Michael Madigan

mikelispopple@gmail.com

A picture containing clipart

Description automatically generated

Data analysis of the economic and societal impact of covid 19

Project Report

Table of Contents

[Abstract 2](#_Toc69740846)

[Introduction 2](#_Toc69740847)

[Datasets 2](#_Toc69740848)

[Implementation Process 3](#_Toc69740849)

[First data set 5](#_Toc69740850)

[Second data set 6](#_Toc69740851)

[Third data set 7](#_Toc69740852)

[Merging 8](#_Toc69740853)

[Merge-1 8](#_Toc69740854)

[Merge-2 8](#_Toc69740855)

[Plotting 9](#_Toc69740856)

[Results 10](#_Toc69740857)

[Result 1 10](#_Toc69740858)

[Insight-1 10](#_Toc69740859)

[Result 2 11](#_Toc69740860)

[Insight-2 11](#_Toc69740861)

[Result 3 12](#_Toc69740862)

[Insight-3 12](#_Toc69740863)

[Result 4 13](#_Toc69740864)

[*Insight-4* 13](#_Toc69740865)

[Result 5 14](#_Toc69740866)

[Insight-5 14](#_Toc69740867)

[Result 6 15](#_Toc69740868)

[Insight-6 15](#_Toc69740869)

[Insights 16](#_Toc69740870)

[References 17](#_Toc69740871)

[Bibliography 17](#_Toc69740872)

**GitHub** URL <https://github.com/MM2020-11/UCDPA_Michael_Madigan>

# Abstract

The arrival of Covid 19 virus has impacted Ireland’s society and economy. It has invaded our daily lives and is constantly in the airways, news and internet. As covid (SARS-CoV-2) is a highly contagious epidemic the full resources of the medical, research and pharmaceutical industry are driving for solutions. In writing this report I hope to expand on this knowledge.

This report applies data analytics to visualise the societal and economic impacts.

The python programming language is used to process, clean and display data graphically allowing data-driven conclusions. The report uses three main data sources; covid new daily cases, staying local data and weekly pandemic support payments. These data sources are compared and analysed.

# Introduction

As the world reacted to covid a vast array of data is collected daily and worldwide. This large data set allows comparison and contrast between different data sets using the Python programming language. Data collection starts in March 2020 as the World Health Organisation (WHO) declared covid as a pandemic

Using learnings from the UCD PA course and online training allowed the processing of data using python tools. Python has an extensive library of add-on features which makes data manipulation all the easier. Also python libraries include plot tools matplotlib and Seaborn. This project uses the Matplotlib library.

This report delves into Irish covid reports and completes the following:

1. Presents and analyses Irish covid case data.
2. Analyses data from ‘staying local’ metrics collected from mobile phone GPS location data
3. Shows impact on government support payments.
4. Checks ratio between male/female payments
5. Checks for correlation between trends

# Datasets

The project used three separate data sources.

The first is a direct link to covid world data via URL link to Python data frame. This data source is very comprehensive and updated daily, hence a very useful data set. This data contains worldwide data and includes Irish covid case numbers. This data is collated on a daily bases at a web site “our world in data” [1]

The second data source is from the Irish government site www.CSO.ie and called the “Staying Local Indicator”. This data was collected from a mobile phone company "Three Ireland" and analysed. Anonymised data is calculated for people staying with 10km of the home, based on phone GPS data. It is believed that increased people movement should correlate strongly to increased transmission of covid. This data is collected at [1] and [2]. This data is collected daily by county and state

A third data set came from Irish government records of Pandemic Unemployment Payments, this data is collected weekly and categories by age and sex. To align the data sets this data was interpolated into daily records to match with the two previous data sets and in order to make direct comparisons.

# Implementation Process

Implementation; high-level graphical overview.

