

MODULE 1 – EVALUATION 2

DINH – NGUYEN – 10 OCTOBRE 2025

0. Import the data set – SAS

Comment : Before importing the file into SAS, we converted the CSV file into an XLSX format. In the French version of Excel, we noticed that when numbers use dots (.) as decimal separators, they are not recognized as numeric values but as text. To fix this issue, we replaced the dots with commas (,), which is the standard decimal separator in French Excel. After making these corrections, we proceeded to import the file into SAS to ensure the numeric variables were correctly interpreted.

```
libname Project 'C:\Users\nguyli22\Documents\SAS';
proc import datafile= 'C:\Users\nguyli22\Documents\SAS\temperat.xlsx'
    out=Project.temperat
    dbms=xlsx replace;
    getnames=yes;
run;
```

	City	Janvier	Février	Mars	Avril	Mai	Jun	Juillet	Août	Septembre	Octobre	Novembre	Décembre	Moyenne	Amplitude	Latitude	Longitude	Région
1	Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7	4.4	9.9	14.6	52.2	4.5 Ouest	
2	Athènes	9.1	9.7	11.7	15.4	20.1	24.5	27.4	27.2	23.8	19.2	14.6	11	17.8	18.3	37.6	23.5 Sud	
3	Berlin	-0.2	0.1	4.4	8.2	13.8	16	18.3	18	14.4	10	4.2	1.2	9.1	18.5	52.3	13.2 Ouest	
4	Bruxelles	3.3	3.3	6.7	8.9	12.8	15.6	17.8	17.8	15	11.1	6.7	4.4	10.3	14.4	50.5	4.2 Ouest	
5	Budapest	-1.1	0.8	5.5	11.6	17	20.2	22	21.3	16.9	11.3	5.1	0.7	10.9	23.1	47.3	19 Est	
6	Copenhague	-0.4	-0.4	1.3	5.8	11.1	15.4	17.1	16.6	13.3	8.8	4.1	1.3	7.8	17.5	55.4	12.3 Nord	
7	Dublin	4.8	5	5.9	7.8	10.4	13.3	15	14.6	12.7	9.7	6.7	5.4	9.3	10.2	53.2	6.1 Nord	
8	Helsinki	-5.8	-6.2	-2.7	3.1	10.2	14	17.2	14.9	9.7	5.2	0.1	-2.3	4.8	23.4	60.1	25 Nord	
9	Kiev	-5.9	-5	-0.3	7.4	14.3	17.8	19.4	18.5	13.7	7.5	1.2	-3.6	7.1	25.3	50.3	30.3 Est	
10	Cracovie	-3.7	-2	1.9	7.9	13.2	16.9	18.4	17.6	13.7	8.6	2.6	-1.7	7.7	22.1	50	19.6 Est	
11	Lisbonne	10.5	11.3	12.0	14.5	16.7	19.4	21.5	21.9	20.4	17.4	13.7	11.1	15.9	11.4	38.4	9.1 Sud	
12	Londres	3.4	4.2	5.5	8.3	11.9	15.1	16.9	16.5	14	10.2	6.3	4.4	9.7	13.5	51.4	0 Nord	
13	Madrid	5	6.6	9.4	12.2	16	20.8	24.7	24.3	19.8	13.9	8.7	5.4	13.9	19.7	40.2	3.4 Sud	
14	Minsk	-6.9	-6.2	-1.9	5.4	12.4	15.9	17.4	16.3	11.6	5.8	0.1	-4.2	5.5	24.3	53.5	27.3 Est	
15	Moscou	-9.3	-7.6	-2	6	13	16.6	18.3	16.7	11.2	5.1	-1.1	-6	5.1	27.6	45.2	1.5 Est	
16	Oslo	-4.3	-3.8	-0.6	4.4	10.3	14.9	16.9	15.4	11.1	5.7	0.5	-2.9	5.6	21.2	59.5	10.5 Nord	
17	Paris	3.7	3.7	7.3	9.7	13.7	16.5	19	18.7	16.1	12.5	7.3	5.2	11.2	49.5	2.2 Ouest		
18	Prague	-1.3	0.2	3.6	8.8	14.3	17.6	19.3	18.7	14.9	9.4	3.8	0.3	9.2	20.6	50	14.2 Est	
19	Reykjavik	-0.3	0.1	0.8	2.9	6.5	9.3	11.1	10.6	7.9	4.5	1.7	0.2	4.6	11.4	64.1	21.6 Nord	

0. Import the data set – R

```
> # Set the path
> path <- "C:/Users/Thinkpad/Desktop/SAS/draft hw 2/temperat.csv"
> print(path)
[1] "C:/Users/Thinkpad/Desktop/SAS/draft hw 2/temperat.csv"
>
> # Read the CSV with semicolon separator
> temperatures_df <- read.csv(
+   path,
+   sep = ";",
+   header = TRUE,
+   row.names = 1
+ )
>
> # View the data
> View(temperatures_df)
```

