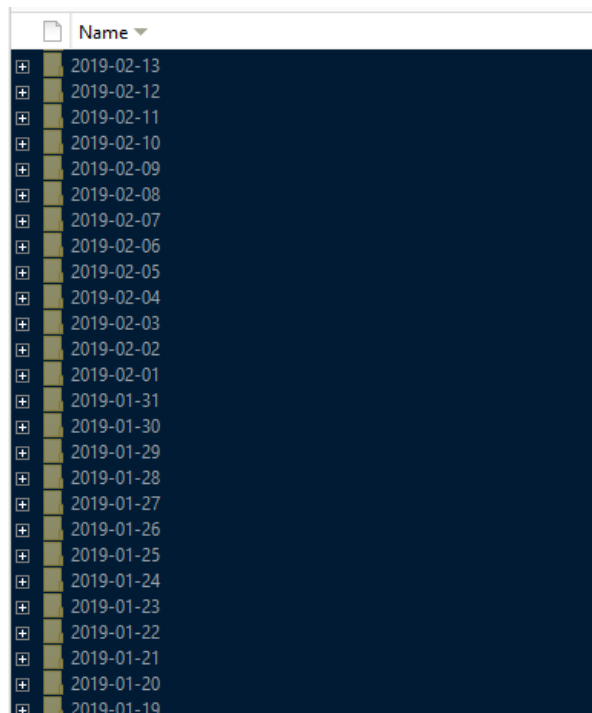


## Περιγραφή υπολογιστικών προϊόντων

Το λογισμικό που αναπτύχθηκε στα πλαίσια του ΠΕ 2.1.1 έχει βασικό στόχο την υποστήριξη της διαδικασίας συνεχούς ροής μετρήσεων από τους αισθητήρες υπαίθρου του επιχειρησιακού δικτύου Μετεωρολογικών σταθμών Ιονίου προς την βάση δεδομένων. Για το λόγο αυτό πραγματοποιήθηκε συγγραφή μονάδων κώδικα (scripts για στοχευμένους Μαθηματικούς και Στατιστικούς υπολογισμούς και σχετικά προϊόντα πραγματικού χρόνου, σε περιβάλλον Matlab.

Αρχικά τρέχει το πρώτο script (**MergeMonthly.m**) του οποίου η λειτουργία είναι να δημιουργεί μηνιαία αρχεία από τα ημερήσια. Αναλυτικότερα, για κάθε έναν από τους 14 μετεωρολογικούς σταθμούς του δικτύου υπάρχει ένα ημερήσιο αρχείο με 37 παραμέτρους καθώς και ημερομηνία και ώρα. Η συχνότητα δειγματοληψίας είναι μία τιμή ανά λεπτό. Επομένως για κάθε σταθμό υπάρχουν στην ιδανική περίπτωση 365 φάκελοι με 365 ημερήσια αρχεία (βλ. Εικόνα 1). Σε κάθε ημερήσιο directory (ανεξαρτήτως ημέρας του έτους), κάθε σταθμός διατηρεί διαρκώς το ίδιο όνομα αρχείου (πχ. ο σταθμός Λευκάδας έχει διαρκώς το 041A0259 ανεξαρτήτως ημέρας). Αρα για να μπορεί ο κώδικας να αναγνωρίζει τα σωστά αρχεία που πρέπει να επεξεργαστεί, θα πρέπει να γνωρίζει ποιό αλφαριθμητικό αντιστοιχεί σε ποιόν σταθμό και για ποιά χρονική περίοδο. Ο χρήστης του κώδικα ενημερώνει χειροκίνητα στην κατάλληλη γραμμή εντολής κάθε φορά το αλφαριθμητικό του εκάστοτε σταθμού.



Εικόνα 1.

Ειδικότερα, προσομοιώθηκαν καταστάσεις όπου ημερήσια αρχεία σταθμών μπορεί να είναι είτε κενά (δηλ. να μην έχουν καθόλου μετρήσεις, απο δυσλειτουργίες, οργάνων, μετάδοσης, κλπ) ως και να μην υπάρχουν καθόλου ως αρχεία (απο εμπλοκή όλου του server για διαστήματα 1 ως και 2-3 ημερών). Η επιλογή λοιπόν της αριθμητικής προσέγγισης (δηλαδή οι πρώτοι 31 φάκελοι να αντιστοιχούν στον Ιανουάριο, οι επόμενοι 28 στον Φεβρουάριο κ.ο.κ.) δεν ήταν η βέλτιστη, αφού θα οδηγούσε σε λανθασμένη συγχώνευση των καταγραφών σε περιπτώσεις απώλειας δεδομένων. Έτσι, το script που αναπτύχθηκε σκανάρει το filename (yyyy-MM-dd), παίρνει την πληροφορία του μήνα (MM) και όσοι φάκελοι έχουν το 01 είναι Ιανουάριος, το 02 Φεβρουάριος κ.ο.κ., ώστε να είναι χειρίσιμες τέτοιες καταστάσεις και να μην κρασάρει ο κώδικας. Τα μηνιαία αρχεία που δημιουργούνται έχουν το εξής format: **station\_month\_year.xls**. Για παράδειγμα, το μηνιαίο αρχείο για το σταθμό Λευκάδας, για το έτος 2019 και το μήνα Αύγουστο θα έχει το εξής όνομα: **LFK-1\_August\_2019.xls**. Τα τελικά προϊόντα από το πρώτο script (*MergeMonthly.m*) είναι συνολικά 12 μηνιαία αρχεία. Στη συνέχεια, το δεύτερο script (*MergeAnnually.m*) αναλαμβάνει να ενώσει τα 12 μηνιαία αρχεία που δημιουργήθηκαν σε ένα ετήσιο με το εξής format: **station\_year.xls**. Για παράδειγμα, το ετήσιο αρχείο για το σταθμό Λευκάδας, για το έτος 2015 θα έχει το εξής όνομα: **LFK-1\_2019.xls**.

Το προηγουμένως δημιουργηθέν ετήσιο αρχείο αποτελεί πλέον τη βάση με την οποία είναι δυνατή η επεξεργασία και ανάλυση των μετρούμενων παραμέτρων. Έτσι, μέσω του script (*Krokidis\_A.m*) καθίσταται δυνατή η επεξεργασία των ετήσιων αρχείων όλων των σταθμών με τους βασικούς κλιματικούς δείκτες για κάθε έτος. Αρχικά εισάγονται όλες οι στήλες με τις μεταβλητές μαζί με το χρονοσημείωμα (timestamp), ενώ στο τέλος των στηλών προστίθεται η στήλη του timestamp σε 5 χωριστές στήλες (Έτος | Μήνας | Ημέρα | Ώρα | Λεπτά). Στο πεδίο station δηλώνεται ο κωδικός του εκάστοτε σταθμού, στο πεδίο year το έτος και τέλος στο πεδίο MINUTEmatrixDIR ο φάκελος στον οποίο επιλέγει ο χρήστης να αποθηκευτούν τα παραγόμενα αρχεία excel βροχόπτωσης, θερμοκρασίας, ανέμου, πίεσης, υγρασίας και ηλιακής ακτινοβολίας (βλ. Εικόνα 2). Στη συνέχεια, πραγματοποιούνται οι απαραίτητοι υπολογισμοί προκειμένου να αποθηκευτούν τα αρχεία, με το εξής format: **station\_Περιγραφή\_year.xls**.

```

Editor - G:\ΕΡΕΥΝΗΤΙΚΑ\ΛΑΕΡΤΗΣ\ΥΠΟΕΡΓΟ 2\ΠΑΡΑΔΟΤΕΑ\CLIMATIC_BASE\2019\Krokidis_A.m*
Krokidis_A.m* x +
1 ***** Import Annually Excel file *****
2 filename='041A0259_LFK-1_2019.xlsx';
3 station='LFK-1';
4 year='2019';
5 MINUTEMATRIXDIR='G:\ΕΡΕΥΝΗΤΙΚΑ\ΛΑΕΡΤΗΣ\ΥΠΟΕΡΓΟ 2\ΠΑΡΑΔΟΤΕΑ\CLIMATIC_BASE\2019\MONTHLY_MATRIXES\Per MINUTE Monthly Matrices\';
6 data = readtable(filename);
7
8 DN = table2array((data(:,2))); Rain = table2array((data(:,3))); Vel_avg = table2array((data(:,4)));
9 Velmin = table2array((data(:,5))); Velmax = table2array((data(:,6))); Velsdv = table2array((data(:,7)));
10 Diravg = table2array((data(:,8))); Dirmin = table2array((data(:,9))); Dirmax = table2array((data(:,10)));
11 Dirsdv = table2array((data(:,11))); Pressure = table2array((data(:,12))); Temperature = table2array((data(:,13)));
12 RH = table2array((data(:,14))); SolarIrradiance = table2array((data(:,15))); UV = table2array((data(:,16)));
13 UVB = table2array((data(:,17))); PMI = table2array((data(:,18))); PM25 = table2array((data(:,19)));
14 PM4 = table2array((data(:,20))); PM10 = table2array((data(:,21))); Total = table2array((data(:,22)));
15 MS = table2array((data(:,23))); FP = table2array((data(:,24))); ES = table2array((data(:,25)));
16 HWSP = table2array((data(:,26))); PMWP = table2array((data(:,27))); SH = table2array((data(:,28)));
17 AH = table2array((data(:,29))); K = table2array((data(:,30))); TDEN = table2array((data(:,31)));
18 TVIR = table2array((data(:,32))); RHO = table2array((data(:,33))); APWR = table2array((data(:,34)));
19 EAN = table2array((data(:,35))); SEN = table2array((data(:,36)));
20
21 ***** R A I N *****
22 % Κατανομή συχνότητας ημερήσιων βροχόπτωσης (κλάσεων εύρους ανά 0.1 mm/min) ανά μήνα
23 t=timetable(date_time, Rain);
24 t.date_time.Format = 'dd-Mmm-yy HH-mm-ss';
25 for a = 1:12
26     % Monthly timetables from t
27     %Month(a,:) = t(MIdx,:);
28 end
29 %Month
30 classes = [0:1:0.1:4.2];
31 for b = 1:numel(Month)
32     file_name = sprintf('Monthly Rain %d', b);

```

Εικόνα 2.