Pandemic Payments data

Staying local data

Covid data

Read

data

from local CSV file

Read

data

from local CSV file

Download data

Direct from url

Correct date data

Interpolate missing daily data

Merged

Merged

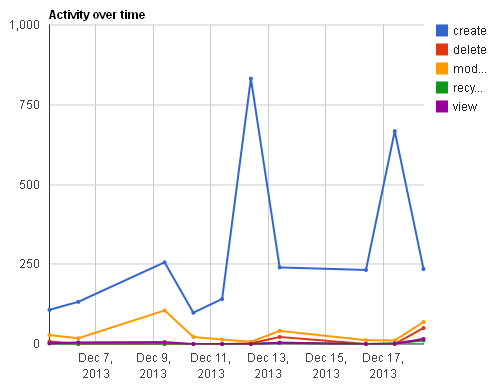
Data cleaned, check, sliced and diced

Data cleaned, check, sliced and diced

Data cleaned, check, sliced and diced

Graphical Display

Graphical Display

Graphical Display

Graphical Display

Calculate data points

Figure High-level overview

Implementation details

This project was implemented using the Pycharm IDE and python. The program is contained in one file “main.py”.

A high-level process graphically Figure 1 on previous page. The project reads files both local and URL link. Cleans, modifies and merges data sets. The called functions are

1. Pandas as pd
   1. used to efficiently manipulate the datasets
2. Matplotlib.pyplot as plt
   1. Graphically represent data
   2. Selected the OO style for plotting controls
3. numpy as np
   1. For numerical calculations
   2. This function was only used during development phase
4. datetime as dt
   1. Used to process and clean date/time formats

To help in the debugging process the code saves data frames to CSV files for checking and uses print commands to check calculation and data properties. The debugging CSV files are named “export\_xxx” for consistency. Print commands include line number or parameter name.

## First data set

The first data source contains covid pandemic data from our World in data and a pandas connect directly to the website containing the file URL ‘owid-covid-data.csv'. The file was inspected with .head and .tail, .shape and .dttypes to determine the dataframe design.

This data file contained a long list of columns. To review the list a python “FOR” loop was used to print the column list ( as an neater alternative to using the .columns property) list using

for idx, column in enumerate(df1.columns):

print(idx, column)

On reviewing the resulting columns were selected and just the Irish iso\_code was extracted for country to allow select of IRL for Ireland, date and number of new cases. In both instances “.loc” was used.

Selected\_cols = ['iso\_code', 'date', 'new\_cases']

The .loc pandas function was used to separate the selected columns

df\_iso\_date\_new\_cases = df1.loc[:, selected\_cols]

Once the data frame was created it had to be cleaned as follows; Nan validation and fill with zeros. The data was cleaned to remove missing points which read as NaN and replace new cases with zeros.

Using the Pandas function *df.fillna(0)*

The dataframe was tested for NaNs using a isna() and sum() Pandas function, then printing the result.  
count\_NaN = df\_iso\_date\_new\_cases\_no\_NaN[selected\_cols].isna().sum()

On reviewing the data types I found that the ‘date, column was not in a date data type. pd.to\_datetime function was used to convert to date

pd.to\_datetime(df\_iso\_date\_new\_cases['date'], yearfirst=True, format="%d/%m/%Y")

The data was reduced to select only data rows relevant to Ireland

df\_IRL = df\_iso\_date\_new\_cases\_no\_NaN.loc[df\_iso\_date\_new\_cases\_no\_NaN["iso\_code"] == 'IRL', :]

During debugging I found that the online data was not always available, A local version is included but will not be as up to date as the online version

## Second data set

Second dataset from [www.cso.ie](http://www.cso.ie) was the Staying local indicator data which was manually downloaded as a CSV file and read into the Python code.

A pandas command was used to read from a local downloaded CSV file. Data was inspected, checked for NaN’s

Data was inspected for columns using a “for loop” and “.head()” functions as small number of columns.

