OwnProject

January 17, 2022

1 Introduction

2 Overview of topic

For this study, I am going to explore and describe which people are involved in using the internet as well as the people that are not involved in it. It will be inclusive to

The United Kingdom in a period from 2013 up to 2020 (January to March). This study will notably focus on these factors:

- age groups
- gender
- disability
- ethnicity
- economic activity

3 Relevant news or research articles

On the process of searching for a topic I have realised that it could be intresting to do a topic about something that is inseparable part of our everyday life in twenty first century.

I have started to look for articles related to internet use, for example: Internet use surges at lockdown.

However, my intention was not to limit myself to what people do on the internet, but rather who are those people and what are their characteristics.

That could be observable in this article that tackles the teenagers: British teenagers among world's most extreme internet users

From that point forward I have decided to search for a dataset that will include up to date data with details about

users/non-users of internet in UK, which I found there Internet users in UK.

4 Data

The data from this source is well separated based on the characteristics of the user we would like to go

about, thus there are many pages of data in it.

The pages in this set have both numbers and percentages. I will focus myself on actual numbers for the purpose of visualisation

and the percentages will be used as an addition.

The data itself is structured well, however, it is not tidy data. There are many tables and unnecessary headings and rows. For

now, I believe that during cleaning I will have a problem with headers as some of them are in one position, whereas, others are

completely in different rows. The tables will have to be cut off from this set and then transformed to be useful. There is some

extra characters on some pages or lack of values on some rows as well. The data itself is very specific as it it always related

to time.

5 Research objectives and motivation

My motivation for this study is to describe situation on the ground of who are the people involved or not involved in the internet usage.

My goal is to get a better perspective about those people and what is it like for internet users based on the

different factors. I would like to go about these factors to explore the details and maybe present some key findings.

6 Research question

My research questions are:

- What people are using the internet?
- What are their characteristics?
- Do their characteristics impact the quantity of internet usage?

7 Population

Population for this data set is repondents aged 16 years old and over. The population statistics were gathered annualy from 2013

up to 2020, January to March.

The data about those people was separated using the factors specified in “Overview of topic”.

The population is present in thousands, but the total number of responses was not specified. For some data sheets, the years of

data collection may change to 2011-2020 and some numbers may go below thousands and even hundreds.

Imports and disabled warnings

[940]:
```
#IMPORTS
#DISABLE WARNINGS
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.filterwarnings("ignore")
```

Helper functions

[941]:
```
default = None

def openFile(io, sheet_name, header, names,
             index_col, usecols, squeeze,
             dtype, engine, converters,
             true_values, false_values, skiprows,
             nrows):
    return pd.read_excel(io, sheet_name, header, names,
                         index_col, usecols, squeeze,
                         dtype, engine, converters,
                         true_values, false_values, skiprows,
                         nrows)

#Source code: Mayank Porwal
#https://stackoverflow.com/questions/61789700/
 ↪creating-function-to-rename-columns-in-pandas-dataframe
def rename_column_name(df, old_column_list, new_column_list):
    df = df.rename(columns=dict(zip(old_column_list, new_column_list)), inplace␣
 ↪= True)
    return df
```

Variables used

[942]:
```
#VARS
ageCols = ["lightgreen", "#90DA90", "#6EC96E"]
titleOne = 'People that used internet \n in last 3 months'
titleTwo = 'People that used internet \n over 3 months ago'
titleThree = 'People that never used internet'
labels = ['16-24', '25-34', '35-44', '45-54', '55-64', '65-74', '75+']
colorsPie = ['#CCFFE5', '#66FFB2', '#00FF80', '#00CC66', '#00CC00', '#009900',␣
 ↪'#666600']
colors = ["deepskyblue", 'red']
col = 'orange'
```

```
x_labels = [2012,2013,2014,2015,2016,2017,2018,2019, 2020]
colorsEth = ["#FF0000", "#FF8000", "#FFFF00", "#00CC00", "#0080FF", "#009999",␣
 ↪"#FF99FF" ]
mSizeEth = 4
valOfSize = 19
label_size = 16
title_size = 18
style = 's-'
mSize = 9
economic_title = 24
```

8 Age

Data loading and cleaning

[943]:
```
#DATAFRAME
#PEOPLE THAT USED INTERNET IN LAST 3 MONTHS

#READING CORRECT NON-EMPTY COLUMNS & ROWS
df = openFile('./internetusers2020.xlsx',
              sheet_name = '1a',
              header = 0,
              names = default,
              index_col = default,
              usecols='A, C:J',
              squeeze = False,
              dtype = default,
              engine = default,
              converters = default,
              true_values = default,
              false_values = default,
              skiprows = 7,
              nrows = 7)

#RENAME EMPTY HEADERS
list1 = ["Age group (years)", "Unnamed: 2", "Unnamed: 3", "Unnamed: 4",␣
 ↪"Unnamed: 5", "Unnamed: 6",
         "Unnamed: 7", "Unnamed: 8", "Unnamed: 9"]
list2 = ["age", "2013", "2014", "2015", "2016", "2017", "2018", "2019", "2020"]
rename_column_name(df, list1, list2)

#SET DTYPES
df['age'] = df['age'].astype('category')
# df.iloc[0:7:, 0:9]
```

[944]:
```
#DATAFRAME
#PEOPLE THAT USED INTERNET OVER 3 MONTHS AGO
```

```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
three_months_agoDF = openFile('./internetusers2020.xlsx',
                              sheet_name = '1a',
                              header = 0,
                              names = default,
                              index_col = default,
                              usecols='A, L:S',
                              squeeze = False,
                              dtype = default,
                              engine = default,
                              converters = default,
                              true_values = default,
                              false_values = default,
                              skiprows = 7,
                              nrows = 7)

#RENAME EMPTY HEADERS
list1 = ["Age group (years)", "Unnamed: 11", "Unnamed: 12", "Unnamed: 13",␣
 ↪"Unnamed: 14", "Unnamed: 15",
         "Unnamed: 16", "Unnamed: 17", "Unnamed: 18"]
rename_column_name(three_months_agoDF, list1, list2)

#SET DTYPES
three_months_agoDF['age'] = three_months_agoDF['age'].astype('category')
# three_months_agoDF.iloc[0:7:, 0:9]
```

[945]:
```
#DATAFRAME
#PEOPLE THAT NEVER USED INTERNET

#READING CORRECT NON-EMPTY COLUMNS & ROWS
never_usedDF = openFile('./internetusers2020.xlsx',
                              sheet_name = '1a',
                              header = 0,
                              names = default,
                              index_col = default,
                              usecols='A, U:AB',
                              squeeze = False,
                              dtype = default,
                              engine = default,
                              converters = default,
                              true_values = default,
                              false_values = default,
                              skiprows = 7,
                              nrows = 7)

#RENAME EMPTY HEADERS
```

```
list1 = ["Age group (years)", "Unnamed: 20", "Unnamed: 21", "Unnamed: 22",␣
↪"Unnamed: 23", "Unnamed: 24",
        "Unnamed: 25", "Unnamed: 26", "Unnamed: 27"]
rename_column_name(never_usedDF, list1, list2)

#SET DTYPES
never_usedDF['age'] = never_usedDF['age'].astype('category')
# never_usedDF.iloc[0:7:, 0:9]
```

[946]:
```
#MELT COLUMNS INTO ROWS (YEARS UNDER THE 'YEAR' COLUMN, VALUES UNDER THE␣
↪'POPULATION')
ts = df.melt(id_vars= ['age'],
             var_name= 'year',
             value_name= 'population')
ts_three_months_ago = three_months_agoDF.melt(id_vars= ['age'],
                                              var_name= 'year',
                                              value_name= 'population')
ts_never_usedDF = never_usedDF.melt(id_vars= ['age'],
                                              var_name= 'year',
                                              value_name= 'population')
```

[947]:
```
#CREATE A DATETIME SERIES BY PARSING THE YEARS
ts['year'] = pd.to_datetime(ts['year'], format='%Y')
ts_three_months_ago['year'] = pd.to_datetime(ts_three_months_ago['year'],␣
↪format='%Y')
ts_never_usedDF['year'] = pd.to_datetime(ts_never_usedDF['year'], format='%Y')

#MAKE YEAR AN INDEX COLUMN (DATETIMEINDEX)
ts.set_index('year', inplace=True)
ts_three_months_ago.set_index('year', inplace=True)
ts_never_usedDF.set_index('year', inplace=True)

#CONVERT THE DATETIMEINDEX TO A PERIODINDEX OF YEAR FREQUENCY
ts = ts.to_period('Y')
ts_three_months_ago = ts_three_months_ago.to_period('Y')
ts_never_usedDF = ts_never_usedDF.to_period('Y')
```

[948]:
```
#CREATE THE HIERARCHICAL INDEXES
ts.set_index(['age'], inplace=True, append=True)
ts.sort_index(inplace=True)

ts_three_months_ago.set_index(['age'], inplace=True, append=True)
ts_three_months_ago.sort_index(inplace=True)

ts_never_usedDF.set_index(['age'], inplace=True, append=True)
ts_never_usedDF.sort_index(inplace=True)
```

8.1 For years 2013, 2016, 2020.

8.2 1. People that used internet in last 3 months.

8.3 2. People that used internet over 3 months ago.

8.4 3. People that never used internet.

[949]:
```
#USED IN LAST 3 MONTHS FOR CHOSEN YEARS
subsetInitial = ts.loc[('2013'),:]
subset_Mid = ts.loc[('2016'),:]
subsetEnd = ts.loc[('2020'),:]

#USED OVER 3 MONTHS AGO TABLE FOR CHOSEN YEARS
subset_Table_two_Initial = ts_three_months_ago.loc[('2013'),:]
subset_Table_two_Mid = ts_three_months_ago.loc[('2016'),:]
subset_Table_two_End = ts_three_months_ago.loc[('2020'),:]

#NEVER USED FOR CHOSEN YEARS
subset_Table_three_Initial = ts_never_usedDF.loc[('2013'),:]
subset_Table_three_Mid = ts_never_usedDF.loc[('2016'),:]
subset_Table_three_End = ts_never_usedDF.loc[('2020'),:]
```

[950]:
```
#SETTING UP DATA FOR FIRST PLOT FOR EACH ROW
axInitial = subsetInitial.reset_index()
axMid = subset_Mid.reset_index()
axEnd = subsetEnd.reset_index()

#SETTING UP DATA FOR SECOND PLOT FOR EACH ROW
ax2Initial = subset_Table_two_Initial.reset_index()
ax2Mid = subset_Table_two_Mid.reset_index()
ax2End = subset_Table_two_End.reset_index()

#SETTING UP DATA FOR THIRD PLOT FOR EACH ROW
ax3Initial= subset_Table_three_Initial.reset_index()
ax3Mid = subset_Table_three_Mid.reset_index()
ax3End = subset_Table_three_End.reset_index()


#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 3, figsize = (19,7), sharey = True, sharex = True)

#########################################################################################
#FIRST ROW
#FIRST PLOT
axes[0, 0].barh(axInitial['age'], axInitial['population'], color = ageCols[0])
axes[0, 0].set_title(titleOne, fontsize = title_size)
axes[0, 0].xaxis.grid(True, color ="grey", alpha=0.2)
axes[0, 0].set_ylabel('2013 age', fontsize = label_size)
```

```
#SECOND PLOT
axes[0, 1].barh(ax2Initial['age'], ax2Initial['population'], color = ageCols[1])
axes[0, 1].set_title(titleTwo, fontsize = title_size)
axes[0, 1].xaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
axes[0, 2].barh(ax3Initial['age'], ax3Initial['population'], color = ageCols[2])
axes[0, 2].set_title(titleThree, fontsize = title_size)
axes[0, 2].xaxis.grid(True, color ="grey", alpha=0.2)

#############################################################################
#SECOND ROW
#FIRST PLOT
axes[1, 0].barh(axMid ['age'], axMid['population'], color = ageCols[0])
axes[1, 0].xaxis.grid(True, color ="grey", alpha=0.2)
axes[1, 0].set_ylabel('2016 age', fontsize = label_size)

#SECOND PLOT
axes[1, 1].barh(ax2Mid ['age'], ax2Mid['population'], color = ageCols[1])
axes[1, 1].xaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
axes[1, 2].barh(ax3Mid['age'], ax3Mid['population'], color = ageCols[2])
axes[1, 2].xaxis.grid(True, color ="grey", alpha=0.2)

#############################################################################
#THIRD ROW
#FIRST PLOT
axes[2, 0].barh(axEnd['age'], axEnd['population'],color = ageCols[0])
axes[2, 0].xaxis.grid(True, color ="grey", alpha=0.2)
axes[2, 0].set_ylabel('2020 age', fontsize = label_size)
axes[2, 0].set_xlabel('population', fontsize = label_size)

#SECOND PLOT
axes[2, 1].barh(ax2End['age'], ax2End['population'],color = ageCols[1])
axes[2, 1].xaxis.grid(True, color ="grey", alpha=0.2)
axes[2, 1].set_xlabel('population', fontsize = label_size)

#THIRD PLOT
axes[2, 2].barh(ax3End['age'], ax3End['population'],color = ageCols[2])
axes[2, 2].xaxis.grid(True, color ="grey", alpha=0.2)
axes[2, 2].set_xlabel('population', fontsize = label_size)

#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
```

```
                    right=0.9,
                    top=2.4,
                    wspace=0.1,
                    hspace=0.1)
plt.show()

#SHOW TABULAR DATA
axEnd.head(15)
```

[950]:

	year	age	population
0	2020	16-24	6844
1	2020	25-34	8908
2	2020	35-44	8339
3	2020	45-54	8716
4	2020	55-64	7796
5	2020	65-74	5504
6	2020	75+	2933

8.5 Conclusion from the above plots.

On the plots generated we can observe three rows for the initial, mid and end year of data collected from sheet '1a'. The data presents the population of each age group depending on how active they are with using the internet.

Firstly, comparing the rows we can see that people, which used the internet in *'last 3 months'* are the majority, whereas the people that used the internet *'over 3 months ago'* are the minority. The first group of people in the first column that utilizes the internet has a population in age groups from '16-24' to '55-64', which goes mostly beyond the 60 thousand mark. It emphasises that those users of this technology in many cases do not decrease but increase. For example, the age '25-34' has increased from 2013 to 2020 going over 90 thousand.

The problem with the middle column is that the people that used the internet *'over 3 months ago'* do not present much information. However, I have realised that based on the 2nd column people have a tendency to either use the internet or not use it at all, when we compare the 3 columns together where the second one is the least chosen option.

In addition, the interesting part of not using the internet is that its population in column 3 especially at the age from '45-54' to '75+' has decreased and probably chosen to use the internet in latest years at least couple of times in 3 months. The pattern here shows that the technological improvement and the need to use the internet minimalizes the population of people from column 3.

8.6 As a percentage for years 2013, 2016, 2020.

8.7 1. People that used internet in last 3 months.

8.8 2. People that used internet over 3 months ago.

8.9 3. People that never used internet.