Αναλυτικά η λίστα αρχείων που θα εμφανίζεται στον σκληρό δίσκο αυτόματα κατά ομάδες, στο παράδειγμα για το σταθμό Λευκάδας και το έτος 2019, είναι η ακόλουθη:

USB Drive (G:) > ΕΡΕΥΝΗΤΙΚΑ > ΛΑΕΡΤΗΣ > ΥΠΟΕΡΓΟ 2 > ΠΑΡΑΔΟΤΕΑ > A\_CLIMATIC\_BASE > 2019 > MONTHLY\_MATRIXES > Per MINUTE Monthly Matrices

Name	Date modified	Type	Size
LFK-1_1a_Annual RAIN Height_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1b_Semester RAIN Height_Cold-Warm Period_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1c_Seasonal RAIN Height_per 3 months_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1d_Monthly RAIN Height_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1e_Monthly RAIN Height_PerHour_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	27 KB
LFK-1_1f_Monthly RAIN Height_10classes of daily heights_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1g_Monthly RAIN_Nof Days per Rain Intensity Class_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	2 KB
LFK-1_1h_Monthly RAIN Height_Max Daily Height_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1i_Monthly RAIN Height_Time of Max Rain Rate_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1j_Monthly RAIN Nof Rainy Days_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1k_Monthly RAIN Nof consecutive Dry-Wet Days_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1m_Annual RAIN_Max Consecutive Rain Days_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_1n_Annual RAIN_Max Consecutive Dry Days_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_2a_Monthly TEMP_AV_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_2b_Monthly TEMP_Times of Min-MIN_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_2c_Monthly TEMP_Times of Max-MAX_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_2d_Monthly TEMP_MeanMIN_MeanMAX_and SDVs_2019.csv	25-Feb-23 20:49	Microsoft Excel C...	1 KB
LFK-1_2e_Monthly TEMP_NofDays with Thres Exceed_2019.csv	25-Feb-23 20:50	Microsoft Excel C...	1 KB
LFK-1_2f_Monthly TEMP_Min-Max-AV Daily Range_2019.csv	25-Feb-23 20:50	Microsoft Excel C...	1 KB
LFK-1_3a_Monthly WIND Means and SDVs_2019.csv	25-Feb-23 20:50	Microsoft Excel C...	1 KB
LFK-1_3c_Monthly WIND_10 Max Gusts_2019.csv	25-Feb-23 20:50	Microsoft Excel C...	4 KB
LFK-1_3d_Monthly AV WIND_Distribution Classes_2019.csv	25-Feb-23 20:50	Microsoft Excel C...	3 KB
LFK-1_3e_Monthly WIND GUST Distribution Classes_2019.csv	25-Feb-23 20:50	Microsoft Excel C...	3 KB
LFK-1_3f_Monthly WIND_Weibull Param_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_4a_SOLAR_Time of Max Daily Irradiance_2019.csv	25-Feb-23 20:52	Microsoft Excel C...	9 KB
LFK-1_5a_Monthly HUM_Times of Min-Max_RH_AH_WindDir_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	2 KB
LFK-1_5b_Monthly HUM_Mean_SDV_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_5c_Monthly HUM_NofDays of AH 2sigma_Exceed_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_5d_Monthly HUM_NofDays with RH Thresh Exceed_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_6a_Monthly PRES_Times of Min-Max_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_6b_Monthly PRES_NofDays of UpperThres Exceed_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_6c_Monthly PRES_NofDays of LowThres Exceed_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_6d_Monthly PRESS_Nof Days_with Thres Exceed_2019.csv	25-Feb-23 20:51	Microsoft Excel C...	1 KB
LFK-1_7a_Wind Velocity Histograms_2019.jpg	25-Feb-23 20:50	JPG File	77 KB
LFK-1_7b_Monthly Wind Velocity Histograms_2019.jpg	25-Feb-23 20:50	JPG File	71 KB
LFK-1_7c_Weibull_2019.jpg	25-Feb-23 20:50	JPG File	162 KB

Εικόνα 3.

## ΠΑΡΑΡΤΗΜΑ:

### A1.MergeMonthly.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Create Monthly Excel Files from Daily ones %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
path = pwd;
% use current folder
% path = 'D:\ΕΡΕΥΝΗΤΙΚΑ\ΛΑΕΡΤΗΣ\ΥΠΟΕΡΓΟ 2\ΠΑΡΑΔΟΤΕΑ\A_CLIMATIC_BASE\2019';
% define a specific folder
station='LFK-1';
year='2019';
S = dir(fullfile(path, '*', '041A0259.xls'));
folders_to_export = numel(S);
% export folders by month
columns_to_keep = (1:37);
% select columns (indexes) to keep --> all columns
% main loop
month_number_old = 1;
out = [];
for ci = 1:folders_to_export
% export folders according to values in vector 'folders_to_export'
    folder_name = S(ci).folder;
    file_name = S(ci).name;
    month_number = extractBetween(folder_name, '-', '-');
% detect month
    month_number = str2num(month_number{:});

    if month_number > month_number_old
% switch to next month
        % 1 / save previous month data
        month_name = datestr(datetime(1,month_number_old,1),'mmm');
        filename_out = [station '_' month_name '_' year '.xls'];
% the excel output name contains the selected folders
        writetable(out,fullfile(path,filename_out));
        % 2 / start a new data concat (1st iteration)
        month_number_old = month_number;
% update month_number_old
        F = fullfile(folder_name,file_name);
        data = readtable(F);
        out = [data(:,columns_to_keep)]; % vertical concatenation
    else
% we are still in the same month so keep concatenate data
        F = fullfile(folder_name,file_name);
        data = readtable(F);
        out = [out;data(:,columns_to_keep)]; % vertical concatenation
    end
    % if we reach the last folder, then save the data (because this last case is not covered
    by the if/else loop above)
    if ci == folders_to_export
        month_name = datestr(datetime(1,month_number,1),'mmm');
        filename_out = [station '_' month_name '_' year '.xls'];
% the excel output name contains the selected folders
        writetable(out,fullfile(path,filename_out));
    end
end
end
```

## A2.MergeAnnually.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Create Annual Excel Files from Monthly ones %%%%%%%%%%
station='LFK-1';
year='2019';
path = pwd;
S = dir([path '*.xls']);
% create "list" ( month numbers matching month position )
n = 0;
list = [];
list_month =
{'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'}; % Ημερολογιακό έτος
% list_month =
{'July', 'August', 'September', 'October', 'November', 'December', 'January', 'February', 'March', 'April', 'May', 'June'}; % Υδρολογικό έτος
for k = 1:numel(S)
    tmp = split(S(k).name,{'_', '_', '.'}); % keep only the month string extracted from excel file filename
    [tf,idx] = ismember(tmp{2},list_month); % look for membership with month list (idx contains position in list)
    list(k) = idx;
end
list = list(list>0); % remove non valid xls files
[newlist,order] = sort(list);
out = [];
for k = 1:numel(order)
    filename = S(order(k)).name % see command window - excel files are processed in correct order
    F = fullfile(path,filename);
    data = readtable(F);
    out = [out;data];
    %progressbar(k/numel(order))
end
data = out;
Date = datetime(data(:,1), 'InputFormat','yyyy-MM-dd');
Time = datetime(data(:,2), 'InputFormat','HH:mm');
date_time = table(Date+timeofday(Time), 'VariableNames',{'DateTime'}); %% Date and Time columns combined
datesonly = table(Date.Day,Date.Month,Date.Year, 'VariableNames',{'DD','MM','YYYY'});
timesOnly = table(Time.Hour,Time.Minute, 'VariableNames',{'HH','mm'});
data = [date_time data(:,3:end) datesonly timesOnly]; %% Ξεχωριστές στήλες το TIMESTAMP
date_time = table2array(date_time);
filename_new = [station '_' year '.xlsx'];
writetable(data,filename_new,'WriteVariableNames',1);
```

### A3. Krokidis\_A.m

```
%%%%%%%%%% Import Anually Excel File %%%%%%%%%%
% filename='041A0259_LFK-1_2019.xlsx';
% station='LFK-1';
% year='2019';
MINUTEmatrixDIR='G:\ΕΡΕΥΝΗΤΙΚΑ\ΛΑΕΡΤΗΣ\ΥΠΟΕΡΓΟ2\ΠΑΡΑΔΟΤΕΑ\A_CLIMATIC_BASE\2019\MONTHLY_MATRICE
S\Per MINUTE Monthly Matrices\';
% data = readtable(filename);
DN = table2array((data(:,2))); Rain = table2array((data(:,3)));
Vel_avg = table2array((data(:,4)));
Velmin = table2array((data(:,5))); Velmax = table2array((data(:,6)));
Velsdv = table2array((data(:,7)));
Diravg = table2array((data(:,8))); Dirmin = table2array((data(:,9)));
Dirmax = table2array((data(:,10)));
Dirsdv = table2array((data(:,11))); Pressure = table2array((data(:,12)));
Temperature = table2array((data(:,13)));
RH = table2array((data(:,14))); SolarIrradiance = table2array((data(:,15)));
UVA = table2array((data(:,16)));
UVB = table2array((data(:,17))); PM1 = table2array((data(:,18)));
PM25 = table2array((data(:,19)));
PM4 = table2array((data(:,20))); PM10 = table2array((data(:,21)));
Total = table2array((data(:,22)));
MS = table2array((data(:,23))); FP = table2array((data(:,24)));
ES = table2array((data(:,25)));
MVAP = table2array((data(:,26))); PVAP = table2array((data(:,27)));
SH = table2array((data(:,28)));
AH = table2array((data(:,29))); K = table2array((data(:,30)));
TDEW = table2array((data(:,31)));
TVIR = table2array((data(:,32))); RHO = table2array((data(:,33)));
APWR = table2array((data(:,34)));
EAN = table2array((data(:,35))); SEN = table2array((data(:,36)));
%%%%%%%%%% R A I N: %%%%%%%%%%
% Κατανομή συχνοτήτων ρυθμών βροχόπτωσης (κλάσεων εύρους ανά 0.1 mm/min) ανά μήνα
t=timetable(date_time, Rain);
t.date_time.Format = "dd-MMM-yy HH-mm-ss";
for a = 1:12
    MMidx = month(t.date_time) == a; % Monthly timetables from t
    tMonth{a,:} = t(MMidx,:);
end
tMonth
classes = [0.1:0.1:4.2];
for b = 1:numel(tMonth)
    [BinCounts,~,Bin] = histcounts(tMonth{b}.Rain, classes, 'Normalization','count');
    Nvr{b,:} = BinCounts;
    Binv{:,b} = Bin;
    MMM{b,:} = month(tMonth{b}.date_time(1,:), 'shortname');
end
MMM
TQ = cell2table(MMM)
Nvr;
Frequencies = array2table(cell2mat(Nvr), 'VariableNames',{'(0.1 - 0.2]','(0.2 - 0.3]','(0.3 -
0.4]','(0.4 - 0.5]','(0.5 - 0.6]','(0.6 - 0.7]','(0.7 - 0.8]','(0.8 - 0.9]','(0.9 -
1.0]','(1.0 - 1.1]','(1.1 - 1.2]','(1.2 - 1.3]','(1.3 - 1.4]','(1.4 - 1.5]','(1.5 -
1.6]','(1.6 -1.7]','(1.7 - 1.8]','(1.8 - 1.9]','(1.9 - 2.0]','(2.0 - 2.1]','(2.1 - 2.2]','(2.2
```

```

- 2.3]', '(2.3 - 2.4]', '(2.4 - 2.5]', '(2.5 - 2.6]', '(2.6 - 2.7]', '(2.7 - 2.8]', '(2.8 -
2.9]', '(2.9 - 3.0]', '(3.0 - 3.1]', '(3.1 - 3.2]', '(3.2 - 3.3]', '(3.3 - 3.4]', '(3.4 -
3.5]', '(3.5 - 3.6]', '(3.6 - 3.7]', '(3.7 - 3.8]', '(3.8 - 3.9]', '(3.9 - 4.0]', '(4.0 -
4.1]', '(4.1 - 4.2]')
MMYYYYtbl=retime(t, 'monthly', 'mean')
MMYYYYtbl=datevec(MMYYYYtbl.date_time(:,1))
MMYYYYtbl=MMYYYYtbl(:,1:2);
Frequencies=[(array2table(MMYYYYtbl)) Frequencies(:,1:41)];
Frequencies.Properties.VariableNames{1} = 'Year';
Frequencies.Properties.VariableNames{2} = 'Month';
FNAMTable1 = [station '_1g_Monthly RAIN_Nof Days per Rain Intensity Class_' year '.csv'];
FNOUTable1=[MINUTEmatrixDIR FNAMTable1];
writetable(Frequencies, FNOUTable1);
% Ενδοημερήσια κατανομή υψών βροχόπτωσης (σε ωριαίες κλάσεις), για κάθε μήνα.
rainPerHour = retime(t, 'hourly', 'sum');
for c = 1:12
    MMidx = month(rainPerHour.date_time) == c;
    rainPerHourMonth{c,:} = rainPerHour(MMidx,:);
end

for dd = 1:12
    Rn = rainPerHourMonth{dd};% Create Temporary 'timetable'
    Hours = hour(Rn.date_time);% Create 'Hours' Variable
    [y,m,d] = ymd(Rn.date_time); % Begin To Create 'Date' Variable
    Date = datetime(y,m,d);% Finish Creating 'Date' Variable
    Rn = addvars(Rn, Date, Hours, 'Before', 'Rain');% Add 'Hours' & 'Date' Variables
    Rn.Properties.VariableNames(1:2) = {'Date', 'Hours'};% Name 'Hours' & 'Date' Variables
    RnT = timetable2table(Rn); % Convert To 'table'
    RainPerHourMonthT{dd,:} = unstack(RnT(:,2:end), 'Rain', 'Hours'); % Unstack & Write To Cell
Array
    RainPerHourMonthT{dd,:}.Properties.VariableNames (2:25)=
{'00:00', '01:00', '02:00', '03:00', '04:00', '05:00', '06:00', '07:00', '08:00', '09:00', '10:00', '11:0
0', '12:00', '13:00', '14:00', '15:00', '16:00', '17:00', '18:00', '19:00', '20:00', '21:00', '22:00', '23
:00'}
end
RainPerHourMonthT=cell2table(RainPerHourMonthT);
RainPerHourMonthT=vertcat(RainPerHourMonthT{1,1},RainPerHourMonthT.RainPerHourMonthT{2,1},RainPerHourMonthT.RainPerHourMonthT{3,1},RainPerHourMonthT.RainPerHourMonthT{4,1},RainPerHourMonthT.RainPerHourMonthT{5,1},RainPerHourMonthT.RainPerHourMonthT{6,1},RainPerHourMonthT.RainPerHourMonthT{7,1},RainPerHourMonthT.RainPerHourMonthT{8,1},RainPerHourMonthT.RainPerHourMonthT{9,1},RainPerHourMonthT.RainPerHourMonthT{10,1},RainPerHourMonthT.RainPerHourMonthT{11,1},RainPerHourMonthT.RainPerHourMonthT{12,1});
res = datevec(RainPerHourMonthT.Date(:,1));
res = res(:,1:3);
res = array2table(res);
res.Properties.VariableNames{1} = 'Year';
res.Properties.VariableNames{2} = 'Month';
res.Properties.VariableNames{3} = 'Day';
RainPerHourMonthT = [RainPerHourMonthT(:,1) res RainPerHourMonthT(:,2:25)];
FNAMTable2 = [station '_1e_Monthly RAIN_Height_PerHour_' year '.csv'];
FNOUTable2=[MINUTEmatrixDIR FNAMTable2];
writetable(RainPerHourMonthT, FNOUTable2);
% Μέγιστος ρυθμός βροχόπτωσης και χρόνος που σημειώθηκε για κάθε μήνα.
MaxRainRatePerMonth = retime(t, 'monthly', 'max'); %%% Μέγιστη μηνιαία βροχόπτωση

