

tlse_3scen_2006-2100

July 24, 2019



Cahier d'exercices pour l'enseignement du changement climatique ou l'apprentissage de programmation issu de la collection "Climat et météo tremplin pour l'enseignement des sciences" (PIA IFÉ ENS de Lyon - Météofrance ENM Toulouse). Le dispositif clef en main repose sur l'utilisation d'une RaspberryPi chargée avec le système d'exploitation Debian enrichi, produit par le projet. Les sources et les exécutables sont accessibles dans [l'espace collaboratif du projet à l'IFÉ ENS de Lyon](#) et une copie se trouve dans [l'espace collaboratif de la forge github](#); plus d'information sur les [blogs d'accompagnement](#) systèmes d'exploitation sur [la page des OS de Raspberries Pi](#). Toutes les ressources issues du projet sont fournies sous licence [Creative Commons](#) ou sous les licences libres d'origine des outils utilisés.

Les ressources du projet **peuvent être utilisées dans tout autre environnement compatible**, notamment tous les cahiers d'exercices peuvent être exécutés sur toute machine disposant d'un python3 et des bibliothèques jupyter, jupyterlab, numpy, netcdf4.

Les données *pré-traitées* utilisées ci-dessous sont **accessibles en ligne** sur le [serveur de données géolocalisées](#) opendap du projet tremplin.

Auteur : G. Vidal



licence : Creative Commons

1 Une approche des enseignements autour du changement climatique : mitigation et adaptation au changement

2 Approche du changement climatique à Toulouse : évolution des températures maximales, minimales et de la pluviométrie jusqu'en 2100

Ce cahier d'exercices utilise les données collectées par le projet Tremplin sur la ville de Toulouse et ses environs à partir du [projet DRIAS](#), converties pré-traitées et mises en ligne sur le [serveur de données climatologiques](#) du projet Tremplin des sciences. Les conversions et pré-traitements ont été réalisés avec les cahiers de programme ipython des phases 1 & 2 de cette série disponibles sur la [forge du projet](#). Ce cahier propose plusieurs voies d'exploration du jeu de données *température max / température min / pluviométrie* sur une grille de 104 x 104 km autour de Toulouse.

L'étude aborde l'évolution des variables moyennées sur N années (N a été fixé à 30 par défaut mais peut être modifié) sur 4 noeuds centrés sur la ville de Toulouse.

Le lot utilisé est issu des trois modélisation RCP 2.6 4.5 et 8.5 fournies par météoFrance. Ce cahier est immédiatement opérationnel et peut directement être exécuté sur jupyter ou jupyterlab, toutefois il manipule des données multidimensionnelles et doit être réservé à des étudiants avancés si l'on souhaite manipuler le code. Par contre l'utilisation des blocs concernant le tracé des courbes ou des cartes est d'un usage facile d'accès et permet d'obtenir simplement des figures. Attention le dessin des figures a été désactivé dans le dépôt sur la forge pour ne pas alourdir inutilement le fichier transféré, il suffit de décommenter la dernière ligne de chacun des blocs de dessin en enlevant le ".

2.1 Préparation de l'environnement et ouverture du fichier de données

Importer d'abord le module netcdf4 et numpy, attention les majuscules sont impératives pour le nom netCDF4. Ces deux modules permettent de traiter les fichiers multidimensionnels au format netCDF utilisés dans le monde de la météorologie et de l'océanographie principalement.

```
In [1]: import netCDF4 as nc
import numpy as np
from datetime import datetime
from array import array
import sys, datetime, os
```

Importer ensuite les données de sortie de modèle depuis le fichier obtenu auprès du [serveur de données climatologiques](#) du projet Tremplin des sciences extrait du site [DRIAS](#) sur la région toulousaine.

L'exemple utilisé ici a été réalisé avec une grille de 13 x 13 noeuds centrés sur la ville de Toulouse, pour obtenir un jeu de données se reporter au manuel numérique réalisé par Éric le Jan et Carole Larose dans le cadre du projet "Climat et Météo Tremplin pour l'enseignement des sciences". Les instructions d'affichage commentées (pour la plupart) permettent de vérifier les propriétés du fichier obtenu ainsi que les variables qui pourront être utilisées. Ces affichages sont facultatifs et peuvent être commentés sans conséquence pour la suite.

```
In [2]: dataSetTlse = nc.Dataset('http://geoloc-tremplin.ens-lyon.fr/climato-data/Toulouse-1/t.
# La ligne ci-dessous permet d'exploiter des données locales si l'utilisateur réalise lui-m
#dataSetLyon = nc.Dataset('t_min-t_max-rstr_thisrun_26-45-85.nc') #to use local file
```

```

print('Description des données issues du modèle : \n',dataSetTlse,'\n')
print('Variables disponibles : ',dataSetTlse.variables.keys()) # get all variable names
#print('Taille du tableau tasmx : ',dataSetTlse.variables['tasmx'].shape ,'\n')

```

Description des données issues du modèle :

```

<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3_CLASSIC data model, file format DAP2):
  title: Extrait TSMx par moyenne mensuelle de 2006 a 2100 Lyon et sa region
  institution: ENS de Lyon
  institute_id: IFE Institut Francais de l Education
  project_id: Climat et meteo tremplin pour l enseignement des sciences
  model_id: CNRM-ALADIN52
  product: output derived from MeteoFrance DRIAS data
  contact: gerard.vidal@ens-lyon.fr
  creation_date: 2019-07-23 19:49:02.710491
  driving_experiment_name: DRIAS2014
  experiment: RCP2.6 RCP4.5 RCP8.5
  model: ALADIN-Climat
  author: Gerard Vidal
  comment: Extraction des moyennes de la region Lyonnaise de 2006 a 2100 et changement des
  dimensions(sizes): i(13), j(13), maxStrlen64(64), month(13), year(95)
  variables(dimensions): int32 i(i), int32 j(j), |S1 month(month,maxStrlen64), int32 year(year)
  groups:

```

Variables disponibles : `odict_keys(['i', 'j', 'month', 'year', 'lat', 'lon', 'x', 'y', 't_max_2100'])`

2.2 Liste des dimensions et des variables du système de données

À partir de la liste des variables obtenue ci-dessus on renomme les jeux de données de chacune des variables qui seront exploitées apour effectuer les calculs et contrôler la taille des échantillons. Les affichages proposés permettent de contrôler que les paramètres présents sont effectivement ceux qui sont attendus.

```

In [3]: for dim in dataSetTlse.dimensions.items():
        print(dim)
        print ('Variables \t Forme \t\t Taille \t type : \n')
        for var in dataSetTlse.variables.keys() :
            print (var, '\t\t',
                    dataSetTlse.variables[var].dimensions, '\t\t',
                    dataSetTlse.variables[var].shape, '\t',
                    dataSetTlse.variables[var].dtype)

('i', <class 'netCDF4._netCDF4.Dimension'>: name = 'i', size = 13
)
('j', <class 'netCDF4._netCDF4.Dimension'>: name = 'j', size = 13
)
('maxStrlen64', <class 'netCDF4._netCDF4.Dimension'>: name = 'maxStrlen64', size = 64
)

```

```

)
('month', <class 'netCDF4._netCDF4.Dimension'>: name = 'month', size = 13
)
('year', <class 'netCDF4._netCDF4.Dimension'>: name = 'year', size = 95
)

```

Variables	Forme	Taille	type :
i	('i',)	(13,)	int32
j	('j',)	(13,)	int32
month	('month', 'maxStrlen64')		(13, 64) S1
year	('year',)	(95,)	int32
lat	('j', 'i')	(13, 13)	float32
lon	('j', 'i')	(13, 13)	float32
x	('i',)	(13,)	int32
y	('j',)	(13,)	int32
t_max_26	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
t_max_45	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
t_max_85	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
t_min_26	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
t_min_45	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
t_min_85	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
rstr_26	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
rstrc_26	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
rstr_45	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
rstrc_45	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
rstr_85	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
rstrc_85	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
delta_t_26	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
delta_t_45	('year', 'month', 'j', 'i')		(95, 13, 13, 13)
delta_t_85	('year', 'month', 'j', 'i')		(95, 13, 13, 13)