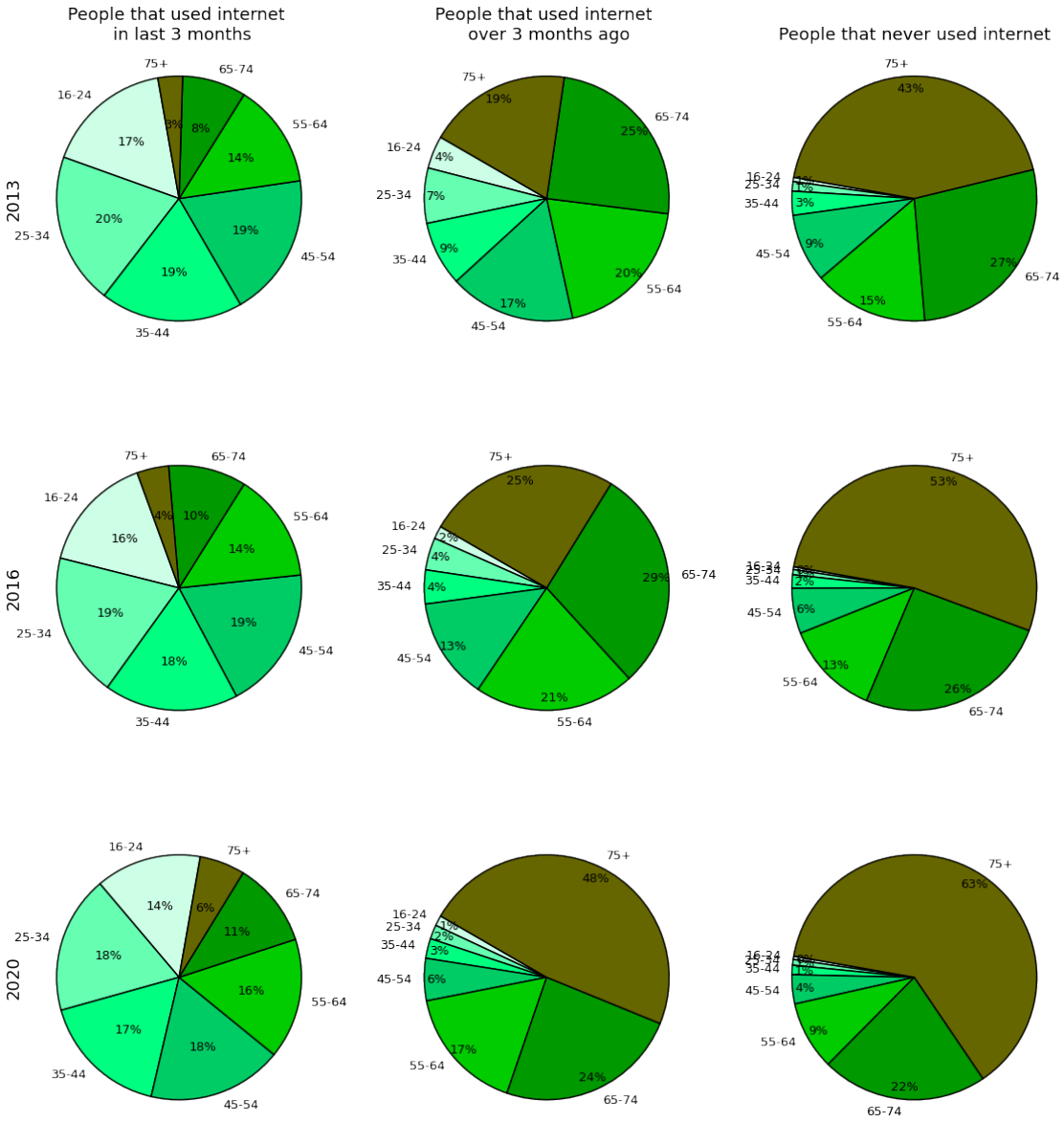
[951]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 3, figsize = (19,7))

##########################################################################
#FIRST ROW
#FIRST PLOT
ax = subsetInitial['population'].plot.pie(
    ax = axes[0, 0],
    autopct= '%1.0f%%',
    fontsize = 13,
```

```
    startangle = 100,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels
)
axes[0, 0].set_title(titleOne, fontsize = 18)
axes[0, 0].set_ylabel('2013', fontsize = 18)

#SECOND PLOT
ax = subset_Table_two_Initial['population'].plot.pie(
    ax = axes[0, 1],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 150,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels,
    pctdistance = 0.9
)
axes[0, 1].set_title(titleTwo, fontsize = 18)
axes[0, 1].set_ylabel('')

#THIRD PLOT
ax = subset_Table_three_Initial['population'].plot.pie(
    ax = axes[0, 2],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 170,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels,
    pctdistance = 0.9
)
axes[0, 2].set_title(titleThree, fontsize = 18)
axes[0, 2].set_ylabel('')

#############################################################################
#SECOND ROW
#FIRST PLOT
ax = subset_Mid['population'].plot.pie(
    ax = axes[1, 0],
```

```
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 110,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels
)
axes[1, 0].set_ylabel('')
axes[1, 0].set_ylabel('2016', fontsize = 18)

#SECOND PLOT
ax = subset_Table_two_Mid['population'].plot.pie(
    ax = axes[1, 1],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 150,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels,
    pctdistance = 0.9
)
axes[1, 1].set_ylabel('')

#THIRD PLOT
ax = subset_Table_three_Mid['population'].plot.pie(
    ax = axes[1, 2],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 170,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels,
    pctdistance = 0.9
)
axes[1, 2].set_ylabel('')

################################################################################
#THIRD ROW
#FIRST PLOT
ax = subsetEnd['population'].plot.pie(
    ax = axes[2, 0],
```

```
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 80,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels
)
axes[2, 0].set_ylabel('')
axes[2, 0].set_ylabel('2020', fontsize = 18)

#SECOND PLOT
ax = subset_Table_two_End['population'].plot.pie(
    ax = axes[2, 1],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 150,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels,
    pctdistance = 0.9
)
axes[2, 1].set_ylabel('')

#THIRD PLOT
ax = subset_Table_three_End['population'].plot.pie(
    ax = axes[2, 2],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 170,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    colors = colorsPie,
    labels = labels,
    pctdistance = 0.9
)
axes[2, 2].set_ylabel('')

##############################################################################
#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
```

```
                    top=2.4,
                    wspace=0.2,
                    hspace=0.2)
plt.show()
```



8.10 Conclusion from the above plots.

For the percentage pie charts, the message is very clear for each column.

The first column shows that the variety of each age does not differ that much from the others, maximum by 17% in year *'2013'* and this difference decreases in year *'2016'* and *'2020'*. The example taken was age *'25-34'* and *'75+'*.

The second column, on the other hand, makes it very clear that older age groups are the ones that used the internet less, comparing it to age groups younger than *'35-44'*. This trend of the higher population that does not use the internet is even higher for the older age groups in the third column.

To summarise it I believe that younger age groups are the ones that use the internet more and the older age groups such as *'35-44'* and above can probably not use it for many months or not at all.

8.11 Difference (annually) of first, middle and end age groups as an example which:

8.12 1. People that used internet in last 3 months.

8.13 2. People that used internet over 3 months ago.

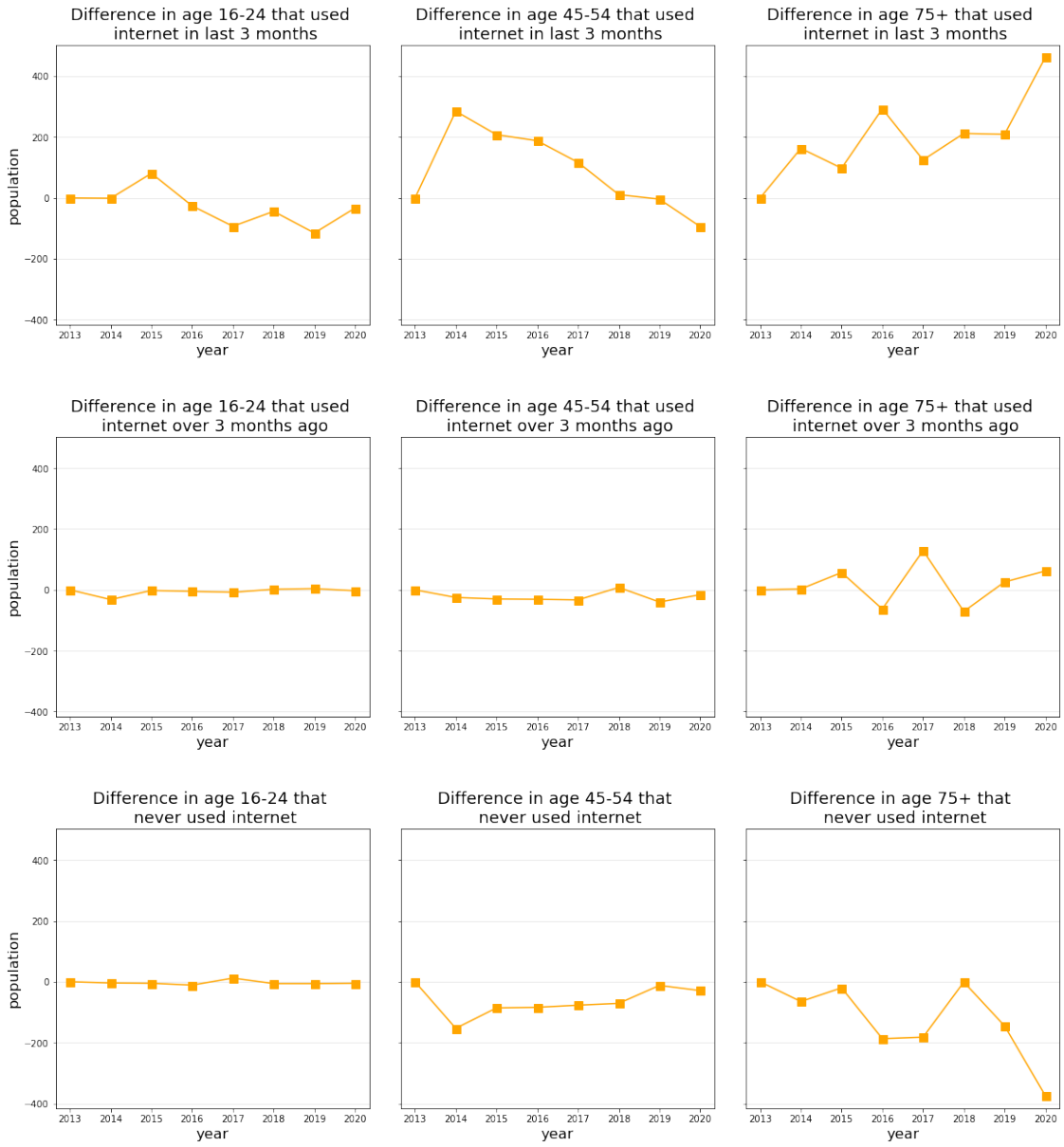
8.14 3. People that never used internet.

[952]:
```
subsetAge1Table1 = ts.xs(['16-24'], level=['age'])
subsetAge2Table1 = ts.xs(['45-54'], level=['age'])
subsetAge3Table1 = ts.xs(['75+'], level=['age'])

subsetAge1Table2 = ts_three_months_ago.xs(['16-24'], level=['age'])
subsetAge2Table2 = ts_three_months_ago.xs(['45-54'], level=['age'])
subsetAge3Table2 = ts_three_months_ago.xs(['75+'], level=['age'])

subsetAge1Table3 = ts_never_usedDF.xs(['16-24'], level=['age'])
subsetAge2Table3 = ts_never_usedDF.xs(['45-54'], level=['age'])
subsetAge3Table3 = ts_never_usedDF.xs(['75+'], level=['age'])
```

[953]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 3, figsize = (19,7), sharey=True)


#FIRST ROW
#FIRST PLOT
tableFirstPlot = subsetAge1Table1.diff().fillna(0).reset_index()
axes[0, 0].plot(tableFirstPlot['population'], style, ms = mSize, color = col)
axes[0, 0].set_xticklabels(x_labels)
axes[0, 0].yaxis.grid(True, color ="grey", alpha=0.2)
axes[0, 0].set_title('Difference in age 16-24 that used \n internet in last 3␣
 ↪months', fontsize = title_size)
axes[0, 0].set_ylabel('population', fontsize = label_size)
axes[0, 0].set_xlabel('year', fontsize = label_size)

#SECOND PLOT
tableSecondPlot = subsetAge2Table1.diff().fillna(0).reset_index()
axes[0, 1].plot(tableSecondPlot['population'], style, ms = mSize, color = col)
axes[0, 1].set_xticklabels(x_labels)
axes[0, 1].yaxis.grid(True, color ="grey", alpha=0.2)
```

```
axes[0, 1].set_title('Difference in age 45-54 that used \n internet in last 3␣
 ↪months', fontsize = title_size)
axes[0, 1].set_xlabel('year', fontsize = label_size)

#THIRD PLOT
tableThirdPlot = subsetAge3Table1.diff().fillna(0).reset_index()
axes[0, 2].plot(tableThirdPlot['population'], style, ms = mSize, color = col)
axes[0, 2].set_xticklabels(x_labels)
axes[0, 2].yaxis.grid(True, color ="grey", alpha=0.2)
axes[0, 2].set_title('Difference in age 75+ that used \n internet in last 3␣
 ↪months', fontsize = title_size)
axes[0, 2].set_xlabel('year', fontsize = label_size)


#SECOND ROW
#FOURTH PLOT
tableFourthPlot = subsetAge1Table2.diff().fillna(0).reset_index()
axes[1, 0].plot(tableFourthPlot['population'], style, ms = mSize, color = col)
axes[1, 0].set_xticklabels(x_labels)
axes[1, 0].yaxis.grid(True, color ="grey", alpha=0.2)
axes[1, 0].set_title('Difference in age 16-24 that used \n internet over 3␣
 ↪months ago', fontsize = title_size)
axes[1, 0].set_ylabel('population', fontsize = label_size)
axes[1, 0].set_xlabel('year', fontsize = label_size)

#FIFTH PLOT
tableFifthPlot = subsetAge2Table2.diff().fillna(0).reset_index()
axes[1, 1].plot(tableFifthPlot['population'], style, ms = mSize, color = col)
axes[1, 1].set_xticklabels(x_labels)
axes[1, 1].yaxis.grid(True, color ="grey", alpha=0.2)
axes[1, 1].set_title('Difference in age 45-54 that used \n internet over 3␣
 ↪months ago', fontsize = title_size)
axes[1, 1].set_xlabel('year', fontsize = label_size)

#SIXTH PLOT
tableSixthPlot = subsetAge3Table2.diff().fillna(0).reset_index()
axes[1, 2].plot(tableSixthPlot['population'], style, ms = mSize, color = col)
axes[1, 2].set_xticklabels(x_labels)
axes[1, 2].yaxis.grid(True, color ="grey", alpha=0.2)
axes[1, 2].set_title('Difference in age 75+ that used \n internet over 3 months␣
 ↪ago', fontsize = title_size)
axes[1, 2].set_xlabel('year', fontsize = label_size)


#THIRD ROW
#SEVENTH PLOT
tableSeventhPlot = subsetAge1Table3.diff().fillna(0).reset_index()
```

```
axes[2, 0].plot(tableSeventhPlot['population'], style, ms = mSize, color = col)
axes[2, 0].set_xticklabels(x_labels)
axes[2, 0].yaxis.grid(True, color ="grey", alpha=0.2)
axes[2, 0].set_title('Difference in age 16-24 that \n never used internet',␣
 ↪fontsize = title_size)
axes[2, 0].set_ylabel('population', fontsize = label_size)
axes[2, 0].set_xlabel('year', fontsize = label_size)

#EIGHTH PLOT
tableEighthPlot = subsetAge2Table3.diff().fillna(0).reset_index()
axes[2, 1].plot(tableEighthPlot['population'], style, ms = mSize, color = col)
axes[2, 1].set_xticklabels(x_labels)
axes[2, 1].yaxis.grid(True, color ="grey", alpha=0.2)
axes[2, 1].set_title('Difference in age 45-54 that \n never used internet',␣
 ↪fontsize = title_size)
axes[2, 1].set_xlabel('year', fontsize = label_size)

#NINETH PLOT
tableNinethPlot = subsetAge3Table3.diff().fillna(0).reset_index()
axes[2, 2].plot(tableNinethPlot['population'], style, ms = mSize, color = col)
axes[2, 2].set_xticklabels(x_labels)
axes[2, 2].yaxis.grid(True, color ="grey", alpha=0.2)
axes[2, 2].set_title('Difference in age 75+ that \n never used internet',␣
 ↪fontsize = title_size)
axes[2, 2].set_xlabel('year', fontsize = label_size)


#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=2.4,
                    wspace=0.1,
                    hspace=0.4)
plt.show()
```



8.15 Conclusion from the above plots.

Above we can see a set of graphs that presents three age groups as an example of how people tend to use the internet/not use it.

On the second and third plot of row one, I have defined two patterns.

The first one shows that middle age group and the difference between decreases over the years. This indicates that the population of the middle age group will go towards higher or lesser use of the internet.

The second finding is that the oldest age group has increased indifference throughout the years,

which suggest that the necessity of using the internet made them use it more often.

Nextly, on the second row the situation of rare using is quite stable with some exception for year *'2017'* for age *'75+'*, where it may be possible that something impactful has happened for older age groups that made them adopt it recurrently.

Moreover, the stable situation continues for row 3 where initial and middle age groups do not vary that much. On the other hand, the difference of the oldest age group decreases which may mean that the third plot of row 3 and the third plot of row 1 has some correlation, especially when we look from year *'2018'* onwards. It brings attention as when one decreases the other one improves.