```

```

MaxRainRatePerMinute = retime(t, 'minutely', 'max');           %% Μέγιστη βροχόπτωση ανά
λεπτό
[yhx,mhx,dhx] = ymd(MaxRainRatePerMinute.date_time);         %% Year, Month, Day Of Daily
Maxima
for f = 1:numel(MaxRainRatePerMonth)
    [ym,mm,dm] = ymd(MaxRainRatePerMonth.date_time(f));
    Lv = ismember([yhx,mhx], [ym,mm], 'rows');
    idx = MaxRainRatePerMinute.Rain(Lv) == MaxRainRatePerMonth.Rain(f);
    Mv = find(Lv);
    Dv = find(idx);
    HM = MaxRainRatePerMinute(Mv(Dv(1)),:);
    RainRateMonthMax(f,:) = timetable2table(HM);
end
RainRateMonthMaxT=table(RainRateMonthMax.date_time.Year, RainRateMonthMax.date_time.Month,
RainRateMonthMax.date_time.Day, RainRateMonthMax.date_time.Hour,
RainRateMonthMax.date_time.Minute,
RainRateMonthMax.Rain,VariableNames={'Year','Month','Day','Hour','Minute','Rain Rate
(mm/min)'});
FNAMETable3 = [station '_1i_Monthly RAIN Height_Time of Max Rain Rate_' year '.csv'];
FNOUTable3=[MINUTEmatrixDIR FNAMETable3];
writetable(RainRateMonthMaxT, FNOUTable3);
% Μέγιστο ημερήσιο ύψος βροχόπτωσης ανά μήνα και ημερομηνία που σημειώθηκε.
idx_time = month(date_time) == 1;
[g, id] = findgroups(month(date_time));
rainPerDay = retime(t, 'daily', 'sum');
g = findgroups(month(rainPerDay.date_time));
rainiestDayPerMonth = splitapply(@iMaxAndLoc, rainPerDay.date_time, rainPerDay.Rain, g);
rainiestDayPerMonthT = table (rainiestDayPerMonth.Time.Year, rainiestDayPerMonth.Time.Month,
rainiestDayPerMonth.Time.Day,
rainiestDayPerMonth.Value,VariableNames={'Year','Month','Day','Value'})
FNAMETable4 = [station '_1h_Monthly RAIN Height_Max Daily Height_' year '.csv'];
FNOUTable4=[MINUTEmatrixDIR FNAMETable4];
writetable(rainiestDayPerMonthT, FNOUTable4);
% Αριθμός βροχοφόρων ημερών μήνα.
rainyDays = timetable(rainPerDay.date_time, double(rainPerDay.Rain > 0));
rainyDays_per_Month=retime(rainyDays, 'monthly', 'sum');
rainyDays_per_MonthT = table (rainyDays_per_Month.Time.Year, rainyDays_per_Month.Time.Month,
rainyDays_per_Month.Var1,VariableNames={'Year','Month','Nof Rainy Days'})
FNAMETable5 = [station '_1j_Monthly RAIN Nof Rainy Days_' year '.csv'];
FNOUTable5=[MINUTEmatrixDIR FNAMETable5];
writetable(rainyDays_per_MonthT, FNOUTable5);
% Κατανομή διάρκειας βροχοφόρων επεισοδίων, δηλαδή αριθμού διαδοχικών βροχοφόρων ημερών σε 14
οριζόμενες απο τον χρήστη κλάσεις (ενδεικτικά 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, >30
days).
%%% Βροχοφόρες-ένα(1)
f1 = find(diff([0 ;rainyDays.Var1 ;0]==1));
From1 = f1(1:2:end-1);           % Start indices
To1 = f1(2:2:end)-1;           % Consecutive ones' counts
rainy_events= [From1 To1 To1-From1+1];
rainy_events_period_duration=rainy_events(:,3);
YEAR = sscanf(year, '%d');
From1 = datetime(YEAR,1,From1);
To1= datetime(YEAR,1,To1);
rainy_events= timetable(From1,To1,(rainy_events_period_duration));
h = [1:30];

```



```

for i=h
    rainy_period(i)= sum(rainy_events_period_duration>i);
end
rainy_period=table(rainy_period(1),rainy_period(2),rainy_period(3),rainy_period(4),rainy_perio
d(5),rainy_period(6),rainy_period(7),rainy_period(8),rainy_period(9),rainy_period(10),rainy_pe
riod(15),rainy_period(20),rainy_period(25),rainy_period(30));
rainy_period=(table2array(rainy_period))';
classes=[1;2;3;4;5;6;7;8;9;10;15;20;25;30];
rainy_period=[classes rainy_period];
t2 = table(classes,rainy_period(:,2), 'VariableNames',{'Class (Max Nof Days)', 'Consecutive
Rainy Days'});
FNAMetable6 = [station '_1m_Annual RAIN_Max Consecutive Rain Days_' year '.csv'];
FNOUtable6=[MINUTEmatrixDIR FNAMetable6];
writetable(t2, FNOUtable6);
% Κατανομή ξηρών περιόδων, δηλαδή αριθμού διαδοχικών ημερών χωρίς βροχή σε 14 οριζόμενες απο
τον χρήστη κλάσεις (ενδεικτικά 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, >30 days).
%% Μηδέν(0)
f0 = find(diff([1 ;rainyDays.Var1 ;1]==0));
From0 = f0(1:2:end-1); % Start indices
To0 = f0(2:2:end)-1; % Consecutive zero counts
dry_events= [From0 To0 To0-From0+1];
dry_events_period_duration=dry_events(:,3);
From0 = datetime(YEAR,1,From0);
To0= datetime(YEAR,1,To0);
dry_events= timetable(From0,To0,dry_events_period_duration);
h = [1:30];
for j=h
    dry_period(j)= sum(dry_events_period_duration>j);
end
dry_period=table(dry_period(1),dry_period(2),dry_period(3),dry_period(4),dry_period(5),dry_perio
d(6),dry_period(7),dry_period(8),dry_period(9),dry_period(10),dry_period(15),dry_period(20),
dry_period(25),dry_period(30));
dry_period=(table2array(dry_period))';
classes=[1;2;3;4;5;6;7;8;9;10;15;20;25;30];
dry_period=[classes dry_period];
t3 = table(classes,dry_period(:,2), 'VariableNames',{'Class (Max Nof Days)', 'Consecutive Dry
Days'});
FNAMetable7 = [station '_1n_Annual RAIN_Max Consecutive Dry Days_' year '.csv'];
FNOUtable7=[MINUTEmatrixDIR FNAMetable7];
writetable(t3, FNOUtable7);
% Μέγιστος αριθμός διαδοχικών βροχοφόρων και ξηρών ημερών του μήνα.
[~,m,~,~,~,~]=datevec(rainyDays.Time);
y=unique([1 find(diff(m)~=0)'+1 numel(m)+1])
rainyDaysperMonth=mat2cell(rainyDays.Var1,diff(y),1);
Datespermonth=mat2cell(rainyDays.Time,diff(y),1);
rainyDaysperMonth_ = {};
for k = 1:12
    rainyDaysperMonth_{k} = timetable(Datespermonth{k,1},rainyDaysperMonth{k,1})
end
rainyDaysperMonth_=rainyDaysperMonth_';
%% Βροχοφόρες-ένα(1)
for m = 1:numel(rainyDaysperMonth) % loop over month
    f_r= find(diff([0; rainyDaysperMonth{m} ; 0]==1));
    rainy_events_this_month= [f_r(1:2:end-1) f_r(2:2:end)-1 (f_r(2:2:end)-1-(f_r(1:2:end-
1))+1)];