	Janvier	Février	Mars	Avril	Mai	Jun	Juillet	Août	Septembre	Octobre	Novembre
Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7
Athènes	9.1	9.7	11.7	15.4	20.1	24.5	27.4	27.2	23.8	19.2	14
Berlin	-0.2	0.1	4.4	8.2	13.8	16.0	18.3	18.0	14.4	10.0	4
Bruxelles	3.3	3.3	6.7	8.9	12.8	15.6	17.8	17.8	15.0	11.1	6
Budapest	-1.1	0.8	5.5	11.6	17.0	20.2	22.0	21.3	16.9	11.3	5
Copenhague	-0.4	-0.4	1.3	5.8	11.1	15.4	17.1	16.6	13.3	8.8	4
Dublin	4.8	5.0	5.9	7.8	10.4	13.3	15.0	14.6	12.7	9.7	6
Helsinki	-5.8	-6.2	-2.7	3.1	10.2	14.0	17.2	14.9	9.7	5.2	0
Kiev	-5.9	-5.0	-0.3	7.4	14.3	17.8	19.4	18.5	13.7	7.5	1
Cracovie	-3.7	-2.0	1.9	7.9	13.2	16.9	18.4	17.6	13.7	8.6	2
Lisbonne	10.5	11.3	12.0	14.5	16.7	19.4	21.5	21.9	20.4	17.4	13
Londres	3.4	4.2	5.5	8.3	11.9	15.1	16.9	16.5	14.0	10.2	6
Madrid	5.0	6.6	9.4	12.2	16.0	20.8	24.7	24.3	19.8	13.9	8
Minsk	-6.9	-6.2	-1.9	5.4	12.4	15.9	17.4	16.3	11.6	5.8	0
Moscou	-9.3	-7.6	-2.0	6.0	13.0	16.6	18.3	16.7	11.2	5.1	-1
Oslo	-4.3	-3.8	-0.6	4.4	10.3	14.9	16.9	15.4	11.1	5.7	0

1. Keep only the temperature columns into a dataset called TEMPERATURE – SAS

```
data project.temperatbis;
  set project.temperat(
    rename=(
      R_gion      = Region
      F_vrier     = Fevrier
      Ao_t        = Aout
      D_cembre    = Decembre
    )
  );
run;
```

Comment : As shown in the SAS log, SAS automatically replaces accented characters with underscores (_) when reading variable names from imported files. Consequently, “Région”, “Février”, “Août”, and “Décembre” become “R_gion”, “F_vrier”, “Ao_t”, and “D_cembre”. We then rename these variables to cleaner, more readable names (“Region”, “Fevrier”, “Aout”, “Decembre”) to keep naming consistent and make the rest of the project easier.

```
Journal - (Sans titre)
437  Libname Project 'C:\Users\nguy1i22\Documents\SAS';
NOTE: Libref PROJECT was successfully assigned as follows:
  Engine: V9
  Physical Name: C:\Users\nguy1i22\Documents\SAS
438  proc import datafile= 'C:\Users\nguy1i22\Documents\SAS\temperat.xlsx'
439    out=Project.temperat
440    dbms=xlsx replace;
441    getnames=yes;
442  run;

NOTE: Variable Name Change. Février -> F_vrier
NOTE: Variable Name Change. Août -> Ao_t
NOTE: Variable Name Change. Décembre -> D_cembre
NOTE: Variable Name Change. Région -> R_gion
```

```
data project.temperature;
  set project.temperatbis;
  drop moyenne amplitude latitude longitude region;
run;
```