9 Gender

Data loading and cleaning

[954]:
```
#DATAFRAME
#GENDER THAT USED INTERNET IN LAST 3 MONTHS

#READING CORRECT NON-EMPTY COLUMNS & ROWS
genderDF = openFile('./internetusers2020.xlsx',
                                sheet_name = '1a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A, C:J',
                                squeeze = False,
                                dtype = default,
                                engine = default,
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = 16,
                                nrows = 2)

#RENAME EMPTY HEADERS
list1 = ["Sex", "Unnamed: 2", "Unnamed: 3", "Unnamed: 4", "Unnamed: 5",␣
 ↪"Unnamed: 6",
         "Unnamed: 7", "Unnamed: 8", "Unnamed: 9"]
list2 = ["sex", "2013", "2014", "2015", "2016", "2017", "2018", "2019", "2020"]
rename_column_name(genderDF, list1, list2)

#RENAME EMPTY HEADERS
genderDF.rename(columns={'Sex': 'sex',
                         'Unnamed: 2' : '2013',
                         'Unnamed: 3' : '2014',
                         'Unnamed: 4' : '2015',
                         'Unnamed: 5' : '2016',
                         'Unnamed: 6' : '2017',
                         'Unnamed: 7' : '2018',
```

```
                          'Unnamed: 8' : '2019',
                          'Unnamed: 9' : '2020'}, inplace = True)

#SET DTYPES
genderDF['sex'] = genderDF['sex'].astype('category')
```

[955]:
```
#DATAFRAME
#GENDER THAT USED INTERNET OVER 3 MONTHS AGO

#READING CORRECT NON-EMPTY COLUMNS & ROWS
genderDF_2 = openFile('./internetusers2020.xlsx',
                              sheet_name = '1a',
                              header = 0,
                              names = default,
                              index_col = default,
                              usecols = 'A, L:S',
                              squeeze = False,
                              dtype = default,
                              engine = default,
                              converters = default,
                              true_values = default,
                              false_values = default,
                              skiprows = 16,
                              nrows = 2)

#RENAME EMPTY HEADERS
list1 = ["Sex", "Unnamed: 11", "Unnamed: 12", "Unnamed: 13", "Unnamed: 14",␣
 ↪"Unnamed: 15",
         "Unnamed: 16", "Unnamed: 17", "Unnamed: 18"]
rename_column_name(genderDF_2, list1, list2)

#SET DTYPES
genderDF_2['sex'] = genderDF_2['sex'].astype('category')
```

[956]:
```
#DATAFRAME
#GENDER THAT NEVER USED INTERNET

#READING CORRECT NON-EMPTY COLUMNS & ROWS
genderDF_3 = openFile('./internetusers2020.xlsx',
                              sheet_name = '1a',
                              header = 0,
                              names = default,
                              index_col = default,
                              usecols = 'A, U:AB',
                              squeeze = False,
                              dtype = default,
                              engine = default,
```

```
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = 16,
                                nrows = 2)

#RENAME EMPTY HEADERS
list1 = ["Sex", "Unnamed: 20", "Unnamed: 21", "Unnamed: 22", "Unnamed: 23",␣
 ↪"Unnamed: 24",
         "Unnamed: 25", "Unnamed: 26", "Unnamed: 27"]
rename_column_name(genderDF_3, list1, list2)

#SET DTYPES
genderDF_3['sex'] = genderDF_3['sex'].astype('category')
```

[957]:
```
#MELT COLUMNS INTO ROWS (WOMEN/MEN UNDER THE 'SEX' COLUMN, VALUES UNDER THE␣
 ↪'POPULATION')
ts_Gender = genderDF.melt(id_vars = ['sex'], var_name='year', value_name =␣
 ↪'population')
ts_three_months_ago_Gender = genderDF_2.melt(id_vars = ['sex'],␣
 ↪var_name='year', value_name = 'population')
ts_never_used_Gender = genderDF_3.melt(id_vars = ['sex'], var_name='year',␣
 ↪value_name = 'population')
```

[958]:
```
#CREATE A DATETIME SERIES BY PARSING THE YEARS
ts_Gender['year'] = pd.to_datetime(ts_Gender['year'], format ='%Y')
ts_three_months_ago_Gender['year'] = pd.
 ↪to_datetime(ts_three_months_ago_Gender['year'], format='%Y')
ts_never_used_Gender['year'] = pd.to_datetime(ts_never_used_Gender['year'],␣
 ↪format='%Y')

#MAKE YEAR AN INDEX COLUMN (DATETIMEINDEX)
ts_Gender.set_index(['year'], inplace = True)
ts_three_months_ago_Gender.set_index(['year'], inplace = True)
ts_never_used_Gender.set_index(['year'], inplace = True)

#CONVERT THE DATETIMEINDEX TO A PERIODINDEX OF YEAR FREQUENCY
ts_Gender = ts_Gender.to_period('Y')
ts_three_months_ago_Gender = ts_three_months_ago_Gender.to_period('Y')
ts_never_used_Gender = ts_never_used_Gender.to_period('Y')
```

[959]:
```
#CREATE THE HIERARCHICAL INDEXES
ts_Gender.set_index(['sex'], inplace=True, append = True)
ts_three_months_ago_Gender.set_index(['sex'], inplace=True, append = True)
ts_never_used_Gender.set_index(['sex'], inplace=True, append = True)
```

```
ts_Gender.sort_index(inplace=True)
ts_three_months_ago_Gender.sort_index(inplace=True)
ts_never_used_Gender.sort_index(inplace=True)
```

9.1 Gender (annually):

9.2 1. People that used internet in last 3 months.

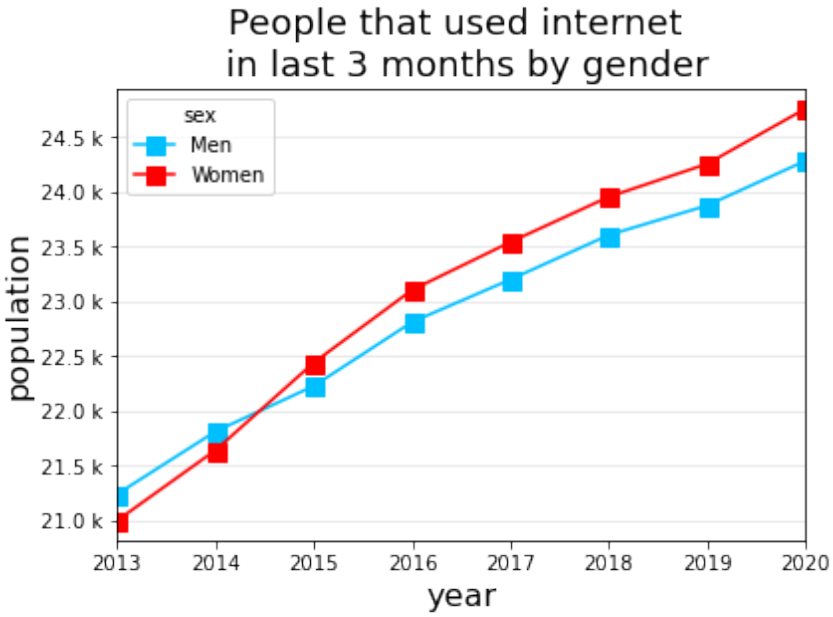
9.3 2. People that used internet over 3 months ago.

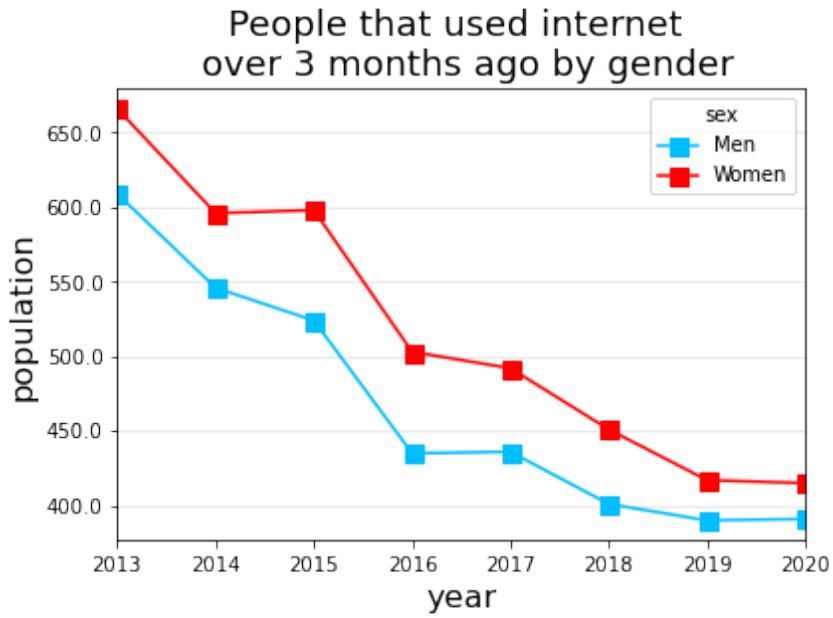
9.4 3. People that never used internet.

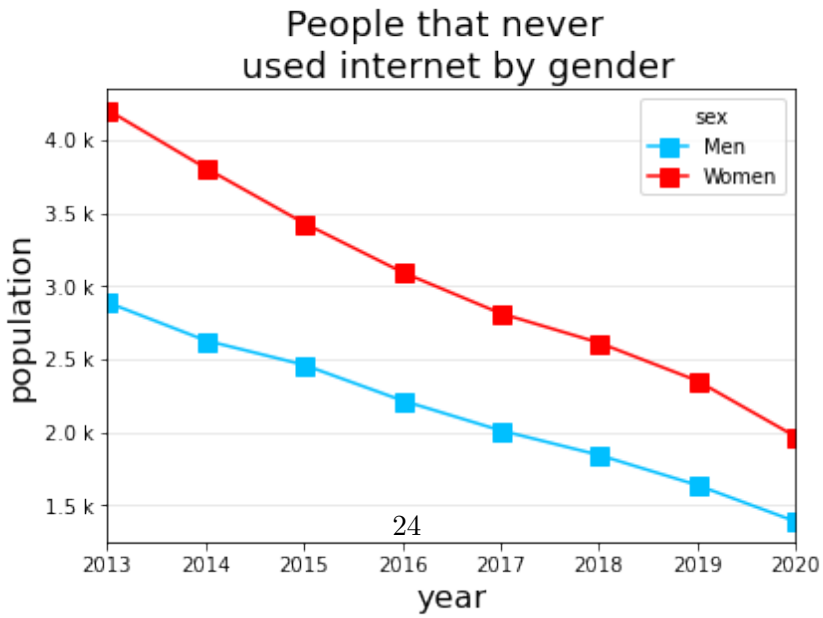
[960]:
```
in_out = ts_Gender['population'].groupby(level=['year', 'sex']).sum()
in_out2 = ts_three_months_ago_Gender['population'].groupby(level=['year',␣
↪'sex']).sum()
in_out3 = ts_never_used_Gender['population'].groupby(level=['year', 'sex']).
↪sum()
```

[961]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 1, figsize = (6, 15))

#FIRST PLOT
in_out.unstack('sex').plot(ax = axes[0], marker = 's', ms = mSize, color =␣
↪colors)
axes[0].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[0].set_title('People that used internet \n in last 3 months by gender',
             fontsize = title_size)
axes[0].set_ylabel('population', fontsize = label_size)
axes[0].set_xlabel('year', fontsize= label_size)
axes[0].yaxis.grid(True, color ="grey", alpha=0.2)

#SECOND PLOT
in_out2.unstack('sex').plot(ax=axes[1], marker = 's', ms = mSize, color =␣
↪colors)
axes[1].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[1].set_title('People that used internet \n over 3 months ago by gender',␣
↪fontsize = title_size)
axes[1].set_ylabel('population', fontsize = label_size)
axes[1].set_xlabel('year', fontsize= label_size)
axes[1].yaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
in_out3.unstack('sex').plot(ax=axes[2], marker = 's', ms = mSize, color =␣
↪colors)
axes[2].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[2].set_title('People that never \n used internet by gender', fontsize =␣
↪title_size)
```

```
axes[2].set_ylabel('population', fontsize = label_size)
axes[2].set_xlabel('year', fontsize= label_size)
axes[2].yaxis.grid(True, color ="grey", alpha=0.2)

#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=0.9,
                    wspace=0.1,
                    hspace=0.4)
plt.show()

#SHOW TABULAR DATA
in_out3.head(15)
```


People that used internet
in last 3 months by gender
sex
Men
Women
population
24.5 k
24.0 k
23.5 k
23.0 k
22.5 k
22.0 k
21.5 k
21.0 k
2013
2014
2015
2016
2017
2018
2019
2020
year


People that used internet
over 3 months ago by gender
sex
Men
Women
population
650.0
600.0
550.0
500.0
450.0
400.0
2013
2014
2015
2016
2017
2018
2019
2020
year


People that never
used internet by gender
sex
Men
Women
population
4.0 k
3.5 k
3.0 k
2.5 k
2.0 k
1.5 k
2013
2014
2015
2016
2017
2018
2019
2020
year

[961]:
```
year  sex
2013  Men      2890
      Women    4209
2014  Men      2629
      Women    3810
2015  Men      2462
      Women    3433
2016  Men      2218
      Women    3097
2017  Men      2013
      Women    2815
2018  Men      1845
      Women    2614
2019  Men      1639
      Women    2354
2020  Men      1390
Name: population, dtype: int64
```

9.5 Conclusion from the above plots.

The first plot allows us to see that 'Men' was a larger population of people that used the internet from 2013 up to mid of 2014. This situation has changed, and the Women population was always higher from that point forward, when it comes to using the internet in *'last 3 months'*.

The second and third graph displays that both genders have gone towards a drop over the years for both rare/no usage. It may suggest the correlation of the 2nd and 3rd plot to a situation of gender increase in plot one but to make it fair we would need more respondents for using internet *'3 months ago'* and *'never used internet'*. It is because, the population of the 2nd plot is 700 at the highest with 4,5 thousand for the 3rd plot, whereas the first plot has 24,5 thousand respondents. It would be difficult to strongly state the the population of hundreds has an effect on tens of thousands.

9.6 Gender as a percentage for years 2013, 2016, 2020:

9.7 1. People that used internet in last 3 months.

9.8 2. People that used internet over 3 months ago.

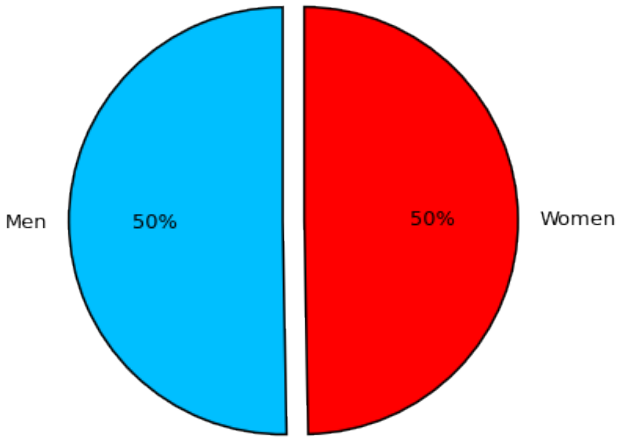
9.9 3. People that never used internet.

[962]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 1, figsize = (10,7))
explode = [0.05, 0.05]
labels = ['Men', 'Women']

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF['2013'].plot.pie(
    ax = axes[0],
```

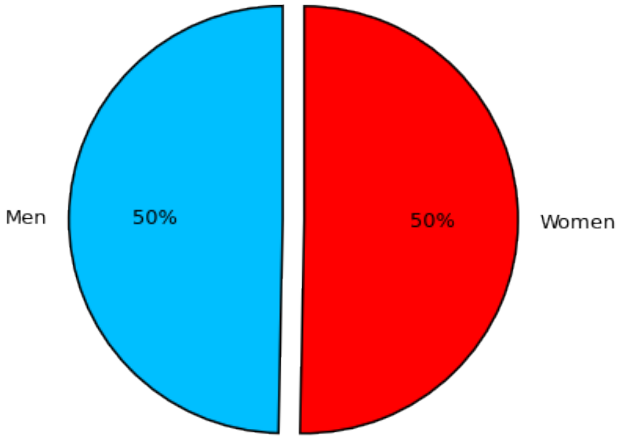
```
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[0].set_title("Gender of survey particpants that used internet in last 3␣
 ↪months \n in % for 2013", fontsize = 18)
axes[0].set_ylabel('')

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF['2016'].plot.pie(
    ax = axes[1],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[1].set_title("Gender of survey particpants that used internet in last 3␣
 ↪months \n in % for 2016", fontsize = 18)
axes[1].set_ylabel('')

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF['2020'].plot.pie(
    ax = axes[2],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[2].set_title("Gender of survey particpants that used internet in last 3␣
 ↪months \n in % for 2020", fontsize = 18)
axes[2].set_ylabel('')
```

```
#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=2.4, )
plt.show()

#SHOW TABULAR DATA
genderDF['2020'].head()
```

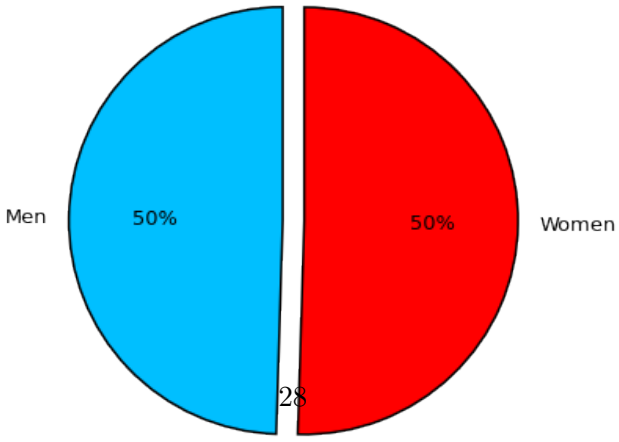
Gender of survey particpants that used internet in last 3 months in % for 2013



Gender of survey particpants that used internet in last 3 months in % for 2016



Gender of survey particpants that used internet in last 3 months in % for 2020

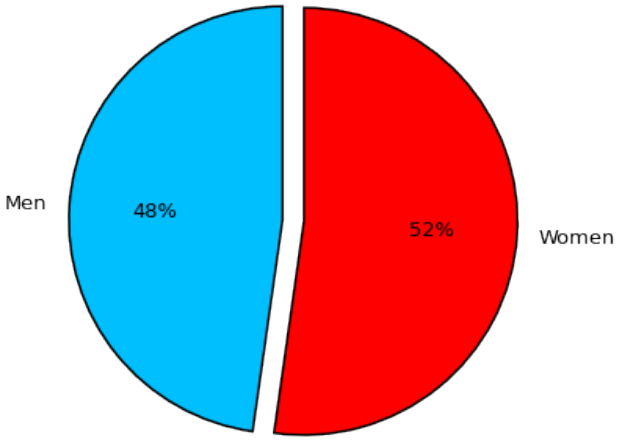


[962]:
```
0    24283
1    24758
Name: 2020, dtype: int64
```

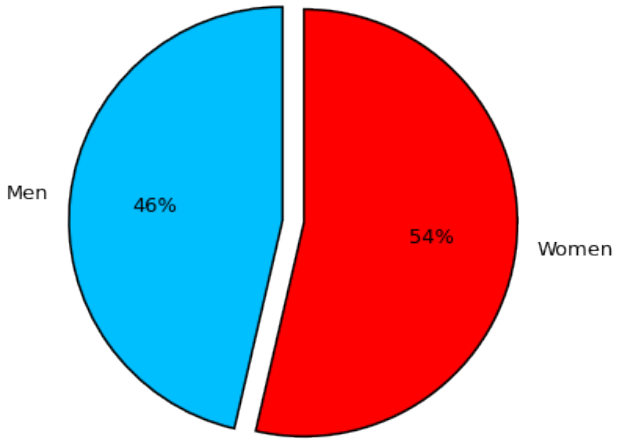
[963]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 1, figsize = (10,7))