```

```

rain_tmp = rainy_events_this_month(:,3);
if ~isempty(rain_tmp)
    Max_conseq_rain(m) = max(rain_tmp);
else
    Max_conseq_rain(m) = 0;
end
end
Max_Rainy_consecutives_per_month=Max_conseq_rain';
    %% Ξηρές- μηδέν(0)
for l = 1:numel(rainyDaysperMonth) % loop over month
    f_d= find(diff([1; rainyDaysperMonth{1} ; 1]==0));
    dry_events_this_month= [f_d(1:2:end-1) f_d(2:2:end)-1 (f_d(2:2:end)-1-(f_d(1:2:end-
1))+1)];
    dry_tmp = dry_events_this_month(:,3);
    if ~isempty(dry_tmp)
        Max_conseq_dry(l) = max(dry_tmp);
    else
        Max_conseq_dry(l) = 0;
    end
end
Max_Dry_consecutives_per_month= Max_conseq_dry';
t4 = table(rainyDays_per_Month.Time.Year,
rainyDays_per_Month.Time.Month,Max_Dry_consecutives_per_month,Max_Rainy_consecutives_per_month
,'VariableNames',{'Year','Month','Max Nof Consecutive Wet Days','Max Nof Consecutive Dry
Days'});
FNAMetable8 = [station '_1k_Monthly RAIN Nof consecutive Dry-Wet Days_' year '.csv'];
FNOUtable8=[MINUTEmatrixDIR FNAMetable8];
writetable(t4, FNOUtable8);
%Αθροιστικά ύψη (μηνιαίο, εποχικό, εκτεταμένο εποχικό, ετήσιο).
    %μηνιαίο
Cumulative_rain_monthly=retime(t, 'monthly', 'sum');
Cumulative_rain_monthlyT=
table(Cumulative_rain_monthly.date_time.Year,Cumulative_rain_monthly.date_time.Month,Cumulativ
e_rain_monthly.Rain,'VariableNames',{'Year','Month','Rain'});
FNAMetable9 = [station '_1d_Monthly RAIN Height_' year '.csv'];
FNOUtable9=[MINUTEmatrixDIR FNAMetable9];
writetable(Cumulative_rain_monthlyT, FNOUtable9);
    %εποχικό
Cumulative_rain_seasonal=retime(t, 'quarterly', 'sum');
Season = {'Jan-March';'April-June';'July-Sept';'Oct-Dec'};
Cumulative_rain_seasonal=table(Season,Cumulative_rain_seasonal.Rain,'VariableNames',{'Season',
'Rain'});
FNAMetable10 = [station '_1c_Seasonal RAIN Height_per 3 months_' year '.csv'];
FNOUtable10=[MINUTEmatrixDIR FNAMetable10];
writetable(Cumulative_rain_seasonal, FNOUtable10);
    %εκτεταμένο εποχικό
warm = {'Apr-Sept'};
Cumulative_rain_extended_seasonal_warm = Cumulative_rain_seasonal(2:3,1:2);
Cumulative_rain_extended_seasonal_warm=sum(Cumulative_rain_extended_seasonal_warm{:},2:end);
Cumulative_rain_extended_seasonal_warm =
table(warm,Cumulative_rain_extended_seasonal_warm,'VariableNames',{'Season','Rain Height'});
cold = {'Oct-March'};
Cumulative_rain_extended_seasonal_cold = Cumulative_rain_seasonal([1,end],1:2);
Cumulative_rain_extended_seasonal_cold=sum(Cumulative_rain_extended_seasonal_cold{:},2:end);

```

```

Cumulative_rain_extended_seasonal_cold =
table(cold,Cumulative_rain_extended_seasonal_cold,'VariableNames',{'Season','Rain Height'});
Cumulative=vertcat(Cumulative_rain_extended_seasonal_cold,Cumulative_rain_extended_seasonal_wa
rm);
FNAMEtable11 = [station '_1b_Semester RAIN Height_Cold-Warm Period_' year '.csv'];
FNOUtable11=[MINUTEmatrixDIR FNAMEtable11];
writetable(Cumulative, FNOUtable11);
    %Ετήσιο
Cumulative_rain_annual=retime(t, 'yearly', 'sum');
Cumulative_rain_annual=table(YEAR, Cumulative_rain_annual.Rain,'VariableNames',{'Year','Rain
Height'});
FNAMEtable12 = [station '_1a_Annual RAIN Height_' year '.csv'];
FNOUtable12=[MINUTEmatrixDIR FNAMEtable12];
writetable(Cumulative_rain_annual, FNOUtable12);
%Κατανομή ημερήσιας ραγδιότητας για κάθε μήνα σε 10 οριζόμενες απο τον χρήστη κλάσεις
(ενδεικτικά μηνιαία βροχόπτωση <0.2, 1.0, 2.5, 5.0, 10, 25, 50, 100, 250, >250 mm).
rainDailyRapidty = retime(t, 'daily', 'sum');
for n = 1:12
    MMidx = month(rainDailyRapidty.date_time) == n;
    rainDailyRapidty_Month{n,:} = rainDailyRapidty(MMidx,:); %% Monthly timetables from
rainDailyRapidty
end
Rapidty_classes = [0.1,0.2,1,2.5,5,10,25,50,100,250,Inf];
Ctrs = Rapidty_classes(1:end-1)+diff(Rapidty_classes)/2;
for o = 1:numel(rainDailyRapidty_Month)
    [BinCounts,~,Bin] = histcounts(rainDailyRapidty_Month{o}.Rain, Rapidty_classes,
'Normalization','count');
    Nv{o,:} = BinCounts;
    Rpd{o,:}=[Rapidty_classes(1:10);BinCounts]
    Binv{:,o} = Bin;
    MMM{o} = month(rainDailyRapidty_Month{k}.date_time(1,:), 'shortname');
end
Rpd
Nv=cell2table(Nv);Nv=splitvars(Nv);
NvT=table(Frequencies.Year,Frequencies.Month, Nv)
NvT=[(array2table(MMYYYTtbl)) Nv(:,1:10)];
NvT.Properties.VariableNames{1} = 'Year';NvT.Properties.VariableNames{2} = 'Month';
NvT.Properties.VariableNames{3} = '(0.1 - 0.2]';NvT.Properties.VariableNames{4} = '(0.2 -
1.0]';
NvT.Properties.VariableNames{5} = '(1.0 - 2.5]';NvT.Properties.VariableNames{6} = '(2.5 -
5.0]';
NvT.Properties.VariableNames{7} = '(5.0 - 10.0]';NvT.Properties.VariableNames{8} = '(10.0 -
25.0]';
NvT.Properties.VariableNames{9} = '(25.0 - 50.0]';NvT.Properties.VariableNames{10} = '(50.0 -
100.0]';
NvT.Properties.VariableNames{11} = '(100.0 - 250.0]';NvT.Properties.VariableNames{12} =
'(250.0 - Inf]';
FNAMEtable13 = [station '_1f_Monthly RAIN Height_10classes of daily heights_' year '.csv'];
FNOUtable13=[MINUTEmatrixDIR FNAMEtable13];
writetable(NvT, FNOUtable13);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% T E M P E R A T U R E : %%%%%%%%%%%
% Μέση μηνιαία θερμοκρασία και τυπική απόκλιση.
t5=timetable(date_time, Temperature);format long g
MeanTemperaturePerMonth = retime(t5, 'monthly', 'mean'); %% Μέση μηνιαία θερμοκρασία
MeanTemperaturePerDay = retime(t5, 'daily', 'mean'); %% Μέση ημερήσια θερμοκρασία
MeanTemperaturePerMonth_STD= retime(MeanTemperaturePerDay, 'monthly', @std); %% Calculate
the standard deviation
MesiThermokrasia = table(MeanTemperaturePerMonth.date_time.Year,
MeanTemperaturePerMonth.date_time.Month,MeanTemperaturePerMonth.Temperature,
MeanTemperaturePerMonth_STD.Temperature , 'VariableNames',{'Year','Month','AV Temp','SDV'})
FNAMetable14 = [station '_2a_Monthly TEMP_AV_' year '.csv'];
FNOUtable14=[MINUTEmatrixDIR FNAMetable14];
writetable(MesiThermokrasia, FNOUtable14);
% μέση ελάχιστη του μήνα, μέση μέγιστη του μήνα θερμοκρασία, και τυπικές αποκλίσεις.
DailyMin = retime(t5, 'daily', 'min'); %% Ελάχιστη ημερήσια θερμοκρασία
MeanMonthlyMin = retime(DailyMin, 'monthly', 'mean') %% Μέση ελάχιστη θερμοκρασία του μήνα
MeanMonthlyMin_SD= retime(DailyMin, 'monthly', @(x)std(x)); %% Τυπική απόκλιση
DailyMax = retime(t5, 'daily', 'max'); %% Μέγιστη ημερήσια θερμοκρασία
MeanMonthlyMax = retime(DailyMax, 'monthly', 'mean') %% Μέση μέγιστη θερμοκρασία του μήνα
MeanMonthlyMax_SD= retime(DailyMax, 'monthly', @(x)std(x)); %% Τυπική απόκλιση
MesiElax_kai_Megisti = table(MeanMonthlyMin.date_time.Year, MeanMonthlyMin.date_time.Month,
MeanMonthlyMin.Temperature, MeanMonthlyMin_SD.Temperature , MeanMonthlyMax.Temperature,
MeanMonthlyMax_SD.Temperature , 'VariableNames',{'Year','Month','Mean MIN Temp','SDV Mean
MIN','Mean MAX Temp','SDV Mean MAX'})
FNAMetable15 = [station '_2d_Monthly TEMP_MeanMIN_MeanMAX_and SDVs_' year '.csv'];
FNOUtable15=[MINUTEmatrixDIR FNAMetable15];
writetable(MesiElax_kai_Megisti, FNOUtable15);
% Απολύτως μηνιαια Ελάχιστη, χρόνος και συνθήκες ανέμου που σημειώθηκε.
MinMinTemperaturePerDay=retime(t5, 'monthly', 'min'); %% ελαχιστη των ελαχιστων
MinTemperaturePerMinute = retime(t5, 'minutely', 'min');
[yhn,mhn,dhn] = ymd(MinTemperaturePerMinute.date_time); %% Year, Month, Day Of Daily Minima
for r = 1:numel(MinMinTemperaturePerDay)
[ym,mm,dm] = ymd(MinMinTemperaturePerDay.date_time(r));
Lv = ismember([yhn,mhn], [ym,mm], 'rows');
idx = MinTemperaturePerMinute.Temperature(Lv) == MinMinTemperaturePerDay.Temperature(r);
Mv = find(Lv);
Dv = find(idx);
HM = MinTemperaturePerMinute(Mv(Dv(1)),:);
AbsMonthlyMin(r,:) = timetable2table(HM);
end
AbsMonthlyMin
t12=timetable(date_time, Diravg);
DT = t12.date_time;
WindDir = t12.Diravg;
TT = timetable(DT,WindDir);
WindConditions_AbsMonthlyMin=TT.WindDir(TT.DT(AbsMonthlyMin.date_time));
t8 =
table(AbsMonthlyMin.date_time.Year,AbsMonthlyMin.date_time.Month,AbsMonthlyMin.date_time.Day,A
bsMonthlyMin.date_time.Hour,AbsMonthlyMin.date_time.Minute,
AbsMonthlyMin.Temperature,WindConditions_AbsMonthlyMin, 'VariableNames',{'Year','Month','Day','
Hour','Minute','MinMIN Temp','Dir'})
FNAMetable16 = [station '_2b_Monthly TEMP_Times of Min-MIN_' year '.csv'];
FNOUtable16=[MINUTEmatrixDIR FNAMetable16];
writetable(t8, FNOUtable16);