	City	Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
1	Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7	4.4
2	Athènes	9.1	9.7	11.7	15.4	20.1	24.5	27.4	27.2	23.8	19.2	14.6	11
3	Berlin	-0.2	0.1	4.4	8.2	13.8	16	18.3	18	14.4	10	4.2	1.2
4	Bruxelles	3.3	3.3	6.7	8.9	12.8	15.6	17.8	17.8	15	11.1	6.7	4.4
5	Budapest	-1.1	0.8	5.5	11.6	17	20.2	22	21.3	16.9	11.3	5.1	0.7
6	Copenhague	-0.4	-0.4	1.3	5.8	11.1	15.4	17.1	16.6	13.3	8.8	4.1	1.3
7	Dublin	4.8	5	5.9	7.8	10.4	13.3	15	14.6	12.7	9.7	6.7	5.4
8	Helsinki	-5.8	-6.2	-2.7	3.1	10.2	14	17.2	14.9	9.7	5.2	0.1	-2.3
9	Kiev	-5.9	-5	-0.3	7.4	14.3	17.8	19.4	18.5	13.7	7.5	1.2	-3.6
10	Cracovie	-3.7	-2	1.9	7.9	13.2	16.9	18.4	17.6	13.7	8.6	2.6	-1.7
11	Lisbonne	10.5	11.3	12.8	14.5	16.7	19.4	21.5	21.9	20.4	17.4	13.7	11.1
12	Londres	3.4	4.2	5.5	8.3	11.9	15.1	16.9	16.5	14	10.2	6.3	4.4
13	Madrid	5	6.6	9.4	12.2	16	20.8	24.7	24.3	19.8	13.9	8.7	5.4
14	Minsk	-6.9	-6.2	-1.9	5.4	12.4	15.9	17.4	16.3	11.6	5.8	0.1	-4.2
15	Moscou	-9.3	-7.6	-2	6	13	16.6	18.3	16.7	11.2	5.1	-1.1	-6
16	Oslo	-4.3	-3.8	-0.6	4.4	10.3	14.9	16.9	15.4	11.1	5.7	0.5	-2.9
17	Paris	3.7	3.7	7.3	9.7	13.7	16.5	19	18.7	16.1	12.5	7.3	5.2
18	Prague	-1.3	0.2	3.6	8.8	14.3	17.6	19.3	18.7	14.9	9.4	3.8	0.3
19	Reykjavik	-0.3	0.1	0.8	2.9	6.5	9.3	11.1	10.6	7.9	4.5	1.7	0.2

1. Keep only the temperature columns into a dataset called TEMPERATURE – R

```
> # Keep only the temperature
columns.
> # The temperature columns are
the first 12 columns:
> # Janvier through Décembre (1-
12),
> TEMPERATURE <-
temperatures_df[, 1:12]
> View(TEMPERATURE)
```

	Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre
Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7
Athènes	9.1	9.7	11.7	15.4	20.1	24.5	27.4	27.2	23.8	19.2	14.6
Berlin	-0.2	0.1	4.4	8.2	13.8	16.0	18.3	18.0	14.4	10.0	4
Bruxelles	3.3	3.3	6.7	8.9	12.8	15.6	17.8	17.8	15	11.1	6
Budapest	-1.1	0.8	5.5	11.6	17.0	20.2	22.0	21.3	16.9	11.3	5
Copenhague	-0.4	-0.4	1.3	5.8	11.1	15.4	17.1	16.6	13.3	8.8	4
Dublin	4.8	5.0	5.9	7.8	10.4	13.3	15.0	14.6	12.7	9.7	6
Helsinki	-5.8	-6.2	-2.7	3.1	10.2	14.0	17.2	14.9	9.7	5.2	0
Kiev	-5.9	-5.0	-0.3	7.4	14.3	17.8	19.4	18.5	13.7	7.5	1
Cracovie	-3.7	-2.0	1.9	7.9	13.2	16.9	18.4	17.6	13.7	8.6	2
Lisbonne	10.5	11.3	12.8	14.5	16.7	19.4	21.5	21.9	20.4	17.4	13
Londres	3.4	4.2	5.5	8.3	11.9	15.1	16.9	16.5	14.0	10.2	6
Madrid	5.0	6.6	9.4	12.2	16.0	20.8	24.7	24.3	19.8	13.9	8
Minsk	-6.9	-6.2	-1.9	5.4	12.4	15.9	17.4	16.3	11.6	5.8	0
Moscou	-9.3	-7.6	-2.0	6.0	13.0	16.6	18.3	16.7	11.2	5.1	-1
Oslo	-4.3	-3.8	-0.6	4.4	10.3	14.9	16.9	15.4	11.1	5.7	0

2. Subset the data to only include the cities of Paris and Marseille into a dataset called TEMPERATURE_PM – SAS

```
data project.temperature_pm;
  set project.temperature;
  if city in ("Paris", "Marseille")
  then output;
run;
```

	City	Janvier	Février	Mars	Avril	Mai	Jun	Juillet	Août	Septembre	Octobre	Novembre	Décembre	
1	Paris	3.7	3.7	7.3	9.7	13.7	16.5	19	18.7	16.1	12.5	7.3	5.2	

2. Subset the data to only include the cities of Paris and Marseille into a dataset called TEMPERATURE_PM – R

```
# Filter only the rows where city is Paris or Marseille
temperature_pm <- subset(temperatures_df, rownames(temperatures_df) %in% c("Paris",
"Marseille"))
# View the result
view(temperature_pm)
```

	Janvier	Février	Mars	Avril	Mai	Jun	Juillet	Août	Septembre	Octobre	Novembre	Décembre	
Paris	3.7	3.7	7.3	9.7	13.7	16.5	19	18.7	16.1	12.5	7.3	5.2	

Comment : Here, the result only contains one row corresponding to Paris. This is expected, since in the original dataset there is no record for Marseille. Therefore, the query correctly returns only the existing city present in the data.