2.3 Création des tableaux de calcul

```

In [31]: Tlse_tmax26 = dataSetTlse.variables['t_max_26'] # variable temperature
         Tlse_tmax45 = dataSetTlse.variables['t_max_45'] # variable temperature
         Tlse_tmax85 = dataSetTlse.variables['t_max_85'] # variable temperature
         Tlse_tmin26 = dataSetTlse.variables['t_min_26'] # variable temperature
         Tlse_tmin45 = dataSetTlse.variables['t_min_45'] # variable temperature
         Tlse_tmin85 = dataSetTlse.variables['t_min_85'] # variable temperature
         Tlse_rstr26 = dataSetTlse.variables['rstr_26']  # variable rainfall
         Tlse_rstr45 = dataSetTlse.variables['rstr_45']  # variable rainfall
         Tlse_rstr85 = dataSetTlse.variables['rstr_85']  # variable rainfall
         Tlse_rstrc26 = dataSetTlse.variables['rstrc_26'] # variable cumulated rainfall
         Tlse_rstrc45 = dataSetTlse.variables['rstrc_45'] # variable cumulated rainfall
         Tlse_rstrc85 = dataSetTlse.variables['rstrc_85'] # variable cumulated rainfall
         Tlse_delta26 = dataSetTlse.variables['delta_t_26'] # variable tmax -tmin daily
         Tlse_delta45 = dataSetTlse.variables['delta_t_45'] # variable tmax -tmin daily

```

```

Tlse_delta85 = dataSetTlse.variables['delta_t_85'] # variable tmax -tmin daily
Tlse_month = nc.chartostring(dataSetTlse.variables['month'][:]) # variable temps
# use the following syntax on local files
# Tlse_month = dataSetTlse.variables['month'] #if local file type = string
Tlse_year = dataSetTlse.variables['year'] # variable temps

#test = nc.chartostring(dataSetTlse.variables['month'][:])
#print(test)
#test[:] = nc.chartostring(dataSetTlse.variables['month'],encoding='utf-8')
#print(test)

Tlse_lat,Tlse_lon = dataSetTlse.variables['lat'], dataSetTlse.variables['lon'] # lat
Tlse_x,Tlse_y = dataSetTlse.variables['x'], dataSetTlse.variables['y'] # coordonnées
Tlse_gridi,Tlse_gridj = dataSetTlse.variables['i'], dataSetTlse.variables['j'] # coord

lenmonths = Tlse_month.shape[0]

#print ('\n Taille des tableaux de calcul : \n',
#       '\ntmax26 : ', Tlse_tmax26.shape, Tlse_tmax26[44,7,5,5],dataSetTlse.variables
#       '\ntmax45 : ', Tlse_tmax45.shape, Tlse_tmax45[44,7,5,5],dataSetTlse.variables
#       '\ntmax85 : ', Tlse_tmax85.shape, Tlse_tmax85[44,7,5,5],dataSetTlse.variables
#       '\n\ntyear : ', Tlse_year.shape, Tlse_year[:],
#       '\n\ntmonth : ', Tlse_month.shape, Tlse_month[:],
#       '\n\ntLat : ', Tlse_lat.shape, Tlse_lat[0,:],
#       '\n\ntLon : ', Tlse_lon.shape, Tlse_lon[0,:],
#       '\n\ntX : ', Tlse_x.shape, Tlse_x[:],
#       '\ntY : ', Tlse_y.shape, Tlse_y[:],
#       '\n\nti : ', Tlse_gridi.shape, Tlse_gridi[:],
#       '\ntj : ', Tlse_gridj.shape, Tlse_gridj[:])

```

Définition et affectation des variables où sont copiées les données conservées et où seront stockés les résultats des calculs. Les années seront calculées pendant le calcul principal, les affichages permettent de vérifier la validité des données utilisées.

2.4 Calcul principal des moyennes par mois pour chaque noeud et toutes les années

Le premier bloc de code permet de fixer les paramètres qui seront utilisés pour les calculs. les commentaires donnent des indications sur les valeurs possibles.

2.4.1 Calculs pour une seule période de yearInterval années

Calcul de la moyenne de températures d'une sélection de mois sur un intervalle de yearInterval années à partir de l'année yearBegin sur les noeuds allant de (startj,starti) de taille (intervalj,intervali)

préparation de la création de figures On importe les bibliothèques plotly

```

In [6]: import plotly.offline as py
import plotly.graph_objs as go

```

```

from plotly import tools
import plotly.io as pio
py.init_notebook_mode(connected=True)

```

2.5 Courbes d'évolution de la température de 2036 à 2100

2.5.1 Calculs pour UNE SÉRIE DE PÉRIODES de yearInterval années

Calcul de la moyenne de températures des mois de calcMonth (il peut y en avoir un seul ou un choix), sur yearInterval années à partir de l'année yearBegin jusqu'à l'année yearBegin + yearPeriod sur les noeuds à partir de (startj, starti) de taille (intervalj, intervali). La valeur trouvée est affectée à la dernière année de la plage de calcul.

```

In [20]: # Variables for the computation
# Séquence d'années pendant lesquelles les calculs sont effectués
# * Choix de l'année de départ : yearBegin
# * Choix du nombre d'années sur lesquelles est effectuée la moyenne glissante : yearPeriod
# * Choix de l'intervalle de calcul yearPeriod
# * Choix des mois choisis pour le calcul 0 = Jan; 7 = Aout ...
# yearbegin + yearperiod must be <= 2101
driasOrigin = 2006
driasEnd = 2100
nbYears = driasEnd - driasOrigin + 1
# ===== Choice of parameters below =====
yearBegin = 2006
yearInterval = 30
yearPeriod = 64
calcMonth = [0, 7, 12] # Calcul effectué pour Janvier (0) Aout (7) et moyenne annuel
lencalcMonths = len(calcMonth)
# Grille i j
# sous espace de la grille 10X10 utilisé
# le premier échantillon est un carré 2X2 au centre de la grille
# Le second échantillon est toute la grille entière
starti = 5
intervali = 3
startj = 5
intervalj = 3 # Calcul sur un carré de 16km de côté au centre de la carte (ville de ...
# ===== END of Choice =====
#starti = 0
#intervali = 10
#startj = 0
#intervalj = 10
# Variables de calcul
startYear = yearBegin - driasOrigin
endYear = startYear + yearPeriod
#print(yearPeriodInterval)
endi = starti + intervali
endj = startj + intervalj

```

```

if not ((startYear >= 0) and (yearBegin + yearPeriod + yearInterval) <= driasEnd):
    print('starting year or finishing year out of bounds')
    sys.exit('giving up on year bounds')
#print(gridj[loc_j])
#print(gridi[loc_i])
#print(startYear, ': ', endYear, ', ', calcMonth, ', ', startj, ': ', endj, ', ', starti, ': ', endi)
#print(temp.shape)
#print(np.mean(temp[startYear:endYear, calcMonth, startj:endj, starti:endi]))

```

2.5.2 Calcul pour n mois sur p périodes de q années pour la zone

Calcul de la moyenne des températures mensuelles sur yearInterval ans pendant une période de yearPeriod années successives pour une sélection de calcMonth mois.

```

In [23]: # Using online file takes a little more time
# Calculate the number of years yielding a result
moyMAreaInterval_26 = np.zeros((lencalcMonths, nbYears))
moyMAreaInterval_45 = np.zeros((lencalcMonths, nbYears))
moyMAreaInterval_85 = np.zeros((lencalcMonths, nbYears))
# Tmax, i and j are computed to localize the mean
for p in range(startYear, endYear) :
    a = startYear + p
    b = a + yearInterval
    moyMAreaInterval_26[:, p] = np.mean(Tlse_tmax26[a:b, calcMonth, startj:endj, starti:endi])
    moyMAreaInterval_45[:, p] = np.mean(Tlse_tmax45[a:b, calcMonth, startj:endj, starti:endi])
    moyMAreaInterval_85[:, p] = np.mean(Tlse_tmax85[a:b, calcMonth, startj:endj, starti:endi])
firstPlotYear = startYear + yearInterval
lastPlotYear = firstPlotYear + yearPeriod
data = [] ## lencalcMonths
trace = [] ## lencalcMonths

```

Le diagramme ci-dessous représente l'évolution sur yearPeriod années de la température dans la région Lyonnaise.(Décommenter la dernière ligne du bloc pour afficher la figure)

```

In [24]: q = 0
trace26_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]]
)
trace45_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]]
)
trace85_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_85[q,:],
    name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]]
)