The required columns were extracted using “.loc”

required\_cols = ['Date', 'County', 'VALUE']

df\_SLI\_Date\_County\_Value = df\_SLI.loc[:, required\_cols]

For unknown reason the data contained an incorrectly formatted date on 20 December21, missing a space. This was fixed with code to replace the error as this alarmed when completing the date data type conversion.

df\_SLI\_Date\_County\_Value['Date'] = df\_SLI\_Date\_County\_Value['Date'].str.replace('2020 December20', '2020 December 20')

This data has 4 incorrect missing data point so the graph dropped to zero at this data points see Figures 1 and 2 below. To improve the presentation the data was interpolated using pandas .interpolate funcition.

df\_pup2 = df\_cases\_pup1.interpolate(axis=0)

This code removed the zero points shown in Figure1 for a smooth graph as shown in Figure 2.

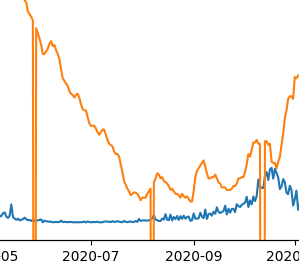
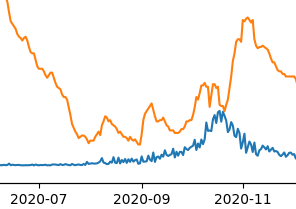
 

Figure graph before interpolation . Figure graph after interpolation

A subset by ‘state’ using the .loc function was used to create a new data frame. The comparison in country wide at ‘state’ and not a specific county level.

df\_SLI\_state = df\_SLI\_Date\_County\_Value\_No\_NaN.loc[df\_SLI\_Date\_County\_Value\_No\_NaN["County"] == 'State', :]

## Third data set

Data was downloaded from CSO.ie containing the Irish pandemic payments. As per the previous data set the data was read in from a local CSV file, cleaned off NaN’s. These files contained dates in the format as weeks in format YYYYWXX eg 2020W12. Code was required to convert this format to a date format.

The code added an extra day format for the first Monday of the week

df\_pup['first\_Monday'] = df\_pup['Week'] + '-1'

Then date formatted with to\_date function

df\_pup['first\_Monday\_date'] = pd.to\_datetime(df\_pup['first\_Monday'], yearfirst=True, format="%YW%W-%w")

The required columns were extracted

required\_cols = ['Statistic', 'Age Group', 'Sex', 'VALUE', 'first\_Monday\_date']

df\_pup1 = df\_pup.loc[:, required\_cols]

then the relevant rows were extacted in a multi-step process

df\_pup1 = df\_pup1.loc[df\_pup1["Statistic"] == 'Persons in receipt of the Pandemic Unemployment Payment', :]

df\_pup1 = df\_pup1.loc[df\_pup1["Age Group"] == 'All ages', :]

df\_pup1 = df\_pup1.loc[df\_pup1["Sex"] == 'Both sexes', :]

Second steps

required\_cols = ['Statistic', 'VALUE', 'first\_Monday\_date']

df\_pup1 = df\_pup1.loc[:, required\_cols]

## Merging

### Merge-1

Date from covid cases and staying local were merged

Data was merged using a left join matching the two data frames dates as indices

df\_cases\_County = pd.merge(df\_IRL, df\_SLI\_state, left\_on='date', right\_on='New\_Date', how='left')

The merged data frame was further reduced and cleaned

df\_cases\_County\_cleaned = (df\_cases\_County\_2[(df\_cases\_County\_2[date\_cols] != 0).all(axis=1)])

After merging and graphing, a calculation to check for correlation was performed to compare the pup recipients to cases.

correlation = df\_cases\_County\_2['VALUE'].corr( df\_cases\_County\_2['new\_cases'], method='pearson')

print('\n Pearson correlation coefficient for new cases and SLI = ', correlation)

The Pearson’s coefficient data is discussed in the results section.

