**synthetic\_population\_dev**

This repository contains the following jupyter notebooks:

**SPENSER\_notebook\_1:**

* Merges all persons and households of the study area in two dataframes (df\_persons\_NE and df\_households\_NE),
* Generates a unique primary key for each table (persons (PID\_AreaMSOA) and households (HID\_AreaOA))
* Removes unnecessary persons (those that were not assigned to any households (HIP = (-1)) and those that are assigned to buildings that are not residential (QS420\_CELL = -2) or education (QS420\_CELL = 26).
* Removes unnecessary households (those buildings that are not residential (QS420\_CELL = -2) or education (QS420\_CELL = 26).

**SPENSER\_notebook\_2:**

Calculates the following columns:

* Total people in household: count the number of persons in each household
* Total children in household: count the number of children in each household
* Same ethnic: check if there are people with the same ethnicity
* Adult similar age: check if there are people within +- 10 years
* Children\_dependency: Boolean value that tells if a person has a children dependency
* Marital\_status: classifies each individual in married, couple or single based on their socio-demographic characteristics

**SPENSER\_notebook\_3:**

Calculates the attribute: driving licence (True / False)

**SPENSER\_notebook\_4:**

Calculates the attribute: Economic activity (Employed, Unemployed, Inactive) based on age, gender and OA area. Census 2011 data is projected to 2019 using Regional labour market statistics: HI01 Headline indicators for the North East

**SPENSER\_notebook\_5:**

Calculates the attribute: Occupation for employed and unemployed people only. This column classifies these people in 9 categories

**SPENSER\_notebook\_6:**

Calculates the attribute: Occupation for inactive people only. This column classifies these people in 5 categories: student, looking after family or home, sick, retired and other

**SPENSER\_notebook\_7:**

Calculates the attribute: Income for employed and unemployed people, based on their sex, age and occupation type

**SPENSER\_notebook\_8:**

Swap income values for employed and unemployed people. This is required because it was observed (outputs from SPENSER\_notebook\_7) that in general, people aged between 40-49 were earning more than expected, while people between 18-21, 22-29, 30-39, 50-59 and 60+ were earning less than expected

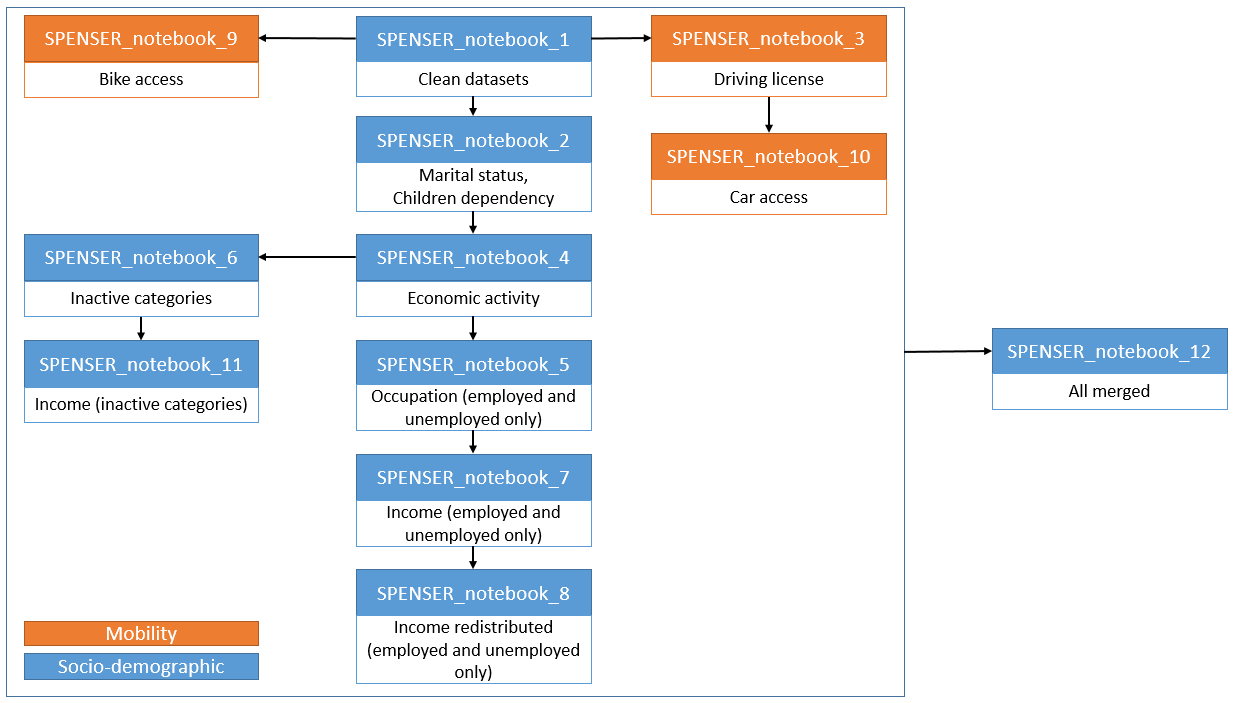
**SPENSER\_notebook\_9:**

Calculates the attribute: Bike access based on data from National Travel Survey.

**SPENSER\_notebook\_10:**

Calculates the attribute: Car access.

Image that shows the order in which notebooks need to be used:



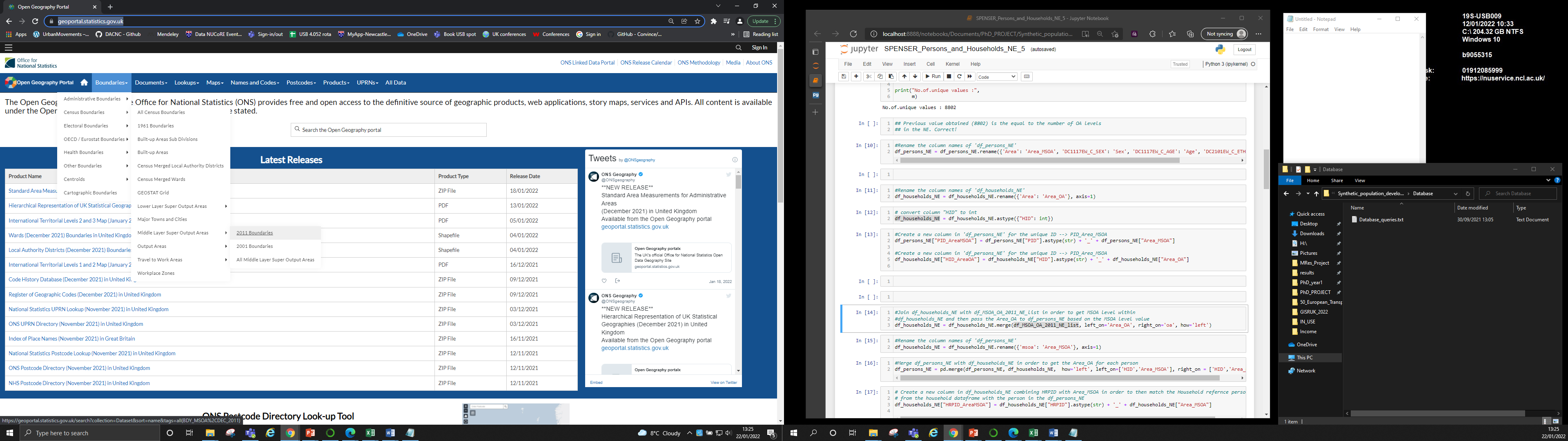
**File name: SPENSER\_notebook\_1**

**This Jupyter notebook:**

* Merges all persons and households of the study area in two dataframes (df\_persons\_NE and df\_households\_NE),
* Generates a unique primary key for each table (persons (PID\_AreaMSOA) and households (HID\_AreaOA))
* Removes unnecessary persons (those that were not assigned to any households (HIP = (-1)) and those that are assigned to buildings that are not residential (QS420\_CELL = -2) or education (QS420\_CELL = 26).
* Removes unnecessary households (those buildings that are not residential (QS420\_CELL = -2) or education (QS420\_CELL = 26).

**Dependencies:**

1. Libraries shown in cell 1
2. Households csv files from SPENSER
3. Persons csv files from SPENSER
4. MSOA and OA areas from <https://geoportal.statistics.gov.uk/> boundaries-census boundaries-Middle layer Super Output Areas – Census 2011



**File name: SPENSER\_notebook\_2**

**This Jupyter notebook:**

* Calculates the following columns:
  + Total people in household: count the number of persons in each household
  + Total children in household: count the number of children in each household
  + Same ethnic: check if there are people with the same ethnicity
  + Adult similar age: check if there are people within +- 10 years
  + Children\_dependency: Boolean value that tells if a person has a children dependency
  + Marital\_status: classifies each individual in married, couple or single based on their socio-demographic characteristics

**Dependencies:**

1. Libraries shown in cell 1
2. Households cvs file exported from SPENSER\_notebook\_1 (df\_households\_NE\_clean.csv)
3. Persons cvs file exported from SPENSER\_notebook\_1 (df\_persons\_NE\_clean.csv)

The first four columns are calculated by looping through each person from the same household and comparing their sociodemographic characteristics.

Then, there are calculated:

* the amount of people in the household (Total people in household);
* the number of people younger than 18 in the household (Total children in household);
* if there is at least another person in the household with the same ethnicity (Same ethnic);
* if there is at least one person in the household 10 years older or younger (Adult similar age) (this value is to identify potential couples and married people)

Once these attributes are calculated, it is possible to assign to each person a marital status and a children dependency, based on the following assumptions:

**Marital status (function marital\_status):**

* if the attribute related to the domestic situation of the household reference person (LC4408\_C\_AHTHUK11\_x) is equal to 2 (Married or same-sex civil partnership couple household), the person is older than 18 and there is another adult with a similar age (+- 10 years), then the person is considered as MARRIED
* if the attribute related to the domestic situation of the household reference person (LC4408\_C\_AHTHUK11\_x) is equal to 3 (Cohabiting couple household), the person is older than 18 and there is another adult with a similar age (+- 10 years), then the person is considered as COUPLE
* In the remaining cases, the person is considered SINGLE

**Children dependency (function children\_dependency):**

* if the attribute related to the domestic situation of the household reference person (LC4408\_C\_AHTHUK11\_x) is equal to 2 (Married or same-sex civil partnership couple household), 3 (Cohabiting couple household) or 4 (Lone parent household), the person is older than 18 years and there is at least one child in the household (Total\_Children\_in\_household > 0), then the person has a child
* In the remaining cases, the person does not have a children dependency

**File name: SPENSER\_notebook\_3**

**This Jupyter notebook:**

* Calculates the attribute: driving licence (True / False)

**Dependencies:**

1. Libraries shown in cell 1
2. Households cvs file exported from SPENSER\_notebook\_2 (df\_households\_NE\_clean.csv)
3. Persons cvs file exported from SPENSER\_notebook\_2 (df\_persons\_NE\_Household\_composition\_updated.csv)
4. RUC11\_OA11\_EW (csv file containing the type of rural/urban for each OA area). Data obtained from <https://geoportal.statistics.gov.uk/datasets/3ce248e9651f4dc094f84a4c5de18655/about>

(This table is for external validation only)

1. NTS0201 (<https://www.gov.uk/government/collections/national-travel-survey-statistics>)
   1. range values of groups of age
   2. % of men by age that have a driving licence
   3. % of women by age that have a driving licence
2. NTS9901 for validation and data collection (<https://www.gov.uk/government/collections/national-travel-survey-statistics>)

**Main function. Code explanation:**

Firstly, it is forced that one person in a household with at least one car, is assigned a driving licence. This is to avoid having households with vehicles and no one having a driving licence (e.g., vehicles would not be able to be used).

Additionally, there was defined a list containing weighted values about the number of households with cars (household\_car\_weight\_list). It was decided that people living in households with more than one car will have a higher probability to have a driving licence, than those living in a household with one car, and even more than those without a car in the household. This values were assigned firstly randomly and were adjusted after a few iterations. The best results were obtained when these values were 0.2, 0.3 and 0.5 for households without a car, with one car and with more than one, respectively.

Table NTS0201 provides information of full car driving licence holders by age and gender: England, 1975/76 onwards

Table NTS9901 provides information of full car driving licence holders by gender, region and Rural-Urban Classification1: 17 years old and over, England, 2002/03 and 2020

By comparing values of the North East in Tab NTS9901\_TimeSeries (based on the sex) against values from NTS0201, it is possible to estimate the % of males and females by range of age that have a driving licence for the year you want to get the data (in this example, 2019).

* % of males with a driving licence 2018/2019 in the NE is 79% in NTS9901, while in NTS0201, the total % of people with driving licence is 80%. Based on that these two values are very similar, it was assumed that the mean value of England driving licences per range of ages can be applied to the NE
* % of females with a driving licence 2018/2019 in the NE is 63% in NTS9901, while in NTS0201, the total % of people with driving licence is 71%.In this case, the values were pondered for each range of age:
  + Example: females(age30-39)NE = (% females(age30-39)England / (total % females England / total % females NE))
    - females(age30-39)NE = (74/(71/63)) = 66%

The % of males and females per range of age ((18,20), (21,29), (30,39), (40,49), (50,59), (60,69), (70,120)) obtained were the following:

men\_driving\_percentage\_list = [0.34,0.65,0.83,0.89,0.89,0.90,0.81]

women\_driving\_percentage\_list = [0.31,0.54,0.66,0.74,0.74,0.70,0.49]

The code starts looping through the people and selecting those of a specific sex and range of age. Based on the sex and range of age selected, the % of people with a driving licence in this subgroup is selected (driving\_percentage).

