Technical validation of CoM dataset 2019

Contents

Introduction

The procedure to verify and improve the coherence of the dataset starts with the extraction of complete emission inventories stored in a PostgreSQL database. At the closing date of this study, (September 2019) 6,249 climate action plans with complete inventories had been submitted by cities in the EU27, EFTA countries and UK, Western Balkans, Eastern and Southern EU neighbourhoods. Inventories and other data are self-reported to the online platform and must accurately reflect the content of the official climate action plan %(SECAP) document. The SECAP document is a separated file, usually in PDF format and publicly available that represents the official action plan endorsed and signed by the local council.

The CoM dataset analysis

As a first step to address the quality of the data reported, yearly GHG emission per capita are plotted for each signatory. Let's first upload and reorganise the CoM dataset 2019.

```
%  Let's first upload and reorganise the CoM dataset 2019.



CoM_dataset2019_all=readtable('CoM_dataset2019.csv','delimiter',',');

T = CoM_dataset2019_all(:,{'emission_inventory_id','city'});  %'emission_inventory_id'
[G,total] = findgroups(T);
total.PC_emissions=splitapply(@sum,CoM_dataset2019_all.PC_emissions(rows),G);

EDGAR_RCO_TCO_2005=readtable('EDGAR_RCO_TCO_20052.csv','delimiter',',');

CoM_dataset2019=(CoM_dataset2019_all(:,1:7));
CoM_dataset2019 = removevars(CoM_dataset2019, 'emission_inventory_sector','type_of_emissio
ns');
CoM_dataset2019=unique(CoM_dataset2019);

% create the variable of activity and emission per capita in the
% Energy in buildings sector residential/municipal/istitutional, Transportation and Waste
sector
```

```
Error using tabular/dotParenReference (line 69)
Unrecognized table variable name 'PC_emissions'.

Error in TechnicalValidationDataset (line 24)
total.PC_emissions=splitapply(@sum,CoM_dataset2019_all.PC_emissions(rows),G);
```

```
CoM_dataset2019_all.activity_data(isnan(CoM_dataset2019_all.activity_data))=0;
CoM_dataset2019_all.emissions(isnan(CoM_dataset2019_all.emissions))=0;

CoM_dataset2019_all.PC_emissions=CoM_dataset2019_all.emissions;
CoM_dataset2019_all.PC_activity_data=CoM_dataset2019_all.activity_data./CoM_dataset2019_al
l.population_in_the_inventory_year;

rows=(CoM_dataset2019_all.emission_inventory_sector~={'Manufacturing and construction indu
stries'}& CoM_dataset2019_all.type_of_emission_inventory=={'baseline_emission_inventory'}&
 CoM_dataset2019_all.city~={'City or Greater city'});
% total2=CoM_dataset2019_all(rows,:);
CoM_dataset2019_all.PC_emissions(rows)=CoM_dataset2019_all.emissions(rows)./CoM_dataset201
9_all.population_in_the_inventory_year(rows);
CoM_dataset2019_all.PC_activity_data(rows)=CoM_dataset2019_all.activity_data(rows)./CoM_da
taset2019_all.population_in_the_inventory_year(rows);
```

```
rows=(CoM_dataset2019_all.emission_inventory_sector~={'Manufacturing and construction indu
stries'}& CoM_dataset2019_all.type_of_emission_inventory=={'baseline_emission_inventory'})
;
T = CoM_dataset2019_all(rows,{'emission_inventory_id','city'});  %'emission_inventory_id'
[G,total] = findgroups(T);
total.PC_emissions=splitapply(@sum,CoM_dataset2019_all.PC_emissions(rows),G);
total.PC_activity_data=splitapply(@sum,CoM_dataset2019_all.PC_activity_data(rows),G);

PC_emissions2=total.PC_emissions(total.PC_emissions<100);
```

```
createFit(PC_emissions2);
```