```

```

%Απολύτως μηνιαία Μέγιστη και χρόνος και συνθήκες ανέμου που σημειώθηκε.
MaxMaxTemperaturePerDay=retime(t5, 'monthly', 'max'); %% μέγιστη των μεγιστων
MaxMaxTemperaturePerDay.date_time.Format = "dd-MMM-yy";
MaxTemperaturePerMinute = retime(t5, 'minutely', 'max');
[yhn,mhn,dhn] = ymd(MaxTemperaturePerMinute.date_time);%% Year, Month, Day Of Daily Maxima
for r = 1:numel(MaxMaxTemperaturePerDay)
[ym,mm,dm] = ymd(MaxMaxTemperaturePerDay.date_time(r));
Lv = ismember([yhn,mhn], [ym,mm], 'rows');
idx = MaxTemperaturePerMinute.Temperature(Lv) == MaxMaxTemperaturePerDay.Temperature(r);
Mv = find(Lv);
Dv = find(idx);
HM = MaxTemperaturePerMinute(Mv(Dv(1)),:);
AbsMonthlyMax(r,:) = timetable2table(HM);
end
AbsMonthlyMax
WindConditions_AbsMonthlyMax=TT.WindDir(TT.DT(AbsMonthlyMax.date_time));
t9 =
table(AbsMonthlyMax.date_time.Year,AbsMonthlyMax.date_time.Month,AbsMonthlyMax.date_time.Day,AbsMonthlyMax.date_time.Hour,AbsMonthlyMax.date_time.Minute,AbsMonthlyMax.Temperature,WindConditions_AbsMonthlyMax,'VariableNames',{'Year','Month','Day','Hour','Minute','MaxMAX Temp','Dir'});
FNAMEtable17 = [station '_2c_Monthly TEMP_Times of Max-MAX_' year '.csv'];
FNOUtable17=[MINUTEmatrixDIR FNAMEtable17];
writetable(t9, FNOUtable17);
% Μέσο, μέγιστο και ελάχιστο ημερήσιο θερμοκρασιακό εύρος κάθε μήνα (με χρονικό προσδιορισμό για το μέγιστο και ελάχιστο).
DailyMin = retime(t5, 'daily', 'min');
MinutelyMin = retime(t5, 'minutely', 'min');
MonthDayMin = DailytoMinutely(DailyMin,MinutelyMin)
DailyMax = retime(t5, 'daily', 'max');
MinutelyMax = retime(t5, 'minutely', 'max');
MonthDayMax =DailytoMinutely(DailyMax,MinutelyMax)
DailyRange = MonthDayMax.Temperature - MonthDayMin.Temperature;
DailyRange = table(MonthDayMin.date_time, DailyRange);
DailyRange.Properties.VariableNames = {'date_time','Temperature'};
DailyRange = table2timetable(DailyRange);
MeanDailyRange = retime(DailyRange, 'monthly', 'mean')
MaxDailyRange = retime(DailyRange, 'monthly', 'max')
MinDailyRange = retime(DailyRange, 'monthly', 'min')
t10 =
table(MeanDailyRange.date_time.Year,MeanDailyRange.date_time.Month,MeanDailyRange.Temperature,MaxDailyRange.Temperature,MinDailyRange.Temperature,'VariableNames',{'Year','Month','AV Daily Range','MAX Daily Range','MIN Daily Range'});
FNAMEtable18 = [station '_2f_Monthly TEMP_MIn-Max-AV Daily Range_' year '.csv'];
FNOUtable18=[MINUTEmatrixDIR FNAMEtable18];
writetable(t10, FNOUtable18);
%Αριθμός ημερών του μήνα με θερμοκρασία μικρότερη 6 κατωφλίων (ενδεικτικά των 10,5,0,-1,-2,-5 °C) οριζόμενων από τον χρήστη για κάθε μήνα ξεχωριστά.
MinTemperaturePerDay=retime(t5,'daily','min')
Threshold_cold = [-5;-2;-1;0;5;10;0]; %% Το τελευταίο να είναι πάντα 0 !
for k3 = 1:12
Lv = month(MinTemperaturePerDay.date_time) == k3;
Month = retime(MinTemperaturePerDay(Lv,:), 'daily', 'mean');
MMM{k3,:} = month(MinTemperaturePerDay(Lv,:).date_time(1), 'shortname');
for k4 = 1:numel(Threshold_cold)-1

```

```

Temp_c = Month.Temperature;
DaysMeetingThreshold_cold(k3,k4) = nnz((Temp_c >= Threshold_cold(k4)) & (Temp_c <
Threshold_cold(k4+1)));
end
end
Total_DaysThreshold_cold = array2table(DaysMeetingThreshold_cold,
'VariableNames',compose('%.2f',Threshold_cold(1:end-1)));
%Αριθμός ημερών του μήνα με θερμοκρασία μεγαλύτερη 6 κατωφλίων (ενδεικτικά των
25,30,35,38,40,42 °C) οριζόμενων απο των χρήστη για κάθε μήνα ξεχωριστά.
MaxTemperaturePerDay=retime(t5,'daily','max')
Threshold_warm = [25;30;35;38;40;42;0]; %% Το τελευταίο να είναι πάντα 0 !
for k1 = 1:12
Lv = month(MaxTemperaturePerDay.date_time) == k1;
Month = retime(MaxTemperaturePerDay(Lv,:), 'daily','mean');
MMM{k1,:} = month(MaxTemperaturePerDay(Lv,:).date_time(1),'shortname');
for k2 = 1:numel(Threshold_warm)-1
Hmdt = Month.Temperature;
DaysMeetingThreshold_warm(k1,k2) = nnz((Hmdt >= Threshold_warm(k2)) & (Hmdt <
Threshold_warm(k2+1)));
end
end
Total_DaysThreshold_warm = array2table(DaysMeetingThreshold_warm,
'VariableNames',compose('%.2f',Threshold_warm(1:end-1)));
TempThresholds = [(array2table(MMYYYTtbl)) Total_DaysThreshold_cold Total_DaysThreshold_warm];
TempThresholds.Properties.VariableNames{1} = 'Year';
TempThresholds.Properties.VariableNames{2} = 'Month';
FNAMEtable19 = [station '_2e_Monthly TEMP_NofDays with Thres Exceed_' year '.csv'];
FNOUtable19=[MINUTEmatrixDIR FNAMEtable19];
writetable(TempThresholds, FNOUtable19);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% W I N D: %%%%%%%%%%%%%%
% Μέσος και τυπική απόκλιση των μέσων ανά λεπτό ταχυτήτων ανέμου και ριπών, για κάθε μήνα
t11=timetable(date_time, Vel_avg); %% Wind speed σε m/s = μέτρο ταχύτητας Vel avg (τυπικά
στην στήλη E στα ημερήσια αρχεία), καθε λεπτο βγαινει μια μεση ταχυτητα ανεμου (από 60
μετρησεις ανα λεπτο)
MeanWindSpeed_Per_Month = retime(t11, 'monthly', 'mean'); %% Μέση μηνιαία ταχυτητα ανεμου
MeanWindSpeed_Per_Minute = retime(t11, 'minutely', 'mean'); %% Μέση ημερήσια ταχυτητα ανεμου
MeanWindSpeed_Per_Minute = rmmissing(MeanWindSpeed_Per_Minute); %% Remove possible NaN
MeanWindSpeed_Per_Month_STD= retime(MeanWindSpeed_Per_Minute, 'monthly', @std); %% Calculate
the standard deviation
Mean_Vel_avg = table(MeanWindSpeed_Per_Month.date_time.Year,
MeanWindSpeed_Per_Month.date_time.Month,MeanWindSpeed_Per_Month.Vel_avg,
MeanWindSpeed_Per_Month_STD.Vel_avg , 'VariableNames',{'Year','Month','Vel_avg Mean','Vel_avg
SDV'})
Mean_Vel_avg=splitvars(Mean_Vel_avg);
t12=timetable(date_time, Diravg); %% Wind direction σε deg = κατεύθυνση ταχύτητας Dir avg
(τυπικά στην στήλη I στα ημερήσια αρχεία)
MeanWindGust_Per_Month = retime(t12, 'monthly', 'mean'); %% Μέση μηνιαία ριπή ανεμου
MeanWindGust_Per_Minute = retime(t12, 'minutely', 'mean'); %% Μέση ημερήσια ριπή ανεμου
MeanWindGust_Per_Minute = rmmissing(MeanWindGust_Per_Minute); %% Remove possible NaN
MeanWindGust_Per_Month_STD= retime(MeanWindGust_Per_Minute, 'monthly', @std); %% Calculate
the standard deviation
Mean_WindGust= table(MeanWindGust_Per_Month.Diravg, MeanWindGust_Per_Month_STD.Diravg ,
'VariableNames',{'Dir_avg Mean','Dir_avg SDV'})

```

```

Mean_WindGust=splitvars(Mean_WindGust);
Mean_Vel_avg_WindGust = [Mean_Vel_avg Mean_WindGust]
FNAMEtable20 = [station '_3a_Monthly WIND Means and SDVs_' year '.csv'];
FNOUtable20=[MINUTEmatrixDIR FNAMEtable20];
writetable(Mean_Vel_avg_WindGust, FNOUtable20);
%▪ ok ▪ Προσδιορισμός των 10 ισχυρότερων ριπών ανέμου (μέτρο - κατεύθυνση) του μήνα και χρόνος
που σημειώθηκαν.
t111=timetable(date_time, Velmax);
Tx1=timetable2table(t111);
for k = 1:12
    Lv1 = month(Tx1.date_time) == k;
    M1 = Tx1(Lv1,:);
    [~,idx] = maxk(M1{:},2),10);
    Max10_Velmax{k} = M1(idx,:);
end
Max10_Velmax= [Max10_Velmax{1, 1}(:,1:2);Max10_Velmax{1, 2}(:,1:2);Max10_Velmax{1,
3}(:,1:2);Max10_Velmax{1, 4}(:,1:2);Max10_Velmax{1, 5}(:,1:2);Max10_Velmax{1,
6}(:,1:2);Max10_Velmax{1, 7}(:,1:2);Max10_Velmax{1, 8}(:,1:2);Max10_Velmax{1,
9}(:,1:2);Max10_Velmax{1, 10}(:,1:2);Max10_Velmax{1, 11}(:,1:2);Max10_Velmax{1, 12}(:,1:2)]
Max10_Velmax.Properties.VariableNames = {'date_time', 'Velmax'};
DT = t12.date_time;
WindDir = t12.Diravg;
TT = timetable(DT,WindDir);
Dir_Max10Velmax=TT.WindDir(TT.DT(Max10_Velmax.date_time));
Max10_VelmaxT = table2timetable (Max10_Velmax)
Max10 =
table(Max10_VelmaxT.date_time.Year,Max10_VelmaxT.date_time.Month,Max10_VelmaxT.date_time.Day,M
ax10_VelmaxT.date_time.Hour,Max10_VelmaxT.date_time.Minute,Max10_Velmax.Velmax,Dir_Max10Velmax
, 'VariableNames',{'Year', 'Month', 'Day', 'Hour', 'Minute', 'Velmax', 'Dir'})
FNAMEtable21 = [station '_3c_Monthly WIND_10 Max Gusts_' year '.csv'];
FNOUtable21=[MINUTEmatrixDIR FNAMEtable21];
writetable(Max10, FNOUtable21);
%Κατανομή μέσων ανα λεπτό ταχυτήτων ανέμου και ριπών λεπτού σε κλάσεις 1 m/s, κατά μήνα, και
παράμετροι κατανομής Weibull.
MM = month(t11.date_time);
T11 = timetable2table(t11);
Vmax = 60;
edgev = linspace(0, Vmax, Vmax+1);           %% Edge Vector
edgem = [edgev(1:end-1); edgev(2:end)].';    %% Bin Limits (For Output Table)
ctrs = mean(diff(edgev))/2 + edgev(1:end-1); %% Bin Centers
[N,Edges,mps] = histcounts(T11.Vel_avg, edgev); %% Call 'histcounts'
WindVelCounts = table(edgem(:,1),edgem(:,2)); %% Initialise Table
for k = 1:12
    Month = T11(MM==k,:);
    [N,Edges,mps] = histcounts(T11.Vel_avg(MM==k,:), edgev); %% Call 'histcounts'
    Nc{k} = N;
    Dch{k+2} = mps;
    WindVelCounts{:,k+2} = N(:);
    MMM{k} = month(Month{1,1}, 'shortname');
end
end
MMMv = [MMM{:}];
WindVelCounts.Properties.VariableNames = {'Start', 'End', MMMv{:}}
VN = WindVelCounts.Properties.VariableNames;
WindVelCountsT = array2table(table2array(WindVelCounts(:,3:end)).', 'RowNames',VN(3:end));
WindVelCountsT.Properties.VariableNames = compose('%d-%d',edgem)