```

```

)

q = 1
trace26_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]]
)
trace45_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]]
)

trace85_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_85[q,:],
    name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]]
)

q = 2
trace26_2 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_26[q,:],
    name = "Moyenne 2.6 de l'année"
)
trace45_2 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_45[q,:],
    name = "Moyenne 4.5 de l'année"
)

trace85_2 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyMAreaInterval_85[q,:],
    name = "Moyenne 8.5 de l'année"
)

fig = tools.make_subplots(rows=3,
                           cols=1,
                           subplot_titles=("mois de janvier ", "mois d'aout", "max de l'année"))

fig.append_trace(trace26_0, 1, 1)
fig.append_trace(trace45_0, 1, 1)
fig.append_trace(trace85_0, 1, 1)
fig.append_trace(trace26_1, 2, 1)
fig.append_trace(trace45_1, 2, 1)
fig.append_trace(trace85_1, 2, 1)

```



```

fig.append_trace(trace26_2, 3, 1)
fig.append_trace(trace45_2, 3, 1)
fig.append_trace(trace85_2, 3, 1)

fig['layout']['xaxis1'].update(title='Year')
fig['layout']['xaxis2'].update(title='Year')
fig['layout']['xaxis3'].update(title='Year')
fig['layout']['yaxis1'].update(title='Degrees C')
fig['layout']['yaxis2'].update(title='Degrees C')
fig['layout']['yaxis3'].update(title='Degrees C')
fig['layout']['margin'] = {'l': 50, 'r': 10, 'b': 50, 't': 50}
fig['layout'].update(autosize = False, height=1200, width=900, title='Températures ma

py.iplot(fig, filename='basic-line')

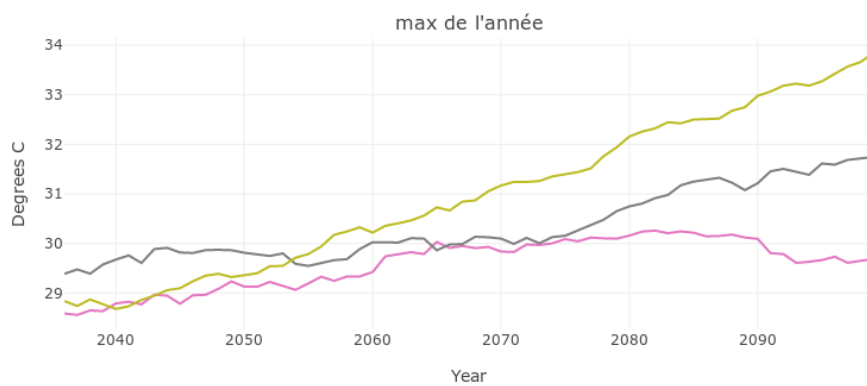
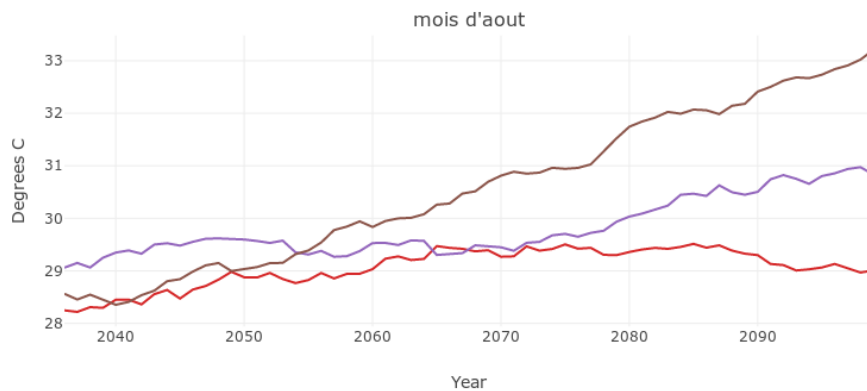
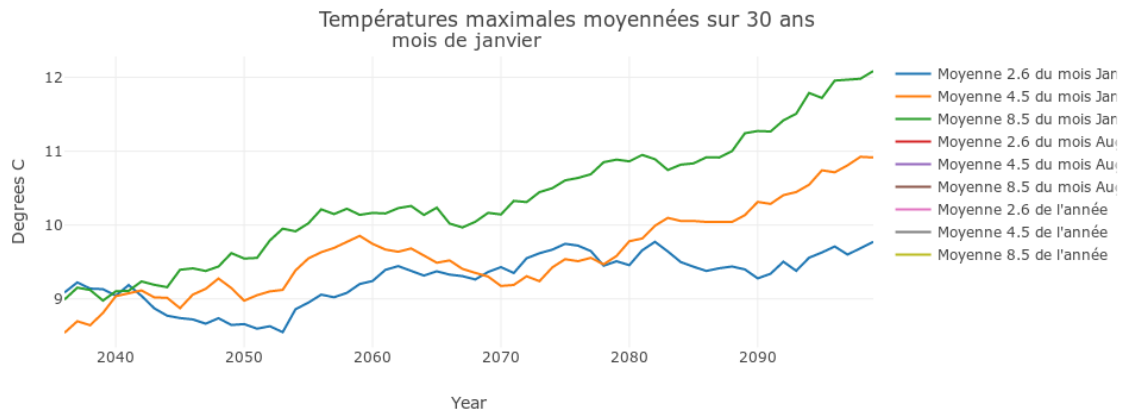
```

This is the format of your plot grid:

```

[ (1,1) x1,y1 ]
[ (2,1) x2,y2 ]
[ (3,1) x3,y3 ]

```



Modifications et enrichissement de la figure Les deux blocs ci-dessous permettent d'améliorer le rendu et d'annoter la figure. Le premier produit un fichier plotly qui peut être ouvert et modifié avec l'éditeur puis sauvegardé sous un nouveau nom par le second.

```
In [11]: pio.write_json(fig, 'scatter1.plotly')
```

```

In [28]: fig_styled = pio.read_json('scatter1+.plotly', output_type='FigureWidget')
         #py.ipplot(fig_styled, filename='styled-basic-line')

In [28]: # Using online file takes a little more time
         # Calculate the number of years yielding a result
moymAreaInterval_26 = np.zeros((lencalcMonths, nbYears))
moymAreaInterval_45 = np.zeros((lencalcMonths, nbYears))
moymAreaInterval_85 = np.zeros((lencalcMonths, nbYears))
         # Tmax, i and j are computed to localize the mean
for p in range(startYear, endYear) :
    a = startYear + p
    b = a + yearInterval
    moymAreaInterval_26[:,p] = np.mean(Tlse_tmin26[a:b,calcMonth,startj:endj,starti:endi])
    moymAreaInterval_45[:,p] = np.mean(Tlse_tmin45[a:b,calcMonth,startj:endj,starti:endi])
    moymAreaInterval_85[:,p] = np.mean(Tlse_tmin85[a:b,calcMonth,startj:endj,starti:endi])
firstPlotYear = startYear + yearInterval
lastPlotYear = firstPlotYear + yearPeriod
#data = [] ## lencalcMonths
#trace = [] ## lencalcMonths# Using online file takes a little more time

```

Le diagramme ci-dessous représente l'évolution sur yearPeriod années de la température dans la région Lyonnaise.(Décommenter la dernière ligne du bloc pour afficher la figure)

```

In [29]: q = 0
         trace26_0 = go.Scatter(
             x = Tlse_year[firstPlotYear:lastPlotYear],
             y = moymAreaInterval_26[q,:],
             name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]]
         )
         trace45_0 = go.Scatter(
             x = Tlse_year[firstPlotYear:lastPlotYear],
             y = moymAreaInterval_45[q,:],
             name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]]
         )
         trace85_0 = go.Scatter(
             x = Tlse_year[firstPlotYear:lastPlotYear],
             y = moymAreaInterval_85[q,:],
             name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]]
         )

         q = 1
         trace26_1 = go.Scatter(
             x = Tlse_year[firstPlotYear:lastPlotYear],
             y = moymAreaInterval_26[q,:],
             name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]]
         )
         trace45_1 = go.Scatter(
             x = Tlse_year[firstPlotYear:lastPlotYear],