Then, people forced at the beginning to have a driving licence are removed from the selection

The remaining persons are split in three blocks (one for those with no cars in the household, those with one car, and those with more than one)

From each block, there are selected randomly the number of people to be assigned a driving licence, based on the weight value that is applied to each block

The selected ones in each block, are merged in one dataframe (df\_persons\_driving\_selection).

Then, it is checked the number of chosen people to be assigned a driving licence. This value is compared against the value that should be reached.

* If the value is lower, the remaining should be collected randomly from the dataframe that contains people with the same gender and age range
* If the value is the expected, then no more persons are selected

All persons that have been assigned a driving licence are updated their Driving licence column to True. The others remain as False.

Finally, a dataframe containing all persons from the synthetic population are merged in a single dataframe.

One the calculation is finished, a single dataframe (df\_persons\_NE\_after\_driving) is obtained. In this dataframe, the column Driving licence was updated to those selected.

Then, some analyses based on sex and range of age are developed in order to check if the % achieved are similar to those from the NTS tables.

Finally, once the results are acceptable, the dataframe is exported as csv file.

**File name: SPENSER\_notebook\_4**

**This Jupyter notebook:**

* Calculates the column: Economic activity (Employed, Unemployed, Inactive) based on age, gender and OA area. Census 2011 data is projected to 2019 using Regional labour market statistics: HI01 Headline indicators for the North East (<https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/datasets/headlinelabourforcesurveyindicatorsforthenortheasthi01> )

**Dependencies:**

1. Libraries shown in cell 1
2. Households cvs file exported from SPENSER\_notebook\_2 (df\_households\_NE\_clean.csv)
3. Persons cvs file exported from SPENSER\_notebook\_2
4. LC6107EW (Economic Activity by sex by age) - Census 2011 (<https://www.nomisweb.co.uk/census/2011/lc6107ew>)

This table needs to be cleaned and column names need to be modified.

Example:

Sex: **Males**; Age: Age **16** to **24**; Economic Activity: Economically active: **In employment**: Total; measures: Value

 is transformed into 

**1**\_**16**\_**24**\_**Employed**

1. Regional labour market statistics: HI01 Headline indicators for the North East (<https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/datasets/headlinelabourforcesurveyindicatorsforthenortheasthi01> )

Tables:

2a rates  Economic activity by age - Rates - People

2b rates  Economic activity by age - Rates - Men

2c rates  Economic activity by age - Rates - Women

This file has to be used to convert data from 2011 to 2019 by comparing values from both years (by sex and range of age)

Conversor value was included directly into the code

**Main function. Code explanation:**

**Inactive people:**

In this category, it was forced that persons which NSSEC is null (e.g., those that are not considered household reference persons with a specific NSSEC value) or NSSEC = 9 (students), were considered as inactive. This is because people which NSSEC is not null are associated to a specific profession (therefore, they can be considered as employed or unemployed) and students are mainly considered as an inactive part of the population (actually, there is a specific section for them in the inactive type of values).

Then, there are defined the range of ages ((16, 24), (25, 34), (35, 49), (50, 64), (65, 120)) and the inactive rates for males (1) and females (2) per range of age.

inactive\_rate\_1\_2019\_list = [38.3,10.0,9.1,26.8,90.3]

inactive\_rate\_2\_2019\_list = [44.5,23.1,19.3,31.6,94.8]

These values (related to the year 2019) were obtained from the “Regional labour market statistics: HI01 Headline indicators for the North East”.

Then, from the “Regional labour market statistics: HI01 Headline indicators for the North East”, there were obtained some converter values to transform the values from 2011 to 2019. In this case of study, there were compared values from (April18-Mar19) and (April10-Mar11) per sex and range of age. Values of (April18-Mar19) were divided by (April10-Mar11) values. Obtained values that are greater than 1 mean that there are more people in that category in 2019 than in 2011, while if the values are lower than 1 mean that there are less. (Later in the code, these values will be updated if the obtained results per sex and group of age differ in more than +-1% from the value in the “Regional labour market statistics: HI01 Headline indicators for the North East” of the year 2019).

The code starts looping through the gender type and range of ages, in order to select only those persons with the specified characteristics (e.g., males aged between 25 and 34).

The percentage of inactive persons of this sex and age range is initialised to 0. This values will be updated once people are selected as inactive in their sex and group of age, and then compared against the expected value from the “Regional labour market statistics: HI01 Headline indicators for the North East”.

Then, a while loop is initialised and will work until the % of inactive people selected is within 1% when compared against the value from the “Regional labour market statistics: HI01 Headline indicators for the North East”.

Then, the code loops through each OA area in the study area

The first OA area is selected and a dataframe (df\_economic\_activity\_area) is generated containing the values from the df\_economic\_activity (values from the Census 2011 that contains the number of people per age and sex considered as inactive, employed and unenmployed) with the values of inactive, employed and unemployed people of the OA area in groups of age.

Then, all people of the selected sex and group of age are stored in a dataframe (globals()[f"df\_{gender}\_{age\_range[0]}\_{age\_range[1]}"])

Another variable (globals()[f"total\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_in\_OA\_2019"]) counts the number of people selected previously.

3 variables are created, containing the type of economic activity based on sex and range of age. These values will be used to find the specific value of people of a specific sex, range of age and OA area that are considered as inactive, employed or unemployed in the Census 2011 table.

Then, the value of people of the selected range of age, sex and OA area that are inactive in 2011 are stored in the variable “globals()[f"inactive\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_LC6107EW"]”. The same for unemployed and employed:

“globals()[f"employed\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_LC6107EW"] = df\_economic\_activity\_area.iloc[0,df\_economic\_activity\_area.columns.get\_loc(col\_gender\_age0\_age1\_employed)]”

“globals()[f"unemployed\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_LC6107EW"] = df\_economic\_activity\_area.iloc[0,df\_economic\_activity\_area.columns.get\_loc(col\_gender\_age0\_age1\_unemployed)]”

Then these three values are combined to know the total number of people with the specific age range and sex that live in the selected OA area during 2011.

Then, this value is used to calculate the ratio of people between 2019 and 2011 with the specific sex, age range and OA area. The number of people selected from the synthetic population is divided by the total number of people from the census 2011 table.