```

```

FNAMEtable22 = [station '_3e_Monthly WIND GUST Distribution Classes_' year '.csv'];
FNOUtable22=[MINUTEmatrixDIR FNAMEtable22];
writetable(WindVelCountsT, FNOUtable22);

MM12 = month(MeanWindGust_Per_Minute.date_time);
T12 = timetable2table(MeanWindGust_Per_Minute);
Dirmax = 60;
edgev = linspace(0, Dirmax, Dirmax+1);           %% Edge Vector
edgem = [edgev(1:end-1); edgev(2:end)].';       %% Bin Limits (For Output Table)
ctrs = mean(diff(edgev))/2 + edgev(1:end-1);    %% Bin Centers
[N,Edges,mps] = histcounts(T12.Diravg, edgev);   %% Call 'histcounts'
Windir_counts = table(edgem(:,1),edgem(:,2));    %% Initialise Table
for k = 1:12
    Month = T12(MM12==k,:);
    [N,Edges,mps] = histcounts(T12.Diravg(MM12==k,:), edgev); %% Call 'histcounts'
    Nc{k} = N;
    Dch{k+2} = mps;
    Windir_counts{:,k+2} = N(:);
    MMM{k} = month(Month{1,1}, 'shortname');
end
MMMv = [MMM{:}];
Windir_counts.Properties.VariableNames = {'Start', 'End', MMMv{:}}
VN = WindVelCounts.Properties.VariableNames;
WindDirCountsT = array2table(table2array(Windir_counts(:,3:end)).', 'RowNames', VN(3:end));
WindDirCountsT.Properties.VariableNames = compose('%d-%d', edgem)
FNAMEtable23 = [station '_3d_Monthly AV WIND Distribution Classes_' year '.csv'];
FNOUtable23=[MINUTEmatrixDIR FNAMEtable23];
writetable(WindDirCountsT, FNOUtable23);
    %Ιστόγραμμα συχνοτήτων ταχυτήτων ανεμου ανά μήνα:
figure(1)
TL = tiledlayout(4,3);
for k = 1:12
    nexttile
    bar(ctrs, WindVelCounts{:,k+2})
    grid
    ylim([0 1.8E+4])
    title(MMM{k})
end
FNAMEtable24 = [station '_7b_Monthly Wind Velocity Histograms_' year];
FNOUtable24=[MINUTEmatrixDIR FNAMEtable24];
saveas(figure(1), FNOUtable24, 'jpg')
    %Ιστόγραμμα συχνοτήτων ταχυτήτων ανεμου για όλο το έτος:
figure(2)
bar(ctrs, N)
grid
xlabel('Velocity')
ylabel('Counts')
FNAMEtable25 = [station '_7a_Wind Velocity Histograms_' year];
FNOUtable25=[MINUTEmatrixDIR FNAMEtable25];
saveas(figure(2), FNOUtable25, 'jpg')
% WEIBULL distribution
probplot('weibull', Diravg);figure(gcf)
set(gca, 'yminorgrid', 'on')
set(gca, 'xminorgrid', 'on')
legend('Diravg', 'Location', 'best')

```



```

figure(gcf)
grid on
FNAMEtable26 = [station '_7c_Weibull_' year];
FNOUtable26=[MINUTEmatrixDIR FNAMEtable26];
saveas(gcf, FNOUtable26, 'jpg')
savefig('weibull.fig')
h=openfig('weibull.fig');
handles=findobj(h,'Type','line');
x=get(handles(1),'Xdata');
y=get(handles(1),'Ydata');
[param,ci] = wblfit(MeanWindSpeed_Per_Month.Vel_avg);
wbl1=[param;ci];
wbl1=array2table(wbl1);
wbl1.Properties.VariableNames = {'param','ci'};
for aa = 1:12
    MMMidx = month(MeanWindGust_Per_Minute.date_time) == aa; % Monthly timetables from
MeanWindGust_Per_Minute
    MeanWindGust_Per_MinuteMonth{aa,:} = MeanWindGust_Per_Minute(MMMidx,:);
end
MeanWindGust_Per_MinuteMonth{1}.Diravg
for i = 1:12
    weibull{i}=wblfit(MeanWindGust_Per_MinuteMonth{i}.Diravg);
end
weibull=weibull';
weibull=cell2table(weibull);
weibull=splitvars(weibull);
weibullT=table(MeanWindGust_Per_Month.date_time.Year,MeanWindGust_Per_Month.date_time.Month,we
ibull.weibull_2,weibull.weibull_1, 'VariableNames', {'Year','Month','Shape','Scale'})
FNAMEtable27 = [station '_3f_Monthly WIND_Weibull Param_' year '.csv'];
FNOUtable27=[MINUTEmatrixDIR FNAMEtable27];
writetable(weibullT, FNOUtable27);
% Κατανομή συχνότητας κατευθύνσεων σε 16 τομείς, ανά μήνα.
DT = t12.date_time;
WindDir = t12.Diravg;
TT = timetable(DT,WindDir)
MM = month(TT.DT);
T = timetable2table(TT);
edgev = linspace(0, 360, 17);
edgem = [edgev(1:end-1); edgev(2:end)].';
ctrs = mean(diff(edgev))/2 + edgev(1:end-1);
[N,Edges,Decihexctant] = histcounts(WindDir, edgev);
WindDirCounts = table(edgem(:,1),edgem(:,2),'VariableNames',{'Start','End'});
for k = 1:12
    Month = T(MM==k,:);
    [N,Edges,Decihexctant] = histcounts(T.WindDir(MM==k,:), edgev);
    Nc{k} = N;
    Dch{k+2} = Decihexctant;
    WindDirCounts(:,k+2) = N(:);
    MMM{k} = month(Month{1,1},'shortname');
end
WindDirCounts.Properties.VariableNames =
{'Start','End','Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'}
FNAMEtable28 = [station '_3b_Monthly WIND_16 Dir Sectors_' year '.csv'];
FNOUtable28=[MINUTEmatrixDIR FNAMEtable28];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% H U M I D I T Y: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Μέση σχετική, Μέση απόλυτη, και τυπικές αποκλίσεις ανά μήνα.
t13=timetable(date_time, RH);      %% Η σχετική υγρασία είναι στην στήλη 0 με ονομα RH σε %.
MeanRelativeHumidityPerMonth = retime(t13, 'monthly', 'mean'); %% Μέση μηνιαία σχετική
υγρασία
MeanRelativeHumidityPerDay = retime(t13, 'daily', 'mean');      %% Μέση ημερήσια σχετική
υγρασία
MeanRelativeHumidityPerMonth_STD= retime(MeanRelativeHumidityPerDay, 'monthly', @std); %%
Calculate the standard deviation per month
t14=timetable(date_time, AH); %% Η απόλυτη υγρασία (όπου υπάρχει) βρίσκεται στην στήλη AD (υπο
τον τίτλο AH ).
MeanAbsoluteHumidityPerMonth = retime(t14, 'monthly', 'mean'); %% Μέση μηνιαία απόλυτη
υγρασία
MeanAbsoluteHumidityPerDay = retime(t14, 'daily', 'mean');      %% Μέση ημερήσια σχετική υγρασία
MeanAbsoluteHumidityPerMonth_STD= retime(MeanAbsoluteHumidityPerDay, 'monthly', @std); %%
Calculate the standard deviation per month
MeanRH_AH_SDs = table
(MeanRelativeHumidityPerMonth.date_time.Year,MeanRelativeHumidityPerMonth.date_time.Month,Mean
RelativeHumidityPerMonth.RH,
MeanRelativeHumidityPerMonth_STD.RH,MeanAbsoluteHumidityPerMonth.AH,MeanAbsoluteHumidityPerMon
th_STD.AH, 'VariableNames', {'Year', 'Month', 'RH_Mean', 'RH_SD', 'AH_Mean', 'AH_SD'})
FNAMetable29= [station '_5b_Monthly_HUM_Mean_SDV_' year '.csv'];
FNOUTable29=[MINUTEmatrixDIR FNAMetable29];
writetable(MeanRH_AH_SDs, FNOUtable29);

% Μέγιστη σχετική, ελάχιστη σχετική και απόλυτη του μήνα, χρόνοι και συνθήκες ανέμου που
σημειώθηκαν.
MaxRelativeHumidityPerMonth = retime(t13, 'monthly', 'max'); %% Μέγιστη μηνιαία σχετική
υγρασία
MinRelativeHumidityPerMonth = retime(t13, 'monthly', 'min'); %% Ελάχιστη μηνιαία σχετική
υγρασία
MaxAbsoluteHumidityPerMonth = retime(t14, 'monthly', 'max'); %% Μέγιστη μηνιαία απόλυτη
υγρασία
MinAbsoluteHumidityPerMonth = retime(t14, 'monthly', 'min'); %% Ελάχιστη μηνιαία απόλυτη
υγρασία
MaxRelativeHumidityPerMinute = retime(t13, 'minutely', 'max'); %% Μέγιστη ωριαία σχετική
υγρασία
MinRelativeHumidityPerMinute = retime(t13, 'minutely', 'min'); %% Ελάχιστη ωριαία σχετική
υγρασία
[yhd,mhd,dhd] = ymd(MinRelativeHumidityPerMinute.date_time); %% Year, Month, Day Of Daily
Maxima
for k = 1:numel(MinRelativeHumidityPerMonth)
    [ym,mm,dm] = ymd(MinRelativeHumidityPerMonth.date_time(k));
    Lv = ismember([yhd,mhd], [ym,mm], 'rows');
    idx = MinRelativeHumidityPerMinute.RH(Lv) == MinRelativeHumidityPerMonth.RH(k);
    Mv = find(Lv);
    Dv = find(idx);
    HM = MinRelativeHumidityPerMinute(Mv(Dv(1)),:);
    RH_MonthDayMin(k,:) = timetable2table(HM);
end
RH_MonthDayMin      %% Ελάχιστη μηνιαία σχετική υγρασία
WindConditionsRH_MonthDayMin=TT.WindDir(TT.DT(RH_MonthDayMin.date_time)) % Συνθήκες ανέμου που
σημειώθηκε Ελάχιστη μηνιαία σχετική υγρασία
RH_MonthDayMinT =
table(RH_MonthDayMin.date_time.Year,RH_MonthDayMin.date_time.Month,RH_MonthDayMin.date_time.Da