### Merge-2

Data from Irish covid cases and pandemic payments were merged

Data was merged using a left join matching the two data frames dates as indices

df\_cases\_pup1 = pd.merge(df\_IRL, df\_pup1, left\_on='date', right\_on='first\_Monday\_date', how='left')

As the pup data was in a weekly format is was necessary to interpolate to report on a daily basis in line with the previous two data sets

df\_pup2 = df\_cases\_pup1.interpolate(axis=0)

Again this data was tested for correlation against the cases data

correlation = df\_pup2['VALUE'].corr( df\_pup2['new\_cases'],method='Pearson')

print('\n coor2 = ', correlation )

The Pearson’s coefficient data is discussed in results section

## Plotting

Data was plotted and displayed in the reports section. This used matplotlib for plots and bar graphs. Also annotations were added to expand the information. Dual-axis were used with limits when the data ranges were different. The recommended object-oriented approach was used

The data was extracted from the data frames and converted to lists. Whilst there are many methods to plot data this solution was found to be convenient by not memory efficient.

Titles, axis labels are included in this code.

The data was arranged using ‘grouped by’ features in preparation for plotting on bar charts

# Results

## Result 1

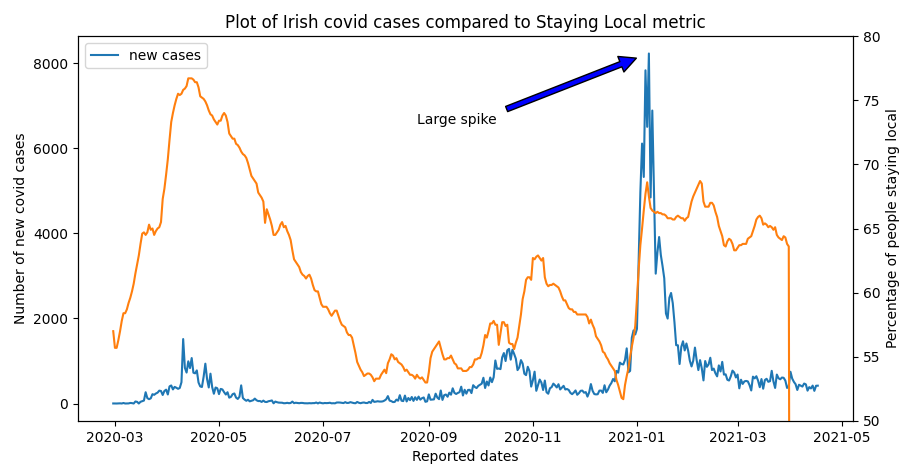


Figure plot of Irish covid case and staying local % by date

Figure 3 above displays a trend of Irish covid cases recorded from February 2020 until present day (blue) along with staying local indicator (orange) over the same period.

### Insight-1

Three key insights from this graph:

1. The large decrease in staying local in December 2020 did forecast the large spiked in new covid cases.
2. The January spike is the only event in which the staying local data yielded a significant correlation. To check for statistical correlation the program ran a python ‘.corr’ test which confirmed a poor correlation as 0.12. A value of 1 would be a perfect correlation.
3. In light of this poor correlation I would advise against using GPS tracking data for covid forecasting as the personal intrusion while data is anonymised does not out weight the insignificant benefit

