_variables)

# Function to plot explained variance
def plot_explained_variance(pca, title):
    plt.figure(figsize=(8, 5))
    plt.bar(range(1, len(pca.explained_variance_ratio_) + 1), pca.explained_variance_ratio_, alpha=0.5, align='center')
    plt.step(range(1, len(pca.explained_variance_ratio_) + 1), np.cumsum(pca.explained_variance_ratio_), where='mid')
    plt.xlabel('Principal Components')
    plt.ylabel('Explained Variance Ratio')
    plt.title(title)
    plt.show()
```

Figure 12: First Stage PCA

```python
print("Ownership Variables Loadings:")
print(ownership_loadings_arg)
```

```
Ownership Variables Loadings:
                    PC1       PC2
Account_15+     0.506091  0.406756
Fin_Inst_Acc_15+ 0.511893  0.289735
Owns_CC_15+     0.462887 -0.862024
Owns_DC_15+     0.517275  0.086707
```

```python
print("\nAvailability Variables Loadings:")
print(availability_loadings_arg)
```

```
Availability Variables Loadings:
                       PC1       PC2
atm_100k          0.647463  0.345288
bank_branches_100k -0.350282  0.926558
insurance_100k    -0.676827 -0.149219
```

```python
print("\nUsage Variables Loadings:")
print(usage_loadings_arg)
```

```
Usage Variables Loadings:
                                PC1       PC2
dep_account_cbank_1000      0.320554  0.145396
out_dep_cbank_GDP           0.294861  0.335617
out_dep_other_GDP           0.253516  0.031012
out_loan_cbank_GDP         -0.242157  0.637525
number_deposit_cbank_account 0.321625  0.122134
Saved_Fin_Inst_15+          0.319993  0.162494
Borrowed_15+                0.325173 -0.061247
Saved_15+                   0.319494  0.170076
Saved_Fin_Inst2_15+         0.319993  0.162494
Borrowed_Fin_Inst_15+       0.256493 -0.593951
Utility_Pay_Acc_15+         0.324894 -0.075109
```

Figure 13: Loadings of each sub-index

```python
# Ownership Subindex
print("Penetration (Argentina):")
print("Eigenvalues:", ownership_pca.explained_variance_)
print("Proportion of Explained Variance:", ownership_pca.explained_variance_ratio_)

# Availability Subindex
print("\nAvailability (Argentina):")
print("Eigenvalues:", availability_pca.explained_variance_)
print("Proportion of Explained Variance:", availability_pca.explained_variance_ratio_)

# Usage Subindex
print("\nUsage (Argentina):")
print("Eigenvalues:", usage_pca.explained_variance_)
print("Proportion of Explained Variance:", usage_pca.explained_variance_ratio_)
```

Figure 14: Eigenvalues

```python
# Calculate sub-indices for Argentina
df_arg['ownership'], var_pen_arg = calculate_sub_index(df_arg, ownership_variables)
df_arg['availability'], var_avail_arg = calculate_sub_index(df_arg, availability_variables)
df_arg['usage'], var_usage_arg = calculate_sub_index(df_arg, usage_variables)

# Calculate sub-indices for Ireland
df_irl['ownership'], var_pen_irl = calculate_sub_index(df_irl, ownership_variables)
df_irl['availability'], var_avail_irl = calculate_sub_index(df_irl, availability_variables)
df_irl['usage'], var_usage_irl = calculate_sub_index(df_irl, usage_variables)

# Standardize the sub-indices to have mean 0 and standard deviation 1
scaler = StandardScaler()
df_arg[['ownership', 'availability', 'usage']] = scaler.fit_transform(df_arg[['ownership', 'availability', 'usage']])
df_irl[['ownership', 'availability', 'usage']] = scaler.fit_transform(df_irl[['ownership', 'availability', 'usage']])

# Assume equal weights for simplicity, or derive from PCA if needed
weights_arg = [1, 1, 1]
weights_irl = [1, 1, 1]

# Calculate the overall Financial Inclusion Index (FII)
df_arg['FII'] = (df_arg['ownership'] * weights_arg[0] +
                 df_arg['availability'] * weights_arg[1] +
                 df_arg['usage'] * weights_arg[2])

df_irl['FII'] = (df_irl['ownership'] * weights_irl[0] +
                 df_irl['availability'] * weights_irl[1] +
                 df_irl['usage'] * weights_irl[2])
```

Figure 15: Second Stage PCA (FFI with equally weights)

```python
import statsmodels.formula.api as smf

# Combine data for fixed effects regression
df_arg['Country'] = 'Argentina'
df_irl['Country'] = 'Ireland'
combined_data = pd.concat([df_arg, df_irl])

# Prepare formula for the regression
formula = 'FII ~ Population_15 + Individuals_using_internet + Life_satisfaction + HDI + Economic_freedom + GDP_per_capita + Inflation_GDP_deflator + C(Country)'