```

```

        y = moymAreaInterval_45[q,:],
        name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]]
    )

    trace85_1 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moymAreaInterval_85[q,:],
        name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]]
    )

    q = 2
    trace26_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moymAreaInterval_26[q,:],
        name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]]
    )
    trace45_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moymAreaInterval_45[q,:],
        name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]]
    )

    trace85_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moymAreaInterval_85[q,:],
        name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]]
    )

    fig = tools.make_subplots(rows=3,
                              cols=1,
                              subplot_titles=("mois de janvier",
                                              "mois d'aout",
                                              "minimum de l'année"))

    fig.append_trace(trace26_0, 1, 1)
    fig.append_trace(trace45_0, 1, 1)
    fig.append_trace(trace85_0, 1, 1)
    fig.append_trace(trace26_1, 2, 1)
    fig.append_trace(trace45_1, 2, 1)
    fig.append_trace(trace85_1, 2, 1)
    fig.append_trace(trace26_2, 3, 1)
    fig.append_trace(trace45_2, 3, 1)
    fig.append_trace(trace85_2, 3, 1)

    fig['layout']['xaxis1'].update(title='Year')
    fig['layout']['xaxis2'].update(title='Year')
    fig['layout']['xaxis3'].update(title='Year')

```

```

fig['layout']['yaxis1'].update(title='Degrees C')
fig['layout']['yaxis2'].update(title='Degrees C')
fig['layout']['yaxis3'].update(title='Degrees C')
fig['layout']['margin'] = {'l': 50, 'r': 10, 'b': 50, 't': 50}
fig['layout'].update(height=1200, width=900, title='Températures minimales moyennées :

py.ipplot(fig, filename='basic-line')

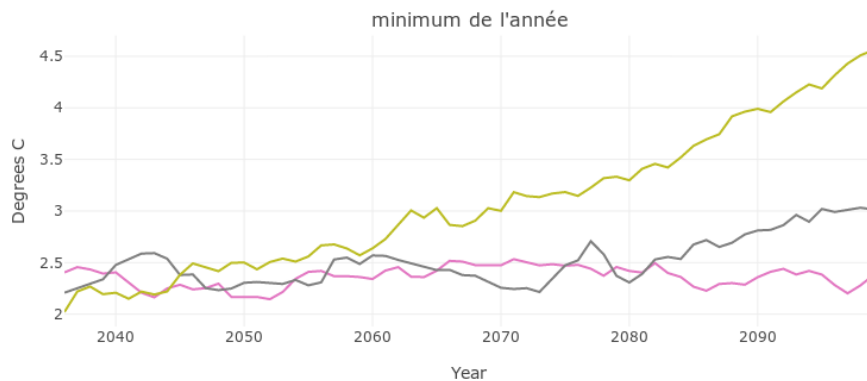
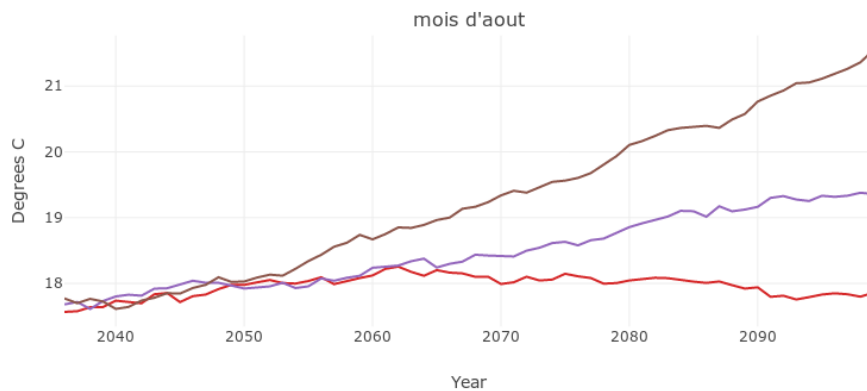
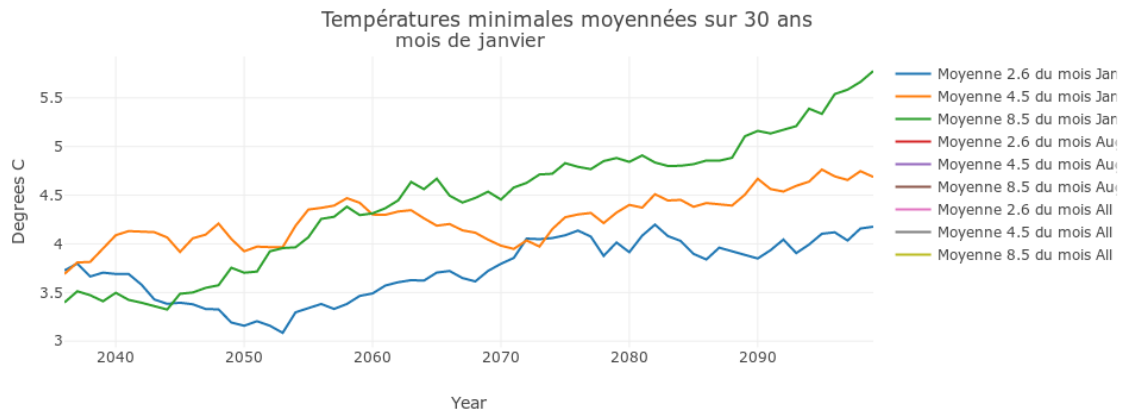
```

This is the format of your plot grid:

```

[ (1,1) x1,y1 ]
[ (2,1) x2,y2 ]
[ (3,1) x3,y3 ]

```



Modifications et enrichissement de la figure

```
In [15]: pio.write_json(fig, 'scatter2.plotly')
```

```
In [27]: fig_styled = pio.read_json('scatter2+.plotly', output_type='FigureWidget')
         #py.iplot(fig_styled, filename='styled-basic-line')
```

```

In [32]: # Using online file takes a little more time
# Calculate the number of years yielding a result
moyDAreaInterval_26 = np.zeros((lencalcMonths, nbYears))
moyDAreaInterval_45 = np.zeros((lencalcMonths, nbYears))
moyDAreaInterval_85 = np.zeros((lencalcMonths, nbYears))
# Tmax, i and j are computed to localize the mean
for p in range(startYear, endYear) :
    a = startYear + p
    b = a + yearInterval
    moyDAreaInterval_26[:,p] = np.mean(Tlse_delta26[a:b,calcMonth,startj:endj,starti:
    moyDAreaInterval_45[:,p] = np.mean(Tlse_delta45[a:b,calcMonth,startj:endj,starti:
    moyDAreaInterval_85[:,p] = np.mean(Tlse_delta85[a:b,calcMonth,startj:endj,starti:
firstPlotYear = startYear + yearInterval
lastPlotYear = firstPlotYear + yearPeriod
#data = [] #* lencalcMonths
#trace = [] #* lencalcMonths# Using online file takes a little more time

In [35]: q = 0
trace26_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]]
)
trace45_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]]
)
trace85_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_85[q,:],
    name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]]
)

q = 1
trace26_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]]
)
trace45_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]]
)

trace85_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],