## Result 2

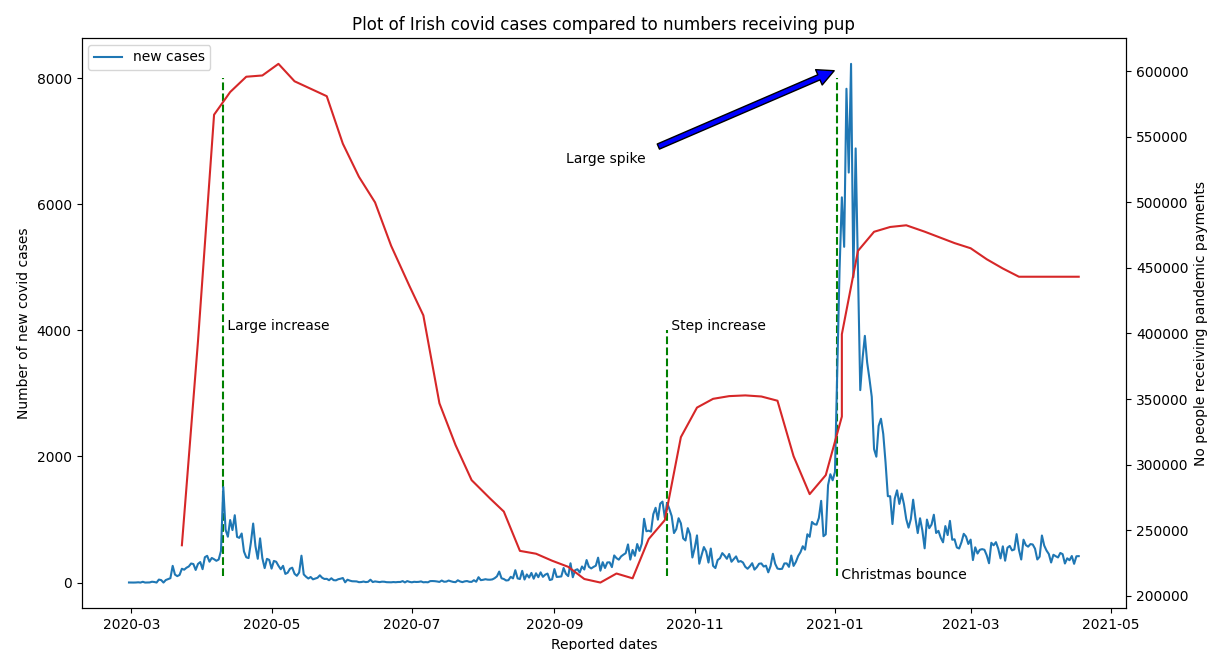


Figure Plot of covid cases vs pup

This bar graph illustrates the trend of people receiving payment vs new cases of covid

### Insight-2

There is a relationship between covid cases and the number of people receiving payments. Increased cases in Nov 2020 say an increase in payments and again in January 2021

## Result 3

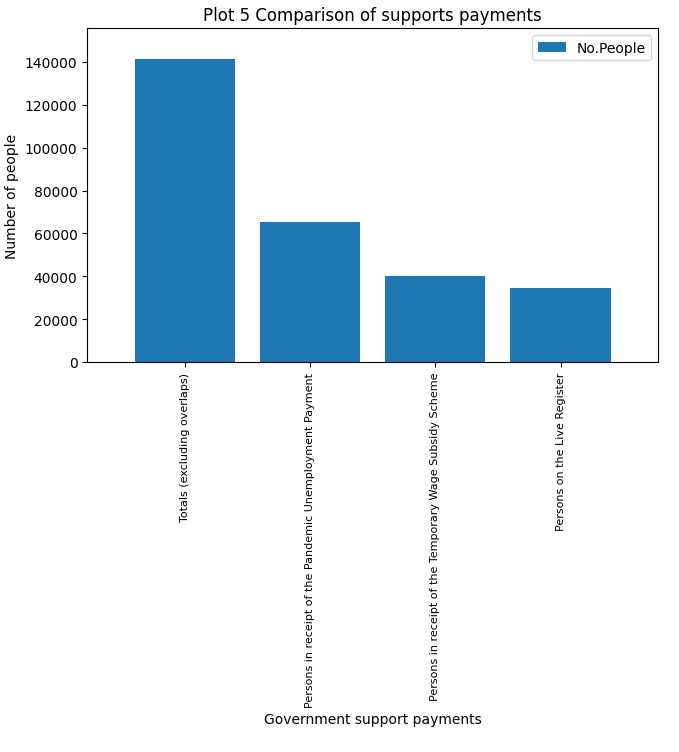


Figure Government Support payments

This bar graph illustrates the different Irish supports used during the pandemic

### Insight-3

From the bar graph, we see that the PUP payments are approximately 50% of the total payments and well exceed the number of people on the live register.

