Part 2: Analysing the “UN Wage” Dataset

1) Find the number of unique countries in the dataset.

A:Result 95

In the Dataset there are 95 different countries.

Task1.pig

SCRIPT:

```
Data =LOAD 'UN_wage_data_with_currency.csv' using PigStorage(',') AS
(CountryArea,Year,Sex,Classification,Subclassification,Coverage,Scope,Source,SourceID,Value,Currency,Value_Footnotes);


CountryArea = FOREACH Data GENERATE CountryArea;


distCountry = DISTINCT CountryArea;


loopdistCountry = group distCountry all;



country_count = foreach loopdistCountry  Generate COUNT(distCountry);


Dump country_count;
```

2) Find countries which have used more than 1 currency. List the countries together with the number of currencies used.

A:

Results:

```
(Malta,2)
(Croatia,3)
(Slovenia,3)
(San Marino,2)
(El Salvador,2)
(Netherlands,2)
(Serbia and Montenegro,2)
```

In the dataset the preciding seven countries have used more than one currencies.

In the result you will find the name of the country and then the number of currencies used.

Task2.pig

SCRIPT:

```
Data =LOAD 'UN_wage_data_with_currency.csv' using PigStorage(',') AS (CountryArea,Year,Sex,Classification,Subclassification,Coverage,Scope,Source,SourceID,Value,Currency,Value_Footnotes);

CountryAreaCur = FOREACH Data GENERATE CountryArea,Currency;

distCountryCur = DISTINCT CountryAreaCur;

countryGroup = group distCountryCur by CountryArea;

country_count = foreach  countryGroup  Generate group,COUNT(distCountryCur.CountryArea) as count;

filter_country = FILTER country_count by count>1;

Dump filter_country;
```

3) Using only *monthly wage* data, calculate the female-to-male monthly wage ratio of each country in each year. List the top 10 countries in descending order of the ratio.

A:

- Some countries may have multiple entries for the same year from

different sources. These are distinguished by different source IDs.

You should have 1 result per country per year per source ID.

- A ratio of “X to Y” means the value of “X divided by Y”.
- Ignore all *hourly* and *weekly wage* data. Use only *monthly wage*

data. Countries with no *monthly wage* entry (e.g. the U.K.) will not appear in the result.

Task3.pig

Results:

```
((Bahrain,2008,# 0),1.4514767932489452)
((Bahrain,2007,# 0),1.4)
((Bahrain,1987,# 0),1.3855932203389831)
((Bahrain,1988,# 0),1.3640350877192982)
((Bahrain,1989,# 0),1.3632286995515694)
((Swaziland,1989,# 2),1.3571428571428572)
((Bahrain,1990,# 0),1.3111111111111111)
((Bahrain,2006,# 0),1.265)
((Swaziland,1989,# 1),1.2483221476510067)
((Bahrain,1991,# 0),1.2478632478632479)
```

The resuls are selected without filtering the coverage column.

The preceding 10 results are the countries with the maximg rate for the value of the row with sex Female divided by the row with the sex male grouped by country and year and source ID.

If you take all rows of the result and not only the top ten, it has one row per country,per year,per source id with the female/male salary rate.

SCRIPT:

```
Data =LOAD 'UN_wage_data_with_currency.csv' using PigStorage(',') AS (CountryArea,Year,Sex,Classification,Subclassification,Coverage,Scope,Source,SourceID,Value:double,Currency,Value_Footnotes);

filtered_scope_e = FILTER Data by Scope=='Earnings per month';
filtered_scope_w= FILTER Data by Scope=='Wage rates per month';

filtered_scope = UNION filtered_scope_e,filtered_scope_w;
```

```
filtered_sex = FILTER filtered_scope by Sex =='Women' OR Sex =='Men';



Data_proj = FOREACH filtered_sex GENERATE CountryArea, Year, Sex, SourceID,Value;


Group_Data_proj = group Data_proj by (CountryArea,Year,SourceID);


country_count = foreach Group_Data_proj
  {
  male= filter Data_proj by Sex =='Men';
  female = filter Data_proj by Sex=='Women';
  Generate group,flatten(female.Value) as female,flatten(male.Value) as male;
  };


country_div = foreach country_count
  {
        Generate group,(float)(female)/(male) as SexWageRatio;
  };


 ordered_country = ORDER country_div BY SexWageRatio DESC;


 Top10_ordered_country = LIMIT ordered_country 10;



dump Top10_ordered_country;
```

4) Using both *weekly* and *monthly wage*, calculating the average monthly wage of each country in each year across both sexes and all data sources. List the result in country alphabetical and descending year, together with the currency.

A:

- Ignore *hourly wage* data as we cannot assume the number of hours per week in each country.
- For *weekly wage* data, we need to convert them to monthly wage. We assume there are 4 weeks in 1 month.
- You will have multiple entries for a country in a year (from

different data sources, etc.). Take the average.

- You result should have 1 entry per country per year.

Task4.pig

part-r-00000

Results: The result for the script is generated in a different file and placed on my VM **RG-N530-C04** and the file name is **part-r-00000** as it is a very big list the path of the part-r-00000 in VM is **Training-desktop-Task4**

The resuls are selected without filtering the coverage column.

The results are calculated by using the following filter :

FILTER Data by Scope=='Earnings per week'

to filter weekly wages and multiply that rows to get the monthly wage.

SCRIPT:

```
Data =LOAD 'UN_wage_data_with_currency.csv' using PigStorage(',') AS (CountryArea,Year,Sex,Classification,Subclassification,Coverage,Scope,Source,SourceID,Value:double,Currency,Value_Footnotes);

filtered_scope_week= FILTER Data by Scope=='Earnings per week';



Data_proj_week = FOREACH filtered_scope_week GENERATE CountryArea, Year,Currency,Value*4;

filtered_scope_e = FILTER Data by Scope=='Earnings per month';
filtered_scope_w= FILTER Data by Scope=='Wage rates per month';

filtered_scope = UNION filtered_scope_e,filtered_scope_w;



Data_proj_m = FOREACH filtered_scope GENERATE CountryArea, Year,Currency,Value;
```

```
Data_proj =UNION Data_proj_m,Data_proj_week;



Group_Data_proj = group Data_proj by (CountryArea,Year,Currency);

country_count = foreach Group_Data_proj
  {
   Generate group.CountryArea,group.Year,group.Currency,flatten(AVG(Data_proj.Value)) as avg;
  };

 ordered_country = ORDER country_count BY CountryArea ASC,Year DESC;



dump Top10_ordered_country;
```

5) Using your answer from question 4, calculate the top 10 countries with the highest percentage change in average monthly wage per year since their data began.

A:

- As some countries changed their currencies in the dataset, you

should have 1 result per country per currency[1].

- If the data of country X go from year 2000 to 2010, the average

monthly wage was 500.0 in year 2000, and 1000.0 in year 2010, then the percentage change per year = (1000.0-500.0)/500.0/(2010-2000)=10%.

- You should not fix the years but use the script to find the earliest and most recent data for a “country+currency” combination

Result:

```
(Croatia,YUM,443.03703703703707)
(Slovenia,YUM,270.3053191489362)
(Kyrgyzstan,KGS,222.73793490460156)
(Kazakhstan,KZT,31.397743055555555)
(Ghana,GHS,21.598425196850396)
(Armenia,AMD,21.327339787920703)
(Uzbekistan,UZS,19.316423357664235)
(Latvia,LVL,17.22450532724505)
(Costa Rica,CRC,2.1311838685578732)
(Peru,PEN,1.4129257958409989)
```

Task5.pig

SCRIPT:

The script is same as task 4 but with some more lines.
In the loop

```
country_year = foreach Group_Data_Country
  {
    Generate group.CountryArea,group.Currency,MAX(ordered_country.Year) as
max_year,MIN(ordered_country.Year) AS min_year;
  };
```

I find the min_year and the max_year per countryArea per Currency and then by joining this country_year with the preceding

```
country_count = foreach Group_Data_proj
  {
    Generate group.CountryArea,group.Year,group.Currency,AVG(Data_proj.Value) as avg;
  };
```

which cointains the avg per CountryArea per Year per Currency I take the related value.

```
max_year_per_country = JOIN country_year BY (CountryArea,max_year,Currency),country_count
by (CountryArea,Year,Currency);

max_year_proj = FOREACH  max_year_per_country GENERATE
country_year::CountryArea AS CountryArea,country_year::Currency AS Currency,
country_year::max_year AS max_year,country_count::avg as max_avg, country_year::min_year AS
min_year;

 max_n_min_year_per_country = JOIN max_year_proj BY
(CountryArea,min_year,Currency),country_count by (CountryArea,Year,Currency);

result = FOREACH   max_n_min_year_per_country GENERATE
max_year_proj::CountryArea AS CountryArea,max_year_proj::Currency as Currency,
(((max_year_proj::max_avg-country_count::avg)/country_count::avg)/(max_year_proj::max_year-
max_year_proj::min_year)) as percentage;
```

Following is the full Task5.pig script.

```
Data =LOAD 'UN_wage_data_with_currency.csv' using PigStorage(',') AS
(CountryArea,Year,Sex,Classification,Subclassification,Coverage,Scope,Source,SourceID,Value:do
uble,Currency,Value_Footnotes);

filtered_scope_week= FILTER Data by Scope=='Earnings per week';



Data_proj_week = FOREACH filtered_scope_week GENERATE CountryArea,
Year,Currency,Value*4;

filtered_scope_e = FILTER Data by Scope=='Earnings per month';
filtered_scope_w= FILTER Data by Scope=='Wage rates per month';

filtered_scope = UNION filtered_scope_e,filtered_scope_w;

Data_proj_m = FOREACH filtered_scope GENERATE CountryArea, Year,Currency,Value;

Data_proj =UNION Data_proj_m,Data_proj_week;



Group_Data_proj = group Data_proj by (CountryArea,Year,Currency);

country_count = foreach Group_Data_proj
  {
   Generate group.CountryArea,group.Year,group.Currency,AVG(Data_proj.Value) as avg;
  };

 ordered_country = ORDER country_count BY CountryArea ASC,Year DESC;



Group_Data_Country = group ordered_country by (CountryArea,Currency);

country_year = foreach Group_Data_Country
  {
   Generate group.CountryArea,group.Currency,MAX(ordered_country.Year) as
max_year,MIN(ordered_country.Year) AS min_year;
  };

max_year_per_country = JOIN country_year BY (CountryArea,max_year,Currency),country_count
by (CountryArea,Year,Currency);

max_year_proj = FOREACH  max_year_per_country GENERATE
country_year::CountryArea AS CountryArea,country_year::Currency AS Currency,
country_year::max_year AS max_year,country_count::avg as max_avg, country_year::min_year AS
min_year;

 max_n_min_year_per_country = JOIN max_year_proj BY
(CountryArea,min_year,Currency),country_count by (CountryArea,Year,Currency);

result = FOREACH   max_n_min_year_per_country GENERATE
max_year_proj::CountryArea AS CountryArea,max_year_proj::Currency as Currency,
(((max_year_proj::max_avg-country_count::avg)/country_count::avg)/(max_year_proj::max_year-
max_year_proj::min_year)) as percentage;

 ordered_result = ORDER result BY percentage  DESC;

 Top10_ordered_result  = LIMIT ordered_result 10;
```

```
dump  Top10_ordered_result;
```

The results will show the top 10 countryArea with with the highest percentage change in average monthly wage per year since their data began.

Part 3: Analysing Datasets of Your Choice

The dataset that I have choosen is the “Air B'nB' listing.csv Dataset” and could be downloaded at the following URL http://insideairbnb.com/get-the-data.html (listing.csv 9-apr-2021).

Airbnb, Inc. is an American company that operates an online marketplace for lodging, primarily homestays for vacation rentals, and tourism activities. Based in San Francisco, California, the platform is accessible via websiteand mobile app. Airbnb does not own any of the listed properties; instead, it profits by receiving commission from each booking.

The selected dataset provides summary information and metrics for listings in Amsterdam.

It provides the following columns

id

name

host_id

host_name

neighbourhood_group

neighbourhood

latitude

longitude

room_type

price

minimum_nights

,number_of_reviews

last_review

reviews_per_month

calculated_host_listings_count

availability_365

It is composed by over 8000 rows, each one describe the attributes of the places that are accessible for bookings in Amsterdam(Netherlands).

The first analysis that could give us useful information is about the average price per zone per night.

It could be useful to plan holidays, or to rent room or the entire house to rest after working meeting.

By having a list with the average price per zone we could improve our research by limiting the range of it without wasting time to search in places that are outside out budget.

ListingsTask1.pig

SCRIPT:

```
Data =LOAD 'listings.csv' using PigStorage(',') AS
(id,name,host_id,host_name,neighbourhood_group,neighbourhood,latitude:float,longitude:float,room_type,price:float,minimum_nights,number_of_reviews,last_review,reviews_per_month,calculated_host_listings_count,availability_365);

room_price = foreach Data

   Generate neighbourhood,room_type,price;

room_price_f = filter room_price by room_type=='Private room' or room_type=='Entire home/apt';


Group_Data_proj = group room_price_f by neighbourhood;
```

```
country_count = foreach Group_Data_proj
  {
    Generate group,flatten(AVG(room_price_f.price)) as avg;
  };



 ordered_country = ORDER country_count BY avg DESC;

dump ordered_country;
```

The script ListingsTask1.pig(which is the script of the first planned analysis) produce the following output :

```
Zone,average_price per night
(Centrum-West,181.75601374570448)
(Centrum-Oost,179.8977879481312)
(IJburg - Zeeburgereiland,174.87278106508876)
(Zuid,170.11713286713288)
(De Pijp - Rivierenbuurt,162.04112441436752)
(Oud-Noord,159.00210970464136)
(De Aker - Nieuw Sloten,158.7171717171717)
(Watergraafsmeer,154.08744394618833)
(Oud-Oost,152.16528066528068)
(Westerpark,150.93562610229276)
(De Baarsjes - Oud-West,148.87975174553918)
(Buitenveldert - Zuidas,141.11)
(Oostelijk Havengebied - Indische Buurt,131.79778393351802)
(Geuzenveld - Slotermeer,126.8010752688172)
(Noord-Oost,124.64622641509433)
(Noord-West,124.58781362007169)
(Bos en Lommer,124.4835039817975)
(Slotervaart,123.78070175438596)
(Osdorp,108.07920792079207)
```

```
(Gaasperdam - Driemond,107.09677419354838)
(Bijlmer-Oost,99.51851851851852)
(Bijlmer-Centrum,93.14285714285714).
```

The zone with the higher rank in this list have the most expensive rooms or apartments and they are located near the Amsterdam city Centre like Centrum-West.

The zone with the lowest rank is located in the suburbs of Amsterdam.

This analysis give a general idea of the average of the price per room or entire apartments per night,but for a single location, there a minimun night that the tourist has to book.

Then the second analysis that could help us while searching is the room type,neighbourhood and the real booking price average calculated by (price* minum_night to book).

This analysis,related to our budget could drive us to rent and entire house in a suburbs or only a private room in the city centre.

In the results we'll find, for one neighbourhood two rows, one with the real average price for the entire house and one with the real average price per room.

ListingsTask2.pig

SCRIPT:

```
Data =LOAD 'listings.csv' using PigStorage(',') AS (id,name,host_id,host_name,neighbourhood_group,neighbourhood,latitude:float,longitude:float,room_type,price:float,minimum_nights:int,number_of_reviews,last_review,reviews_per_month,calculated_host_listings_count,availability_365);



room_price_f = filter Data by room_type=='Private room' or room_type=='Entire home/apt';


room_price = foreach room_price_f


                        Generate neighbourhood,room_type,(price * minimum_nights) as real_price;
```

```
Group_Data_proj = group room_price by (neighbourhood,room_type);



country_count = foreach Group_Data_proj
  {
    Generate group.neighbourhood,group.room_type,flatten(AVG(room_price.real_price)) as avg;
  };


 ordered_country = ORDER country_count BY neighbourhood DESC;

dump ordered_country;
```

The script ListingsTask2.pig(which is the script of the first planned analysis) produce the following output :

```
neighbourhood,room_type,real_price
(Zuid,Entire home/apt,923.8740740740741)
(Zuid,Private room,330.7035175879397)
(Westerpark,Entire home/apt,539.7422266800402)
(Westerpark,Private room,341.3868613138686)
(Watergraafsmeer,Private room,242.9438202247191)
(Watergraafsmeer,Entire home/apt,561.8207282913165)
(Slotervaart,Private room,935.1607142857143)
(Slotervaart,Entire home/apt,988.6347826086957)
(Oud-Oost,Entire home/apt,648.726076555024)
(Oud-Oost,Private room,277.1746031746032)
(Oud-Noord,Entire home/apt,508.14835164835165)
(Oud-Noord,Private room,203.1090909090909)
(Osdorp,Entire home/apt,513.1935483870968)
```

```
(Osdorp,Private room,248.2051282051282)
(Oostelijk Havengebied - Indische Buurt,Entire home/apt,1160.6928446771378)
(Oostelijk Havengebied - Indische Buurt,Private room,894.751677852349)
(Noord-West,Entire home/apt,535.8585365853659)
(Noord-West,Private room,150.85135135135135)
(Noord-Oost,Entire home/apt,610.9230769230769)
(Noord-Oost,Private room,177.95652173913044)
(IJburg - Zeeburgereiland,Entire home/apt,679.3665338645418)
(IJburg - Zeeburgereiland,Private room,431.82758620689657)
(Geuzenveld - Slotermeer,Entire home/apt,558.1637931034483)
(Geuzenveld - Slotermeer,Private room,275.64285714285717)
(Gaasperdam - Driemond,Entire home/apt,2178.7)
(Gaasperdam - Driemond,Private room,367.39622641509436)
(De Pijp - Rivierenbuurt,Private room,555.5866261398177)
(De Pijp - Rivierenbuurt,Entire home/apt,1089.7085427135678)
(De Baarsjes - Oud-West,Private room,217.9749430523918)
(De Baarsjes - Oud-West,Entire home/apt,1087.4184198223468)
(De Aker - Nieuw Sloten,Private room,322.2093023255814)
(De Aker - Nieuw Sloten,Entire home/apt,571.6071428571429)
(Centrum-West,Entire home/apt,749.8996627318718)
(Centrum-West,Private room,266.45178571428573)
(Centrum-Oost,Entire home/apt,821.5740932642487)
(Centrum-Oost,Private room,252.88439306358381)
(Buitenveldert - Zuidas,Private room,215.27083333333334)
(Buitenveldert - Zuidas,Entire home/apt,1049.9473684210527)
(Bos en Lommer,Entire home/apt,439.4407894736842)
(Bos en Lommer,Private room,203.67226890756302)
(Bijlmer-Oost,Entire home/apt,521.0333333333333)
(Bijlmer-Oost,Private room,422.4117647058824)
(Bijlmer-Centrum,Entire home/apt,330.05)
(Bijlmer-Centrum,Private room,172.47727272727272)
```

The third and last analysis retrieves the top 10 customer reviewed rooms. An high number of reviews usually relate to a good cutoff between the place and the cost and other attirbutes that describes a room.

The results header of this analysis is :

ListingsTask3.pig

name,neighbourhood,room_type,price,minimum_nights,number_of_reviews

SCRIPT:

```
Data =LOAD 'listings.csv' using PigStorage(',') AS (id,name,host_id,host_name,neighbourhood_group,neighbourhood,latitude:float,longitude:float,room_type,price:float,minimum_nights,number_of_reviews:int,last_review,reviews_per_month,calculated_host_listings_count,availability_365);

room_price_f = filter Data by room_type=='Private room' or room_type=='Entire home/apt';

 ordered_listings = ORDER room_price_f BY number_of_reviews DESC;


top_ten_ordered = LIMIT ordered_listings  10;


top_ten_ordered_proj =foreach top_ten_ordered

   Generate name,neighbourhood,room_type,price,minimum_nights,number_of_reviews;

dump top_ten_ordered_proj;
```

The script ListingsTask3.pig(which is the script of the first planned analysis) produce the following output :

```
(The Backroom - Central private appt,Centrum-West,Entire home/apt,109.0,2,860)
(Sleeping in a unique ship in the center of A'dam!,Centrum-Oost,Private room,36.0,1,798)
(Amsterdam Houseboat 'Centre',Centrum-Oost,Entire home/apt,200.0,2,783)
(Amsterdam molen,Osdorp,Private room,93.0,1,772)
(Generator - Private 4 bed Room,Oud-Oost,Private room,117.0,1,707)
(Quiet room in Amsterdam Center,Centrum-West,Private room,70.0,1,702)
(HOUSEBOAT NOVA 80m2 + FREE BIKES,Zuid,Entire home/apt,98.0,1,691)
(B&B in de Amsterdamse Pijp,De Pijp - Rivierenbuurt,Private room,39.0,1,686)
(Rebel - Private Room,Centrum-West,Private room,122.0,1,659)
(BBBellamy,De Baarsjes - Oud-West,Private room,99.0,1,620)
```

The most reviewed rooms have in common the lower number of minimum_nights.

It translate in a total lower cost of the rent, so the customers seems to care a lot about the total cost, booking with more frequency cheap rooms, and comparing this top 10 results with the output of the first analysis we could see that this top 10 prices are under the calculated average per zone.