```

```

        y = moyDAreaInterval_85[q,:],
        name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]]
    )

q = 2
trace26_2 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_26[q,:],
    name = "Moyenne 2.6 de l'année"
)
trace45_2 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_45[q,:],
    name = "Moyenne 4.5 de l'année"
)

trace85_2 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moyDAreaInterval_85[q,:],
    name = "Moyenne 8.5 de l'année"
)

fig = tools.make_subplots(rows=3,
                           cols=1,
                           subplot_titles=("mois de janvier ", "mois d'aout", "max de l'année"))

fig.append_trace(trace26_0, 1, 1)
fig.append_trace(trace45_0, 1, 1)
fig.append_trace(trace85_0, 1, 1)
fig.append_trace(trace26_1, 2, 1)
fig.append_trace(trace45_1, 2, 1)
fig.append_trace(trace85_1, 2, 1)
fig.append_trace(trace26_2, 3, 1)
fig.append_trace(trace45_2, 3, 1)
fig.append_trace(trace85_2, 3, 1)

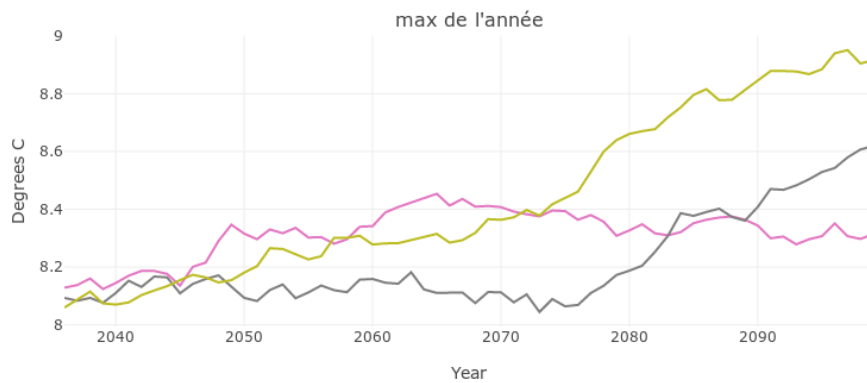
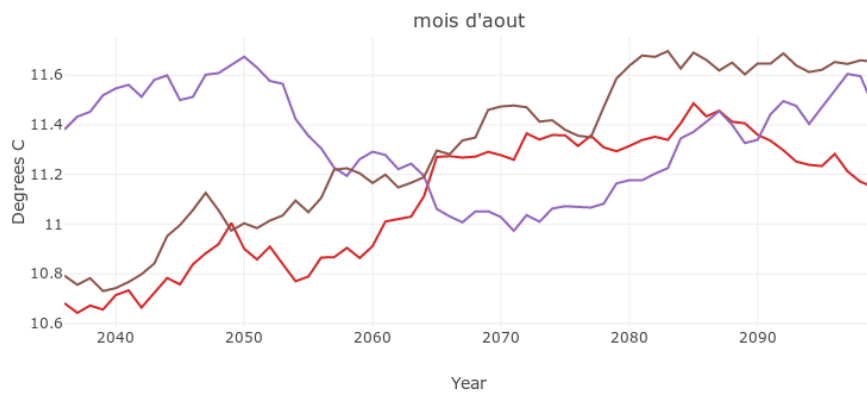
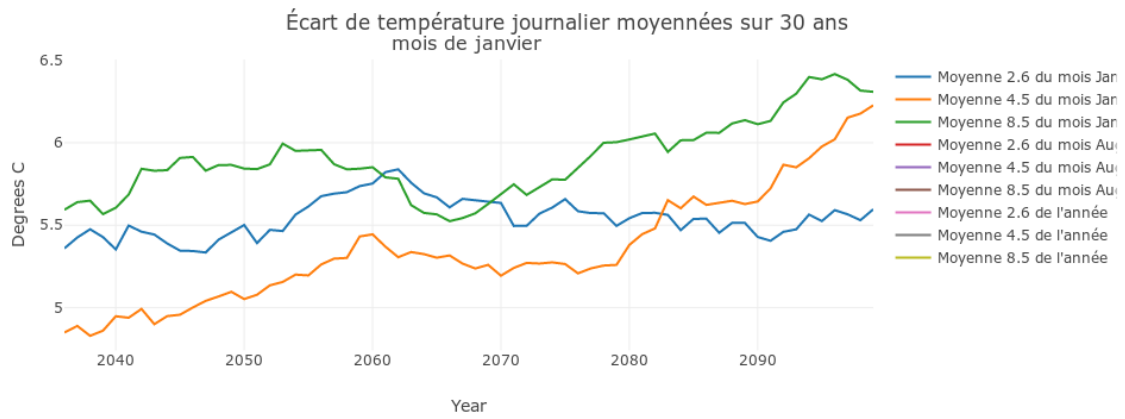
fig['layout']['xaxis1'].update(title='Year')
fig['layout']['xaxis2'].update(title='Year')
fig['layout']['xaxis3'].update(title='Year')
fig['layout']['yaxis1'].update(title='Degrees C')
fig['layout']['yaxis2'].update(title='Degrees C')
fig['layout']['yaxis3'].update(title='Degrees C')
fig['layout']['margin'] = {'l': 50, 'r': 10, 'b': 50, 't': 50}
fig['layout'].update(autosize = False, height=1200, width=900, title='Écart de température')

py.plot(fig, filename='basic-line')

```

This is the format of your plot grid:

[(1,1) x1,y1]
 [(2,1) x2,y2]
 [(3,1) x3,y3]



```

In [36]: # Using online file takes a little more time
# Calculate the number of years yielding a result
moypAreaInterval_26 = np.zeros((lencalcMonths, nbYears))
moypAreaInterval_45 = np.zeros((lencalcMonths, nbYears))
moypAreaInterval_85 = np.zeros((lencalcMonths, nbYears))
# Tmax, i and j are computed to localize the mean
for p in range(startYear, endYear) :
    a = startYear + p
    b = a + yearInterval
    moypAreaInterval_26[:, p] = np.mean(Tlse_rstr26[a:b, calcMonth, startj:endj, startk:endk])
    moypAreaInterval_45[:, p] = np.mean(Tlse_rstr45[a:b, calcMonth, startj:endj, startk:endk])
    moypAreaInterval_85[:, p] = np.mean(Tlse_rstr85[a:b, calcMonth, startj:endj, startk:endk])
firstPlotYear = startYear + yearInterval
lastPlotYear = firstPlotYear + yearPeriod
data = [] ## lencalcMonths
trace = [] ## lencalcMonths# Using online file takes a little more time

```

```

In [37]: q = 0
trace26_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moypAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]],
)
trace45_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moypAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]],
)
trace85_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moypAreaInterval_85[q,:],
    name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]],
)

q = 1
trace26_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moypAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]],
)
trace45_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moypAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]],
)

trace85_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],

```

```

        y = moypAreaInterval_85[q,:],
        name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]],
        opacity=0.75
    )

    q = 2
    trace26_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moypAreaInterval_26[q,:],
        name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]],
    )
    trace45_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moypAreaInterval_45[q,:],
        name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]],
    )

    trace85_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moypAreaInterval_85[q,:],
        name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]],
    )

    fig = tools.make_subplots(rows=3,
                              cols=1,
                              subplot_titles=("mois de janvier",
                                              "mois d'aout",
                                              "moyenne mensuelle sur l'année"))

    fig.append_trace(trace26_0, 1, 1)
    fig.append_trace(trace45_0, 1, 1)
    fig.append_trace(trace85_0, 1, 1)
    fig.append_trace(trace26_1, 2, 1)
    fig.append_trace(trace45_1, 2, 1)
    fig.append_trace(trace85_1, 2, 1)
    fig.append_trace(trace26_2, 3, 1)
    fig.append_trace(trace45_2, 3, 1)
    fig.append_trace(trace85_2, 3, 1)

    fig['layout']['xaxis1'].update(title='Année')
    fig['layout']['xaxis2'].update(title='Année')
    fig['layout']['xaxis3'].update(title='Année')
    fig['layout']['yaxis1'].update(title='mm of rain')
    fig['layout']['yaxis2'].update(title='mm of rain')
    fig['layout']['yaxis3'].update(title='mm of rain')
    fig['layout'].update(height=1200, width=900, title='Précipitations moyennées sur 30 a
    #fig['layout']['margin'] = {'l': 50, 'r': 10, 'b': 50, 't': 50}
    #fig['layout'].update(title='Température moyenne maximale du mois de juillet',

```

```
#                                     autosize=False, width=800, height=380)

py.ipplot(fig, filename='basic-line')
```

This is the format of your plot grid:

```
[ (1,1) x1,y1 ]
[ (2,1) x2,y2 ]
[ (3,1) x3,y3 ]
```



Modifications et enrichissement de la figure

```
In [20]: pio.write_json(fig, 'scatter3.plotly')
```

```
In [26]: fig_styled = pio.read_json('scatter3+.plotly', output_type='FigureWidget')
         #py.iplot(fig_styled, filename='styled-basic-line')
```

```

In [38]: # Using online file takes a little more time
# Calculate the number of years yielding a result
moycAreaInterval_26 = np.zeros((lencalcMonths, nbYears))
moycAreaInterval_45 = np.zeros((lencalcMonths, nbYears))
moycAreaInterval_85 = np.zeros((lencalcMonths, nbYears))
# Tmax, i and j are computed to localize the mean
for p in range(startYear, endYear) :
    a = startYear + p
    b = a + yearInterval
    moycAreaInterval_26[:, p] = np.mean(Tlse_rstrc26[a:b, calcMonth, startj:endj, startk:endk], axis=0)
    moycAreaInterval_45[:, p] = np.mean(Tlse_rstrc45[a:b, calcMonth, startj:endj, startk:endk], axis=0)
    moycAreaInterval_85[:, p] = np.mean(Tlse_rstrc85[a:b, calcMonth, startj:endj, startk:endk], axis=0)
firstPlotYear = startYear + yearInterval
lastPlotYear = firstPlotYear + yearPeriod
data = [] ## lencalcMonths
trace = [] ## lencalcMonths# Using online file takes a little more time