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF_2['2013'].plot.pie(
    ax = axes[0],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[0].set_title("Gender of survey particpants that used internet over 3␣
 ↪months ago \n in % for 2013", fontsize = 18)
axes[0].set_ylabel('')

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF_2['2016'].plot.pie(
    ax = axes[1],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[1].set_title("Gender of survey particpants that used internet over 3␣
 ↪months ago \n in % for 2016", fontsize = 18)
axes[1].set_ylabel('')

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF_2['2020'].plot.pie(
    ax = axes[2],
```

```
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[2].set_title("Gender of survey particpants that used internet over 3␣
 ↪months ago \n in % for 2020", fontsize = 18)
axes[2].set_ylabel('')

#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=2.4, )
plt.show()

#SHOW TABULAR DATA
genderDF_2['2020'].head()
```

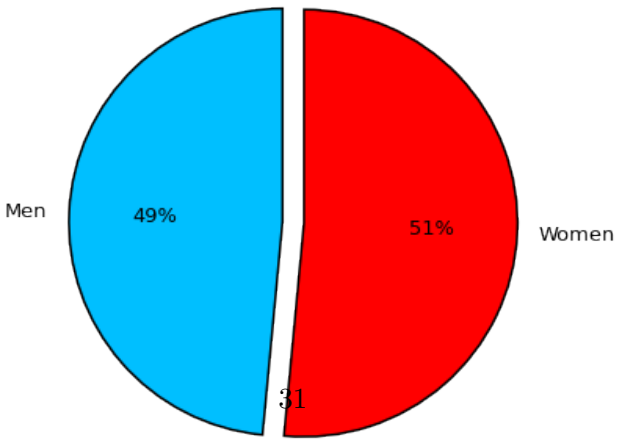
Gender of survey particpants that used internet over 3 months ago in % for 2013



Gender of survey particpants that used internet over 3 months ago in % for 2016



Gender of survey particpants that used internet over 3 months ago in % for 2020

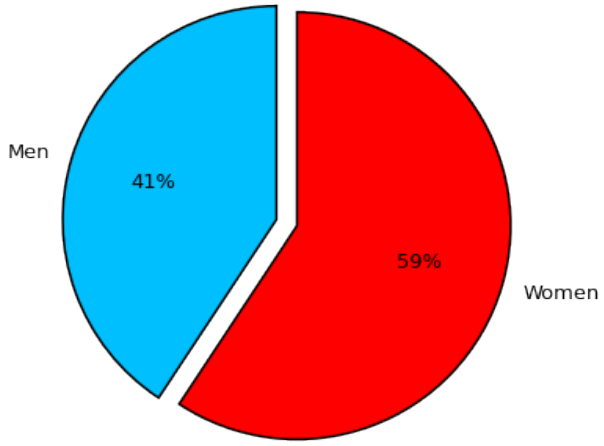


[963]:
```
0    391
1    415
Name: 2020, dtype: int64
```

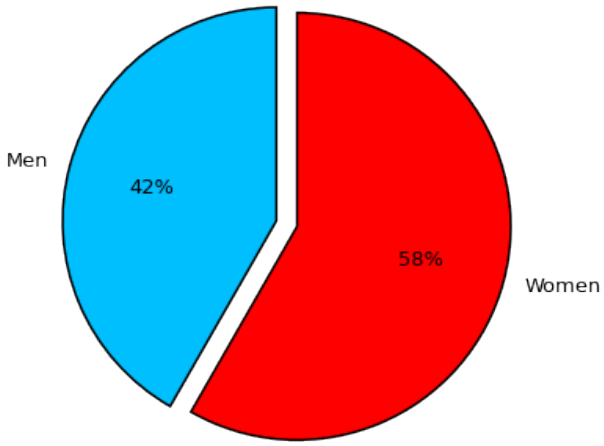
[964]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 1, figsize = (10,7))

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF_3['2013'].plot.pie(
    ax = axes[0],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[0].set_title("Gender of survey particpants that never used internet \n in␣
 ↪% for 2013", fontsize = 18)
axes[0].set_ylabel('')

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF_3['2016'].plot.pie(
    ax = axes[1],
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[1].set_title("Gender of survey particpants that never used internet \n in␣
 ↪% for 2016", fontsize = 18)
axes[1].set_ylabel('')

#PLOTTING A GENDER OF REPONDENTS.
ax = genderDF_3['2020'].plot.pie(
    ax = axes[2],
```

```
    autopct= '%1.0f%%',
    fontsize = 13,
    startangle = 90,
    wedgeprops = {"edgecolor" : "black",
                  'linewidth': 1.5,
                  'antialiased': True},
    explode = explode,
    colors = colors,
    labels = labels
)
axes[2].set_title("Gender of survey particpants that never used internet \n in␣
 ↪% for 2020", fontsize = 18)
axes[2].set_ylabel('')

#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=2.4, )
plt.show()

#SHOW TABULAR DATA
genderDF_3['2020'].head()
```

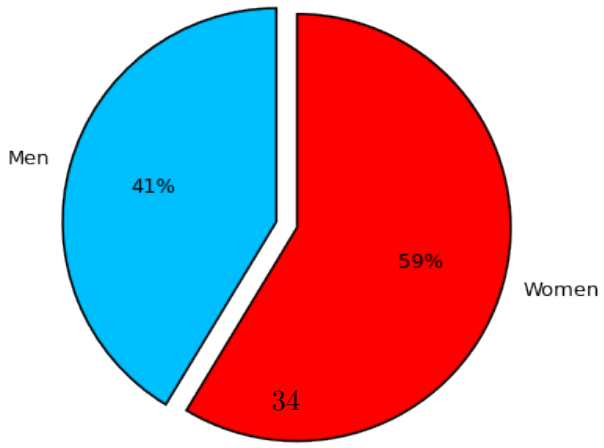
Gender of survey particpants that never used internet in % for 2013



Gender of survey particpants that never used internet in % for 2016



Gender of survey particpants that never used internet in % for 2020



[964]:
```
0    1390
1    1972
Name: 2020, dtype: int64
```

9.10 Conclusion from the above 3 sets of plots.

The usage of internet through the years in *'last three months'* had a little change since the differences between *'Men'* and *'Women'* in percentage was exactly at the mark of 50%. It happened for this internet usage as the amount of both genders was different only by a couple of hundreds, which can be seen a little for 2016 and 2020 when we will keep attention on them.

Nonetheless, in the second set of plots, we start to see a difference of 4%, 8%, and 2% for the side of *'Women'* at the given years. An interesting fact that comes from it is that the difference between males and females has changed to a minimum in 2020 when we look at mentioned percentages. This begs the question why there was a sudden drop from 8% to 2% but we do not have such a details to base any opinion on that.

After all, the lack of usage goes towards females much more in the last set of plots, but this time the changes from beginning to end were not that drastic.

9.11 Annually

9.12 1. Difference between Women and Men that used internet in last 3 months.

9.13 2. Difference between Women and Men that used internet in over 3 months ago.

9.14 3. Difference between Women and Men that never used internet.

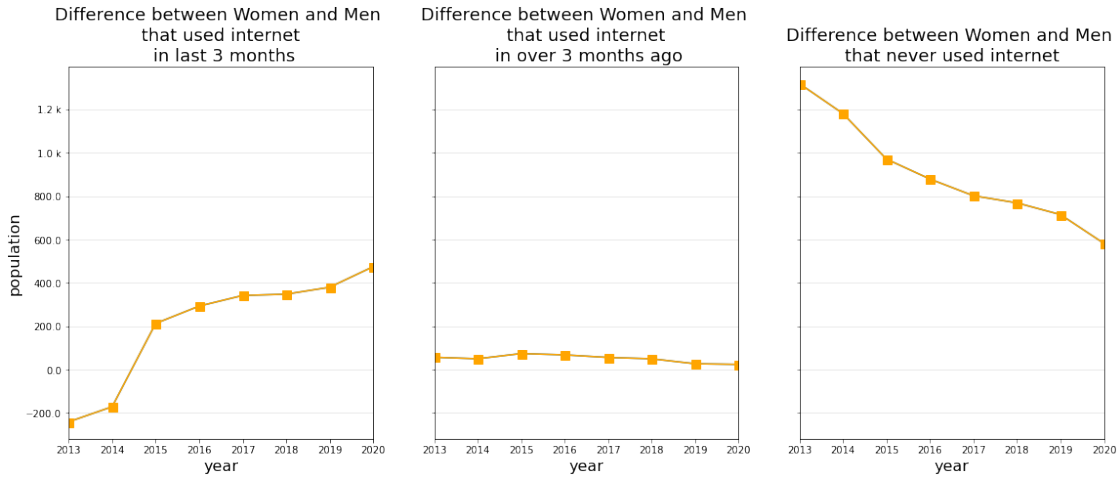
9.14.1 Note

We use larger group to show the difference between genders. In this case Women.

[965]:
```
#HERE I HAVE TAKEN LARGER GROUP AS FIRST (WOMEN) TO SEE MORE ACCURATE␣
↪DIFFERENCES
difference = (in_out.xs('Women', level='sex') - in_out.xs('Men', level='sex'))
difference.name = 'difference'
max_year = difference.idxmax()

#HERE I HAVE TAKEN LARGER GROUP AS FIRST (WOMEN) TO SEE MORE ACCURATE␣
↪DIFFERENCES
differenceTwo = (in_out2.xs('Women', level='sex') - in_out2.xs('Men',␣
↪level='sex'))
differenceTwo.name = 'differenceTwo'
max_yearTwo = differenceTwo.idxmax()

#HERE I HAVE TAKEN LARGER GROUP AS FIRST (WOMEN) TO SEE MORE ACCURATE␣
↪DIFFERENCES
```

```
differenceThree = (in_out3.xs('Women', level='sex') - in_out3.xs('Men',␣
 ↪level='sex'))
differenceThree.name = 'differenceThree'
max_yearThree = differenceThree.idxmax()
```

[966]:
```
#VARS
col = 'orange'

#CREATING SUBPLOTS
# fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(6, 9), sharex=True)
fix, axes = plt.subplots(1, 3, figsize = (19, 7), sharey=True)

#FIRST PLOT
difference.plot(ax = axes[0])
axes[0].plot(difference, style, ms = mSize, color = col)
axes[0].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[0].set_title('Difference between Women and Men \n that used internet \n in␣
 ↪last 3 months', fontsize= title_size)
axes[0].set_xlabel('year', fontsize = label_size)
axes[0].set_ylabel('population', fontsize = label_size)
axes[0].yaxis.grid(True, color ="grey", alpha=0.2)

#SECOND PLOT
differenceTwo.plot(ax = axes[1])
axes[1].plot(differenceTwo, style, ms = mSize, color = col)
axes[1].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[1].set_title('Difference between Women and Men \n that used internet \n in␣
 ↪over 3 months ago', fontsize= title_size)
axes[1].set_xlabel('year', fontsize = label_size)
axes[1].yaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
differenceThree.plot(ax = axes[2])
axes[2].plot(differenceThree, style, ms = mSize, color = col)
axes[2].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[2].set_title('Difference between Women and Men \nthat never used␣
 ↪internet', fontsize= title_size)
axes[2].set_xlabel('year', fontsize = label_size)
axes[2].yaxis.grid(True, color ="grey", alpha=0.2)

plt.show()

#SHOW TABULAR DATA
difference.head(15)
```



[966]:
```
year
2013   -241
2014   -171
2015    213
2016    293
2017    342
2018    348
2019    380
2020    475
Freq: A-DEC, Name: difference, dtype: int64
```

9.15 Conclusion from the above plots.

Initially looking at three graphs, we can see that the leaning of each of them goes toward some other direction.

Moving on to the first plot, we are presented with a plot that appears to give us data about *'Women'* being the larger gender group when it comes to regular usage of the internet. This trend has not stopped going upwards from 2015 up to 2020. This means that this trend might continue slowly rising or ascending by a large amount for next years.

In contrast, the second plot shows little or no difference between each gender. Also is likely that the difference will worsen from 2020 until it hits 0 or negative numbers, since that is what the latest data point presents.

Furthermore, the numbers plunge over the years in the third plot that went from thousands to hundreds. It is very iteresting as it proposes that males are more leaned towards technology and the females had no intention to use the internet until it was more common from 2014 ahead.

9.16 Annually:

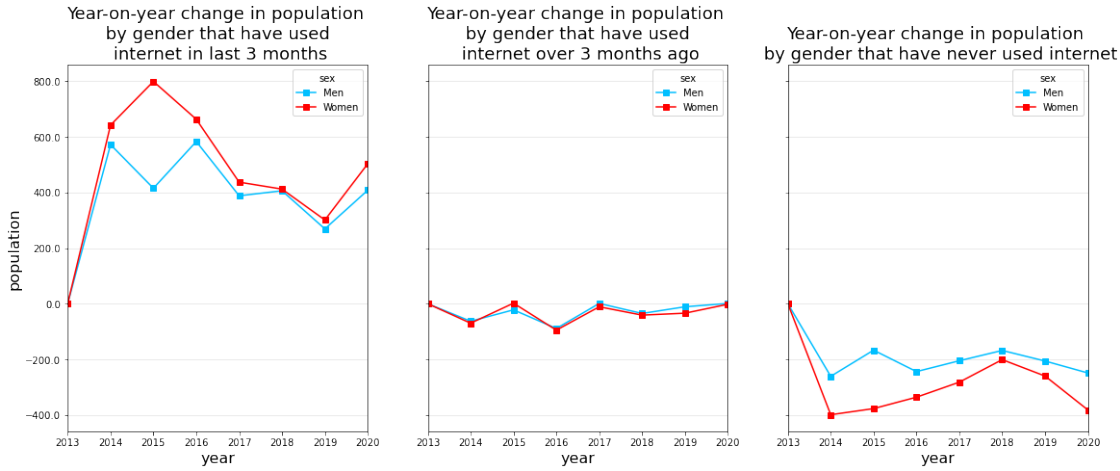
9.17 1. Year-on-year change in population by gender that used internet in last 3 months.

9.18 2. Year-on-year change in population by gender that used internet over 3 months ago.

9.19 3. Year-on-year change in population by gender that never used internet.