## Result 4

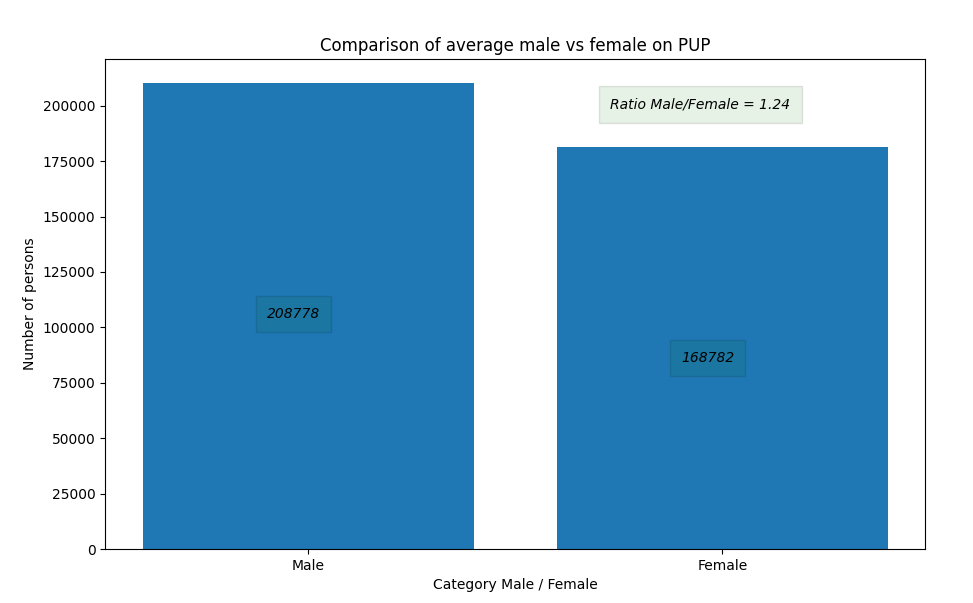
****

Figure Bar chart of male / female payments

Bar chart of the numbers of people on support payments grouped by sex.

### *Insight-4*

The ratio of male to female on the various government payment schemes is 1.24

## Result 5

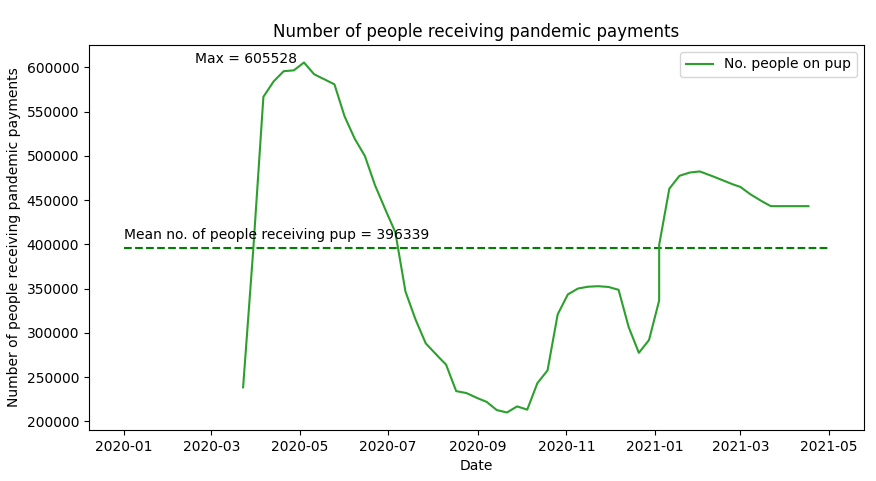
****

Figure plot of the number of people receiving payments

Chart of the number of people on support payments grouped by date. Note the mean is calculated in code as 396,339 and maximum is 605528 and both displayed on chart

### Insight-5

The number of people on pup rapidly rose to a maximum of 605 thousand dropping during Summer of 2020. This prompts the question, is this a seasonal Summer/Winter trend – further analysis with machine learning can be employed by looking at states in the southern hemisphere to determine relationships.