```

```

In [39]: q = 0
trace26_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moycAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]],
)
trace45_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moycAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]],
)
trace85_0 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moycAreaInterval_85[q,:],
    name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]],
)

q = 1
trace26_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moycAreaInterval_26[q,:],
    name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]],
    xaxis='x2',
    yaxis='y2'
)
trace45_1 = go.Scatter(
    x = Tlse_year[firstPlotYear:lastPlotYear],
    y = moycAreaInterval_45[q,:],
    name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]],
    xaxis='x2',

```

```

        yaxis='y2'
    )

    trace85_1 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moycAreaInterval_85[q,:],
        name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]],
        xaxis='x2',
        yaxis='y2'
    )

    q = 2
    trace26_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moycAreaInterval_26[q,:],
        name = 'Moyenne 2.6 du mois %s'%Tlse_month[calcMonth[q]],
        xaxis='x2',
        yaxis='y2'
    )

    trace45_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moycAreaInterval_45[q,:],
        name = 'Moyenne 4.5 du mois %s'%Tlse_month[calcMonth[q]],
        xaxis='x2',
        yaxis='y2'
    )

    trace85_2 = go.Scatter(
        x = Tlse_year[firstPlotYear:lastPlotYear],
        y = moycAreaInterval_85[q,:],
        name = 'Moyenne 8.5 du mois %s'%Tlse_month[calcMonth[q]],
        xaxis='x2',
        yaxis='y2'
    )

    fig = tools.make_subplots(rows=3,
                              cols=1,
                              subplot_titles=(
                                  "Mois de janvier",
                                  "Mois d'aout",
                                  "Cumul annuel de précipitations"
                              ))

    fig.append_trace(trace26_0, 1, 1)
    fig.append_trace(trace45_0, 1, 1)
    fig.append_trace(trace85_0, 1, 1)
    fig.append_trace(trace26_1, 2, 1)
    fig.append_trace(trace45_1, 2, 1)

```

```

fig.append_trace(trace85_1, 2, 1)
fig.append_trace(trace26_2, 3, 1)
fig.append_trace(trace45_2, 3, 1)
fig.append_trace(trace85_2, 3, 1)
fig['layout']['xaxis1'].update(title='Année')
fig['layout']['xaxis2'].update(title='Année')
fig['layout']['xaxis3'].update(title='Année')
fig['layout']['yaxis1'].update(title='mm of rain')
fig['layout']['yaxis2'].update(title='mm of rain')
fig['layout']['yaxis3'].update(title='mm of rain')
fig['layout'].update(height=1200, width=1000, title='Précipitations cumulées moyennées')

py.ipplot(fig, filename='basic-line')

```

This is the format of your plot grid:

```

[ (1,1) x1,y1 ]
[ (2,1) x2,y2 ]
[ (3,1) x3,y3 ]

```




Modifications et enrichissement de la figure

```
In [24]: pio.write_json(fig, 'scatter4.plotly')
```

```
In [25]: fig_styled = pio.read_json('scatter4+.plotly', output_type='FigureWidget')
         #py.iplot(fig_styled, filename='styled-basic-line')
```

2.6 Approche cartographique des températures aux environs de Toulouse

```
In [59]: trace0 = go.Heatmap(z = Tlse_tmax26[34,7,:,:].tolist(), zauto=False, zmin=29,zmax=33)
        trace1 = go.Heatmap(z = Tlse_tmax26[34, 12, :,:].tolist(), zauto=False, zmin=29,zmax=33)

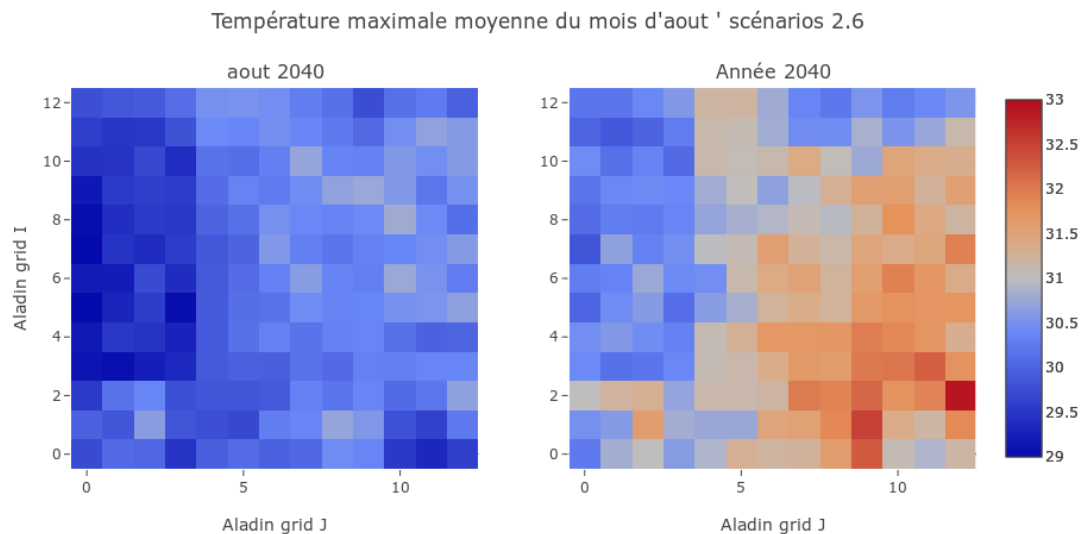
        fig = tools.make_subplots(rows=1, cols=2, subplot_titles=('aout 2040','Année 2040'))

        fig.append_trace(trace0, 1, 1)
        fig.append_trace(trace1, 1, 2)

        fig['layout']['xaxis1'].update(title='Aladin grid J')
        fig['layout']['xaxis2'].update(title='Aladin grid J')
        fig['layout']['yaxis1'].update(title='Aladin grid I')
        fig['layout'].update(title="Température maximale moyenne du mois d'aout ' scénarios 2
                                autosize=False, width=800, height=500)
        py.ipplot(fig, filename='basic-heatmap')
```

This is the format of your plot grid:

```
[ (1,1) x1,y1 ] [ (1,2) x2,y2 ]
```



```
In [78]: trace0 = go.Heatmap(z = Tlse_tmin26[34,11,:,:].tolist(), zauto=False, zmin=-1,zmax=3)
        trace1 = go.Heatmap(z = Tlse_tmin26[34, 12, :,:].tolist(), zauto=False, zmin=-1,zmax=3)

        fig = tools.make_subplots(rows=1, cols=2, subplot_titles=('janvier 2040','Année 2040'))

        fig.append_trace(trace0, 1, 1)
```

```

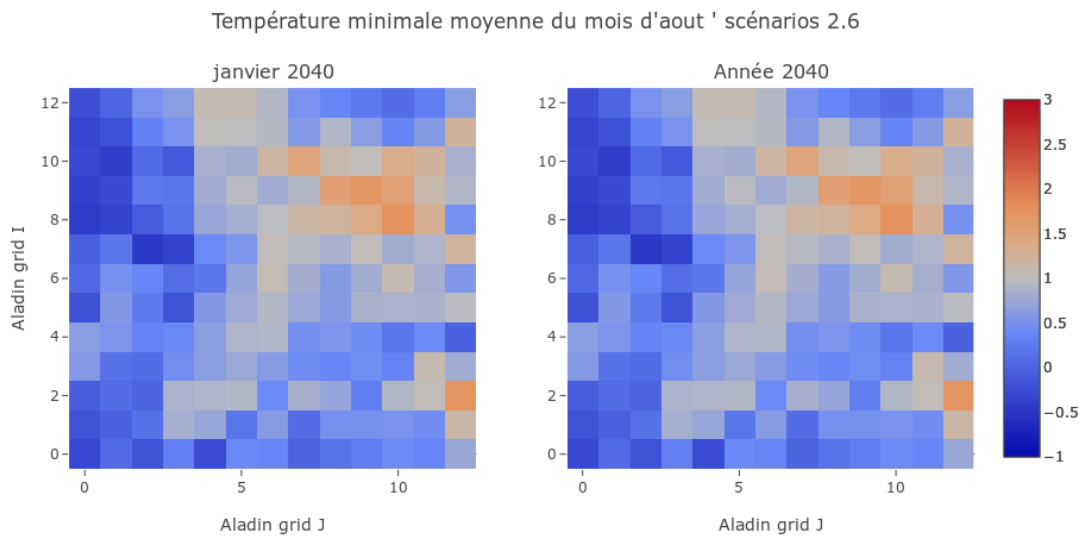
fig.append_trace(trace1, 1, 2)

fig['layout']['xaxis1'].update(title='Aladin grid J')
fig['layout']['xaxis2'].update(title='Aladin grid J')
fig['layout']['yaxis1'].update(title='Aladin grid I')
fig['layout'].update(title="Température minimale moyenne du mois d'aout ' scénarios 2
                        autosize=False, width=800, height=500)
py.iplot(fig, filename='basic-heatmap')