[967]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(1,3, figsize = (19,7), sharey = True)

#FIRST PLOT
in_out.unstack('sex').diff().fillna(0).plot(ax = axes[0], marker = 's', color =␣
↪colors)
axes[0].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[0].set_title('Year-on-year change in population \n by gender that have␣
↪used \n internet in last 3 months',
            fontsize = title_size)
axes[0].set_ylabel('population', fontsize = label_size)
axes[0].set_xlabel('year', fontsize = label_size)
axes[0].yaxis.grid(True, color ="grey", alpha=0.2)

#SECOND PLOT
in_out2.unstack('sex').diff().fillna(0).plot(ax = axes[1], marker = 's', color␣
↪= colors)
axes[1].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[1].set_title('Year-on-year change in population \n by gender that have␣
↪used \n internet over 3 months ago',
            fontsize = title_size)
axes[1].set_ylabel('population', fontsize = label_size)
axes[1].set_xlabel('year', fontsize = label_size)
axes[1].yaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
in_out3.unstack('sex').diff().fillna(0).plot(ax = axes[2], marker = 's', color␣
↪= colors)
axes[2].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[2].set_title('Year-on-year change in population \n by gender that have␣
↪never used internet',
            fontsize = title_size)
axes[2].set_ylabel('population', fontsize = label_size)
axes[2].set_xlabel('year', fontsize = label_size)
axes[2].yaxis.grid(True, color ="grey", alpha=0.2)

plt.show()
```

```
#SHOW TABULAR DATA
in_out.unstack('sex').diff().head(15)
```



[967]:

sex	Men	Women
year		
2013	NaN	NaN
2014	572.0	642.0
2015	415.0	799.0
2016	583.0	663.0
2017	388.0	437.0
2018	406.0	412.0
2019	269.0	301.0
2020	408.0	503.0

9.20 Conclusion from the above plots.

Taking it all together, we see that the changes over the years for both males and females were quite close to each other.

In the first plot, we can see that more superior changes have happened for *'Women'* and they kept being sharper when it comes to regular usage.

Accordingly, the internet usage *'over 3 months ago'* for female and male has barely changed each year and has nearly been the same close to the 0.

In the end for 3rd plot, both male and female change has declined to negative numbers. Perhaps, it shows that both genders from 2014 up to 2018 were against any internet usage, but in the next years, they had to change their view and those numbers kept on falling.

10 Disability

Data loading and cleaning

[968]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
disabilityDF = openFile('./internetusers2020.xlsx',
                                sheet_name = '3a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A:B, D:J',
                                squeeze = False,
                                dtype = default,
                                engine = default,
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = [0,1,2,3,4,5,6, 10, 13, 16, 19, 22,␣
 ↪25, 28],
                                nrows = 20)

#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 1", "Unnamed: 3", "Unnamed: 4", "Unnamed: 5",␣
 ↪"Unnamed: 6",
         "Unnamed: 7", "Unnamed: 8", "Unnamed: 9"]
list2 = ["age", "disability type", "2014", "2015", "2016", "2017", "2018",␣
 ↪"2019", "2020"]
rename_column_name(disabilityDF, list1, list2)

disabilityDF.iat[1, 0] =  '16-24'
disabilityDF.iat[3, 0] =  '25-34'
disabilityDF.iat[5, 0] =  '35-44'
disabilityDF.iat[7, 0] =  '45-54'
disabilityDF.iat[9, 0] =  '55-64'
disabilityDF.iat[11, 0] =  '65-74'
disabilityDF.iat[13, 0] =  '75+'

#SET DTYPES
disabilityDF['disability type'] = disabilityDF['disability type'].str.lower().
 ↪astype('category')
disabilityDF['age'] = disabilityDF['age'].astype('category')
```

[969]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
disabilityDF_2 = openFile('./internetusers2020.xlsx',
                                sheet_name = '3a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A:B, L:R',
                                squeeze = False,
                                dtype = default,
```

```
                                engine = default,
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = [0,1,2,3,4,5,6, 10, 13, 16, 19, 22,␣
 ↪25, 28],
                                nrows = 20)

#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 1", "Unnamed: 11", "Unnamed: 12", "Unnamed:␣
 ↪13", "Unnamed: 14",
         "Unnamed: 15", "Unnamed: 16", "Unnamed: 17"]
rename_column_name(disabilityDF_2, list1, list2)

disabilityDF_2.iat[1, 0] =  '16-24'
disabilityDF_2.iat[3, 0] =  '25-34'
disabilityDF_2.iat[5, 0] =  '35-44'
disabilityDF_2.iat[7, 0] =  '45-54'
disabilityDF_2.iat[9, 0] =  '55-64'
disabilityDF_2.iat[11, 0] =  '65-74'
disabilityDF_2.iat[13, 0] =  '75+'

#SET DTYPES
disabilityDF_2['disability type'] = disabilityDF_2['disability type'].str.
 ↪lower().astype('category')
disabilityDF_2['age'] = disabilityDF_2['age'].astype('category')
```

[970]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
disabilityDF_3 = openFile('./internetusers2020.xlsx',
                                sheet_name = '3a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A:B, T:Z',
                                squeeze = False,
                                dtype = default,
                                engine = default,
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = [0,1,2,3,4,5,6, 10, 13, 16, 19, 22,␣
 ↪25, 28],
                                nrows = 20)

#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 1", "Unnamed: 19", "Unnamed: 20", "Unnamed:␣
 ↪21", "Unnamed: 22",
```

```
        "Unnamed: 23", "Unnamed: 24", "Unnamed: 25"]
rename_column_name(disabilityDF_3, list1, list2)

disabilityDF_3.iat[1, 0] =  '16-24'
disabilityDF_3.iat[3, 0] =  '25-34'
disabilityDF_3.iat[5, 0] =  '35-44'
disabilityDF_3.iat[7, 0] =  '45-54'
disabilityDF_3.iat[9, 0] =  '55-64'
disabilityDF_3.iat[11, 0] =  '65-74'
disabilityDF_3.iat[13, 0] =  '75+'

#SET DTYPES
disabilityDF_3['disability type'] = disabilityDF_3['disability type'].str.
 ↪lower().astype('category')
disabilityDF_3['age'] = disabilityDF_3['age'].astype('category')
```

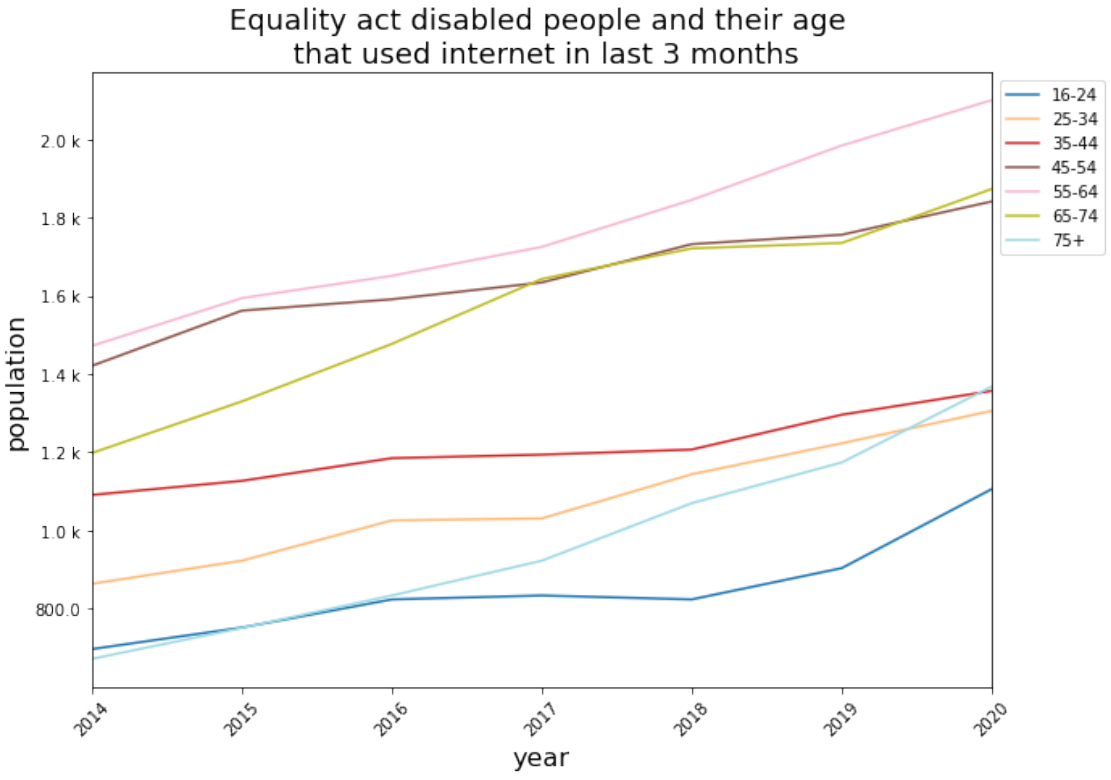
[971]:
```
#MELT COLUMNS INTO ROWS (EACH AGE UNDER 'AGE' COLUMN AND TYPE OF DISABILITY␣
 ↪UNDER 'TYPE OF DISABILITY')
ts1 = disabilityDF.melt(id_vars = ['age', 'disability type'],
                        var_name = 'year', value_name = 'population')
ts2 = disabilityDF_2.melt(id_vars = ['age', 'disability type'],
                        var_name = 'year', value_name = 'population')
ts3 = disabilityDF_3.melt(id_vars = ['age', 'disability type'],
                        var_name = 'year', value_name = 'population')

#CREATE A DATETIME SERIES BY PARSING THE YEARS
ts1['year'] = pd.to_datetime(ts1['year'], format = '%Y')
ts2['year'] = pd.to_datetime(ts2['year'], format = '%Y')
ts3['year'] = pd.to_datetime(ts3['year'], format = '%Y')

#MAKE YEAR AND AGE AN INDEX COLUMN (DATETIMEINDEX)
ts1.set_index('year', inplace = True)
ts2.set_index('year', inplace = True)
ts3.set_index('year', inplace = True)

#CONVERT THE DATETIMEINDEX TO A PERIODINDEX OF YEAR FREQUENCY
ts1 = ts1.to_period('Y')
ts2 = ts2.to_period('Y')
ts3 = ts3.to_period('Y')

#MAKE AGE AND DISABILITY TYPE AN INDEX COLUMN (DATETIMEINDEX)
ts1.set_index(['age', 'disability type'], inplace = True, append = True)
ts2.set_index(['age', 'disability type'], inplace = True, append = True)
ts3.set_index(['age', 'disability type'], inplace = True, append = True)

#SORT THE INDEX
ts1.sort_index(inplace = True)
```

```
ts2.sort_index(inplace = True)
ts3.sort_index(inplace = True)
```

10.1 Equality act disabled people and their age that used internet in last 3 months.

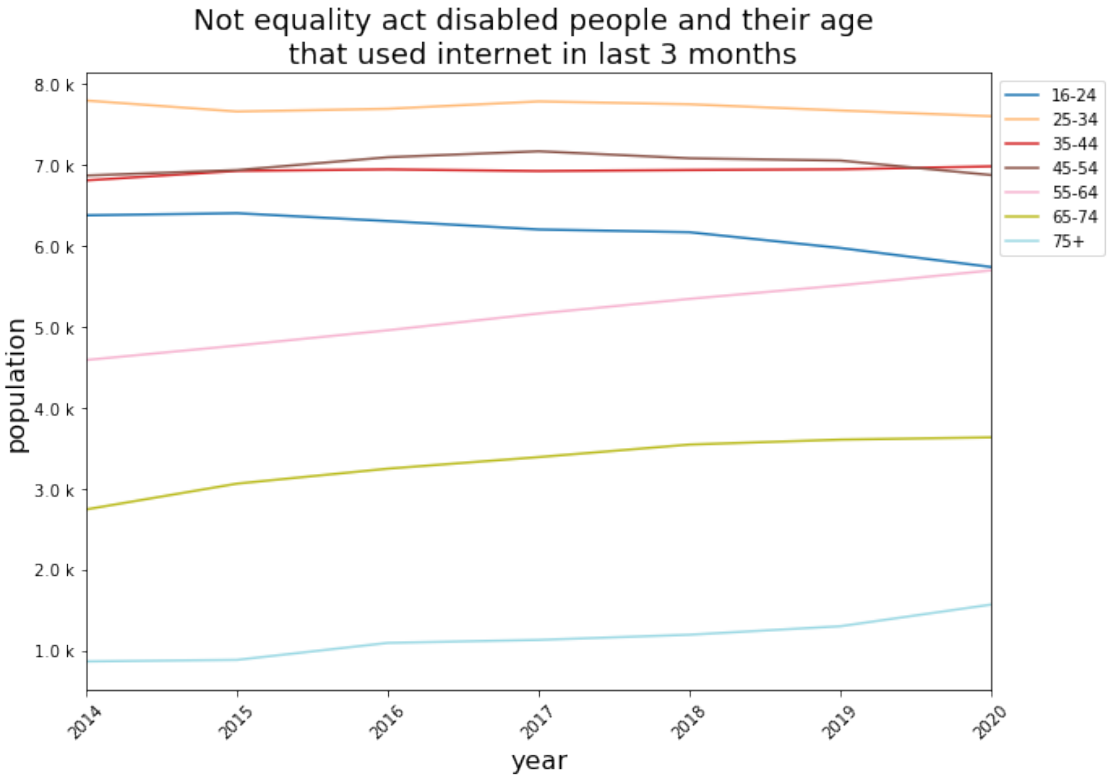
[972]:
```
eqAct = ts1.xs('equality act disabled', level = 'disability type').unstack()
notEqAct = ts1.xs('not equality act disabled', level = 'disability type').
 ↪unstack()
```

[973]:
```
ax = eqAct['population'].plot(rot = 45, colormap = 'tab20',figsize = (10, 7))
ax.yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
ax.legend(bbox_to_anchor=(1.0, 1.0))
ax.set_title('Equality act disabled people and their age \n that used internet␣
 ↪in last 3 months',
             fontsize = title_size)
ax.set_xlabel('year', fontsize = label_size)
ax.set_ylabel('population', fontsize = label_size)
plt.show()
```



10.2 Not equality act disabled people and their age that used internet in last 3 months.

[974]:
```
ax = notEqAct['population'].plot(rot = 45, colormap = 'tab20', figsize = (10,␣
↪7))
ax.yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
ax.legend(bbox_to_anchor=(1.0, 1.0))
ax.set_title('Not equality act disabled people and their age \n that used␣
↪internet in last 3 months',
             fontsize = title_size)
ax.set_xlabel('year', fontsize = label_size)
ax.set_ylabel('population', fontsize = label_size)
plt.show()
```



10.3 Conclusion from the above 2 plots.

To make this conclusion understandable I would like to explain what are the disability types for those plots. Equality act disabled - refers to those people who self-assess that they have a disability following the Equality Act definition of disability. Not equality act disabled - means all of the people who have not declared they have a disability.

On the first plot, we can observe that the equality act disabled population climbed upwards for each age group presented in the plot. As we would expect the lines of the older age groups tend

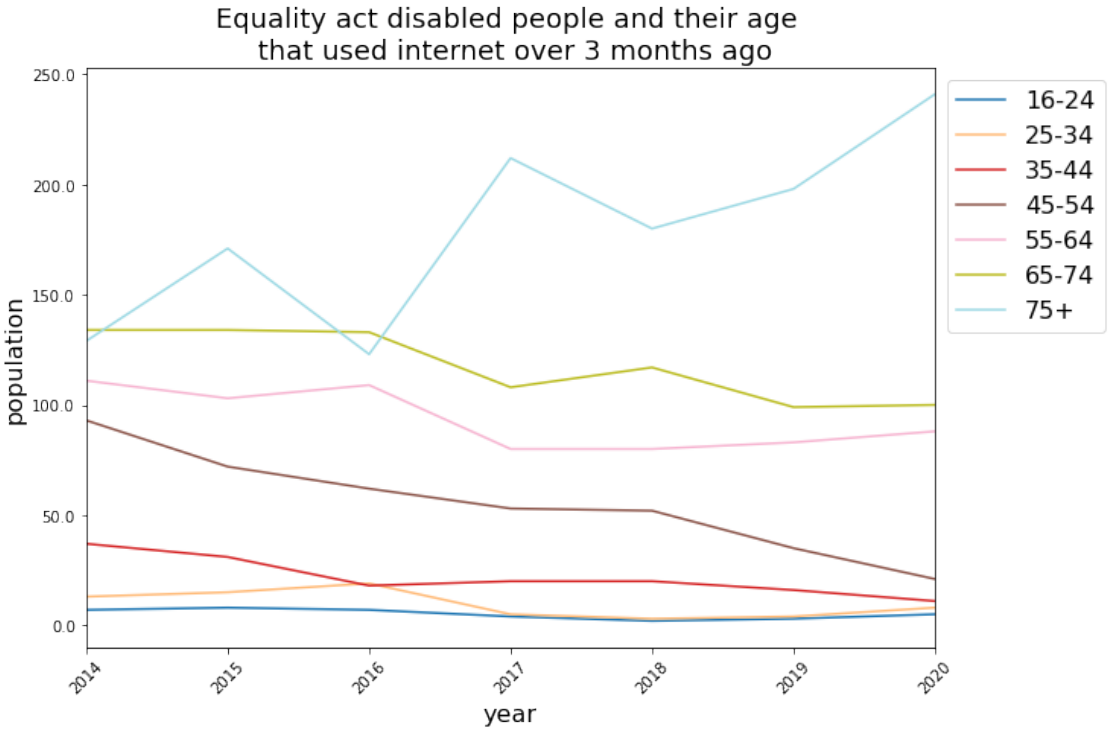
to have a higher population which tells us that more older people say they have a disability than young groups. Younger groups may either have no disabilities or they did not confirm it.

On the second plot, we see that more people, especially the younger have confirmed they have no disability. Even more important is the fact that for each age group in this plot the population line is flat and does not seem to harshly go into ascending or descending direction. For this reason, I think that the number of respondents has not changed much and was not collected from many different people because those results are a very calm for that many years.