```

```

y,RH_MonthDayMin.date_time.Hour,RH_MonthDayMin.date_time.Minute,RH_MonthDayMin.RH,
WindConditionsRH_MonthDayMin,
'VariableNames',{'RH_Min_Year','RH_Min_Month','RH_Min_Day','RH_Min_Hour','RH_Min_Minute','RH_Min',
'WindDir of RH_Min'})

```

```

[yhc,mhc,dhc] = ymd(MaxRelativeHumidityPerMinute.date_time); %% Year, Month, Day Of Daily
Maxima

```

```

for k = 1:numel(MaxRelativeHumidityPerMonth)
    [ym,mm,dm] = ymd(MaxRelativeHumidityPerMonth.date_time(k));
    Lv = ismember([yhc,mhc],[ym,mm],'rows');
    idx = MaxRelativeHumidityPerMinute.RH(Lv) == MaxRelativeHumidityPerMonth.RH(k);
    Mv = find(Lv);
    Dv = find(idx);
    HM = MaxRelativeHumidityPerMinute(Mv(Dv(1)),:);
    RH_MonthDayMax(k,:) = timetable2table(HM);
end

```

```

RH_MonthDayMax %% Μέγιστη μηνιαία σχετική υγρασία

```

```

WindConditionsRH_MonthDayMax=TT.WindDir(TT.DT(RH_MonthDayMax.date_time))%Συνθήκες ανέμου που
σημειώθηκε Μέγιστη μηνιαία σχετική υγρασία

```

```

RH_MonthDayMaxT =

```

```

table(RH_MonthDayMax.date_time.Year,RH_MonthDayMax.date_time.Month,RH_MonthDayMax.date_time.Da
y,RH_MonthDayMax.date_time.Hour,RH_MonthDayMax.date_time.Minute,RH_MonthDayMax.RH,
WindConditionsRH_MonthDayMax,
'VariableNames',{'RH_Max_Year','RH_Max_Month','RH_Max_Day','RH_Max_Hour','RH_Max_Minute','RH_M
ax','WindDir of RH_Max'})

```

```

MinAbsoluteHumidityPerMinute = retime(t14, 'minutely', 'min'); %% Ελάχιστη ωριαία απόλυτη
υγρασία

```

```

[yhh,mhh,dhh] = ymd(MinAbsoluteHumidityPerMinute.date_time); %% Year, Month, Day Of Daily
Maxima

```

```

for k = 1:numel(MinAbsoluteHumidityPerMonth)
    [ym,mm,dm] = ymd(MinAbsoluteHumidityPerMonth.date_time(k));
    Lv = ismember([yhh,mhh],[ym,mm],'rows');
    idx = MinAbsoluteHumidityPerMinute.AH(Lv) == MinAbsoluteHumidityPerMonth.AH(k);
    Mv = find(Lv);
    Dv = find(idx);
    HM = MinAbsoluteHumidityPerMinute(Mv(Dv(1)),:);
    AH_MonthDayMin(k,:) = timetable2table(HM);
end

```

```

AH_MonthDayMin %% Ελάχιστη μηνιαία απόλυτη υγρασία

```

```

WindConditionsAH_MonthDayMin=TT.WindDir(TT.DT(AH_MonthDayMin.date_time) % Συνθήκες ανέμου που
σημειώθηκε Ελάχιστη μηνιαία απόλυτη υγρασία

```

```

AH_MonthDayMinT =

```

```

table(AH_MonthDayMin.date_time.Year,AH_MonthDayMin.date_time.Month,AH_MonthDayMin.date_time.Da
y,AH_MonthDayMin.date_time.Hour,AH_MonthDayMin.date_time.Minute,AH_MonthDayMin.AH,
WindConditionsAH_MonthDayMin,
'VariableNames',{'AH_Min_Year','AH_Min_Month','AH_Min_Day','AH_Min_Hour','AH_Min_Minute','AH_M
in','WindDir of AH_Min'})

```

```

MaxAbsoluteHumidityPerMinute = retime(t14, 'minutely', 'max'); %% Μέγιστη ωριαία απόλυτη
υγρασία

```

```

[yhg,mhg,dhg] = ymd(MaxAbsoluteHumidityPerMinute.date_time); %% Year, Month, Day Of Daily
Maxima

```

```

for k = 1:numel(MaxAbsoluteHumidityPerMonth)
    [ym,mm,dm] = ymd(MaxAbsoluteHumidityPerMonth.date_time(k));
end

```

```

Lv = ismember([yhg,mhg], [ym,mm], 'rows');
idx = MaxAbsoluteHumidityPerMinute.AH(Lv) == MaxAbsoluteHumidityPerMonth.AH(k);
Mv = find(Lv);
Dv = find(idx);
HM = MaxAbsoluteHumidityPerMinute(Mv(Dv(1)),:);
AH_MonthDayMax(k,:) = timetable2table(HM);
end
AH_MonthDayMax    %% Μέγιστη μηνιαία απόλυτη υγρασία
WindConditionsAH_MonthDayMax=TT.WindDir(TT.DT(AH_MonthDayMax.date_time)) % Συνθήκες ανέμου που
σημειώθηκε Μέγιστη μηνιαία απόλυτη υγρασία
AH_MonthDayMaxT =
table(AH_MonthDayMax.date_time.Year,AH_MonthDayMax.date_time.Month,AH_MonthDayMax.date_time.Da
y,AH_MonthDayMax.date_time.Hour,AH_MonthDayMax.date_time.Minute,AH_MonthDayMax.AH,
WindConditionsAH_MonthDayMax,
'VariableNames',{'AH_Max_Year','AH_Max_Month','AH_Max_Day','AH_Max_Hour','AH_Max_Minute','AH_M
ax','WindDir of AH_Max'})
Min_Max_RH_AH_WindDir=table(RH_MonthDayMinT,RH_MonthDayMaxT,AH_MonthDayMinT,AH_MonthDayMaxT);
Min_Max_RH_AH_WindDir = splitvars(Min_Max_RH_AH_WindDir);
FNAMETable30 = [station '_5a_Monthly HUM_Times of Min-Max_RH_AH_WindDir_' year '.csv'];
FNOUTable30=[MINUTEmatrixDIR FNAMETable30];
writetable(Min_Max_RH_AH_WindDir, FNOUTable30);
%Αριθμός ημερών ανά μήνα με σχετική υγρασία μεγαλύτερη ή ίση του 95 και του 100%
MaxRelativeHumidityPerDay = retime(t13, 'daily', 'max'); %% Μέγιστη ημερήσια σχετική υγρασία
MaxRelativeHumidityPerDay.date_time.Format = "MMM-yy"
Threshold_hmdt= [95;100;0]; %% Το τελευταίο να είναι πάντα 0 !
for k1 = 1:12
    Lv = month(MaxRelativeHumidityPerDay.date_time) == k1;
    Month = retime(MaxRelativeHumidityPerDay(Lv,:), 'daily', 'mean');
for k2 = 1:numel(Threshold_hmdt)-1
    Hmdt = Month.RH;
    DaysMeetingThreshold_hmdt(k1,k2) = nnz((Hmdt >= Threshold_hmdt(k2)) & (Hmdt <
Threshold_hmdt(k2+1)));
end
end
DaysMeetingThreshold = array2table(DaysMeetingThreshold_hmdt,
'VariableNames',compose('% .2f',Threshold_hmdt(1:end-1)))
DaysMeetingThreshold = table
(MaxRelativeHumidityPerMonth.date_time.Year,MaxRelativeHumidityPerMonth.date_time.Month,DaysMe
etingThreshold.("95.00"),DaysMeetingThreshold.("100.00"), 'VariableNames',
{'Year','Month','95%','100%'})
FNAMETable31 = [station '_5d_Monthly_HUM_NofDays with RH Thresh Exceed_' year '.csv'];
FNOUTable31=[MINUTEmatrixDIR FNAMETable31];
writetable(DaysMeetingThreshold, FNOUTable31);

%Αριθμός ημερών ανά μήνα με απόλυτη υγρασία μεγαλύτερη του 2σ ανά μήνα.
MaxAbsoluteHumidityPerDay = retime(t14, 'daily', 'max'); %% Μέγιστη ημερήσια απόλυτη υγρασία
MaxAbsoluteHumidityPerDay.date_time.Format = "MMM-yy"
Threshold_hmdt2SD= 2*(MeanAbsoluteHumidityPerMonth_STD.AH)%% εμφάνιση/αναφορά του 2σ ανά μήνα
Threshold_hmdt2SD(13)=0;
for k3 = 1:12
    Lv = month(MaxAbsoluteHumidityPerDay.date_time) == k3;
    Month = retime(MaxAbsoluteHumidityPerDay(Lv,:), 'monthly', 'mean');
for k4 = 1:numel(Threshold_hmdt2SD)-1
    Hmdt = Month.AH;

```

```

DaysMeetingThreshold_2SD(k3,k4) = nnz((Hmdt >= Threshold_hmdt2SD(k4)) & (Hmdt <
Threshold_hmdt2SD(k4+1)));
end
end
DaysMeetingThreshold_hmdt2SD = array2table(DaysMeetingThreshold_2SD,
'VariableNames',compose('% .2f',Threshold_hmdt2SD(1:end-1)))
DMT2SD=table(MaxAbsoluteHumidityPerMonth.date_time.Year,MaxAbsoluteHumidityPerMonth.date_time.
Month,Threshold_hmdt2SD(1:12), 'VariableNames', {'Year','Month','AH 2sigma'})
DaysMeetingThreshold_hmdt2SDT=[DMT2SD DaysMeetingThreshold_hmdt2SD]
FNAMetable32 = [station '_5c_Monthly HUM_NofDays of AH 2sigma_Exceed_' year '.csv'];
FNOUTtable32=[MINUTEmatrixDIR FNAMetable32];
writetable(DaysMeetingThreshold_hmdt2SDT, FNOUTtable32);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% P R E S S U R E : %%%%%%%%%%

```

```

% Μέση, μέγιστη, ελάχιστη πίεση μήνα και χρόνοι που σημειώθηκαν οι ακραίες.