If value > 1, it means there are more people in 2019 in the selected OA area of the specific gender and range of age. If value < 1, the opposite.

Then, a new value of inactive people (specific sex, range of age and OA area) is generated, updating the value from 2011 by multiplying the 2011:

New value for 2019 = (Value from 2011 (table LC6107EW)) \* (inactive conversor value (based on age range and sex)) \* (ratio of people 2019 vs 2011 )

This value is the number of people from the synthetic population that will be considered as "inactive" based on their OA area, range of age and sex.

Then, there is generated a dataframe containing only those persons that are considered students (LC4605\_C\_NSSEC\_x'] == 9): (globals()[f"df\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_students"]).

If there area student in the dataframe, they will be selected. The maximum number of selected students will be the “New value for 2019”.

If there is not any student or if the number of students selected is lower than the “New value for 2019”, then another selection will be done between those persons which NSSEC value is null or 9. A similar procedure is followed as before.

If there are still some people to be assigned as INACTIVE, then their NSSEC can be any value

After these 3 selection types ((LC4605\_C\_NSSEC\_x'] == 9), (NSSEC is\_null or NSSEC = 9), NSSEC any value), the selected persons are concatenated in a dataframe globals()[f"df\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_inactive\_all\_each\_area"] and then appended into a list (persons\_inactive\_list).

Once all OA areas are looped and all member of a gender and group of age are selected as inactive, then the previous list is converted into a dataframe (globals()[f"df\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_inactive\_all"]).

Then, using this dataframe, it is calculated the % of people inactive with the same sex and range of age.

The values is then compare against the one given in table “Regional labour market statistics:HI01 Headline indicators for the North East” related to year 2019. If the difference is not within a 1%, then a new iteration should be done updating the parameter that transform the employment rate from 2011 to 2019 ((globals()[f"inactive\_conversor\_{gender}\_{age\_range[0]}\_{age\_range[1]}"])). This value is increased or decreased by 0.025, depending if the value obtained is greater or lower than the expected.

If the obtained value is not within 1%, the while loop starts again and will stop once the previous difference is within 1%.

Once all groups of sex and range of ages are processed, the function finishes printing the following sentence: print('Job done. Check the results.')

Then, results obtained for each group (group of same sex and range of age) need to be concatenated in order to have all the persons selected as “Inactive” in a single dataframe (df\_persons\_NE\_inactive)

Finally, some analyses are done in order to find the obtained percentage values for each sex and range of ages.

**Employed people:**

The employed piece of code follows the same methodology as the one explained before for inactive people.

Firstly, there was created a dataframe containing all people except those selected before as inactive and those where NSSEC = 8 (8. Never worked and long-term unemployed). People with a NSSEC value equal to 8 were considered as unemployed.

In the code, the number of persons considered as employed (based on the rage of age, sex and OA area) were those based on the “New value for 2019” of employed obtained.

If after this first selection, there are still some people in the OA area to be assigned as "EMPLOYED" but there are no more people in the selected dataframe, then there are considered as well those people which NSSEC = 8 (in some of the cases, the amount of people in the initial dataframe used in this part of the code were not enough to select all the employed. That is why some people NSSEC = 8 needed to be considered as employed)

Once all OA areas are looped and all member of a gender and group of age are selected as employed, then the previous list is converted into a dataframe (globals()[f"df\_{gender}\_{age\_range[0]}\_{age\_range[1]}\_employed\_all"]).

Then, using this dataframe, it is calculated the % of people employed with the same sex and range of age.

The values is then compare against the one given in table “Regional labour market statistics:HI01 Headline indicators for the North East” related to year 2019. If the difference is not within a 1%, then a new iteration should be done updating the parameter that transform the employment rate from 2011 to 2019 ((globals()[f"employed\_conversor\_{gender}\_{age\_range[0]}\_{age\_range[1]}"])). This value is increased or decreased by 0.025, depending if the value obtained is greater or lower than the expected.

If the obtained value is not within 1%, the while loop starts again and will stop once the previous difference is within 1%.

Once all groups of sex and range of ages are processed, the function finishes printing the following sentence: print('Job done. Check the results.')

Then, results obtained for each group (group of same sex and range of age) need to be concatenated in order to have all the persons selected as “Inactive” in a single dataframe (df\_persons\_NE\_inactive)

Finally, some analyses are done in order to find the obtained percentage values for each sex and range of ages.

**Unemployed:**

Unemployed were considered those that were not considered as inactive or employed. The remaining people were then selected as unemployed.

Also, some analyses are done in order to find the obtained percentage values for each sex and range of ages.

##

Finally, each group of people (employed, inactive and unemployed) were updated their column “Economic activity” and saved as csv files.

**File name: SPENSER\_notebook\_5**

**This Jupyter notebook:**

* Calculates the column: Occupation for employed and unemployed people only. This column classifies these people in 9 categories

**Dependencies:**

1. Libraries shown in cell 1
2. Households cvs file exported from SPENSER\_notebook\_2 (df\_households\_NE\_clean.csv)
3. Persons cvs file exported from SPENSER\_notebook\_2
4. LC6112EW (Occupation by age) - Census 2011 – Nomis (<https://www.nomisweb.co.uk/census/2011/lc6112ew>)

This table needs to be cleaned and column names need to be modified.

Example:

Occupation: **2**. Professional occupations; Age: Age **16** to **24**; measures: Value

* Transformed into 

occupation\_**2**\_**16**\_**24**

1. annual population survey - workplace analysis (<https://www.nomisweb.co.uk/datasets/apsw>)

Geography: local authority

Date: the chosen one for your project

Cell: T10b Employment by occupation (SOC2010) and industry (SIC 2007)

This file has to be used to convert data from 2011 to 2019 by comparing values from both years (by local authority and sex but then an average value was used for the whole population)

Converter value was included directly into the code

**Main function. Code explanation:**

A dataframe containing only those people aged 16 or more and categorised as employed or unemployed are considered only.

A list of range of ages is generated based on data from the Census 2011 table LC6112EW ((16, 24), (25, 34), (35, 49), (50, 64), (65, 120))

A list containing the order in which occupations are assigned for males and females are defined. occupation\_list\_males = [2,9,3,1,5,8,4,6,7] #values used in my project

occupation\_list\_females = [2,3,1,9,6,7,4,5,8] #values used in my project

The order of the values in the list varies depending on the % of people associated (in total and by sex) to each occupation in the area of study.

For a second selection of people per occupation, there were created two more list with the order in which occupations are assigned (one per sex). The orders are slightly different

occupation\_list\_males\_last = [2,3,9,4,5,1,8,6,7] #values used in my project

occupation\_list\_females\_last = [2,3,9,1,5,7,8,4,6] #values used in my project

INITIAL values to transform data from 2011 to 2019 based on the relationship (2019/2011).

There is not any difference between sex because LC6112EW census 2011 table does not provide differences in sex.

This values were obtained from the “annual population survey - workplace analysis”, by comparing data from Apr 2010-Mar 2011 and Apr 2018-Mar 2019. Values of males and females for each local authority were aggregated and the converter value was calculated as follows:

occupation\_1\_conversor = (number of males in occupation 1 2019 + number of females in occupation 1 2019) / (number of males in occupation 1 2011 + number of females in occupation 1 2011).

Then, the average of all values (all LAD in the project) was considered as the occupation converter value for each occupation.

Based on data from the “annual population survey - workplace analysis”, there were calculated the % of males and females related to each occupation. These values are useful to identify the order in which occupations are going to be assigned (as it was explained before) for each sex.

The code starts looping through a list of OA areas.

Once one OA area is selected, there are selected the following data:

* LC6112EW row related to this OA area, where information about the number of people in each occupation type are assigned in 2011
* Persons in the synthetic population aged 16 or more, categorised as employed or unemployed and in the OA area selected.

Then, there is another loop for each age range.

Then, there is a selection of males and females within this range of age

For each gender, it is selected the order in which the occupations are assigned.

Then, the occupation assignment starts following the previous order, based on a for loop through the occupations

For each occupation, there is selected the appropriate value related to the sex, age of range and OA area from the LC6112EW table (df\_occupation\_OAarea in the code).

Then, it is calculated the total number of people in the range of age that are found in the census data

After that, a ratio comparing the number of people in 2019 and 2011 is calculated (ratio\_people\_2019\_2011)

Then, there is a new loop through the occupation list.

The number of people to be assigned to each occupation needs to be updated to 2019. This is done as follows:

Update the value to 2019: 2019 = 2011 \* occupation\_"X"\_conversor \* ratio\_people\_2019\_2011

Once the value is updated, the random selection of persons per occupation is done by sex and range of age

Then, if there are still some persons not assigned an occupation, then they are selected based on the second occupation list (occupation\_list\_males\_last and occupation\_list\_females\_last).

If there are still some people to be assigned an occupation but all gaps have been used, then it is necessary to allocate the people one by one base on the order occupations are given [based on occupation\_list\_males\_last and occupation\_list\_emales\_last].

Finally, the dataframes generated are concatenated (first, second and third selection, and then all together).

The code finishes and the following message will appear: 'Code has finished. Check the results'

Then, some analyses are done in order to calculate the % of people (total, males and females) in each occupation type

If the obtained results are not the expected, then the order in which occupations are chosen must be altered for both, males and females in their first and second selection (third one is the same as the second).

Finally, the dataframe is exported as a csv file.

##

**File name: SPENSER\_notebook\_6**

**This Jupyter notebook:**

* Calculates the column: Occupation for inactive people only. This column classifies these people in 5 categories: student, looking after family or home, sick, retired and other

**Dependencies:**

1. Libraries shown in cell 1
2. Households cvs file exported from SPENSER\_notebook\_2 (df\_households\_NE\_clean.csv)
3. Persons cvs file exported from SPENSER\_notebook\_4
4. QS601EW (Economic activity) - Census 2011 – Nomis (<https://www.nomisweb.co.uk/census/2011/qs601ew>)

This table needs to be cleaned and column names need to be modified.

|  |  |
| --- | --- |
| Old names | New names |
| Economic Activity: Economically active: Full-time student; measures: Value | Economically\_active\_Fulltime\_student |
| Economic Activity: Economically inactive: Retired; measures: Value | Economically\_inactive\_Retired |
| Economic Activity: Economically inactive: Student (including full-time students); measures: Value | Economically\_inactive\_Student |
| Economic Activity: Economically inactive: Looking after home or family; measures: Value | Economically\_inactive\_Looking\_after\_home\_or\_family |
| Economic Activity: Economically inactive: Long-term sick or disabled; measures: Value | Economically\_inactive\_Longterm\_sick\_or\_disabled |
| Economic Activity: Economically inactive: Other; measures: Value | Economically\_inactive\_Other |

1. Regional labour market statistics: HI01 Headline indicators for the North East (Tables 10a, 10b, 10c)

(<https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/datasets/headlinelabourforcesurveyindicatorsforthenortheasthi01>)  
This file has to be used to convert data from 2011 to 2019 by comparing values from both years (by sex and range of age)

Conversor value was included directly into the code

**Main function. Code explanation:**

In this code, only people aged between 16 and 64 are grouped in one of the 5 possible categories (student, retired, looking after family/home, sick and other). People older than 64 are considered as “Retired” only. This is because information provided in document “Regional labour market statistics: HI01 Headline indicators for the North East” only shows statistics related to this range of age (16-64).

A dataframe containing only those people in the range of age is used (df\_persons\_NE\_inactive\_16\_64)

Students:

A list containing both genders is created (gender\_list = [1,2] (male (1) and female (2)))

An empty list where the small blocks of dataframes of STUDENTS will be stored is created empty: persons\_NE\_inactive\_students\_chosen\_all\_list = []

An “inactive\_student\_conversor” value is defined. It converts the % of inactive students from 2011 to 2019 based on data from "Regional labour market statistics: HI01 Headline indicators for the North East"

Then, the proportion of males (percentage\_1\_student) and females (percentage\_2\_student) that belong to this category are defined. These values are obtained from the Regional labour market statistics: HI01 Headline indicators for the North East".

The function starts looping through the OA areas of the study area.

The row of the “QS601EW” file that contains information of the looped OA area is selected (total\_QS601EW\_inactive\_in\_OA\_2011).

Then, the people from the synthetic population within this OA area are selected in a dataframe (df\_persons\_NE\_inactive\_16\_64\_OAarea) and the total number of them are stored (total\_inactive\_in\_OA\_2019).

Next, is to calculate the ratio of people 2019 vs 2011 (to increase or decrease the number of people that have to be selected per type of inactivity (retired, student, looking after, disable or other))

(ratio\_people\_2019\_2011 = total\_inactive\_in\_OA\_2019/total\_QS601EW\_inactive\_in\_OA\_2011)

Then, the value of the number of people living in the seleted OA area that are inactive STUDENTS from table QS601EW are stored in a variable (total\_QS601EW\_inactive\_students\_in\_OA\_2011)

The number of inactive students to be assigned in 2019 (inactive\_students\_2019) is based on the following parameters:

* total\_QS601EW\_inactive\_students\_in\_OA\_2011
* inactive\_student\_conversor
* ratio\_people\_2019\_2011

Once it is known the total number of students (projected to 2019) that are going to be assigned in the selected OA area, a loop on the gender list is started. This is to keep the proportion of students to be assigned based on their gender.

Students are selected in various (5) selections. If after the first selection, there are still some remaining students to be selected, then second selection is triggered. Once there are not more students to be assigned in the OA area, people from the other gender are chosen.

The 5 selections are the following:

* Selection 1:
  + 1. QS420\_CELL = 26 (agents living in buildings related to education)
  + 2. Sex: proportions based on variables "percentage\_male\_student" and "percentage\_male\_student" assigned previously
* Selection 2:
  + 1. NSSEC == 9 (Students)
  + 2. Sex: proportions based on variables "percentage\_male\_student" and "percentage\_male\_student" assigned previously
* Selection 3:
  + 1. LC4408\_C\_AHTHUK11 = 5 (Domestic situation = Multi-person household)
  + 2. Age: people 16-35 years old
  + 3. Sex: proportions based on variables "percentage\_male\_student" and "percentage\_male\_student" assigned previously
* Selection 4:
  + 1. LC4408\_C\_AHTHUK11 = 1,2, OR 3 (Domestic situation = one person household, married, cohabitating couple)
  + 2. Age: people 16-35 years old
  + 3. Sex: proportions based on variables "percentage\_male\_student" and "percentage\_male\_student" assigned previously
* Selection 5:
  + Finally, select randomly the remaining ones (leftovers) based on the age only (sort age ascending order and selecte the remaining ones)

Once the code is finished, it is calculated the number of people selected as students (len(df\_inactive\_students\_all)) and it is checked if there is any duplicates (df\_inactive\_students\_all\_1). If the lenght of both dataframes is equal, it means there is not any duplicate row.

Then, some checks are done in order to calculate the % of people selected, the % of males, the % of females and the proportion of males/females selected.

If those values are not close enough to the ones provided in the document "Regional labour market statistics: HI01 Headline indicators for the North East (Tables 10a, 10b, 10c)", then the "inactive\_student\_conversor" parameter value needs to be updated (increased or decreased, based on the calculated and the expected values).

Looking after family/home:

Once students have been selected, then students are removed from the dataframe in order to avoid selecting the same person in different categories (df\_persons\_NE\_inactive\_16\_64\_NO\_students).

The selection of “looking after family/home” people is similar to the one followed with the students. The code is the same until the gender loop starts (before there are only changes about the values of the parameters used (inactive\_loookingafter\_conversor, percentage\_1\_lookingafter, percentage\_2\_lookingafter))

In order to avoid having more than one person per household in the category "LOOKING AFTER FAMILY / HOME", after the first iteration (female loop) and when the second loop starts (male), if a male belongs to a household were a female was selected, then the male is not considered to be in the category "LOOKING AFTER FAMILY / HOME". A new dataframe is created removing the rows which household IDs have already assigned a person as "LOOKING AFTER HOME/FAMILY". The values to be removed are coming from the list "all\_HID\_AreaOA\_x\_list"  
For females, the list is always empty. For males, this list contains the HID\_AreaOA\_x of females that were already assigned as "LOOKING AFTER HOME/FAMILY"

There are 3 selections:

* Selection 1:
  + 1. Marital status: Married or Couple
  + 2. Children dependency: True
  + 3. Sex: proportions based on variables "percentage\_male\_lookingafter" and "percentage\_female\_lookingafter" assigned previously
* Selection 2:
  + 1. Marital status: Married or Couple
  + 2. Children dependency: False
  + 3. Sex: proportions based on variables "percentage\_male\_lookingafter" and "percentage\_female\_lookingafter" assigned previously
  + 4. Age: sort age by 9asecnding order = False) and choose the top 'n values required
  + # In case there are still leftovers:
* Selection 3:
  + 1. Sex: proportions based on variables "percentage\_male\_lookingafter" and "percentage\_female\_lookingafter" assigned previously
  + 2. Marital status: any type (married, couple or single)
  + Age: no age requirements.

Once selections in the OA area between females are done, their households IDs are recorded in a list (all\_HID\_AreaOA\_x\_list).

Once the code is finished, it is calculated the number of people selected as looking after family/home (len(df\_inactive\_lookingafter\_all\_22feb\_3)) and it is checked if there is any duplicates (df\_inactive\_lookingafter\_all\_22feb\_3\_1). If the lenght of both dataframes is equal, it means there is not any duplicate row.

Then, some checks are done in order to calculate the % of people selected, the % of males, the % of females and the proportion of males/females selected.

If those values are not close enough to the ones provided in the document "Regional labour market statistics: HI01 Headline indicators for the North East (Tables 10a, 10b, 10c)", then the " inactive\_loookingafter\_conversor" parameter value needs to be updated (increased or decreased, based on the calculated and the expected values).

Retired:

Once looking after people have been selected, then looking after people are removed from the dataframe in order to avoid selecting the same person in different categories (df\_persons\_NE\_inactive\_16\_64\_NO\_students\_NO\_lookingafter).

The selection of “retired” people is similar to the one followed with the students and looking after. The code is the same until the gender loop starts (before there are only changes about the values of the parameters used (inactive\_retired\_conversor, percentage\_1\_retired, percentage\_2\_retired)).

In this case, there are 4 selections:

* First selection:
  + 1. People aged 60-64
  + 2. Sex: proportions based on variables "percentage\_male\_retired" and "percentage\_female\_retired" assigned previously
* Second selection:
  + 1. People aged 55-59
  + 2. Sex: proportions based on variables "percentage\_male\_retired" and "percentage\_female\_retired" assigned previously
* Third selection:
  + 1. People aged 50-54
  + 2. Sex: proportions based on variables "percentage\_male\_retired" and "percentage\_female\_retired" assigned previously
* Forth selection:
  + 1. People aged < 50
  + 2. Sex: proportions based on variables "percentage\_male\_retired" and "percentage\_female\_retired" assigned previously

Once the code is finished, it is calculated the number of people selected as retired (len(df\_people\_NE\_inactive\_retired\_all\_v2)) and it is checked if there is any duplicates (df\_people\_NE\_inactive\_retired\_all\_v2\_1). If the length of both dataframes is equal, it means there is not any duplicate row.

Then, some checks are done in order to calculate the % of people selected, the % of males, the % of females and the proportion of males/females selected.

If those values are not close enough to the ones provided in the document "Regional labour market statistics: HI01 Headline indicators for the North East (Tables 10a, 10b, 10c)", then the "inactive\_retired\_conversor" parameter value needs to be updated (increased or decreased, based on the calculated and the expected values).

Sick:

Once retired people have been selected, then retired people are removed from the dataframe in order to avoid selecting the same person in different categories (df\_persons\_NE\_inactive\_16\_64\_NO\_students\_NO\_lookingafter\_NO\_retired).

The selection of “sick” people is similar to the one followed with the students, looking after and retired. The code is the same until the gender loop starts (before there are only changes about the values of the parameters used (inactive\_sick\_conversor, percentage\_1\_sick, percentage\_2\_sick)).

In this case, there are two selections:

* First selection:
  + 1. People aged 50-64
  + 2. Sex: proportions based on variables "percentage\_1\_longtermsick" and "percentage\_2\_longtermsick" assigned previously
* Second selection:
  + 1. People aged < 50
  + 2. Sex: proportions based on variables "percentage\_1\_longtermsick" and "percentage\_2\_longtermsick" assigned previously

Once the code is finished, it is calculated the number of people selected as sick (len(df\_people\_NE\_inactive\_longtermsick\_all)) and it is checked if there is any duplicates (df\_people\_NE\_inactive\_longtermsick\_all\_1). If the length of both dataframes is equal, it means there is not any duplicate row.

Then, some checks are done in order to calculate the % of people selected, the % of males, the % of females and the proportion of males/females selected.

If those values are not close enough to the ones provided in the document "Regional labour market statistics: HI01 Headline indicators for the North East (Tables 10a, 10b, 10c)", then the "inactive\_retired\_conversor" parameter value needs to be updated (increased or decreased, based on the calculated and the expected values).

Other:

Finally, category “other” contains the remaining people that were not assigned as student, looking after family/home, retired or sick (df\_persons\_NE\_inactive\_16\_64\_Other).

Some checks are done in order to calculate the % of people selected, the % of males, the % of females and the proportion of males/females selected.

Results obtained should be close enough to those provided by the "Regional labour market statistics: HI01 Headline indicators for the North East (Tables 10a, 10b, 10c)". Some differences may appear due to this is the last selection done and less possibilities to choose people are available.

All inactive people aged more than 64 years old are stored in a dataframe (df\_persons\_NE\_inactive\_65\_120). This group of people will be considered as retired.

Once all inactive categories are defined, the ‘Occupation’ column is updated in each dataframe with the correspondent value.

All updated dataframes are meged in a single file (df\_persons\_NE\_inactive\_5categories).

Finally, the previous dataframe is saved as a csv file.

##

**File name: SPENSER\_notebook\_7**

**This Jupyter notebook:**

* Calculates the column: Income for employed and unemployed people, based on their sex, age and occupation type

**Dependencies:**

1. Libraries shown in cell 1
2. Persons cvs file exported from SPENSER\_notebook\_5
3. ONS data about the annual gross income of the year of analysis:
   1. Source: ONS
   2. Dataset: Earnings and hours worked, region by occupation by two-digit SOC: ASHE Table 3
   3. Link: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/earningsandworkinghours/datasets/regionbyoccupation2digitsocashetable3
   4. Year: 2019 revised edition of this dataset
   5. File downloaded: table32019revised.zip
   6. Table: Work Region Occupation SOC10 (2) Table 3.7a Annual pay - Gross 2019.xls
4. ONS data about the gender pay gap per region:
   1. Dataset: Earnings and hours worked, region by occupation by two-digit SOC: ASHE Table 15
   2. Link: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/earningsandworkinghours/datasets/regionbyoccupation4digitsoc2010ashetable15
   3. Year: 2019 revised edition of this dataset
   4. File downloaded: table152019revised.zip
   5. Table: Work Region Occupation SOC10 (4) Table 15.12 Gender pay gap 2019.xls
5. ONS data about income per range of age:
   1. Earnings and hours worked, age group: ASHE Table 6
   2. Link: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/earningsandworkinghours/datasets/agegroupashetable6
   3. Year: 2019 revised edition of this dataset
   4. File downloaded: table62019revised.zip
   5. Table: Age Group Table 6.7a Annual pay - Gross 2019.xls

**Main function. Code explanation:**

df\_persons\_NE\_occupation contains all persons assigned an occupation type (1-9).

In this notebook, income is calculated per occupation type separately (one cell per occupation type).

Each cel contains several parameters that have to be adjusted until results obtained are close enough to those provided by ONS (Point 3 in dependencies: ONS data about the annual gross income of the year of analysis). These parameters are the following:

* Occupation min, max income (example: occupation\_2\_income). These values are guessed from the “ONS data about the annual gross income of the year of analysis”, just checking the values of the lowest and highest percentile values per occupation in the selected region.
* Gender pay gap: (example: occupation\_2\_gap = 0.114). This value is obtained from Earnings and hours worked, region by occupation by two-digit SOC: ASHE Table 15
* Range of income value that a person can earn depending on their age. A conditional IF clause is developed in the code defining the width of the range of income per age range.

The code works as follows:

* Occupation min and max income values are defined (example: occupation\_2\_income = [14500,69000])
* Gender gap is defined (example: occupation\_2\_gap = 0.114)
* Then, people assigned to a specific occupation are selected in a dataframe ((globals()[f"df\_occupation\_{occupation}"]))
* Previous dataframe is looped in order that each person in the dataframe is assigned an income value.
* Income value depends on the age of the person. Depending on their age and the IF clause defined, the person will get a range of min, max income that can earn.
* Then, a random value between min and max is selected equally distributed.
* Then, the gender gap is applied. If the person is a male, then his income will be increased half of the gender gap. If the person is female, then her income will be decreased half of the gender gap.
* The income value is updated in the column “Income” and another person from the dataframe is selected
* Once all persons in the dataframe are processed, the code is finished with the following print on the screen: 'Job done. check the results'

Once income is calculated for people from a specific occupation type, statistical calculus are shown (min, max, mean, median, histogram and percentiles). These values are compared against those provided by ONS (Earnings and hours worked, region by occupation by two-digit SOC: ASHE Table 3). If results are not close enough to those expected, then it is required to modify the previous 3 explained parameters. This procedure has to be repeated as many times as required until results obtained are good enough.

Once income values are obtained for all occupation types, then the dataframes are combined in a single dataframe (df\_income\_all) and the same statistical calculus are done and compared against global data from ONS (Earnings and hours worked, region by occupation by two-digit SOC: ASHE Table 3).

###

**File name: SPENSER\_notebook\_8**

**This Jupyter notebook:**

* Swap income values for employed and unemployed people. This is required because it was observed (outputs from SPENSER\_notebook\_7) that in general, people aged between 40-49 were earning more than expected, while people between 18-21, 22-29, 30-39, 50-59 and 60+ were earning less than expected

**Dependencies:**

1. Libraries shown in cell 1
2. Persons cvs file exported from SPENSER\_notebook\_7

**Main function. Code explanation:**

It was observed that the median income values per range of age were not proportional to those shown in the ONS data about income per range of age (Earnings and hours worked, age group: ASHE Table 6, file: Age Group Table 6.7a Annual pay - Gross 2019.xls) (See table below).

|  |  |  |  |
| --- | --- | --- | --- |
|  | ONS median annual gross income (UK) | Calculated median gross income | Difference |
| **All employees** | 24,937 | 23,050 | 1,887 |
| 16-17b | 3,198 | 5,500 | -2,302 |
| 18-21 | 10,958 | 8,230 | 2,728 |
| 22-29 | 22,437 | 17,210 | 5,227 |
| 30-39 | 27,358 | 24,940 | 2,418 |
| 40-49 | 28,881 | 39,880 | -10,999 |
| 50-59 | 26,000 | 26,430 | -430 |
| 60+ | 19,252 | 15,560 | 3,692 |

This code tries to reduce the median income of those people between 40 and 49 years, and increase those between 18-21, 22-29, 30-39, 50-59 and 60+. People swap their income value by occupation type and sex. This is done in two steps.

The first step adjusts the median income value of those people aged between 18-21 swapping some of their income values with some people aged 22-29 and 60+.

In this case, there were selected a 6.9% of people aged 22-29 and another 6.9% of people aged 60+. Their incomes were swapped with people aged 18-21 (percentage values need to be adjusted to each case). This iterative process needs to be run as many times as required (adjusting the % of people swapping their incomes) until the calculated median gross income value for the people aged 18-21 is close enough to the data provided by the ONS.

Once people aged 18-21 have fixed the median annual gross income, it is time to fix the remaining range of ages in the second step.

In the case shown in the code, 45% of people aged 40-49 and 15% of people aged 50-59 swap their income value with people aged 22-29, 30-39 and 60+. The proportions given to each range of age are 0.44, 0.24 and 0.32, respectively.

The percentages of people with high income swapping their income values and the proportions of people with low income values swapping theirs with the first ones are obtained after several tries and errors. Each case study should find the best combination to get an optimal result.

Once an optimal combination of the parameters is achieved, the differences between observed data from ONS and calculated data should be constant.

The ONS data refers to the UK, while the calculated data refers only to a specific region of England (North-East), so differences between them are expected.

###

**File name: SPENSER\_notebook\_9**

**This Jupyter notebook:**

* Calculates the column: Bike access based on data from National Travel Survey.

**Dependencies:**

1. Libraries shown in cell 1
2. Persons csv files from SPENSER (spenser\_notebook\_1)
3. NTS table NTS0608 (Bicycle ownership by age: England, 2017/19)  
   link: <https://www.gov.uk/government/statistical-data-sets/nts06-age-gender-and-modal-breakdown>

**Main function. Code explanation:**

This code assign bike access to the population based on data from NTS. Bike access is assigned based on the age of the people.

A specific percentage of people of a specific range of age is randomly selected and assigned the attribute of bike access.

##

**File name: SPENSER\_notebook\_10**

**This Jupyter notebook:**

* Calculates the column: Car access.

**Dependencies:**

1. Libraries shown in cell 1
2. Persons csv files from SPENSER (spenser\_notebook\_1)

**Main function. Code explanation:**

This code assign car access to the population based two condtions:

* If the person lives in a household with at least one car
* If the person has a driving license.

If both conditions are true, then the person is assigned with car access.

If any or both conditions are false, then the person does not have car access.

###

**File name: SPENSER\_notebook\_11**

**This Jupyter notebook:**

* Calculates the column: Income, for inactive people (students, retired, sick, looking after home or family, others)

**Dependencies:**

1. Libraries shown in cell 1
2. Persons csv files from SPENSER (spenser\_notebook\_6)
3. Pensioners' incomes series: An analysis of trends in pensioner incomes: 1994/95 to 2018/19  
   Link: <https://www.gov.uk/government/statistics/pensioners-incomes-series-financial-year-2018-to-2019>
4. Personal Independence Payment (PIP). Link: <https://www.gov.uk/pip>
5. Universal Credit. Link: <https://www.gov.uk/universal-credit>
6. Carers allowance. Link: <https://www.gov.uk/carers-allowance>
7. National Living Wage and National Minimum Wage: government response to the Low Pay Commission’s Autumn 2019 recommendations.   
   Link: <https://www.gov.uk/government/publications/national-living-wage-and-national-minimum-wage-government-response-to-the-low-pay-commissions-autumn-2019-recommendations>

**Main function. Code explanation:**

This notebook calculates the income for inactive people (students, retired, sick, looking after home or family, others), based on the socio-demographic characteristics of each individual, using data from the government.

Retired:

Income value for retired people is based on data from “Pensioners' incomes series: An analysis of trends in pensioner incomes: 1994/95 to 2018/19”.

A median income is given to each person depending if they are married or couple, male single or female single.

Sick:

Sick people get an income value based the PIP (Personal Independence Payment) and Universal Credit. Based on their personal characteristics, they can get the following:

* PIP daily living
* PIP mobility
* Universal credit standard
* Universal credit for children
* Universal credit extra disability or health condition.

Looking after home or family:

Income is based on the carers allowance and the Universal Credit

Other:

Income is based on the Universal Credit

Student:

Students get their income based on their age and the number of hours worked (random value between 5 and 20 per week).

###

**File name: SPENSER\_notebook\_12**

**This Jupyter notebook:**

* Merges all previous generated outputs in one single dataframe

**Dependencies:**

1. Libraries shown in cell 1
2. All output csv files from the following notebooks:
   1. SPENSER\_notebook\_4 (only unemployed people)
   2. SPENSER \_notebook\_8 (income for employed and unemployed people)
   3. SPENSER\_notebook\_9 (bike access)
   4. SPENSER\_notebook\_10 (driving licence and car access)
   5. SPENSER\_notebook\_11 (income for inactive people)

**Main function. Code explanation:**

This code merges all dataframes in one. The goal is to have all calculated attributes together in a single dataframe.

It starts identifying unemployed people under 16 years old and updating their “economic activity”, “occupation” and “income” values.

Then, they are concatenated with people with an occupation and those inactive. The result should contain the whole population of the study area.

Then, some analyses of the attributes are made based on three groups of age (0-15, 16-64, and 65+).

Dataframes related to mobility (car access, driving licence and bike access) are opened.

Similar analyses of the mobility attributes are made based on three groups of age (0-15, 16-64, and 65+) and merged with the previous dataset that contains all socio-demographic attributes calculated (df\_persons\_NE\_sociodemographics\_clean).

Finally, the relevant columns are kept in a new dataframe (df\_persons\_NE\_sociodemographics\_mobility\_clean). Columns are renamed and the dataframe is saved as a CSV file.