3. Calculate the mean temperature for all cities (average of the year) and for each month (average of all cities). Put the results in datasets MEAN_CITY and MEAN_MONTH – SAS

a. Mean_city

```
data project.mean_city;
  set project.temperature;
  moyenne_annuelle =
mean(janvier,fevrier,mars,avril,mai,juin,juillet,aout,septembre,octobre,novembre,decembre
);
  keep city moyenne_annuelle;
run;
```

	City	moyenne_annuelle		City	moyenne_annuelle
1	Amsterdam	9.8416666667	19	Prague	9.1333333333
2	Athènes	17.808333333	20	Reykjavik	4.6083333333
3	Berlin	9.0333333333	21	Rome	15.408333333
4	Bruxelles	10.283333333	22	Sarajevo	9.4666666667
5	Budapest	10.9416666667	23	Sofia	9.675
6	Copenhague	7.8333333333	24	Stockholm	5.875
7	Dublin	9.275	25	Anvers	10.225
8	Helsinki	4.7833333333	26	Barcelone	16.208333333
9	Kiev	7.0833333333	27	Bordeaux	12.7166666667
10	Cracovie	7.7833333333	28	Edimbourg	8.3333333333
11	Lisbonne	15.933333333	29	Francfort	9.7833333333
12	Londres	9.725	30	Genève	9.7166666667
13	Madrid	13.9	31	Gênes	16
14	Minsk	5.475	32	Milan	12.6666666667
15	Moscou	5.075	33	Palerm	17.608333333
16	Oslo	5.6333333333	34	Séville	18.2166666667
17	Paris	11.1166666667	35	St. Pétersbourg	4.5916666667
				Zurich	8.6916666667

```

/* Add the display format */
data project.mean_citybis;
  set project.mean_city;
  format moyenne_annuelle numx8.3;
run ;

```

Comment : By applying the numx8.3 format, we obtain properly formatted data, with values displayed using a comma as the decimal separator and three decimal places.

	City	moyenne_annuelle
1	Amsterdam	9,842
2	Athènes	17,808
3	Berlin	9,033
4	Bruxelles	10,283
5	Budapest	10,942
6	Copenhague	7,833
7	Dublin	9,275
8	Helsinki	4,783
9	Kiev	7,083

b. Mean_month

```

proc summary data=Project.TEMPERATURE;
  var janvier fevrier mars avril mai juin juillet aout septembre octobre novembre
decembre;
  output out=Project.MEAN_MONTH(drop=_TYPE_ _FREQ_) mean=;
run;

```

	Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
1	1.3457142857	2.2171428571	5.2285714286	9.2828571429	13.911428571	17.414285714	19.622857143	18.98	15.631428571	11.002857143	6.0657142857	2.88

```
/* Add the display format */
```

```

data Project.mean_monthbis;
  set Project.MEAN_MONTH;
  format janvier fevrier mars avril mai juin juillet aout septembre octobre novembre
decembre numx8.3;
run;

```

	Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
1	1,346	2,217	5,229	9,283	13,911	17,414	19,623	18,980	15,631	11,003	6,066	2,880

3. Calculate the mean temperature for all cities (average of the year) and for each month (average of all cities). Put the results in datasets MEAN_CITY and MEAN_MONTH – R

a. Mean_city

```

# Mean temperature per city
MEAN_CITY <- data.frame(
  Moyenne_Annuelle =
rowMeans(TEMPERATURE[, 1:12]),
  row.names = rownames(TEMPERATURE)
)
# Print results
print(MEAN_CITY)

```

	Moyenne_Annuelle
Amsterdam	9.841667
Athènes	17.808333
Berlin	9.033333
Bruxelles	10.283333
Budapest	10.941667
Copenhague	7.833333
Dublin	9.275000
Helsinki	4.783333
Kiev	7.083333
Cracovie	7.783333
Lisbonne	15.933333
Londres	9.725000
Madrid	13.900000
Minsk	5.475000
Moscou	5.075000
Oslo	5.633333
Paris	11.116667

b. Mean_month

```

# Mean temperature per month
MEAN_MONTH <- data.frame(
  Moyenne_Ville = colMeans(TEMPERATURE[, 1:12]),
  row.names = colnames(TEMPERATURE)[1:12]
)
# Print results
print(MEAN_MONTH)