## Result 6

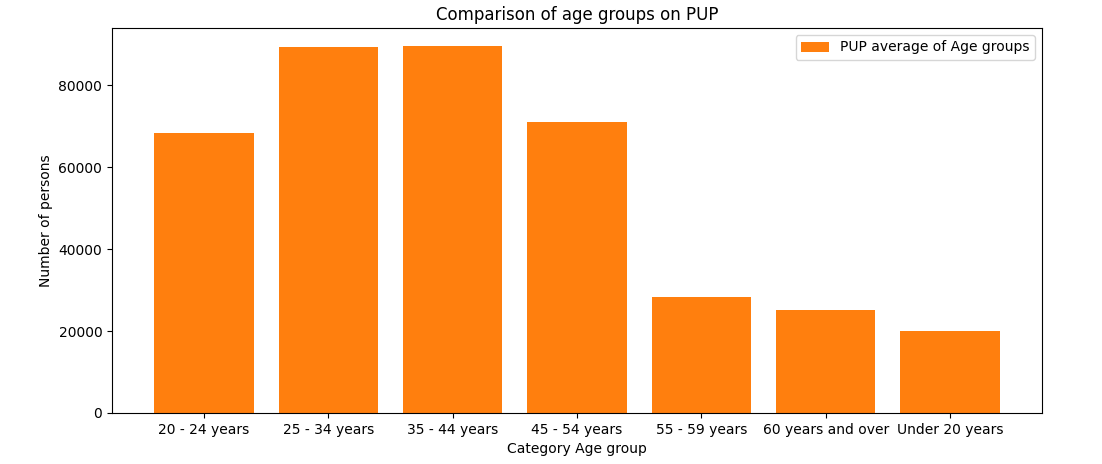
****

Figure comparison of pup by age group

Bar chart of the number of people receiving pandemic payments by age group.

### Insight-6

The bulk of people receiving PUP are in age groups 20 o 54 years of age.

## Result 7

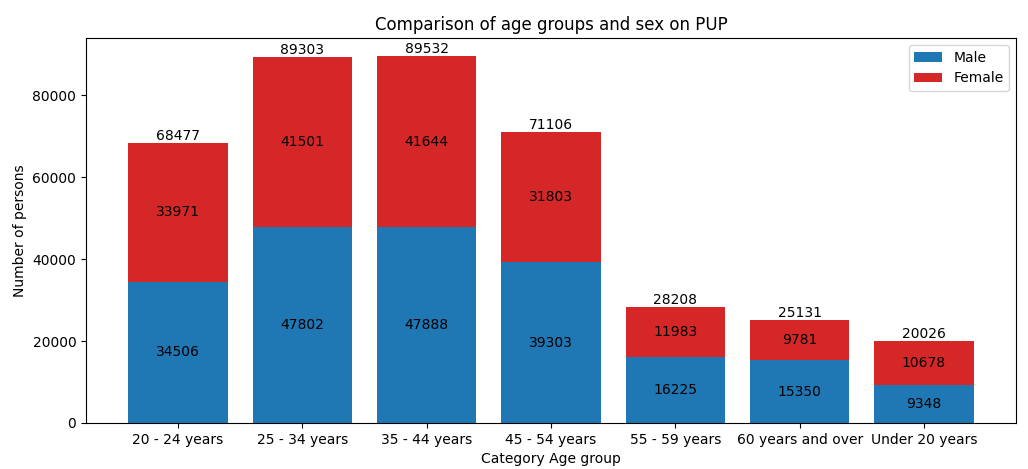


Figure comparison of pup by age group and sex

Bar chart of the number of people receiving pandemic payments by age group and subdivided into male/female.

Calculations of the individual ratios as follows:

Age Group ratio

0 20 - 24 years 0.984495

1 25 - 34 years 0.868185

2 35 - 44 years 0.869612

3 45 - 54 years 0.809175

4 55 - 59 years 0.738552

5 60 years and over 0.637199

6 Under 20 years 1.142276

Data show that women exceed men only in the under 20 years age category

### Insight-7

The distribution of pup payments across age groups and sex show no significant ages where the ratios differ. The ratio is common across all ages.