```

This is the format of your plot grid:

```
[ (1,1) x1,y1 ] [ (1,2) x2,y2 ]
```



```

In [58]: trace0 = go.Heatmap(z = Tlse_tmax26[94,7,:,:].tolist(), zauto=False, zmin=29,zmax=33)
        trace1 = go.Heatmap(z = Tlse_tmax26[94, 12, :,:].tolist(), zauto=False, zmin=29,zmax=

fig = tools.make_subplots(rows=1, cols=2, subplot_titles=('aout 2100','Année 2100'))

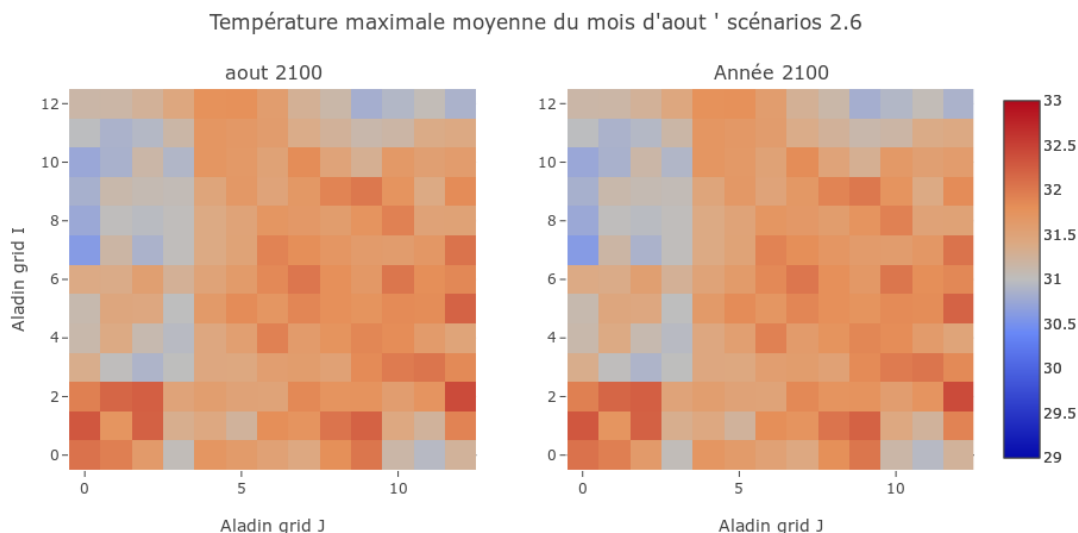
fig.append_trace(trace0, 1, 1)
fig.append_trace(trace1, 1, 2)

fig['layout']['xaxis1'].update(title='Aladin grid J')
fig['layout']['xaxis2'].update(title='Aladin grid J')
fig['layout']['yaxis1'].update(title='Aladin grid I')
fig['layout'].update(title="Température maximale moyenne du mois d'aout ' scénarios 2
                        autosize=False, width=800, height=500)
py.iplot(fig, filename='basic-heatmap')

```

This is the format of your plot grid:

```
[ (1,1) x1,y1 ] [ (1,2) x2,y2 ]
```



```
In [27]: ##### Modifications et enrichissement de la figure
```

```
In [27]: pio.write_json(fig, 'scatter5.plotly')
```

```
In [24]: fig_styled = pio.read_json('scatter5+.plotly', output_type='FigureWidget')
         #py.iplot(fig_styled, filename='styled-basic-line')
```

Comparaison de 3 horizons pour le mois de juillet en région lyonnaise Ce bloc utilise les données mises en forme dans le bloc précédent pour afficher une comparaison entre 3 horizons. (Décommenter la dernière ligne du bloc pour afficher la figure)

```
In [70]: trace00 = go.Heatmap(z=Tlse_tmax26[23,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace01 = go.Heatmap(z=Tlse_tmax26[53,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace02 = go.Heatmap(z=Tlse_tmax26[93,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace10 = go.Heatmap(z=Tlse_tmax45[23,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace11 = go.Heatmap(z=Tlse_tmax45[53,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace12 = go.Heatmap(z=Tlse_tmax45[93,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace20 = go.Heatmap(z=Tlse_tmax85[23,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace21 = go.Heatmap(z=Tlse_tmax85[53,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)
         trace22 = go.Heatmap(z=Tlse_tmax85[93,2,:,:].tolist(), zauto=False, zmin=10,zmax=15)

         fig = tools.make_subplots(rows=3, cols=3, subplot_titles=('Année 2030', 'Année 2050', 'Année 2070'))
```

```

fig.append_trace(trace00, 1, 1)
fig.append_trace(trace01, 1, 2)
fig.append_trace(trace02, 1, 3)
fig.append_trace(trace10, 2, 1)
fig.append_trace(trace11, 2, 2)
fig.append_trace(trace12, 2, 3)
fig.append_trace(trace20, 3, 1)
fig.append_trace(trace21, 3, 2)
fig.append_trace(trace22, 3, 3)

fig['layout']['xaxis1'].update(title='Aladin grid J')
fig['layout']['xaxis2'].update(title='Aladin grid J')
fig['layout']['xaxis3'].update(title='Aladin grid J')
fig['layout']['yaxis1'].update(title='Aladin grid I')
fig['layout'].update(title='Température maximale moyenne du mois de Mars scénarios 2.0',
                    autosize=False, width=800, height=1100)

py.iplot(fig, filename='basic-heatmap')

```

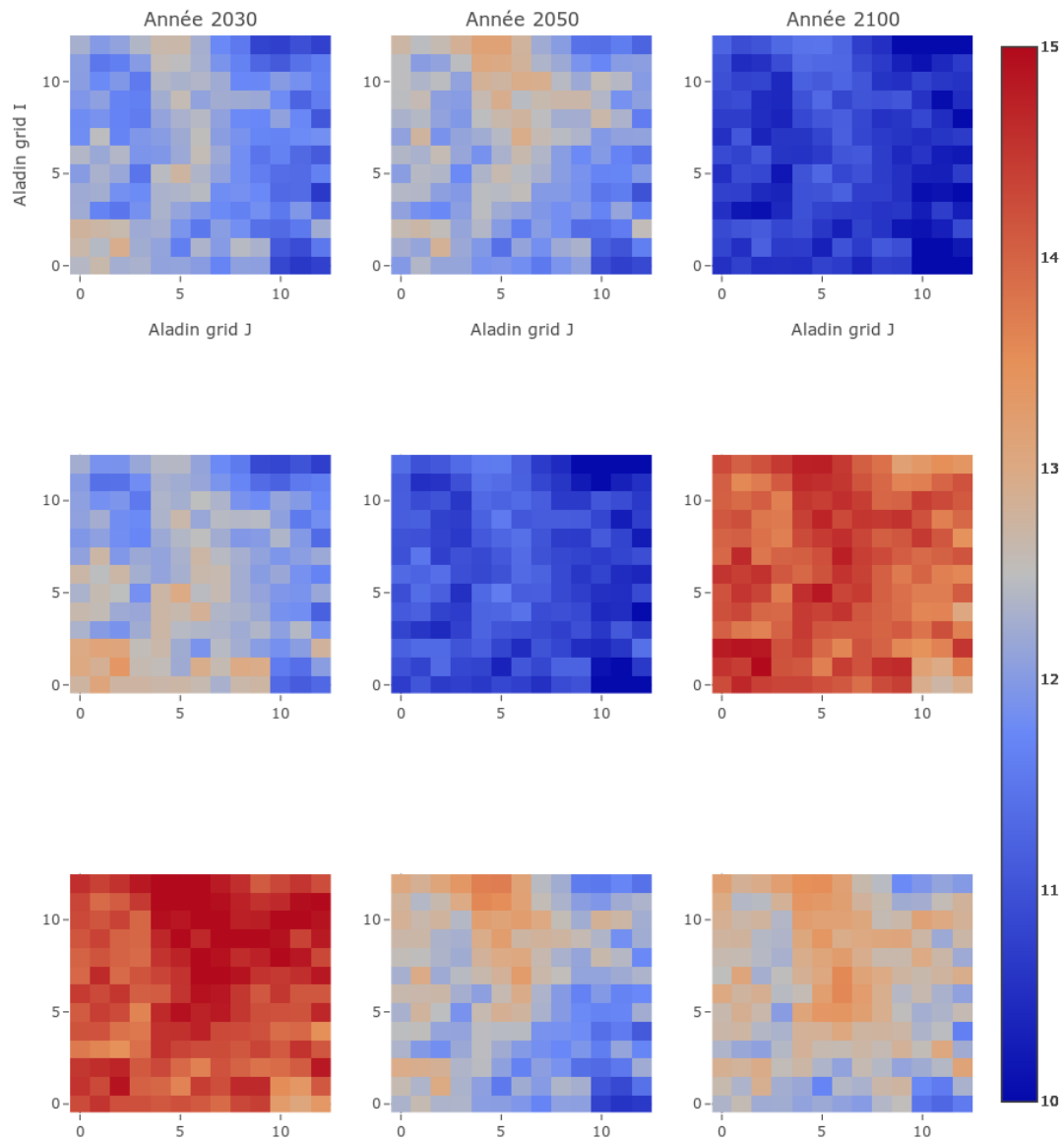
This is the format of your plot grid:

```

[ (1,1) x1,y1 ] [ (1,2) x2,y2 ] [ (1,3) x3,y3 ]
[ (2,1) x4,y4 ] [ (2,2) x5,y5 ] [ (2,3) x6,y6 ]
[ (3,1) x7,y7 ] [ (3,2) x8,y8 ] [ (3,3) x9,y9 ]