```

	Moyenne_Ville
Janvier	1.345714
Février	2.217143
Mars	5.228571
Avril	9.282857
Mai	13.911429
Juin	17.414286
Juillet	19.622857
Août	18.980000
Septembre	15.631429
Octobre	11.002857
Novembre	6.065714
Décembre	2.880000

4. Sort the dataset TEMPERATURE by city (alphabetical order) – SAS

```
proc sort data=project.temperature
    out=project.temperature_sorted;
by city;
run;
```

	City	Janvier	Février	Mars	Avril	Mai	Jun	Juillet	Août	Septembre	Octobre	Novembre	Décembre
1	Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7	4.4
2	Anvers	3.1	2.9	6.2	8.9	12.9	15.5	17.9	17.6	14.7	11.5	6.8	4.7
3	Athènes	9.1	9.7	11.7	15.4	20.1	24.5	27.4	27.2	23.8	19.2	14.6	11
4	Barcelone	9.1	10.3	11.8	14.1	17.4	21.2	24.2	24.1	21.7	17.5	13.1	10
5	Berlin	-0.2	0.1	4.4	8.2	13.8	16	18.3	18	14.4	10	4.2	1.2
6	Bordeaux	5.6	6.7	9	11.9	15	18.3	20.4	20	17.6	13.5	8.5	6.1
7	Bruxelles	3.3	3.3	6.7	8.9	12.8	15.6	17.8	17.8	15	11.1	6.7	4.4
8	Budapest	-1.1	0.8	5.5	11.6	17	20.2	22	21.3	16.9	11.3	5.1	0.7
9	Copenhague	-0.4	-0.4	1.3	5.8	11.1	15.4	17.1	16.6	13.3	8.8	4.1	1.3
10	Cracovie	-3.7	-2	1.9	7.9	13.2	16.9	18.4	17.6	13.7	8.6	2.6	-1.7
11	Dublin	4.8	5	5.9	7.8	10.4	13.3	15	14.6	12.7	9.7	6.7	5.4
12	Edimbourg	2.9	3.6	4.7	7.1	9.9	13	14.7	14.3	12.1	8.7	5.3	3.7
13	Francfort	0.2	1.8	5.4	9.7	14.3	17.5	19	18.3	14.8	9.8	4.9	1.7
14	Genève	0.1	1.9	5.1	9.4	14.8	17.3	19.4	18.5	15	9.8	4.9	1.4
15	Gênes	8.7	9.7	11.4	13.8	17.5	21	24.5	24.6	21.8	17.8	12.2	10
16	Helsinki	5.6	6.2	2.7	3.1	10.2	14	17.2	14.9	9.7	5.2	0.1	2.3
17	Kiev	5.9	5	4.3	7.4	14.3	17.8	19.4	18.5	13.7	7.5	1.2	3.6
18	Ljubljane	10.5	11.3	12.8	14.5	16.7	19.4	21.5	21.9	20.4	17.4	13.7	11.1
19	Londres	3.4	4.2	5.5	8.3	11.9	15.1	16.9	16.5	14	10.2	6.3	4.4
20	Madrid	6	6.6	6.6	10.2	14.6	19.7	24.7	24.9	20.9	17.9	12.7	4.7

4. Sort the dataset TEMPERATURE by city (alphabetical order) – R

```
> # Sort the TEMPERATURE data frame alphabetically by city (row name).
> TEMPERATURE <- TEMPERATURE[order(rownames(TEMPERATURE)), ]
> View(TEMPERATURE)
```

	Janvier	Février	Mars	Avril	Mai	Jun	Juillet	Août	Septembre	Octobre	Novembre
Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7
Anvers	3.1	2.9	6.2	8.9	12.9	15.5	17.9	17.6	14.7	11.5	6.8
Athènes	9.1	9.7	11.7	15.4	20.1	24.5	27.4	27.2	23.8	19.2	14
Barcelone	9.1	10.3	11.8	14.1	17.4	21.2	24.2	24.1	21.7	17.5	13
Berlin	-0.2	0.1	4.4	8.2	13.8	16.0	18.3	18.0	14.4	10.0	4
Bordeaux	5.6	6.7	9.0	11.9	15.0	18.3	20.4	20.0	17.6	13.5	8
Bruxelles	3.3	3.3	6.7	8.9	12.8	15.6	17.8	17.8	15.0	11.1	6
Budapest	-1.1	0.8	5.5	11.6	17.0	20.2	22.0	21.3	16.9	11.3	5
Copenhague	-0.4	-0.4	1.3	5.8	11.1	15.4	17.1	16.6	13.3	8.8	4
Cracovie	-3.7	-2.0	1.9	7.9	13.2	16.9	18.4	17.6	13.7	8.6	2
Dublin	4.8	5.0	5.9	7.8	10.4	13.3	15.0	14.6	12.7	9.7	6
Edimbourg	2.9	3.6	4.7	7.1	9.9	13.0	14.7	14.3	12.1	8.7	5
Francfort	0.2	1.8	5.4	9.7	14.3	17.5	19.0	18.3	14.8	9.8	4
Gênes	8.7	9.7	11.4	13.8	17.5	21.0	24.5	24.6	21.8	17.8	12
Genève	0.1	1.9	5.1	9.4	13.8	17.3	19.4	18.5	15.0	9.8	4