Data show that women exceed men only in the under 20 years age category

# Insights

The principal insights are reported along with the graphs for easier review however repeated here for benefit of review.

1. Three key insights from graph Figure 3:
2. The large decrease in staying local in December 2020 did forecast the large spiked in new covid cases.
3. The January spike is the only event in which the staying local data yielded a significant correlation. To check for statistical correlation the program ran a python ‘.corr’ test which confirmed a poor correlation of 0.12. A value of 1 would be a perfect correlation.
4. In light of this poor correlation (0.12) I would advise against using GPS tracking data for covid forecasting as the personal intrusion while data is anonymised does not out weight the insignificant benefit.
5. There is a relationship between covid cases and the number of people receiving payments. Increased cases in Nov 2020 say an increase in payments and again in January 2021
6. From the bar graph Figure 5, we see that the PUP payments are approximately 50% of the total payments and well exceed the number of people on the live register.
7. The ratio of male to female on the various government payment schemes is 1.24
8. The number of people on pup rapidly rose to a maximum of 605 thousand dropping during Summer of 2020. This prompts the question, is this a seasonal Summer/Winter trend? Further analysis with machine learning can be employed by looking at states in southern hemisphere to determine relationships.
9. The bulk of people receiving PUP are in age groups 20 o 54 years of age.
10. The distribution of pup payments across age groups and sex show no significant ages where the ratios differ. The ratio is common across all ages

# Discussion

Machine learning (ML) may be used to collate the many data sources and using categorical risk data based on age group and to predict the path of covid and the probability of new out-breaks as occurred in Jan 2021.

Simple linear regression is unlikely to predict outbreaks however can be used to allow authorities to make decisions on relaxing covid lockdown requirements. Also, allow authorities to plan and manage demand for support payments.

The is a great deal of opportunity of machine learning to forecast this complex pandemic, on the positive side there is ample data available from many sources. ML has been applied to predict covid risk factors, collect medical data and present pointers to the medical teams. The UK’s NHS in particular benefited from shared knowledge which is centralised and shared. Allowing beneficial medical practices to be analysed and winning methods quickly adopted. ML will only enhance the in the future.

# References

<https://data.cso.ie/table/SLI01>

The Staying Local Indicator (SLI) provides daily estimated percentages of county populations that have stayed local i.e. within 10km of their usual place of residence, averaged over the preceding seven days.

# Bibliography

|  |  |
| --- | --- |
| [1] | H. Ritchie, “Our World in Data,” [Online]. Available: #https://ws.cso.ie/public/api.jsonrpc?data=%7B%22jsonrpc%22:%222.0%22,%22method%22:%22PxStat.Data.Cube\_API.ReadDataset%22,%22params%22:%7B%22class%22:%22query%22,%22id%22:%5B%5D,%22dimension%22:%7B%7D,%22extension%22:%7B%22pivot%22:null,%22codes%22:false,. [Accessed 14 April 2021]. |
| [2] | https://data.cso.ie/, “https://data.cso.ie/,” Staying Local Indicator, [Online]. Available: https://data.cso.ie/. [Accessed 17 April 2021]. |
| [3] | C. I. -. I. Government, “COVID-19 Pandemic Unemployment Payment,” [Online]. Available: https://www.citizensinformation.ie/en/social\_welfare/social\_welfare\_payments/unemployed\_people/covid19\_pandemic\_unemployment\_payment.html. [Accessed 17 April 2021]. |
| [4] | I. Central Statistics Office, “Persons on the Live Register, in receipt of the Pandemic Unemployment Payment , in receipt of the Temporary Wage Subsidy Scheme and Total excluding overlaps,” Central Statistics Office, Ireland, [Online]. Available: https://data.cso.ie/. [Accessed 17 April 2021]. |