```

```

t15=timetable(date_time, Pressure);format long g    %% Preferences -> Variables -> Default

```

```

array format: long g

```

```

MeanPressurePerMonth = retime(t15, 'monthly', 'mean'); %% Μέση μηνιαία πίεση
MaxPressurePerMonth = retime(t15, 'monthly', 'max');    %% Μέγιστη μηνιαία πίεση
MaxPressurePerDay = retime(t15, 'daily', 'max');        %% Μέγιστη ημερήσια πίεση
MaxPressurePerMinute = retime(t15, 'minutely', 'max'); %% Μέγιστη πίεση ανά λεπτό
[yhx,mhx,dhx] = ymd(MaxPressurePerMinute.date_time);    %% Year, Month, Day Of Daily Maxima

```

```

for k = 1:numel(MaxPressurePerMonth)

```

```

    [ym,mm,dm] = ymd(MaxPressurePerMonth.date_time(k));

```

```

    Lv = ismember([yhx,mhx], [ym,mm], 'rows');

```

```

    idx = MaxPressurePerMinute.Pressure(Lv) == MaxPressurePerMonth.Pressure(k);

```

```

    Mv = find(Lv);

```

```

    Dv = find(idx);

```

```

    HM = MaxPressurePerMinute(Mv(Dv(1)),:);

```

```

    PressureMonthDayMax(k,:) = timetable2table(HM);

```

```

end

```

```

PressureMonthDayMax

```

```

MinPressurePerMonth = retime(t15, 'monthly', 'min'); %% Ελάχιστη μηνιαία πίεση

```

```

MinPressurePerDay = retime(t15, 'daily', 'min');      %% Ελάχιστη ημερήσια πίεση

```

```

MinPressurePerMinute = retime(t15, 'minutely', 'min'); %% Ελάχιστη πίεση ανά λεπτό

```

```

[yhn,mhn,dhn] = ymd(MinPressurePerMinute.date_time); %% Year, Month, Day Of Daily Maxima

```

```

for k = 1:numel(MinPressurePerMonth)

```

```

    [ym,mm,dm] = ymd(MinPressurePerMonth.date_time(k));

```

```

    Lv = ismember([yhn,mhn], [ym,mm], 'rows');

```

```

    idx = MinPressurePerMinute.Pressure(Lv) == MinPressurePerMonth.Pressure(k);

```

```

    Mv = find(Lv);

```

```

    Dv = find(idx);

```

```

    HM = MinPressurePerMinute(Mv(Dv(1)),:);

```

```

    PressureMonthDayMin(k,:) = timetable2table(HM);

```

```

end

```

```

PressureMonthDayMin

```

```

MeanMaxMin = table

```

```

(PressureMonthDayMax.date_time.Year,PressureMonthDayMax.date_time.Month,PressureMonthDayMax.da
te_time.Day,PressureMonthDayMax.date_time.Hour,PressureMonthDayMax.date_time.Minute,

```

```

PressureMonthDayMax.Pressure,

```

```

PressureMonthDayMin.date_time.Year,PressureMonthDayMin.date_time.Month,PressureMonthDayMin.dat
e_time.Day,PressureMonthDayMin.date_time.Hour,PressureMonthDayMin.date_time.Minute,PressureMon

```

```

thDayMin.Pressure, 'VariableNames', {'Year', 'Month', 'Day', 'Hour', 'Minute', 'Max
Pressure', 'Year_', 'Month_', 'Day_', 'Hour_', 'Minute_', 'Min Pressure'})
FNAMEtable33 = [station '_6a_Monthly PRES_Times of Min-Max_' year '.csv'];
FNOUtable33=[MINUTEmatrixDIR FNAMEtable33];
writetable(MeanMaxMin, FNOUtable33);
% Αριθμός ημερών μήνα με χαμηλή (<1013.25) και υψηλή (>1013.25) πίεση.
PressurePerDay = retime(t15, 'daily', 'mean');           %% Μέση ημερήσια πίεση
Threshold_pressure= [1013.25;1013.25;0];                %% Το τελευταίο να είναι πάντα 0 !
for k5 = 1:12
    Lv = month(PressurePerDay.date_time) == k5;
    Month = retime(PressurePerDay(Lv,:), 'daily', 'mean');
for k6 = 1:numel(Threshold_pressure)-1
    Prss = Month.Pressure;
    DaysMeetingThreshold_Prss(k5,k6) = nnz((Prss < Threshold_pressure(k6)) & (Prss >
Threshold_pressure(k6+1)));
end
end
DaysHighLow = array2table(DaysMeetingThreshold_Prss)
DaysHighLow = table(MinPressurePerMonth.date_time.Year, MinPressurePerMonth.date_time.Month,
DaysHighLow.DaysMeetingThreshold_Prss1, DaysHighLow.DaysMeetingThreshold_Prss2,
'VariableNames', {'Year', 'Month', '<1013.25', '>1013.25'});
FNAMEtable34 = [station '_6d_Monthly PRESS_Nof Days_with Thres Exceed_' year '.csv'];
FNOUtable34=[MINUTEmatrixDIR FNAMEtable34];
writetable(MeanMaxMin, FNOUtable34);
% Αριθμός ημερών μήνα με πίεση μεγαλύτερη απο 4 οριζόμενα απο τον χρήστη κατώφλια (ενδεικτικά
1025, 1030, 1035, 1040).
Threshold_prs_high= [1025;1030;1035;1040;0]; %% Το τελευταίο να είναι πάντα 0 !
for k7 = 1:12
    Lv = month(PressurePerDay.date_time) == k7;
    Month = retime(PressurePerDay(Lv,:), 'daily', 'mean');
for k8 = 1:numel(Threshold_prs_high)-1
    prss_high = Month.Pressure;
    DaysMeetingThreshold_prss_high(k7,k8) = nnz((prss_high > Threshold_prs_high(k8)));
end
end
Threshold_pressure_high =
array2table(DaysMeetingThreshold_prss_high, 'VariableNames', compose('% .2f', Threshold_prs_high(1
:end-1)))
TPH=table(MinPressurePerMonth.date_time.Year, MinPressurePerMonth.date_time.Month,
'VariableNames', {'Year', 'Month'})
Threshold_pressure_highT=[TPH Threshold_pressure_high]
FNAMEtable35 = [station '_6b_Monthly PRES_NofDays of UpperThresh Exceed_' year '.csv'];
FNOUtable35=[MINUTEmatrixDIR FNAMEtable35];
writetable(Threshold_pressure_highT, FNOUtable35);
% Αριθμός ημερών μήνα με πίεση μικρότερη απο 4 οριζόμενα απο τον χρήστη κατώφλια (ενδεικτικά
1000, 995, 990, 985).
Threshold_prs_low= [1000;995;990;985;0]; %% Το τελευταίο να είναι πάντα 0 !
for k9 = 1:12
    Lv = month(PressurePerDay.date_time) == k9;
    Month = retime(PressurePerDay(Lv,:), 'daily', 'mean');
for k10 = 1:numel(Threshold_prs_low)-1
    prss_low = Month.Pressure;
    DaysMeetingThreshold_prss_low(k9,k10) = nnz((prss_low < Threshold_prs_low(k10)));
end
end

```

```

Threshold_pressure_low =
array2table(DaysMeetingThreshold_prss_low, 'VariableNames', compose('%0.2f', Threshold_prs_low(1:en
nd-1)))
TPL=table(MinPressurePerMonth.date_time.Year,MinPressurePerMonth.date_time.Month,
'VariableNames', {'Year','Month'})
Threshold_pressure_lowT=[TPL Threshold_pressure_low]
FNAMetable36= [station '_6c_Monthly PRES_NofDays of LowThresh Exceed_' year '.csv'];
FNOUtable36=[MINUTEmatrixDIR FNAMetable36];
writetable(Threshold_pressure_low, FNOUtable36);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% S O L A R   I R R A D I A N C E: %%%%%%%%%%%

```

```

% Μέγιστη ημερήσια παροχή ισχύος απο την Ηλιακή ακτινοβολία και χρόνος που σημειώθηκε.

```

```

t16=timetable(date_time, SolarIrradiance);

```

```

MaxSolarIrradiancePerDay = retime(t16, 'daily', 'max'); %% Μέγιστη ημερήσια παροχή ισχύος απο
την Ηλιακή ακτινοβολία

```

```

MaxSolarIrradiancePerMinute = retime(t16, 'minutely', 'max'); %% Μέγιστη ανά λεπτό παροχή
ισχύος απο την Ηλιακή ακτινοβολία

```

```

SolarMonthDayMax = DaillytoMinutely(MaxSolarIrradiancePerDay,MaxSolarIrradiancePerMinute);

```

```

SolarMonthDayMaxT = table

```

```

(SolarMonthDayMax.date_time.Year,SolarMonthDayMax.date_time.Month,SolarMonthDayMax.date_time.D
ay,SolarMonthDayMax.date_time.Hour,SolarMonthDayMax.date_time.Minute,

```

```

SolarMonthDayMax.SolarIrradiance, 'VariableNames',

```

```

{'Year','Month','Day','Hour','Minute','SolarIrradiance'});

```

```

FNAMetable37= [station '_4a_SOLAR_Time of Max Daily Irradiance_2019_' year '.csv'];

```

```

FNOUtable37=[MINUTEmatrixDIR FNAMetable37];

```

```

writetable(SolarMonthDayMaxT, FNOUtable37);

```

```

%%%% Functions %%%

```

```

function out = iMaxAndLoc(t, v)

```

```

[maxV, idx] = max(v);

```

```

out = table(t(idx), maxV, 'VariableNames', {'Time', 'Value'});

```

```

end

```

```

function ExactTime = MonthlyToMinutely(MonthlyTT,MinutelyTT)

```

```

MinutelyVN = MinutelyTT.Properties.VariableNames;

```

```

MonthlyVN = MonthlyTT.Properties.VariableNames;

```

```

if ~strcmp(MinutelyVN{:}, MonthlyVN{:})

```

```

    ExactTime = timetable();

```

```

    fprintf('Variable names %s and %s in the argument timetables are not the

```

```

same.',MonthlyVN{:},MinutelyVN{:})

```

```

    return

```

```

end

```

```

[yhn,mhn,dhn] = ymd(MinutelyTT.date_time);

```

```

% Year, Month, Day Of Daily Maxima

```

```

for k = 1:numel(MonthlyTT)

```

```

    [ym,mm,dm] = ymd(MonthlyTT.date_time(k));

```

```

% Year, Month, Day Of Monthly

```

```

Maxima

```

```

    Lv = ismember([yhn,mhn], [ym,mm], 'rows');

```

```

    idx = MinutelyTT{:,1}(Lv) == MonthlyTT{:,1}(k);

```

```

    Mv = find(Lv);

```

```

    Dv = find(idx);

```

```

    HM = MinutelyTT(Mv(Dv(1)),:);

```

```

    MonthDayTT(k,:) = timetable2table(HM);

```

```

end

```

```

ExactTime = MonthDayTT;

```

```

end

```

```

function [ExactTime,AllMatches] = DailytoMinutely(MonthlyTT,MinutelyTT)
MinutelyVN = MinutelyTT.Properties.VariableNames;
MonthlyVN = MonthlyTT.Properties.VariableNames;
if ~strcmp(MinutelyVN{:}, MonthlyVN{:})
    ExactTime = timetable();
    fprintf('Variable names %s and %s in the argument timetables are not the
same.',MonthlyVN{:},MinutelyVN{:})
    return
end
[yhn,mhn,dhn] = ymd(MinutelyTT.date_time);           % Year, Month, Day Of Daily Maxima
for k = 1:numel(MonthlyTT)
    [ym,mm,dm] = ymd(MonthlyTT.date_time(k));        % Year, Month, Day Of Monthly Maxima
    %   YrMoDa = [ym mm dm]
    Lv = ismember([yhn,mhn,dhn], [ym,mm,dm], 'rows');
    idx = MinutelyTT{:,1}(Lv) == MonthlyTT{:,1}(k);
    %   LvSize = nnz(Lv)
    Mv = find(Lv);
    Dv = find(idx);
    HM = MinutelyTT(Mv(Dv(1)),:);
    AllHM{k,:} = MinutelyTT(Mv(Dv),:);
    MonthDayTT(k,:) = timetable2table(HM);
end
ExactTime = MonthDayTT;
AllMatches = AllHM;
end

```