Showing 1 to 16 of 35 entries, 14 total columns

5. Melt the data such that each city-month combination is in a separate row in the data set TEMP_CITY_MONTH – SAS

```
data project.temp_city_month;
  set project.temperature_sorted;
  /* Tableau contenant les 12 variables de mois */
  array m{12} janvier fevrier mars avril mai juin juillet aout septembre octobre novembre
decembre;
  /* Boucle DO pour créer une ligne par mois */
  do i=1 to 12;
    mois = vname(m{i});
    temperature = m{i};
    output;
  end;
  keep city mois temperature;
run;
```

	City	mois	temperature
1	Amsterdam	Janvier	2.9
2	Amsterdam	Février	2.5
3	Amsterdam	Mars	5.7
4	Amsterdam	Avril	8.2
5	Amsterdam	May	12.5
6	Amsterdam	June	14.8
7	Amsterdam	Juillet	17.1
8	Amsterdam	Août	17.1
9	Amsterdam	Septembre	14.5
10	Amsterdam	Octobre	11.4
11	Amsterdam	Novembre	7
12	Amsterdam	Décembre	4.4
13	Anvers	Janvier	3.1
14	Anvers	Février	2.9
15	Anvers	Mars	6.2
16	Anvers	Avril	8.9
17	Anvers	May	12.9
18	Anvers	June	15.5
19	Anvers	Juillet	17.9
20	Anvers	Août	17.6
21	Anvers	Septembre	14.7
22	Anvers	Octobre	11.5
23	Anvers	Novembre	6.8

```

/* Add the display format */
data project.temp_city_monthbis;
  set project.temp_city_month;
  format temperature numx8.1;
run ;

```

	City	mois	temperature
1	Amsterdam	Janvier	2.9
2	Amsterdam	Fevrier	2.5
3	Amsterdam	Mars	5.7
4	Amsterdam	Avril	8.2
5	Amsterdam	Mai	12.5
6	Amsterdam	Juin	14.8
7	Amsterdam	Juillet	17.1
8	Amsterdam	Aout	17.1
9	Amsterdam	Septembre	14.5

5. Melt the data such that each city-month combination is in a separate row in the data set TEMP_CITY_MONTH – R

```

library(reshape2)
# Add rownames as a column
TEMPERATURE$City <- rownames(TEMPERATURE)
# Melt to long format
TEMP_CITY_MONTH <- melt(
  TEMPERATURE,
  id.vars = "City",
  variable.name = "Month",
  value.name = "Temperature"
)
View(TEMP_CITY_MONTH)

```

	City	Month	Temperature
1	Amsterdam	Janvier	2.9
2	Anvers	Janvier	3.1
3	Athènes	Janvier	9.1
4	Barcelone	Janvier	9.1
5	Berlin	Janvier	-0.2
6	Bordeaux	Janvier	5.6
7	Bruxelles	Janvier	3.3
8	Budapest	Janvier	-1.1
9	Copenhague	Janvier	-0.4

6. Bonus question – SAS

```

data project.bonus;
  set project.temperature_sorted;
  /* 1) Define the arrays:
     - m : monthly values of the current row
     - s : cumulative sums (carried over from one city to another)
     - a_ : cumulative averages to output */
  array m{12} janvier fevrier mars avril mai juin juillet aout septembre octobre novembre
decembre;
  array s{12} sjanvier sfevrier smars savril smai sjuin sjuillet saout sseptembre
soctobre snovembre sdecembre;
  array a{12} a_janvier a_fevrier a_mars a_avril a_mai a_juin a_juillet a_aout
a_septembre a_octobre a_novembre a_decembre;
  /* 2) Keep N and cumulative sums from one observation to the next */
  retain N sjanvier sfevrier smars savril smai sjuin sjuillet saout sseptembre soctobre
snovembre sdecembre;
  /* 3) Initialize at the very beginning (first city) */
  if _N_ = 1 then do i = 1 to 12;
    s[i] = 0;
  end;
  if _N_ = 1 then N = 0;
  /* 4) Increment the number of cities considered */
  N + 1;
  /* 5) Update the cumulative sums, then calculate the averages = sum / N */
  do i = 1 to 12;
    if not missing(m[i]) then s[i] + m[i];
    a[i] = s[i] / N;
  end;
  /* 6) Output one line for each value of N */
  N_First_Cities = N;
  keep N_First_Cities a_:; /* Keep N and all average variables a_* */
  output;
run;