```

Température maximale moyenne du mois de Mars scénarios 2.6 4.5 8.5



Modifications et enrichissement de la figure

```
In [30]: pio.write_json(fig, 'scatter6.plotly')
```

```
In [23]: fig_styled = pio.read_json('scatter6+.plotly', output_type='FigureWidget')
         #py.iplot(fig_styled, filename='styled-basic-line')
```

```
In [64]: startYear = 44
         endYear = 74
```

```

slice = lenmonths - 1 # slice = lenmonths pour traiter en plus la moyenne annuelle
moyAreaInterval1 = [None] * slice
moyAreaInterval2 = [None] * slice
moyAreaInterval3 = [None] * slice
moyAreaInterval1[:] = np.mean(Tlse_tmax26[startYear:endYear,:,startj:endj,starti:endi
moyAreaInterval2[:] = np.mean(Tlse_tmax45[startYear:endYear,:,startj:endj,starti:endi
moyAreaInterval3[:] = np.mean(Tlse_tmax85[startYear:endYear,:,startj:endj,starti:endi
#print(moyAreaInterval[0:slice])
with open('moyMonth.txt', 'w') as file :
    for p in range(slice) :
        file.write(Tlse_month[p])
        file.write(';')
        file.write(str(moyAreaInterval1[p]))
        file.write(str(moyAreaInterval2[p]))
        file.write(str(moyAreaInterval3[p]))
        file.write('\n')

```

```

In [65]: trace0 = go.Scatter(
        x = Tlse_month[0:slice],
        y = moyAreaInterval1[0:slice],
        name = 'scenario 2.6'
    )
    trace1 = go.Scatter(
        x = Tlse_month[0:slice],
        y = moyAreaInterval2[0:slice],
        name = 'scenario 4.5'
    )
    trace2 = go.Scatter(
        x = Tlse_month[0:slice],
        y = moyAreaInterval3[0:slice],
        name = 'scenario 8.5'
    )

    data = [trace0,trace1,trace2]

    layout = dict (
        title = "Température moyenne maximale des mois de l'année 2050",
        xaxis = dict(
            title = 'Mois',
            showline=True,
            showticklabels=True,
            ticklen=5
        ),
        yaxis = dict(
            title = 'Temperature (degrees Celsius)',
            showline=True,
            showticklabels=True
        ),
    )

```

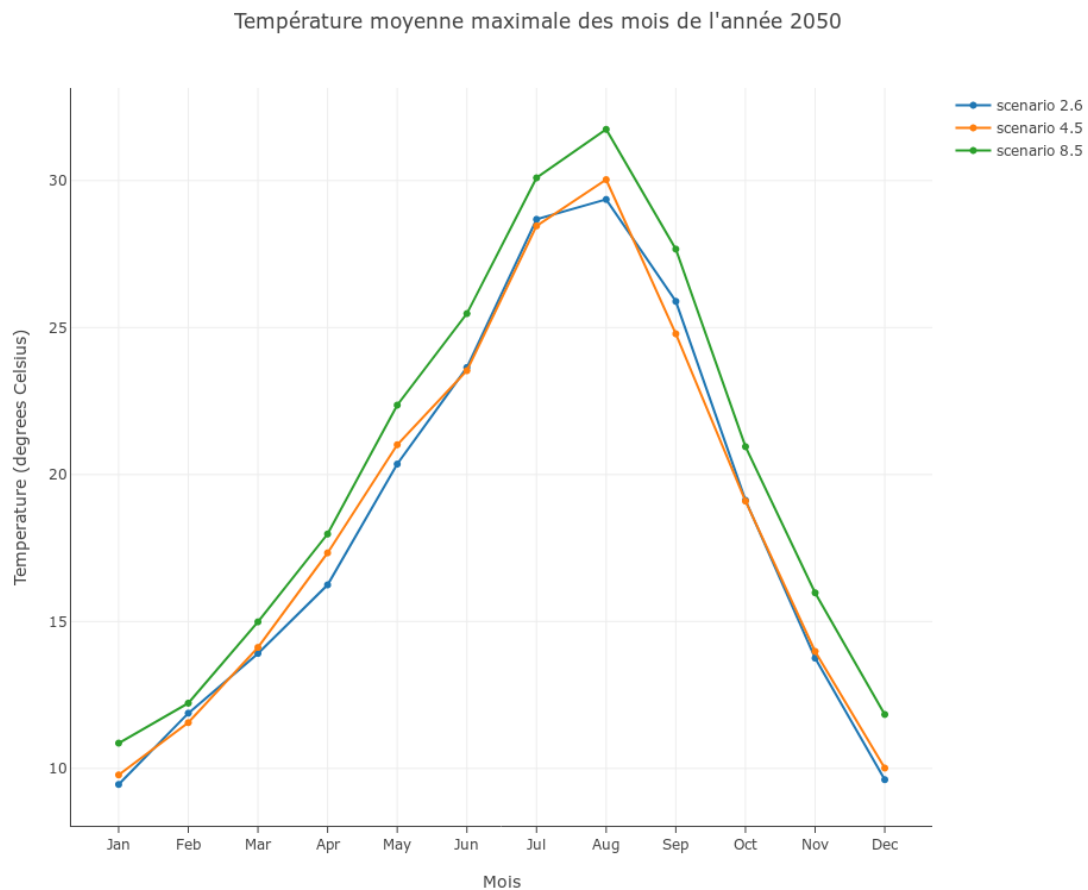
```

        height = 800,
        autosize=False,
        width=850,
    )

    fig = dict(data=data, layout=layout)

    py.iplot(fig, filename='basic-line')

```



Modifications et enrichissement de la figure

```

In [36]: pio.write_json(fig, 'scatter7.plotly')

In [22]: fig_styled = pio.read_json('scatter7+.plotly', output_type='FigureWidget')
          #py.iplot(fig_styled, filename='styled-basic-line')

```


2.6.1 Contrôle du contenu du fichier sauvegardé

Ce bloc permet d'imprimer le contenu du jeu de données qui vient d'être sauvegardé afin de vérifier qu'il contient les données attendues. le suivant ferme le fichier écrit.

```
In [15]: #for dim in extractLyonTempYearMonth.dimensions.items():  
#       print(dim[1])  
#for var in extractLyonTempYearMonth.variables.keys() :  
#       print (var, '\t\t', extractLyonTempYearMonth.variables[var].dimensions, '\t\t',  
#               extractLyonTempYearMonth.variables[var].shape, '\t', extractLyonTempYearM  
  
In [ ]:
```