10.4 Equality act disabled people and their age that used internet over 3 months ago.

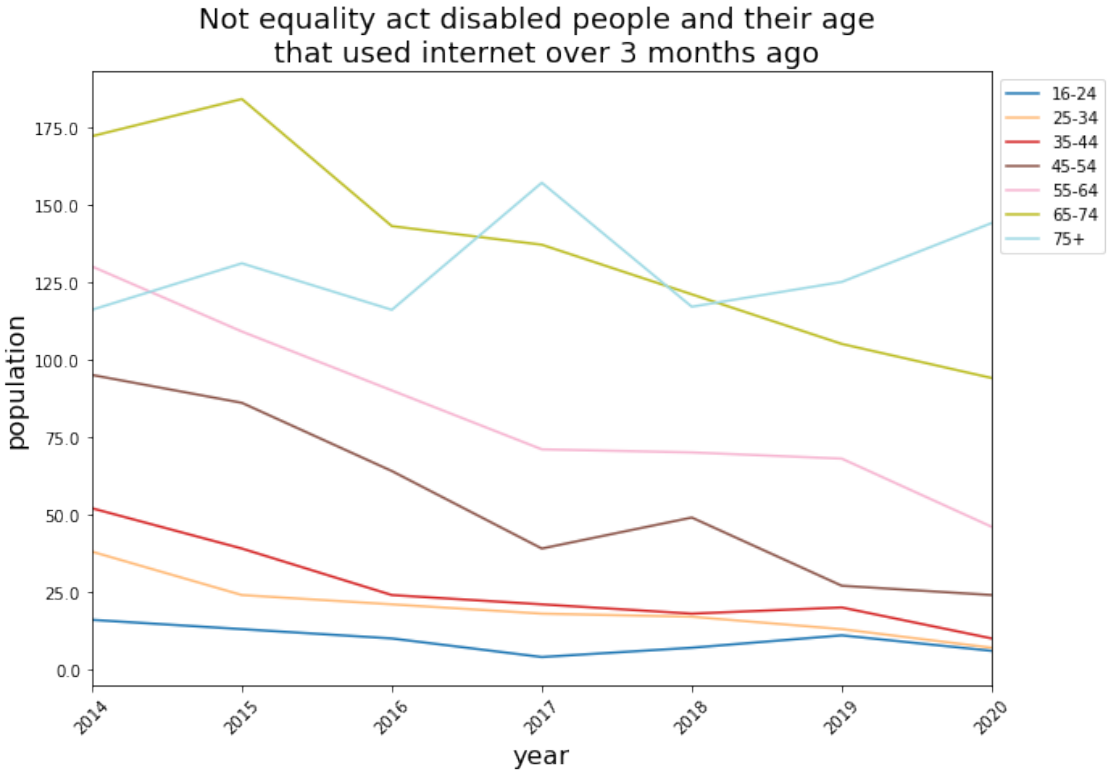
[975]:
```
eqAct_2 = ts2.xs('equality act disabled', level = 'disability type').unstack()
notEqAct_2 = ts2.xs('not equality act disabled', level = 'disability type').
↪unstack()
```

[976]:
```
ax = eqAct_2['population'].plot(rot = 45, colormap = 'tab20',figsize = (10, 7) )
ax.yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
ax.legend(bbox_to_anchor=(1.0, 1.0), prop={'size': 16})
ax.set_title('Equality act disabled people and their age \n that used internet␣
↪over 3 months ago',
             fontsize = title_size)
ax.set_xlabel('year', fontsize = label_size)
ax.set_ylabel('population', fontsize = label_size)
plt.show()
```



10.5 Not equality act disabled people and their age that used internet over 3 months ago.

[977]:
```
ax = notEqAct_2['population'].plot(rot = 45, colormap = 'tab20', figsize = (10,␣
↪7))
ax.yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
ax.legend(bbox_to_anchor=(1.0, 1.0))
ax.set_title('Not equality act disabled people and their age \n that used␣
↪internet over 3 months ago',
             fontsize = title_size)
ax.set_xlabel('year', fontsize = label_size)
ax.set_ylabel('population', fontsize = label_size)
plt.show()
```



10.6 Conclusion from the above 2 plots.

For the majority of age groups, the population decreases when it comes to using the internet in more than 3 months and not using it at all.

On the first plot, we can see that each age that is below the *'75+'* group has declined in population by a small number and the oldest group climbs towards the 250 mark. Nothing is intriguing about

it as older people tend to get more health-related issues, so when the survey, happens there might be more of them each year that use the internet often and are at the same time deprived of the ability to do certain things.

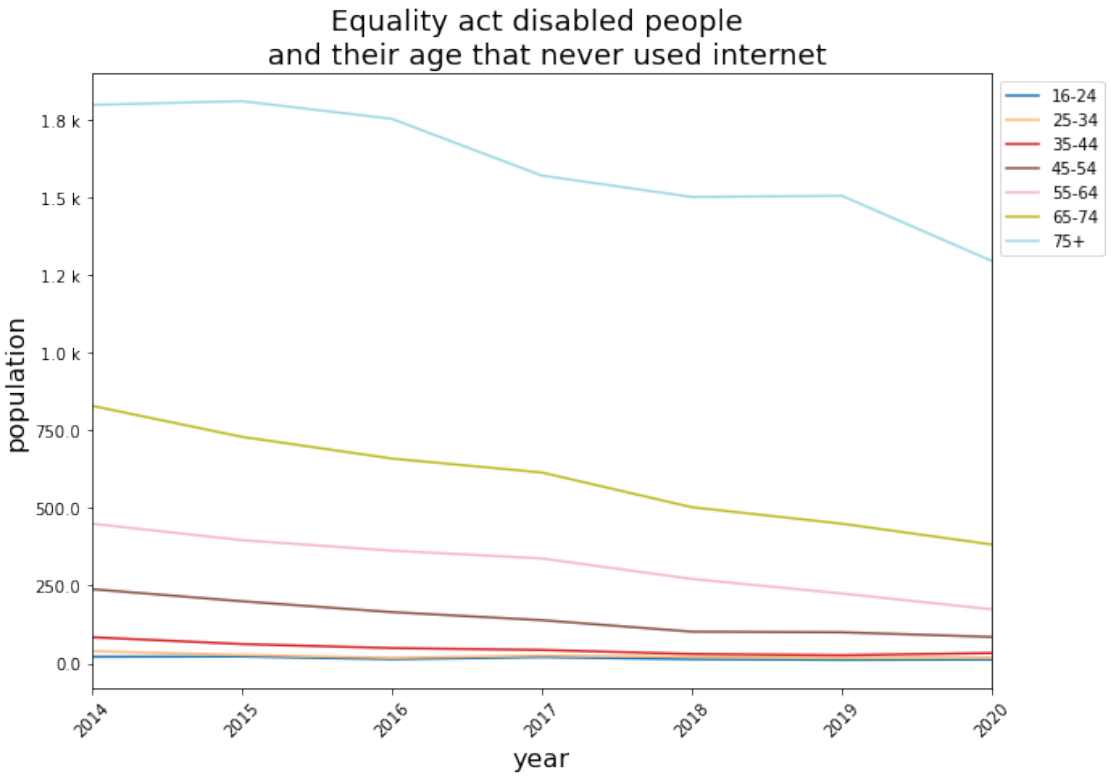
On the second plot, we observe that most population decreases each year and this time, the oldest group seems to follow the trend more with a small increase from 2018 to 2020. As a result, we get more older people that either has no disability or they do not want to share it.

Also, it is important to mention that in the second plot, we can see that younger age groups are deteriorating when it comes to stating that they have no disability. Based on this finding, I believe that younger people do not want to say they have no disability, so people will look at them differently and maybe they will have some advantage. It is hard to specify the reason why a young person does this.

10.7 Equality act disabled people and their age that never used internet.

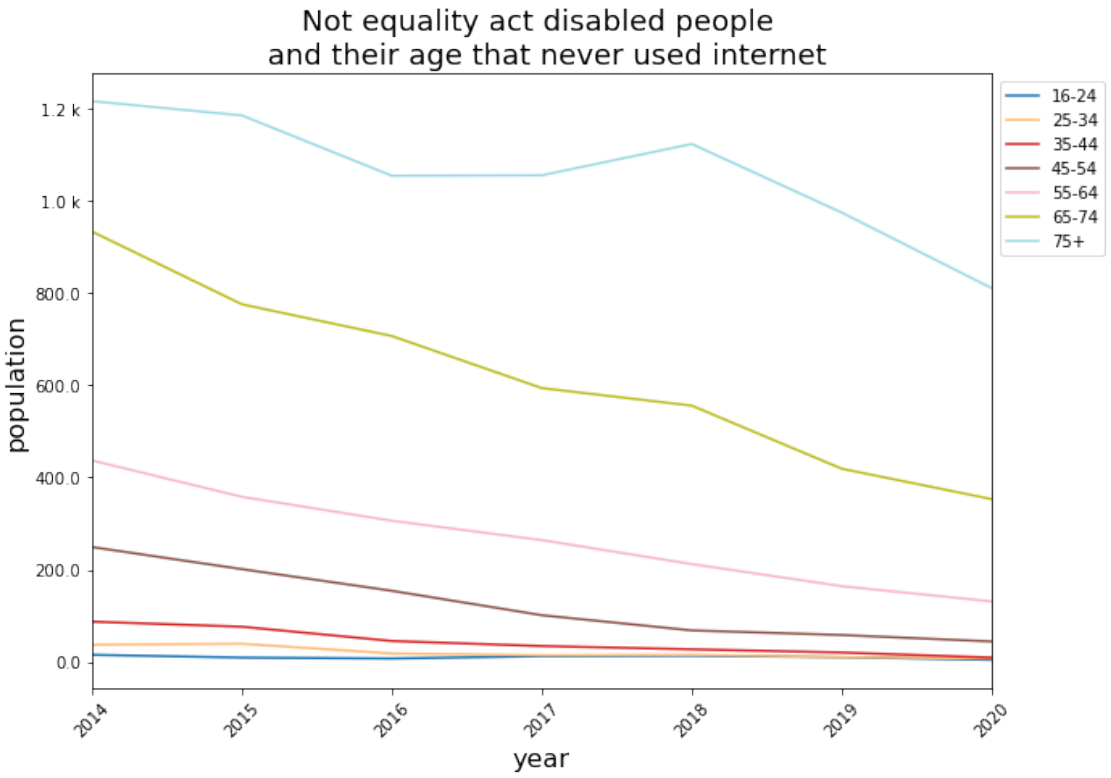
[978]:
```
eqAct_3 = ts3.xs('equality act disabled', level = 'disability type').unstack()
notEqAct_3 = ts3.xs('not equality act disabled', level = 'disability type').
↪unstack()
```

[979]:
```
ax = eqAct_3['population'].plot(rot = 45, colormap = 'tab20',figsize = (10, 7) )
ax.yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
ax.legend(bbox_to_anchor=(1.0, 1.0))
ax.set_title('Equality act disabled people \n and their age that never used␣
↪internet',
             fontsize = title_size)
ax.set_xlabel('year', fontsize = label_size)
ax.set_ylabel('population', fontsize = label_size)
plt.show()
```



10.8 Not equality act disabled people and their age that never used internet.

[980]:
```
ax = notEqAct_3['population'].plot(rot = 45, colormap = 'tab20', figsize = (10,␣
↪7), )
ax.yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
ax.legend(bbox_to_anchor=(1.0, 1.0))
ax.set_title('Not equality act disabled people \n and their age that never used␣
↪internet',
             fontsize = title_size)
ax.set_xlabel('year', fontsize = label_size)
ax.set_ylabel('population', fontsize = label_size)
plt.show()
```



10.9 Conclusion from the above 2 plots.

Both of the plots for equality act disabled and not equality act disabled present that there had to be a change in people attitude when it comes to the internet.

In those modern days, we are presented with doing everything online, so both young and older people have to use the internet to finish off the important things in their lives. For younger age groups, it might be something simple as homework, whereas for older sending some important information to the government or paying for any kind of bill they have or accessing a bank.

Due to all of those reasons the above plots, it is clear that none of the age groups has a possibility not to use the internet for their advantage even when it comes to using it once over three months. The population of each age group with a disability or no disability leans towards zero, and if we would keep a record of it for the next years, I would expect that it to be close to that mark.

11 Ethnicity

11.1 Warning!

In the below section I have decided to drop the 'white' ethnic group as it work as on outlier. Due to this sitaution none of the graphs would present relevant and useful data about other ethnic groups. I have done it purely for better explanation of findings.

For pure inforamtion:

I want to stress that the 'white' is in possesion of largest population, which would compare to others as around 95% to 5%.

Data loading and cleaning

[981]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
ethnicityDF = openFile('./internetusers2020.xlsx',
                                sheet_name = '4a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A, C:J',
                                squeeze = False,
                                dtype = default,
                                engine = default,
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = 4,
                                nrows = 8)

#DROP THE VASTLY HIGH ETHIC GROUP FOR PURPOSE OF BETTER PRESENTATION OF MINOR␣
 ↪GROUPS
ethnicityDF.drop( index = 0, inplace=True)

#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 2", "Unnamed: 3", "Unnamed: 4", "Unnamed: 5",␣
 ↪"Unnamed: 6",
         "Unnamed: 7", "Unnamed: 8", "Unnamed: 9"]
list2 = ["ethnicity", "2013", "2014", "2015", "2016", "2017", "2018", "2019",␣
 ↪"2020"]
rename_column_name(ethnicityDF, list1, list2)

#SET DTYPES
ethnicityDF['ethnicity'] = ethnicityDF['ethnicity'].str.lower().
 ↪astype('category')
```

[982]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
ethnicityDF_2 = openFile('./internetusers2020.xlsx',
                                sheet_name = '4a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A, L:S',
                                squeeze = False,
                                dtype = default,
                                engine = default,
                                converters = default,
```

```
                                true_values = default,
                                false_values = default,
                                skiprows = 4,
                                nrows = 9)

#DROP THE VASTLY HIGH ETHIC GROUP FOR PURPOSE OF BETTER PRESENTATION OF MINOR␣
 ↪GROUPS
ethnicityDF_2.drop( index = 0, inplace=True)

#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 11", "Unnamed: 12", "Unnamed: 13", "Unnamed:␣
 ↪14", "Unnamed: 15",
         "Unnamed: 16", "Unnamed: 17", "Unnamed: 18"]
rename_column_name(ethnicityDF_2, list1, list2)

#SET DTYPES
ethnicityDF_2['ethnicity'] = ethnicityDF_2['ethnicity'].str.lower().
 ↪astype('category')
```

[983]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
ethnicityDF_3 = openFile('./internetusers2020.xlsx',
                                sheet_name = '4a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A, U:AB',
                                squeeze = False,
                                dtype = default,
                                engine = default,
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = 4,
                                nrows = 9)
#DROP THE VASTLY HIGH ETHIC GROUP FOR PURPOSE OF BETTER PRESENTATION OF MINOR␣
 ↪GROUPS
ethnicityDF_3.drop( index = 0, inplace=True)

#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 20", "Unnamed: 21", "Unnamed: 22", "Unnamed:␣
 ↪23", "Unnamed: 24",
         "Unnamed: 25", "Unnamed: 26", "Unnamed: 27"]
rename_column_name(ethnicityDF_3, list1, list2)

#SET DTYPES
ethnicityDF_3['ethnicity'] = ethnicityDF_3['ethnicity'].str.lower().
 ↪astype('category')
```

[984]:
```
#MELT COLUMNS INTO ROWS (EACH ETHNICITY UNDER 'ETHNICITY' COLUMN)
tsEth1 = ethnicityDF.melt(id_vars = ['ethnicity'],
                          var_name = 'year', value_name = 'population')
tsEth2 = ethnicityDF_2.melt(id_vars = ['ethnicity'],
                          var_name = 'year', value_name = 'population')
tsEth3 = ethnicityDF_3.melt(id_vars = ['ethnicity'],
                          var_name = 'year', value_name = 'population')

#CREATE A DATETIME SERIES BY PARSING THE YEARS
tsEth1['year'] = pd.to_datetime(tsEth1['year'], format = '%Y')
tsEth2['year'] = pd.to_datetime(tsEth2['year'], format = '%Y')
tsEth3['year'] = pd.to_datetime(tsEth3['year'], format = '%Y')

#MAKE YEAR AND AGE AN INDEX COLUMN (DATETIMEINDEX)
tsEth1.set_index('year', inplace = True)
tsEth2.set_index('year', inplace = True)
tsEth3.set_index('year', inplace = True)

#CONVERT THE DATETIMEINDEX TO A PERIODINDEX OF YEAR FREQUENCY
tsEth1 = tsEth1.to_period('Y')
tsEth2 = tsEth2.to_period('Y')
tsEth3 = tsEth3.to_period('Y')

#MAKE ETHNICITY TYPE AN INDEX COLUMN (DATETIMEINDEX)
tsEth1.set_index(['ethnicity'], inplace = True, append = True)
tsEth2.set_index(['ethnicity'], inplace = True, append = True)
tsEth3.set_index(['ethnicity'], inplace = True, append = True)