# Fit the Fixed Effects model
fixed_effects_model = smf.ols(formula, data=combined_data).fit()
print(fixed_effects_model.summary())
```

Figure 16: Fixed Effects Regression

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the datasets
subindex_FII_ARG = pd.read_csv('/content/sample_data/subindex_FII_ARG.csv')
subindex_FII_IRL = pd.read_csv('/content/sample_data/subindex_FII_IRL.csv')
demographics_ARG = pd.read_csv('/content/sample_data/year_normalized_ARG_demographics.csv')
demographics_IRL = pd.read_csv('/content/sample_data/year_normalized_IRL_demographics.csv')

# Merge the sub-index data with the demographic data for each country
df_arg = pd.merge(subindex_FII_ARG, demographics_ARG, on='Year')
df_irl = pd.merge(subindex_FII_IRL, demographics_IRL, on='Year')

# Add a Country column to distinguish between the two countries
df_arg['Country'] = 'Argentina'
df_irl['Country'] = 'Ireland'

# Combine the data into a single DataFrame
combined_data = pd.concat([df_arg, df_irl])

# Define the variables for the correlation matrix
variables_to_check = [
    'Population_15',
    'Individuals_using_internet',
    'Life_satisfaction',
    'HDI',
    'Economic_freedom',
    'GDP_per_capita',
    'Inflation_GDP_deflator'
]

# Calculate the correlation matrix
correlation_matrix = combined_data[variables_to_check].corr()
```

Figure 17: Correlation Matrix and Heatmap

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    FII   R-squared:                       0.961
Model:                            OLS   Adj. R-squared:                  0.937
Method:                 Least Squares   F-statistic:                     40.35
Date:                Sun, 11 Aug 2024   Prob (F-statistic):           6.04e-08
Time:                        22:58:20   Log-Likelihood:                -19.136
No. Observations:                  22   AIC:                             56.27
Df Residuals:                      13   BIC:                             66.09
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
Intercept                     -5.1358      0.709     -7.248      0.000      -6.667      -3.605
C(Country)[T.Ireland]          1.1873      0.351      3.386      0.005       0.430       1.945
Population_15                  7.0691      3.119      2.266      0.041       0.330      13.808
Individuals_using_internet     1.4460      2.783      0.520      0.612      -4.566       7.458
Life_satisfaction              0.4334      0.803      0.540      0.599      -1.301       2.168
HDI                           -2.9476      0.815     -3.618      0.003      -4.708      -1.188
Economic_freedom               1.3100      0.884      1.482      0.162      -0.600       3.219
GDP_per_capita                 1.8516      0.904      2.049      0.061      -0.101       3.804
Inflation_GDP_deflator         0.7574      0.899      0.843      0.415      -1.184       2.699
==============================================================================
Omnibus:                        4.594   Durbin-Watson:                   1.453
Prob(Omnibus):                  0.101   Jarque-Bera (JB):                2.823
Skew:                          -0.844   Prob(JB):                        0.244
Kurtosis:                       3.482   Cond. No.                         46.1
==============================================================================

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

Figure 18: Regression Results

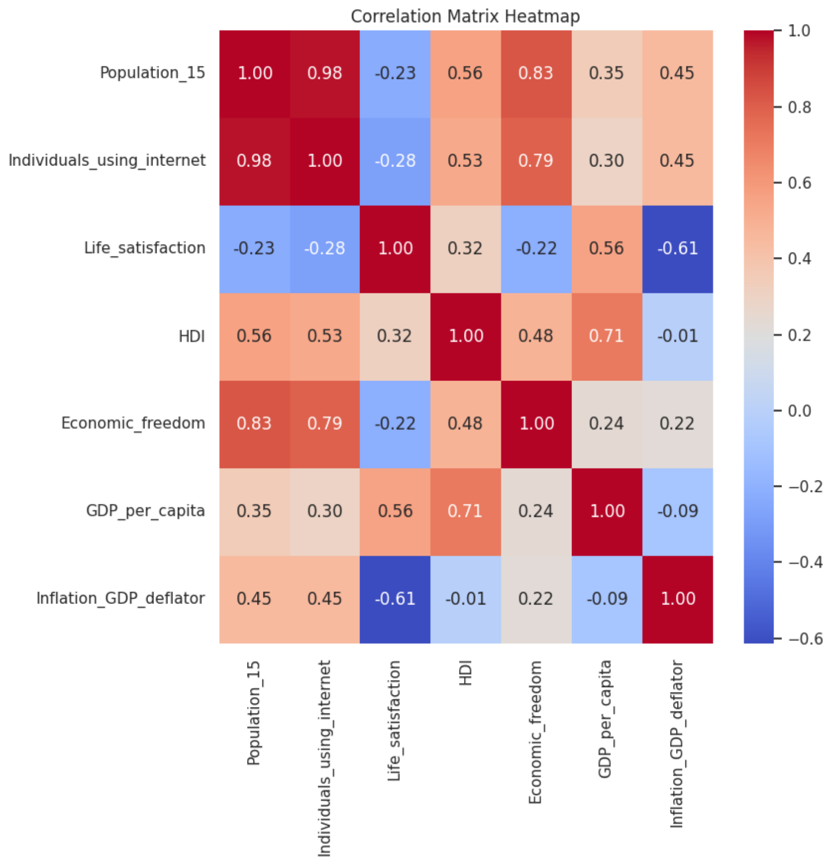

Figure 19: Heatmap

6 References

Built-in, 2023. PCA Using Python: A Tutorial [online]. Built In. URL https://builtin.com/machine-learning/pca-in-python (accessed 6.12.24).

GeeksforGeeks, 2018. Principal Component Analysis with Python [online]. GeeksforGeeks. URL https://www.geeksforgeeks.org/principal-component-analysis-with-python/ (accessed 6.12.24).

Younes, K., Mouhtady, O., Chaouk, H., Obeid, E., Roufayel, R., Moghrabi, A., Murshid, N., 2021. The Application of Principal Component Analysis (PCA) for the Optimization of the Conditions of Fabrication of Electrospun Nanofibrous Membrane for Desalination and Ion Removal. Membranes 11, 979. https://doi.org/10.3390/membranes11120979