```

VIEWTABLE: Project.Bonus														
	a_janvier	a_fevrier	a_mars	a_avril	a_mai	a_juin	a_juillet	a_aout	a_septembre	a_octobre	a_novembre	a_decembre	N_First_Cities	
1	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7	4.4	1	
2	3	2.7	5.95	8.55	12.7	15.15	17.5	17.35	14.6	11.45	6.9	4.55	2	
3	5.0333333333	5.0333333333	7.8666666667	10.8333333333	15.1666666667	18.2566666667	20.8	20.6333333333	17.6666666667	14.0333333333	9.4666666667	6.7	3	
4	6.05	6.35	8.85	11.65	15.725	19	21.65	21.5	18.675	14.9	10.375	7.525	4	
5	4.8	5.1	7.96	10.96	15.34	18.4	20.98	20.8	17.82	13.92	9.14	6.26	5	
6	4.9333333333	5.3666666667	8.1333333333	11.1666666667	15.2833333333	18.3833333333	20.8833333333	20.6666666667	17.7833333333	13.85	9.0333333333	6.2333333333	6	
7	4.7	5.0714285714	7.9285714286	10.8	14.928571429	17.985714296	20.442857143	20.257142857	17.385714286	13.457142857	8.7	5.9714285714	7	
8	3.975	4.5375	7.625	10.9	15.1875	18.2625	20.6375	20.3875	17.325	13.1875	8.25	5.3125	8	
9	3.4888888889	3.9888888889	6.9222222222	10.3333333333	14.7333333333	17.9444444444	20.2444444444	19.9666666667	16.877777778	12.7	7.7888888889	4.8666666667	9	
10	2.77	3.39	6.42	10.09	14.58	17.84	20.06	19.73	16.56	12.29	7.27	4.21	10	
11	2.9545454545	3.5363636364	6.3727272727	9.8818181818	14.2	17.427272727	19.6	19.263636364	16.209090909	12.054545455	7.2181818182	4.3181818182	11	
12	2.95	3.5416666667	6.2333333333	9.65	13.8416666667	17.0583333333	19.1916666667	18.85	15.8666666667	11.75	7.0583333333	4.2666666667	12	
13	2.7384615385	3.4076923077	6.1692307692	9.6538461538	13.876923077	17.092307692	18.807692308	15.784615385	11.623076923	6.6923076923	4.0692307692	13		
14	2.55	3.3	6.0928571429	9.6357142857	13.8771428571	17.107142857	19.192857143	18.785714286	15.785714286	11.492857143	6.75	3.8785714286	14	
15	2.96	3.66	6.4466666667	9.9133333333	14.1133333333	17.3666666667	19.5466666667	19.1733333333	16.1333333333	11.9133333333	7.1133333333	4.2866666667	15	
16	2.4125	3.04735	5.875	9.4875	13.98675	17.15625	19.4	18.50625	15.73125	11.49375	6.675	3.875	16	
17	1.9235294118	2.570582535	5.5117647059	9.3647058254	13.894117647	17.194117647	19.4	18.82352941	15.611764706	11.258023529	6.3524411765	3.4352441176	17	
18	2.4	3.0555555556	5.9166666667	9.65	14.05	17.3166666667	19.5166666667	19.05	15.877777778	11.6	6.7611111111	3.8611111111	18	
19	2.4526315789	3.1157894737	5.8947368421	9.5789473684	13.936842105	17.2	19.378947368	18.915789474	15.778947368	11.526315789	6.7368421053	3.8894736842	19	
20	2.58	3.29	6.07	9.71	14.04	17.38	19.645	19.185	15.98	11.645	6.835	3.965	20	
21	2.5095230095	3.3047619048	6.1619047619	9.8476190476	14.195230095	17.5666666667	19.842857143	19.35714257	16.119047619	11.71425714	6.6380952381	3.9	21	

/* Add the display format */

```

data project.bonusbis;
  set project.bonus;
  format a_janvier a_fevrier a_mars a_avril a_mai a_juin a_juillet a_aout a_septembre
a_octobre a_novembre a_decembre numx8.1;
run;

```

	a_janvier	a_fevrier	a_mars	a_avril	a_mai	a_juin	a_juillet	a_aout	a_septembre	a_octobre	a_novembre	a_decembre	N_First_Cities	
1	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7.0	4.4	1	
2	3.0	2.7	6.0	8.6	12.7	15.2	17.5	17.4	14.6	11.5	6.9	4.6	2	
3	5.0	5.0	7.9	10.8	15.2	18.3	20.8	20.6	17.7	14.0	9.5	6.7	3	
4	6.1	6.4	8.9	11.7	15.7	19.0	21.7	21.5	18.7	14.9	10.4	7.5	4	
5	4.8	5.1	8.0	11.0	15.3	18.4	21.0	20.8	17.8	13.9	9.1	6.3	5	
6	4.9	5.4	8.1	11.1	15.3	18.4	20.9	20.7	17.8	13.9	9.0	6.2	6	
7	4.7	5.1	7.9	10.8	14.9	18.0	20.4	20.3	17.4	13.5	8.7	6.0	7	
8	4.0	4.5	7.6	10.9	15.2	18.3	20.6	20.4	17.3	13.2	8.3	5.3	8	
9	3.5	4.0	6.9	10.3	14.7	17.9	20.2	20.0	16.9	12.7	7.8	4.9	9	
10	2.8	3.4	6.4	10.1	14.6	17.8	20.1	19.7	16.6	12.3	7.3	4.2	10	
11	3.0	3.5	6.4	9.9	14.2	17.4	19.6	19.3	16.2	12.1	7.2	4.3	11	
12	3.0	3.5	6.2	9.7	13.8	17.1	19.2	18.9	15.9	11.8	7.1	4.3	12	

6. Bonus question – R

```
# Number of cities
n_cities <- nrow(TEMPERATURE)
# Calculate cumulative mean per month
CUMULATIVE_MEAN <- as.data.frame(
  apply(TEMPERATURE[, 1:12], 2, cumsum) / (1:n_cities)
)
# Set row and column names
rownames(CUMULATIVE_MEAN) <- paste0("N=", 1:n_cities)
colnames(CUMULATIVE_MEAN) <- colnames(TEMPERATURE)[1:12]
View(CUMULATIVE_MEAN)
```

Screenshot of RStudio showing the 'CUMULATIVE_MEAN' data frame. The table has 16 rows (N=1 to N=16) and 12 columns (Janvier to Décembre). The data shows cumulative mean temperatures over 12 months for 16 different locations.

	Janvier	Février	Mars	Avril	Mai	Juin	JUILLET	Août	Septembre	Octobre	Novembre	Décembre
N=1	2.900000	2.500000	5.700000	8.200000	12.50000	14.80000	17.10000	17.10000	14.50000	11.40000	7.000000	
N=2	3.000000	2.700000	5.950000	8.550000	12.70000	15.15000	17.50000	17.35000	14.60000	11.45000	6.900000	
N=3	5.033333	5.033333	7.866667	10.833333	15.166667	18.266667	20.800000	20.633333	17.666667	14.033333	9.466667	
N=4	6.050000	6.350000	8.850000	11.650000	15.72500	19.00000	21.65000	21.50000	18.67500	14.90000	10.375000	
N=5	4.800000	5.100000	7.960000	10.960000	15.34000	18.40000	20.98000	20.80000	17.82000	13.92000	9.140000	
N=6	4.933333	5.366667	8.133333	11.116667	15.28333	18.38333	20.88333	20.666667	17.78333	13.85000	9.033333	
N=7	4.700000	5.071429	7.928571	10.800000	14.92857	17.98571	20.44286	20.25714	17.38571	13.45714	8.700000	
N=8	3.975000	4.537500	7.625000	10.900000	15.18750	18.26250	20.63750	20.38750	17.32500	13.18750	8.250000	
N=9	3.488889	3.988889	6.922222	10.333333	14.73333	17.94444	20.24444	19.966667	16.87778	12.70000	7.788889	
N=10	2.770000	3.390000	6.420000	10.090000	14.58000	17.84000	20.06000	19.73000	16.56000	12.29000	7.270000	
N=11	2.954545	3.536364	6.372727	9.881818	14.20000	17.42727	19.60000	19.26364	16.20909	12.05455	7.218182	
N=12	2.950000	3.541667	6.233333	9.650000	13.84167	17.05833	19.19167	18.85000	15.866667	11.77500	7.058333	
N=13	2.738462	3.407692	6.169231	9.653846	13.87692	17.09231	19.17692	18.80769	15.78462	11.62308	6.892308	
N=14	3.164286	3.785714	6.542857	9.950000	14.13571	17.37143	19.55714	19.22143	16.21429	12.06429	7.271429	
N=15	2.960000	3.660000	6.446667	9.913333	14.11333	17.36667	19.54667	19.17333	16.13333	11.91333	7.113333	
N=16	2.412500	3.043750	5.875000	9.487500	13.86875	17.15625	19.40000	18.90625	15.73125	11.49375	6.675000	

Showing 1 to 16 of 35 entries, 12 total columns