#SORT THE INDEX
tsEth1.sort_index(inplace = True)
tsEth2.sort_index(inplace = True)
tsEth3.sort_index(inplace = True)
```

11.2 Ethnic group:

11.3 1. People that used internet in last 3 months.

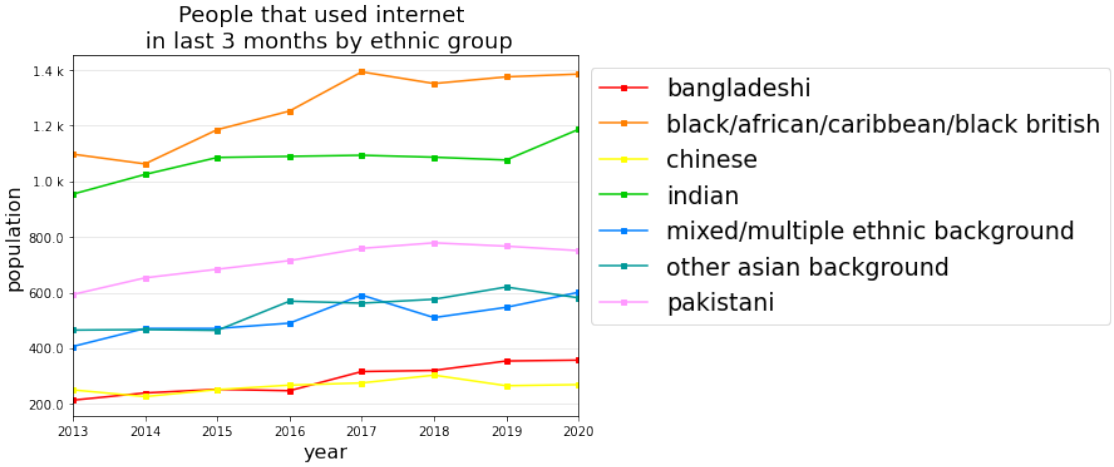
11.4 2. People that used internet over 3 months ago.

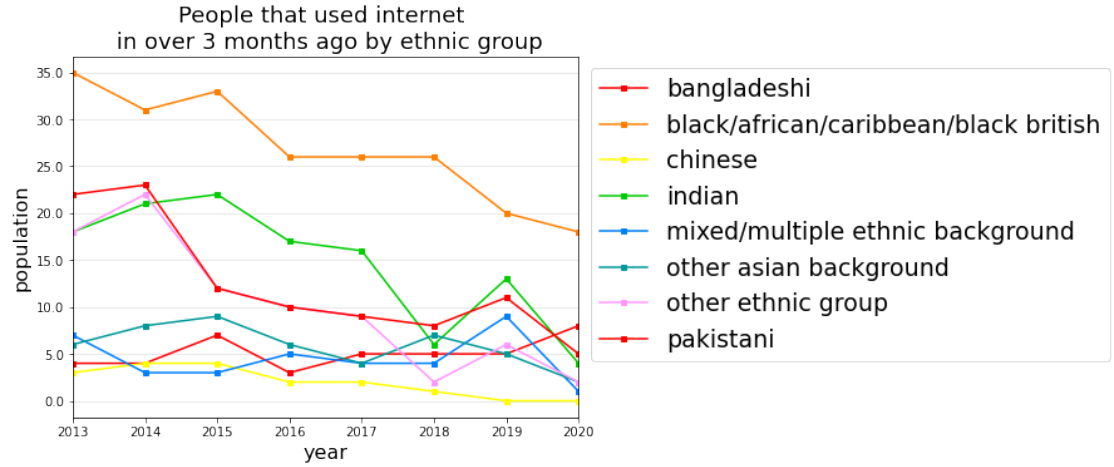
11.5 3. People that never used internet.

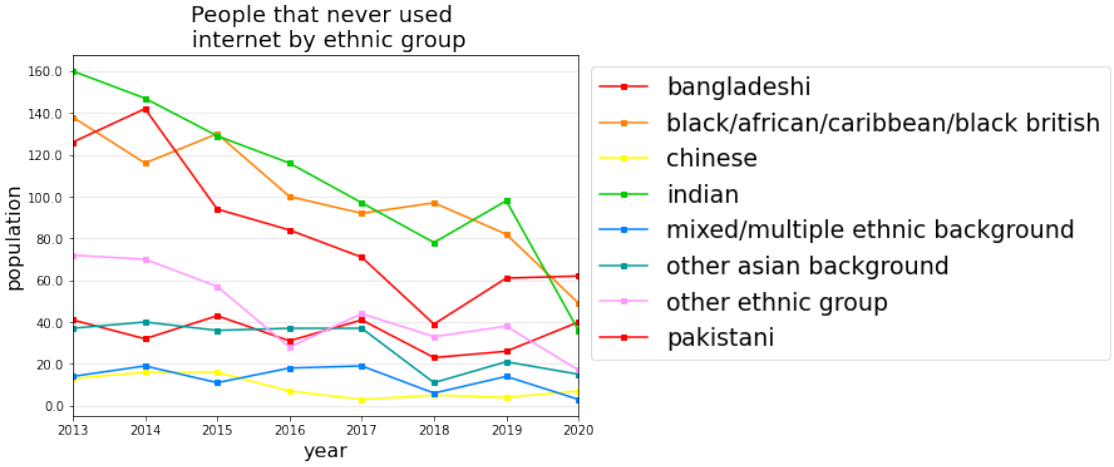
[985]:
```
in_outEth = tsEth1['population'].groupby(level=['year', 'ethnicity']).sum()
in_outEth2 = tsEth2['population'].groupby(level=['year', 'ethnicity']).sum()
in_outEth3 = tsEth3['population'].groupby(level=['year', 'ethnicity']).sum()
```

[986]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 1, figsize = (7, 19))
```

```
#FIRST PLOT
in_outEth.unstack('ethnicity').plot(ax = axes[0], marker = 's', ms = mSizeEth,␣
 ↪color = colorsEth)
axes[0].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[0].legend(bbox_to_anchor=(1.0, 1.0), prop={'size': valOfSize})
axes[0].set_title('People that used internet \n in last 3 months by ethnic␣
 ↪group',
                  fontsize = title_size)
axes[0].set_ylabel('population', fontsize = label_size)
axes[0].set_xlabel('year', fontsize= label_size)
axes[0].yaxis.grid(True, color ="grey", alpha=0.2)

#FIRST PLOT
in_outEth2.unstack('ethnicity').plot(ax = axes[1], marker = 's', ms = mSizeEth,␣
 ↪color = colorsEth)
axes[1].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[1].legend(bbox_to_anchor=(1.0, 1.0), prop={'size': valOfSize})
axes[1].set_title('People that used internet \n in over 3 months ago by ethnic␣
 ↪group',
                  fontsize = title_size)
axes[1].set_ylabel('population', fontsize = label_size)
axes[1].set_xlabel('year', fontsize= label_size)
axes[1].yaxis.grid(True, color ="grey", alpha=0.2)

#FIRST PLOT
in_outEth3.unstack('ethnicity').plot(ax = axes[2], marker = 's', ms = mSizeEth,␣
 ↪color = colorsEth)
axes[2].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[2].legend(bbox_to_anchor=(1.0, 1.0), prop={'size': valOfSize})
axes[2].set_title('People that never used \n internet by ethnic group',␣
 ↪fontsize = title_size)
axes[2].set_ylabel('population', fontsize = label_size)
axes[2].set_xlabel('year', fontsize= label_size)
axes[2].yaxis.grid(True, color ="grey", alpha=0.2)

#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=0.9,
                    wspace=0.1,
                    hspace=0.4)
plt.show()
```


People that used internet
in last 3 months by ethnic group
population
year
bangladeshi
black/african/caribbean/black british
chinese
indian
mixed/multiple ethnic background
other asian background
pakistani


People that used internet
in over 3 months ago by ethnic group
population
year
bangladeshi
black/african/caribbean/black british
chinese
indian
mixed/multiple ethnic background
other asian background
other ethnic group
pakistani


People that never used
internet by ethnic group
population
year
bangladeshi
black/african/caribbean/black british
chinese
indian
mixed/multiple ethnic background
other asian background
other ethnic group
pakistani

11.6 Conclusion from the above plots.

On the first plot, we can see that people that used the internet in *'last 3 months'* are mostly *'african'* and *'indian'* and they possess the largest population in this case. The smallest population is *'chinese'* and *'bangladeshi'*. Undoubtedly the first plot presents that each ethnic groups population climbs when it comes to using the internet but it is a much slower rise on the lower lines. Those lines are telling me that the adaptation of using the internet happens a bit slower for people that have east roots.

Moreover, each group decrease in population when it comes to rare use and the population on the second plot is very small that already presents that there will be fewer and fewer people with such habits. The same situation happens for non-users with different backgrounds where a population of *'indian'* goes from high numbers such as 160 goes to 39 over the years.

11.7 Annually:

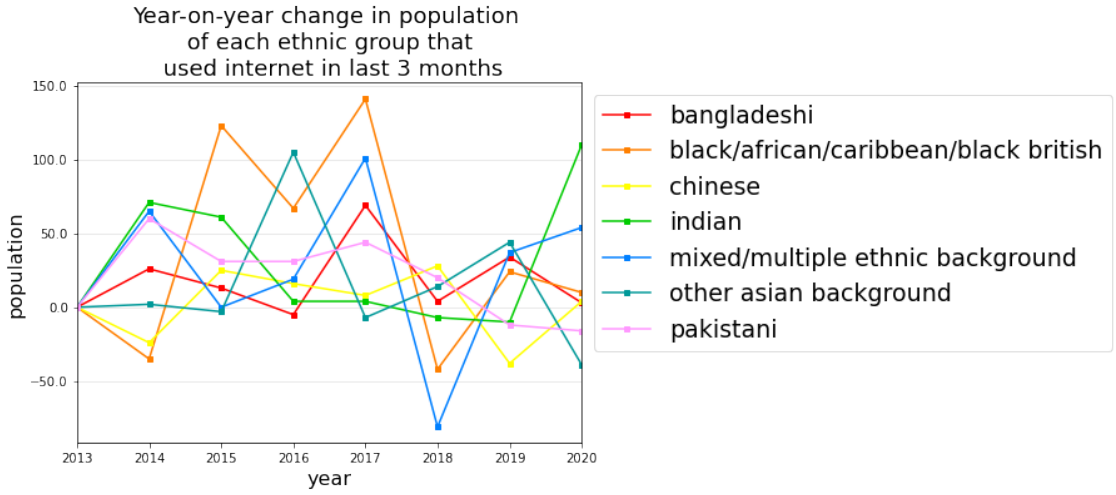
11.8 1. Year-on-year change in population of each ethnic group that used internet in last 3 months.

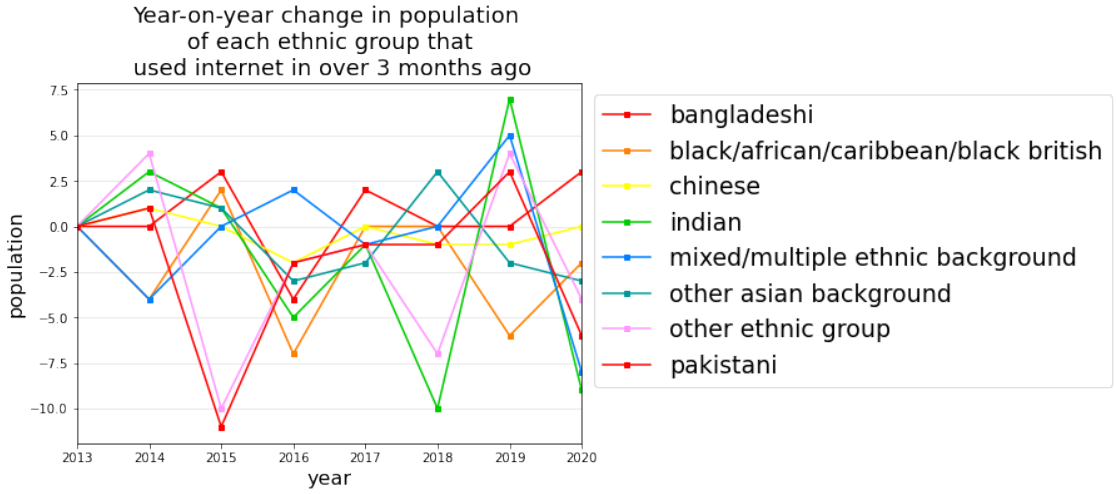
11.9 2. Year-on-year change in population of each ethnic group that used internet in over 3 months ago.

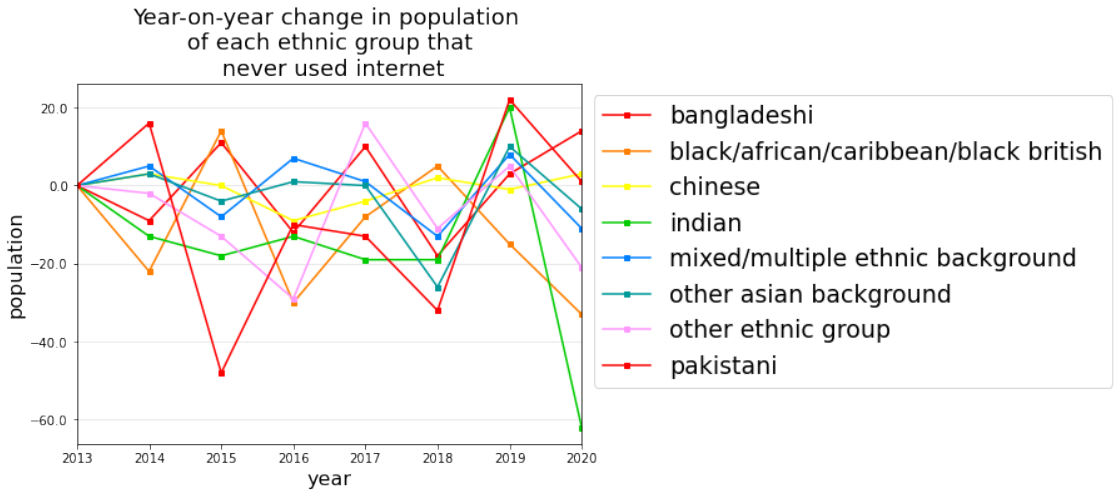
11.10 3. Year-on-year change in population of each ethnic group that never used internet.

[987]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(3, 1, figsize = (7, 19))

#'Year-on-year change in population \n by gender that have used \n internet in␣
 ↪last 3 months'
#FIRST PLOT
in_outEth.unstack('ethnicity').diff().fillna(0).plot(ax = axes[0], marker = 's',
                                                     ms = mSizeEth, color =␣
 ↪colorsEth)
axes[0].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[0].legend(bbox_to_anchor=(1.0, 1.0), prop={'size': valOfSize})
axes[0].set_title('Year-on-year change in population \n of each ethnic group␣
 ↪that \n used internet in last 3 months',
                  fontsize = title_size)
axes[0].set_ylabel('population', fontsize = label_size)
axes[0].set_xlabel('year', fontsize= label_size)
axes[0].yaxis.grid(True, color ="grey", alpha=0.2)

#SECOND PLOT
in_outEth2.unstack('ethnicity').diff().fillna(0).plot(ax = axes[1], marker =␣
 ↪'s',
                                                      ms = mSizeEth, color =␣
 ↪colorsEth)
axes[1].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
```

```
axes[1].legend(bbox_to_anchor=(1.0, 1.0), prop={'size': valOfSize})
axes[1].set_title('Year-on-year change in population \n of each ethnic group␣
 ↪that \n used internet in over 3 months ago',
                  fontsize = title_size)
axes[1].set_ylabel('population', fontsize = label_size)
axes[1].set_xlabel('year', fontsize= label_size)
axes[1].yaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
in_outEth3.unstack('ethnicity').diff().fillna(0).plot(ax = axes[2], marker =␣
 ↪'s',
                                                       ms = mSizeEth, color =␣
 ↪colorsEth)
axes[2].yaxis.set_major_formatter(mpl.ticker.EngFormatter(places=1))
axes[2].legend(bbox_to_anchor=(1.0, 1.0), prop={'size': valOfSize})
axes[2].set_title('Year-on-year change in population \n of each ethnic group␣
 ↪that \n never used internet',
                  fontsize = title_size)
axes[2].set_ylabel('population', fontsize = label_size)
axes[2].set_xlabel('year', fontsize= label_size)
axes[2].yaxis.grid(True, color ="grey", alpha=0.2)

#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=0.9,
                    wspace=0.1,
                    hspace=0.4)
plt.show()
```


Year-on-year change in population
of each ethnic group that
used internet in last 3 months
population
year
bangladeshi
black/african/caribbean/black british
chinese
indian
mixed/multiple ethnic background
other asian background
pakistani


Year-on-year change in population
of each ethnic group that
used internet in over 3 months ago
population
year
bangladeshi
black/african/caribbean/black british
chinese
indian
mixed/multiple ethnic background
other asian background
other ethnic group
pakistani


Year-on-year change in population
of each ethnic group that
never used internet
population
year
bangladeshi
black/african/caribbean/black british
chinese
indian
mixed/multiple ethnic background
other asian background
other ethnic group
pakistani

11.11 Conclusion from the above plots.

Due to similar ranges of respondents of each ethnic group the year-on-year change in population seems drastic for each group.

One the first plot, we see that at least five ethnic groups and their changes got smoother from 2019. They year-on-year change has got lower as there was dozens of respondents from the mentioned year.

On the first plot, we can observe that each year the number of respondents for at least five ethnic groups got more smooth, which emphasise that the change from 2019 to 2020 were only by dozens of people with different background. There was a higher year-on-year change from 2013 to 2017, even by hundreds. It tells us that for some ethnic groups, there were more and more people that used the internet in last 3 motnhs and took part in the survey at those early years of study.

Secondly, another plot for rare usage shows very little variation year-on-year, whereas the last plot implies the worsening population amount for any ethnic group that never used internet.

12 Economic activity

Data loading and cleaning

[988]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
economicDF = openFile('./internetusers2020.xlsx',
                              sheet_name = '7a',
                              header = 0,
                              names = default,
                              index_col = default,
                              usecols = 'A, C:L',
                              squeeze = False,
                              dtype = default,
                              engine = default,
                              converters = default,
                              true_values = default,
                              false_values = default,
                              skiprows = 4,
                              nrows = 8)

#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 2", "Unnamed: 3", "Unnamed: 4", "Unnamed: 5",␣
 ↪"Unnamed: 6",
         "Unnamed: 7", "Unnamed: 8", "Unnamed: 9", "Unnamed: 10", "Unnamed: 11"]
list2 = ["economic status", "2011", "2012", "2013", "2014", "2015",
         "2016", "2017", "2018", "2019", "2020"]
rename_column_name(economicDF, list1, list2)

#SET DTYPES
economicDF['economic status'] = economicDF['economic status'].str.lower().
 ↪astype('category')
```

[989]:
```
#READING CORRECT NON-EMPTY COLUMNS & ROWS
economicDF_2 = openFile('./internetusers2020.xlsx',
                                sheet_name = '7a',
                                header = 0,
                                names = default,
                                index_col = default,
                                usecols = 'A, N:W',
                                squeeze = False,
                                dtype = default,
                                engine = default,
                                converters = default,
                                true_values = default,
                                false_values = default,
                                skiprows = 4,
                                nrows = 8)



#RENAME EMPTY HEADERS
list1 = ["Unnamed: 0", "Unnamed: 13", "Unnamed: 14", "Unnamed: 15", "Unnamed:␣
 ↪16", "Unnamed: 17",
         "Unnamed: 18", "Unnamed: 19", "Unnamed: 20", "Unnamed: 21", "Unnamed:␣
 ↪22"]
rename_column_name(economicDF_2 , list1, list2)

#REMOVING '~' CHARACTER FROM 0
economicDF_2['2020'] = economicDF_2['2020'].replace(['~0'], 0)

#SET DTYPES
economicDF_2 ['economic status'] = economicDF_2 ['economic status'].str.lower().
 ↪astype('category')
```

[990]:
```
#MELT COLUMNS INTO ROWS (EACH ETHNICITY UNDER 'ETHNICITY' COLUMN)
tsEc1 = economicDF.melt(id_vars = ['economic status'],
                        var_name = 'year', value_name = 'population')
tsEc2 = economicDF_2.melt(id_vars = ['economic status'],
                        var_name = 'year', value_name = 'population')

#CREATE A DATETIME SERIES BY PARSING THE YEARS
tsEc1['year'] = pd.to_datetime(tsEc1['year'], format = '%Y')
tsEc2['year'] = pd.to_datetime(tsEc2['year'], format = '%Y')

#MAKE YEAR AND AGE AN INDEX COLUMN (DATETIMEINDEX)
tsEc1.set_index('year', inplace = True)
tsEc2.set_index('year', inplace = True)

#CONVERT THE DATETIMEINDEX TO A PERIODINDEX OF YEAR FREQUENCY
```

```
tsEc1 = tsEc1.to_period('Y')
tsEc2 = tsEc2.to_period('Y')

#MAKE ETHNICITY TYPE AN INDEX COLUMN (DATETIMEINDEX)
tsEc1.set_index(['economic status'], inplace = True, append = True)
tsEc2.set_index(['economic status'], inplace = True, append = True)

#SORT THE INDEX
tsEc1.sort_index(inplace = True)
tsEc2.sort_index(inplace = True)
```

12.1 For years 2013, 2016, 2020.

12.2 Row 1:

12.3 1. People that used internet in last 3 months by economic activity.