Extreme studentized deviate procedure - ESD

```
% In this section we describe the automatic routine implemented to detect and treat the
% outliers in inventories from small medium towns (number of inventories = (5,655 +43)
% covering a total population = 61.824 +0.729million of inhabitants).

% The procedure starts with dividing the data into two groups based on the normalization p
rocess:
% the activity data in the residential/municipal/institutional/tertiary buildings and tran
sport sector and Waste
% were normalised with the population size, whereas the activity data in manufacturing and
 construction industries
% were normalised with the GDP values.

% The method is based on a generalised ESD (extreme studentized deviate) procedure for the
 detection of
% abnormal energy consumptions

% The procedure starts considering the whole data set of GHG emissions per capita. The mea
n, the standard deviation,
% the skewness (the second, the third moment about the mean) are calculated at the beginni
ng for each set of data
```

```
CoM_dataset2019_all.PC_activity_data=CoM_dataset2019_all.activity_data;
rows=(CoM_dataset2019_all.emission_inventory_sector~={'Manufacturing and construction indu
stries'}& CoM_dataset2019_all.type_of_emission_inventory=={'baseline_emission_inventory'}&
```

```
 CoM_dataset2019_all.city~={'City or Greater city'});
CoM_dataset2019_all.PC_activity_data(rows)=CoM_dataset2019_all.activity_data(rows)./CoM_da
taset2019_all.population_in_the_inventory_year(rows);
```

```
rows=(CoM_dataset2019_all.emission_inventory_sector~={'Manufacturing and construction indu
stries'}& CoM_dataset2019_all.type_of_emission_inventory=={'baseline_emission_inventory'}&
 CoM_dataset2019_all.city~={'City or Greater city'});
T = CoM_dataset2019_all(rows,{'emission_inventory_id','city'});  %'emission_inventory_id'
[G,total] = findgroups(T);
total.population_in_the_inventory_year=splitapply(@max,CoM_dataset2019_all.population_in_t
he_inventory_year(rows),G);
total.PC_emissions=splitapply(@sum,CoM_dataset2019_all.PC_emissions(rows),G);
total.PC_activity_data=splitapply(@sum,CoM_dataset2019_all.PC_activity_data(rows),G);
total.emissions=splitapply(@sum,CoM_dataset2019_all.emissions(rows),G);
total.activity_data=splitapply(@sum,CoM_dataset2019_all.activity_data(rows),G);

total_initial=total;
% total=total_initial;
```

```
Median=median(total_initial.PC_activity_data);
Mean=mean(total_initial.PC_activity_data);
SDs=std(total_initial.PC_activity_data); %standard deviation
S=skewness(total_initial.PC_activity_data);%the skewness
```

```
Z=(total.PC_activity_data-mean(total.PC_activity_data))/std(total.PC_activity_data);
total.Z=Z;
n=length(total.PC_activity_data);

alpha=0.05;
p_upper=1-alpha./(2*(n-1));% the upper limit
p_low=alpha/(2*(n-1));
t_upper=tinv(p_upper, n-1-1); %this is the inverse of tstudent distribution
t_low=tinv(p_low, n-1-1); %this is the inverseof tstudent distribution

lambda_upper=t_upper*(n-1-1)/((((n-1)*(n-1-2+t_upper^2)))^0.5);
lambda_low=t_low*(n-1-1)/((((n-1)*(n-1-2+t_low^2)))^0.5);

rows=(total.Z>=lambda_upper);
out=total(rows,:);

rows=(total.Z<lambda_upper);
total=total(rows,:);
```

```
MeanRobust=mean(total.PC_activity_data);
MedianRobust=median(total.PC_activity_data);
SDsRobust=std(total.PC_activity_data); %standard deviation of the robust sample
S=skewness(total.PC_activity_data);%the skewness
```

Median Absolute Deviation (MAD)

```
% To conclude, also a non-parametric statistical procedure, i.e. the Median Absolute Devia
tion (MAD),
% has been applied to identify outliers in dataset that are non normal distributed.
```

```
% This method is more robust than the ESD, but less efficient, and its validity increases
% as data approach normal distribution. Similarly to the ESD, the choice of 5 as a critica
l
% value is motivated by the reasoning that if the observations other than outliers
% have an approximately normal distribution, it picks up as an outlier any observations
% more than about three standard deviations from the means


total2=total_initial;

Median2=median(total2.PC_activity_data);
total2.Z2=abs(total2.PC_activity_data-Median2);
MAD=median(abs(total2.PC_activity_data-Median2));
total2.Z3=abs(total2.PC_activity_data-Median2)/MAD;
% we have the same outlier idenified with the ESD and MAD with a critical of value of 8
```

Assessment of performance indicators

```
rows=(CoM_dataset2019_all.type_of_emission_inventory=={'baseline_emission_inventory'} & Co
M_dataset2019_all.city=={'City or Greater city'});
total_cities=CoM_dataset2019_all(rows,:);

T = total_cities(:,{'emission_inventory_id'});  %'emission_inventory_id'
[G,total_cities2] = findgroups(T);
total_cities2.population_in_the_inventory_year=splitapply(@max,total_cities.population_in_
the_inventory_year,G);
total_cities2.activity_data=splitapply(@sum,total_cities.activity_data,G);
total_cities2.emissions=splitapply(@sum,total_cities.emissions,G);

rows=(CoM_dataset2019_all.type_of_emission_inventory=={'baseline_emission_inventory'} & Co
M_dataset2019_all.city~={'City or Greater city'});
total_towns=CoM_dataset2019_all(rows,:);

T = total_towns(:,{'emission_inventory_id'});  %'emission_inventory_id'
[G,total_towns2] = findgroups(T);
total_towns2.population_in_the_inventory_year=splitapply(@max,total_towns.population_in_th
e_inventory_year,G);
total_towns2.activity_data=splitapply(@sum,total_towns.activity_data,G);
total_towns2.emissions=splitapply(@sum,total_towns.emissions,G);
sum(total_towns2.population_in_the_inventory_year)
total_towns3=innerjoin(total_towns2, total(:,1));

total3=union(total_cities2,total_towns3);

total3.PC_activity_data=total3.activity_data./total3.population_in_the_inventory_year;
total3.PC_emissions=total3.emissions./total3.population_in_the_inventory_year;
all_data_clean=total3;
Mean_clean=mean(total3.PC_emissions);
Median_clean=median(total3.PC_emissions);
SDs_clean=std(total3.PC_emissions); %standard deviation of the robust sample
S_clean=skewness(total3.PC_emissions);%the skewness


total_initial = removevars(total_initial, {'PC_emissions','PC_activity_data','city'});
sum(total_initial.population_in_the_inventory_year)+sum(total_cities2.population_in_the_in
ventory_year)

all_initial=union (total_cities2, total_initial);
all_initial.PC_emissions=all_initial.emissions./all_initial.population_in_the_inventory_ye
ar;
```

```
Mean_all=mean(all_initial.PC_emissions);
Median_all=median(all_initial.PC_emissions);
SDs_all=std(all_initial.PC_emissions); %standard deviation of the robust sample
S_all=skewness(all_initial.PC_emissions);%the skewness

mean(total_initial.PC_emissions);
CoM_dataset2019_clean=innerjoin(CoM_dataset2019_all,all_data_clean(:,1));

rows=(CoM_dataset2019_clean.type_of_emissions=={'direct_emissions'}&(CoM_dataset2019_clean
.emission_inventory_sector=={'Institutional/tertiary buildings and facilities'}|CoM_datase
t2019_clean.emission_inventory_sector=={'Residential buildings and facilities'}|CoM_datase
t2019_clean.emission_inventory_sector=={'Municipal buildings and facilities'}));
T = CoM_dataset2019_clean(rows,{'emission_inventory_id'});  %'emission_inventory_id'
[G,totalbd] = findgroups(T);
totalbd.emissions_BD=splitapply(@sum,CoM_dataset2019_clean.emissions(rows),G);% direct emi
ssion in buildings


rows=(CoM_dataset2019_clean.type_of_emissions=={'direct_emissions'}&CoM_dataset2019_clean.
emission_inventory_sector=={'Transportation'});
T = CoM_dataset2019_clean(rows,{'emission_inventory_id'});  %'emission_inventory_id'
[G,totaltd] = findgroups(T);
totaltd.emissions_TD=splitapply(@sum,CoM_dataset2019_clean.emissions(rows),G);% direct emi
ssion in transport

totald=outerjoin(totalbd,totaltd,'MergeKeys',true);
isnan()

totald.emissions_BD(isnan(totald.emissions_BD))=0;
totald.emissions_BD=totald.emissions_BD*10^-6;
totald.emissions_TD(isnan(totald.emissions_TD))=0;
totald.emissions_TD=totald.emissions_TD*10^-6;
```

Technical Validation with EDGAR

```
EDGAR_RCO_TCO_2005.RCO_LAU2_2005(isnan(EDGAR_RCO_TCO_2005.RCO_LAU2_2005))=0;
EDGAR_RCO_TCO_2005.TCO_LAU2_2005(isnan(EDGAR_RCO_TCO_2005.TCO_LAU2_2005))=0;
Tech_Valid=innerjoin (EDGAR_RCO_TCO_2005,CoM_dataset2019 );

rows=(Tech_Valid.RCO_LAU2_2005==0 & Tech_Valid.TCO_LAU2_2005==0);
Tech_Valid(rows,:)=[];

rows=(Tech_Valid.inventory_year==2005);
Tech_Valid2=Tech_Valid(rows,:);
Tech_Valid3=innerjoin(Tech_Valid2,totald);

Tech_Valid3.RSME_BUILD=(Tech_Valid3.RCO_LAU2_2005-Tech_Valid3.emissions_BD).^2;
RSME_BUILD=(sum(Tech_Valid3.RSME_BUILD)/length(Tech_Valid3.RCO_LAU2_2005))^0.5;
RSQ_BUILD=corr(Tech_Valid3.RCO_LAU2_2005,Tech_Valid3.emissions_BD);

Tech_Valid3.RSME_TRANSP=(Tech_Valid3.TCO_LAU2_2005-Tech_Valid3.emissions_TD).^2;
RSME_TRANSP=(sum(Tech_Valid3.RSME_TRANSP)/length(Tech_Valid3.RCO_LAU2_2005))^0.5;
RSQ_TRANSP=corr(Tech_Valid3.TCO_LAU2_2005,Tech_Valid3.emissions_BD);
```