12.4 Row 2:

12.5 2. People that used internet in over 3 months ago/never by economic activity.

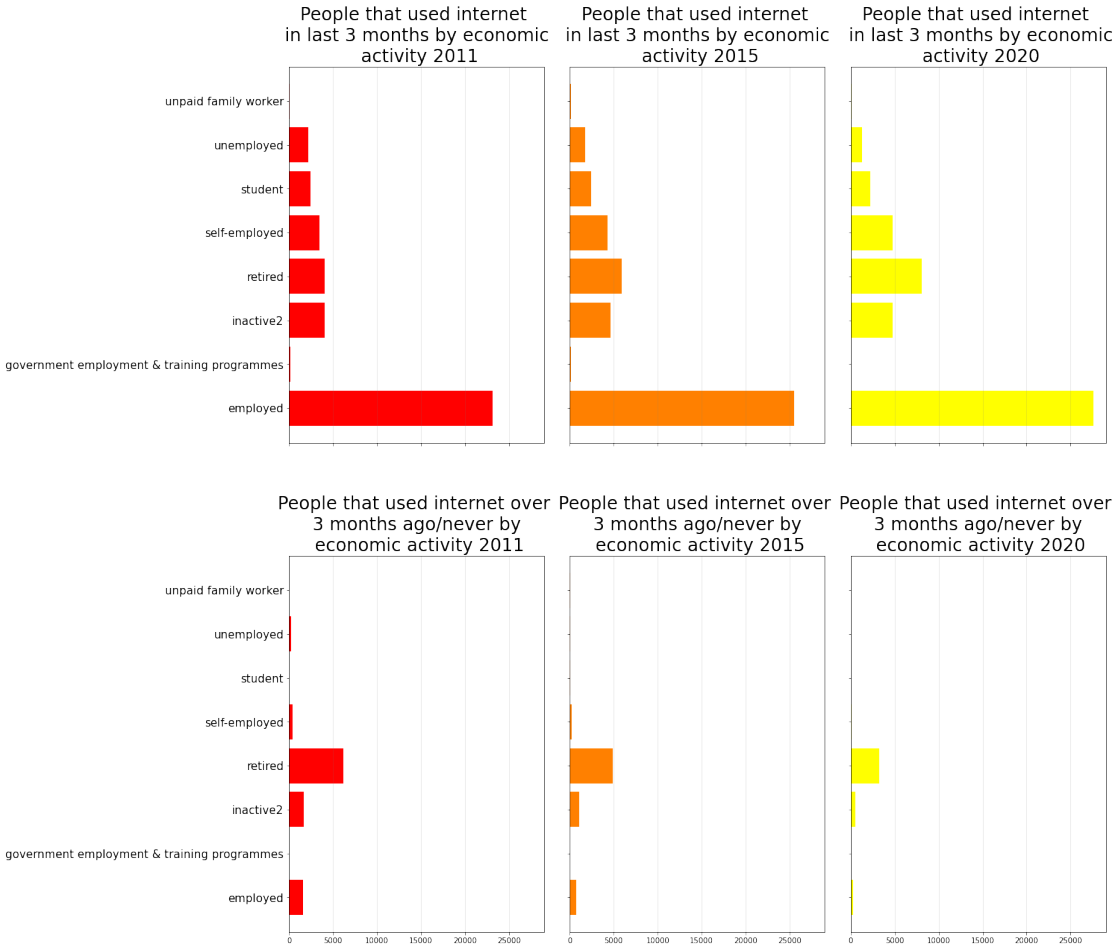
[991]:
```
#ECONOMIC ACTIVITY FOR CHOSEN YEARS FOR FIRST ROW
subsetInitial = tsEc1.loc[('2011'),:]
subsetMid = tsEc1.loc[('2015'),:]
subsetEnd = tsEc1.loc[('2020'),:]

#ECONOMIC ACTIVITY FOR CHOSEN YEARS FOR SECOND ROW
subsetInitial_2 = tsEc2.loc[('2011'),:]
subsetMid_2 = tsEc2.loc[('2015'),:]
subsetEnd_2 = tsEc2.loc[('2020'),:]
```

[992]:
```
#SETTING UP DATA FOR EVERY PLOT FOR FIRST ROW
axInitial = subsetInitial.reset_index()
axMid = subsetMid.reset_index()
axEnd = subsetEnd.reset_index()

#SETTING UP DATA FOR EVERY PLOT FOR SECOND ROW
axInitial_2 = subsetInitial_2.reset_index()
axMid_2 = subsetMid_2.reset_index()
axEnd_2 = subsetEnd_2.reset_index()
```

[1004]:
```
#CREATING SUBPLOTS
fix, axes = plt.subplots(2, 3, figsize = (19,7), sharey = True, sharex = True)

##############################################################################
#FIRST ROW
#FIRST PLOT
```

```
axes[0, 0].barh(axInitial['economic status'], axInitial['population'], color =␣
 ↪colorsEth[0])
axes[0, 0].set_title('People that used internet \n in last 3 months by economic␣
 ↪\n activity 2011', fontsize = economic_title)
axes[0, 0].xaxis.grid(True, color ="grey", alpha=0.2)
axes[0 ,0].set_yticklabels(axInitial['economic status'], fontsize=15)

#SECOND PLOT
axes[0, 1].barh(axMid ['economic status'], axMid['population'], color =␣
 ↪colorsEth[1])
axes[0, 1].set_title('People that used internet \n in last 3 months by economic␣
 ↪\n activity 2015', fontsize = economic_title)
axes[0, 1].xaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
axes[0, 2].barh(axEnd['economic status'], axEnd['population'], color =␣
 ↪colorsEth[2])
axes[0, 2].set_title('People that used internet \n  in last 3 months by␣
 ↪economic \n activity 2020', fontsize = economic_title)
axes[0, 2].xaxis.grid(True, color ="grey", alpha=0.2)

#############################################################################
#SECOND ROW
#FIRST PLOT
axes[1, 0].barh(axInitial_2['economic status'], axInitial_2['population'],␣
 ↪color = colorsEth[0])
axes[1, 0].set_title('People that used internet over \n 3 months ago/never by␣
 ↪\n economic activity 2011', fontsize = economic_title)
axes[1, 0].xaxis.grid(True, color ="grey", alpha=0.2)
axes[1, 0].set_yticklabels(axInitial['economic status'], fontsize=15)

#SECOND PLOT
axes[1, 1].barh(axMid_2['economic status'], axMid_2['population'], color =␣
 ↪colorsEth[1])
axes[1, 1].set_title('People that used internet over \n 3 months ago/never by␣
 ↪\n economic activity 2015', fontsize = economic_title)
axes[1, 1].xaxis.grid(True, color ="grey", alpha=0.2)

#THIRD PLOT
axes[1, 2].barh(axEnd_2['economic status'], axEnd_2['population'], color =␣
 ↪colorsEth[2])
axes[1, 2].set_title('People that used internet over \n 3 months ago/never by␣
 ↪\n economic activity 2020', fontsize = economic_title)
axes[1, 2].xaxis.grid(True, color ="grey", alpha=0.2)

#############################################################################
```

```
#ADJUSTING THE SPACING BETWEEN SUBPLOTS
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=2.4,
                    wspace=0.1,
                    hspace=0.3)
plt.show()

#SHOW TABULAR DATA
axInitial_2.head(15)
```



[1004]:

	year	economic status	population
0	2011	employed	1607
1	2011	government employment & training programmes	12
2	2011	inactive2	1721
3	2011	retired	6152

```
4  2011                          self-employed         390
5  2011                                student          22
6  2011                             unemployed         244
7  2011                   unpaid family worker          16
```

12.6 Conclusion from the above plots.

First of all, I would like to inform you that *'inactive'* person means that a person is looking after family or that they are disabled and therefore cannot be active. It also might be some other reason.

Looking at the first rows, we can see that not much has changed from 2011 up to 2020. However, the biggest change we can see is that *'retired'* people are getting confident or might be encouraged to use the internet more often thus the visible difference across the years. Their population has changed from something like 40 thousand up to 70 thousand what is a huge difference and, I believe it happened because people tend to make the software more accessible and user friendly for elder people. Additionally, the amount of *'self-employed'* has climbed from 2011 up to 2020, which gives me a clue that self-employed people may need internet even more than before in their profession.

The second row makes it even more coherent that no matter what the economic situation of a person is, they tend to use more internet than they would in 2011. I believe so because every single bar has decrease from 2011 to 2015 up to 2020. The only economic group that does it slower are elderly people. It is because they are not likely to change much in what they do on regular basis, and they do not like the changes when it comes to their phones and anything related. Most of them simply think of it as redundant.

13 Justyfing visualisation choices

The choice of my visaulisations were based on the fact that the data I have worked with was strongly related to time so I had to work a lot with time series. I have used as many possible techniques from the videos and the resources to this kind of data. I want to stress that for each factor I have carefully made decision on what would be the good visaulisation and what not. This can be for example seen in some parts where one section has more visualisations than other. I have decided that some techniques do not present any useful results such as flat lines close to zero value so I have avoided them. For additional details, I have added tabulated data that always corresponds to the nearest plot.

The data types assigned to the columns such as 'age' or 'disability' were most of the time a category as they had limited number of possibilities to choose from. The change to categorical type can be seen throught the presentation with many columns.

When it comes to human perception, I have decided to assign *'green'* type of colors when it comes to an *'age'* as green reminds me of leaves that are getting older and thus change color. With this color change, I wanted to stress how the groups of people differ based on their internet usage when it comes to the bar chart. For pie charts, we can see that the younger the age group, the brighter the colour. For *'gender'* I have used more common colors blue and red, which kind of represent both of these genders in today's culture. The *'disability'* plots have random colors that are more different from each other to make it easier to read the plot, same applies to *'ethnic group'* and *'economic activity'*.

14 Main conclusion and evaluation

The key findings from this presentation are: 1. Based on the age people tend to use intrnet oftenly or not use it at all. It means that they are either choose one extreme or the other. For instance, there is very small population of people that would use internet once in couple months. It seems like this was the situation throught the years. 2. Young people are mostly in a group that uses phone regularly, whereas groups like *'35-44'* and above are using it once upon time or never. 3. Men used to use internet more often before the middle of 2014. Women population on the other hand was always larger after 2014. Females population was also higher when it comes to using internet over 3 months ago/never using it. 4. The amount of regular users increases for eqaulity act disabled people and at the same decreases for both eaulity act disabled and not eqaulity disabled when it comes to rare usage or no usage. 5. Based on ethnic group the population of each of them that uses internet at least once in 3 months is increasing over time and decreasing for using interent in over 3 months or never using it. 6. The year-one-year changes for ethnic groups have smoothed from 2018, there is less variation in the most recent years. 7. Retired and self-employed people have changed a lot when it comes to the internet usage from 2011-2020.

The process that I went through was very interesting but I would consider trying to approach this data with more visualisation next time. I would most likely change the visualisation variation for this presentation. In addition, I would for sure try to minimize the repetition of the code that for now I am unable to, except for those helper functions. If i had more time and practice I would put most of the matplotlib relted code into own designed functions that then could be used for entire presentation.

When it comes to visualisation I would probably say that I am satisfied with them. I have tried show message each time we would look at it so things like men and women color would consiously make sense without even reading the labels. I have made sure that the labels, ticks and titles are visible to person reading it and there is no overlapping which was a problem for pie charts. As an addition I also included the square markers so people can easily read the values using them with a help of a grid. If I had an oportunity to change the graphs I would probably work with their apperance to make it more relatable to human perception so it would be very strong each time new person sees it.

I have learnt a lot during this coursework, and those are the key things: 1. The most important part is tidying the data. 2. Quickly getting the wanted data ready in my data frames. 3. Working with data frames quickly and effectively. 4. Using data over time to create visualisations. 5. Working with subplots. 6. Working around inadequacies of the given sheet. 7. Applying some human perception related techniques.

I think that I am quite strict with the work I do and, I always believe there is a place for improvement so, I work with other useful sheets given by this data set that would allow me to present other factors, work on making my visualisations better for a greater audience, trying to get more adequate conclusions.

Arkadiusz Nowacki 17/01/2022 (anowa002)

[4]:
```
import os
!jupyter nbconvert  OwnProject.ipynb --to pdf
```

```
[NbConvertApp] Converting notebook OwnProject.ipynb to pdf
```

```
[NbConvertApp] Support files will be in OwnProject_files\
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Making directory .\OwnProject_files
[NbConvertApp] Writing 288569 bytes to notebook.tex
[NbConvertApp] Building PDF
Traceback (most recent call last):
  File "D:\Programy\Anaconda3\Scripts\jupyter-nbconvert-script.py", line 10, in
<module>
    sys.exit(main())
  File "D:\Programy\Anaconda3\lib\site-packages\jupyter_core\application.py",
line 254, in launch_instance
    return super(JupyterApp, cls).launch_instance(argv=argv, **kwargs)
  File "D:\Programy\Anaconda3\lib\site-
packages\traitlets\config\application.py", line 845, in launch_instance
    app.start()
  File "D:\Programy\Anaconda3\lib\site-packages\nbconvert\nbconvertapp.py", line
350, in start
    self.convert_notebooks()
  File "D:\Programy\Anaconda3\lib\site-packages\nbconvert\nbconvertapp.py", line
524, in convert_notebooks
    self.convert_single_notebook(notebook_filename)
  File "D:\Programy\Anaconda3\lib\site-packages\nbconvert\nbconvertapp.py", line
489, in convert_single_notebook
    output, resources = self.export_single_notebook(notebook_filename,
resources, input_buffer=input_buffer)
  File "D:\Programy\Anaconda3\lib\site-packages\nbconvert\nbconvertapp.py", line
418, in export_single_notebook
    output, resources = self.exporter.from_filename(notebook_filename,
resources=resources)
  File "D:\Programy\Anaconda3\lib\site-
packages\nbconvert\exporters\exporter.py", line 181, in from_filename
    return self.from_file(f, resources=resources, **kw)
```

```
  File "D:\Programy\Anaconda3\lib\site-
packages\nbconvert\exporters\exporter.py", line 199, in from_file
    return self.from_notebook_node(nbformat.read(file_stream, as_version=4),
resources=resources, **kw)
  File "D:\Programy\Anaconda3\lib\site-packages\nbconvert\exporters\pdf.py",
line 183, in from_notebook_node
    self.run_latex(tex_file)
  File "D:\Programy\Anaconda3\lib\site-packages\nbconvert\exporters\pdf.py",
line 153, in run_latex
    return self.run_command(self.latex_command, filename,
  File "D:\Programy\Anaconda3\lib\site-packages\nbconvert\exporters\pdf.py",
line 110, in run_command
    raise OSError("{formatter} not found on PATH, if you have not installed "
OSError: xelatex not found on PATH, if you have not installed xelatex you may
need to do so. Find further instructions at
https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex.
```

[ ]: