

اسکرپت m-file تمرین کاربردی ۷-۱ به نام Shiraz_Annual_Rainy_Days_from_daily.m در پوشه Organizing Data و در مسیر ZIMA_MATLAB_book\Second Volume\Organizing Data قرار داده شده است.

۱-۲-۱- استخراج پارامترهای آماری ماهانه و سالانه از سری های زمانی روزانه با رویکرد جستجو در داده

در بخش قبل، استخراج برخی پارامترهای اقلیمی در مقیاس سالانه و از داده های روزانه توضیح داده شد. در این بخش شیوه استخراج مقادیر ماهانه و سالانه از داده های روزانه آموزش داده خواهند شد.

۱-۲-۱-۴- استخراج بارش ماهانه و سالانه از سری زمانی روزانه

✓تمرین کاربردی

تمرین کاربردی ۸-۱- فایل Shiraz Synoptic Station.xlsx حاوی اطلاعات روزانه برخی پارامترهای هواشناسی در ایستگاه سینوپتیک شیراز می باشد. این فایل در پوشه Organizing Data و در مسیر ZIMA_MATLAB_book\Second Volume\Organizing Data قرار دارد. مقادیر بارش ماهانه و سالانه ایستگاه هواشناسی شیراز را در هر سال محاسبه کنید.

این تمرین از برخی جهات شبیه به تمرین های قبلی است اما تفاوت بارز آن ارائه خروجی ها در دو مقیاس زمانی ماهانه و سالانه است. در ادامه و با تشریح کدهای برنامه نویسی، بیشتر با این تفاوت ها و شباهت ها آشنا خواهید شد. ابتدا اطلاعات فایل Shiraz Synoptic Station.xlsx را با استفاده از فرمان xlsread به محیط MATLAB فراخوانی نمایید. دقت نمایید که فایل Shiraz Synoptic Station.xlsx حتماً در پوشه جاری قرار داشته باشد.

```
clc,clear all
[num,txt] = xlsread('Shiraz Synoptic Station');%It Reads data and strings from the excel file
```

هر چند که با متغیر num می توان فرآیند پیش رو را ادامه داد، اما برای ساده تر شدن و افزایش درک فرآیند برنامه نویسی در این مثال، داده های مربوط به پارامتر بارش را در یک متغیر جدید اضافه کنید. اطلاعات بارش در ستون چهاردهم از متغیر num قرار دارد.

```
Rain = num(:,14);% Locates the rainfall data to the Rain variable
```

در اینجا نیاز بدان است که داده های تاریخ نیز مورد استفاده قرار گیرند. می توانیم آن ها را نیز در یک متغیر جدید قرار دهیم. متغیر Date را ایجاد

می‌کنیم که ستون اول تا سوم آن به ترتیب حاوی اطلاعات سال، ماه و روز میلادی است. البته با توجه به شرایط تمرین حاضر فقط نیاز به داده‌های سال و ماه داریم. همچنین احتیاج به تعیین تعداد سال در فایل Excel ایستگاه سینوپتیک شیراز داریم که همانند تمرین‌های قبل عمل می‌نماییم.

```
Date = num(:,4:6); % Locates the Julian date data to the Date variable
Years = minmax(Date(:,1)'); % Detremine the first and final years
Years = Years(1):Years(2); % Creates the vectors which contains all years
```

حال که اطلاعات اولیه مورد نظر فراهم گردید بایستی حلقه‌های مورد نظر را تشکیل دهیم. قبل از آن بهتر است یک ماتریس NaN به تعداد سطر معادل تعداد سال و تعداد ستون معادل 12 ماه ایجاد نماییم.

```
Monthly_rain = nan(length(Years),12);
```

بهتر است یک بار دیگر نیز بدین موضوع اشاره شود که این ماتریس به دو منظور ایجاد می‌شود، یکی افزایش سرعت محاسبات و دیگری در نظر گرفتن نقص‌های آماری موجود. البته سرعت محاسبات برای چنین ماتریس کوچکی تفاوت معناداری نخواهد داشت و این موضوع در رابطه با ماتریس‌های با ابعاد بزرگ‌تر بیشتر قابل توجه است. با این حال در رابطه با مشخص کردن نواقص آماری این موضوع بسیار حائز اهمیت است. در واقع ما می‌توانستیم از یک ماتریس با اعداد صفر و یا یک نیز به هدف افزایش سرعت محاسبات خود دست یابیم، اما آنچه که نمی‌توانستیم مشخص کنیم مقادیر نواقص اطلاعاتی بود. برای توضیح بیشتر این موضوع را در نظر بگیرید که ممکن است بنا به هر دلیلی، اطلاعات در یک یا چند تاریخ خاص اصلاً وجود نداشته باشد و حتی اطلاعات یک یا چند برداشت نیز در یک ماه خاص موجود نباشد. همان طور که قبلاً نیز بدان اشاره شده است، اگر به فایل Excel داده‌های هواشناسی شیراز مراجعه نمایید مشاهده خواهید کرد که ماه‌های سال میلادی 1985 از ماه 6 به بعد شروع شده است. در واقع در ماه ژانویه تا ژوئن این سال هیچ داده‌ای وجود ندارد. تابع sum یا nansum از MATLAB جمع مقادیر تهی یا خالی را برابر با صفر در نظر می‌گیرد، این در حالی است که تابع mean یا nanmean مقدار NaN را برای یک ورودی تهی باز می‌گرداند. مثال‌های زیر این موضوع را به خوبی نشان می‌دهد.

```
mean([])
ans =
    NaN

sum([])
ans =
    0
```

از آنجایی که جمع بارش‌های روزانه یک ماه مقادیر بارش ماهانه آن ماه را تشکیل می‌دهد، لذا در صورتی که این عمل انجام نشود، برای ماه‌هایی که اصلاً اطلاعات آن موجود نیست، مقدار بارش صفر لحاظ می‌شود که در عمل با مقدار واقعی بارش صفر در تضاد است. البته برای تکمیل این موضوع نیازمند قرار دادن یک شرط نیز هستیم که در گام بعدی بدان اشاره خواهد شد. در این مرحله با نوشتن دو حلقه تو در تو به راحتی داده‌های ماهانه بارش

استخراج می شوند. در این مثال از اندیس گذاری منطقی یا logical استفاده شده است. دقت کنید که مقادیر NAN نیز حذف شده اند.

```
u=1;
for i = Years
    for j = 1:12
        a = Rain(Date(:,1) == i & Date(:,2) == j );
        a(isnan(a)) = []; % It removes the NANs from the a
        if length(a)>25
            Monthly_rain(u,j)= sum(a);
        end
    end
    u=u+1;
end
display(Monthly_rain)
```

در حلقه فوق i نشان دهنده و شمارنده سال و j شماره ماه است. از آنجایی که اندیس ردیف داده های بارش و ردیف داده های مربوط با تاریخ مانند یکدیگر هستند (اندازه ردیف های آنان با هم برابر است) می توان به صورت اندیس دهی منطقی از اندیس های مشترک آنان استفاده نمود. به عبارتی بهتر، هر جا که اعداد منطقی (Logical) برای داده های تاریخ برابر با یک بود همان اندیس هایی از داده های بارش هستند که مد نظر ما می باشند. لذا درون حلقه دوم ابتدا اندیس روزهایی که سال آنان برابر سال معادل شمارنده i و ماه آنان برابر با j است به صورت منطقی (logical) تعیین می شوند. در ابتدا یک متغیر جدید به نام a تولید می شود که شامل داده هایی از بارش هستند که تاریخ سال آن ها برابر با i و ماه آن ها برابر با j است. پس از آن با استفاده از فرمان $a(isnan(a)) = []$ آرایه های NAN از وکتور a برداشته می شوند. در گام بعدی، یک شرط تعریف شده است که اعلام می نماید اگر طول بردار a از 25 (25 روز داده) بیشتر باشد جمع آن با استفاده از تابع sum محاسبه شود در غیر این صورت اندیس مربوط به ماه و سال مورد نظر به صورت NAN باقی می ماند. در اینجا توجه کنید که مقدار 25 بر حسب شرایط و نظر کارشناسی اعلام شده است. اگر هدف تعیین و مشارکت دقیق تر همه روزها در محاسبات باشد می توان مقدار 25 را افزایش داد و حداقل برابر با 28 که برابر با تعداد روزهای ماه ژانویه است قرار داد و یا قبل از آن اقدام به رفع نواقص آماری و همچنین بازسازی اطلاعات نمود. از آنجایی که بارش ماهانه برابر است با مجموع بارش های روزانه و همچنین آرایه های NAN نیز از وکتور a حذف شده اند، لذا از تابع sum برای این منظور استفاده می شود. سپس این مقدار بارش ماهانه به متغیر $Monthly_rain$ تحویل داده می شود. متغیر $Monthly_rain$ دارای دو اندیس سطر و ستون است. اندیس سطر آن برابر u در نظر گرفته شده است که نشان دهنده ردیف سال است. در ابتدای حلقه و بیرون از تمام حلقه ها متغیر u برابر با یک در نظر گرفته شده است.

بعد از اتمام حلقه دوم، دوباره به مقدار u عدد یک اضافه و تبدیل به اندیس سال بعدی می شود. شمارنده j نیز برای تعیین مکان و اندیس ماه ها

در نظر گرفته شده و بدین ترتیب `Monthly_rain` در حین اجرای برنامه تکمیل می‌گردد. البته همان طور که قبلاً بدان اشاره شد، از آنجایی که تابع `sum` و یا `nansum` جمع یک ماتریس تهی را صفر بر می‌گرداند، این شرط لحاظ شده است. به طوری که اگر ماتریس `a` تهی نباشد و طول آن از 25 بیشتر باشد، عملیات جمع بر روی آن انجام می‌شود، در غیر این صورت بدون اجرای هیچ تابعی، حلقه ادامه پیدا می‌کند و `Monthly_rain(u,j)` به صورت `NAN` باقی می‌ماند. خروجی این برنامه به شکل ذیل است.

`Monthly_rain =`

	NaN	NaN	NaN	NaN	NaN	0	0	0	0	0	9.6000	105.7000
4.5000	33.3000	41.8000	33.8000	5.9000	0	0	0	2.0000	0	0	82.2000	243.8000
9.1000	19.2000	99.5000	11.6000	0.6000	0	0.2000	0	0	19.5000	0	33.4000	0
135.7000	112.6000	49.5000	44.5000	0	0	0	0	0	0	0	89.7000	0
15.8000	24.6000	39.6000	24.2000	0	0	0	0	0	0	37.3000	89.0000	0
104.6000	114.9000	10.3000	20.0000	0	0	0	0	0	0	12.7000	17.8000	0
219.9000	45.0000	66.3000	7.1000	0	0	0	0	0	7.9000	0	75.0000	0
109.3000	31.0000	47.5000	13.3000	17.4000	0	0	4.0000	0.1000	0	2.3000	243.5000	0
125.9000	98.5000	88.7000	17.7000	0.5000	0	0	0.2000	0	1.7000	8.0000	0.1000	0
20.7000	7.6000	110.0000	11.7000	29.0000	0	0	0	0.1000	2.2000	108.8000	80.7000	0
15.1000	161.5000	42.4000	60.8000	3.3000	3.0000	0.1000	0	0	28.0000	4.0000	174.8000	0
145.3000	82.0000	123.0000	28.8000	1.2000	2.2000	0.6000	22.5000	0	2.7000	2.2000	12.4000	0
46.5000	5.7000	97.2000	38.8000	3.4000	0	0	0	0	4.9000	30.4000	97.9000	0
100.3000	97.5000	95.7000	23.2000	0.7000	0.4000	0	0	0	0.1000	0	0	0
109.1000	144.2000	54.3000	4.7000	0	0	0	0	0	0	0.4000	12.4000	0
167.0000	10.9000	2.3000	0.3000	0	0	0	0	0	11.1000	72.2000	94.0000	0
20.9000	29.8000	9.0000	0	2.0000	0.2000	0	0	0	0	6.4000	141.5000	0
95.6000	60.2000	31.8000	52.8000	0.1000	0	0	0	0	0	1.9000	30.5000	0
95.6000	79.3000	101.9000	19.0000	0	0	0	0	0	0	0	60.7000	0
234.8000	15.9000	5.5000	29.9000	0.7000	0	1.1000	0	0	0	28.6000	305.2000	0
108.1000	36.5000	31.5000	2.3000	0.1000	0	4.0000	0	0	0	111.5000	32.5000	0

توجه نمایید که در اینجا حد روز بارانی در نظر گرفته نشده است. معمولاً روز بارانی به روزی گفته می‌شود که مقدار بارش در آن بزرگ‌تر مساوی 1 میلیمتر باشد. چنانچه هدف در نظر گرفتن تعداد روز بارانی نیز باشد و بخواهیم این حد روزهای بارانی را در برنامه لحاظ نماییم می‌توانیم به راحتی با استفاده از عملگرهای رابطه‌ای مقدار بارش را به صورت `Monthly_rain(u,j) = sum(a(a>=1))`؛ همچنین دقت کنید که 5 ماه اول سال اول، یعنی سال 1985، داده‌ها به صورت `NAN` باقی مانده‌اند و نشان دهنده آن است که داده‌ها در این ماه‌ها شرط لازم را نداشته‌اند. اولین ستون از اطلاعات مربوط به بارش در ماه ژانویه است و آخرین ستون مربوط به ماه دسامبر می‌شود. حال بایستی داده‌های سالانه استخراج گردند. از آنجایی که مجموع بارش در 12 ماه هر سال به عنوان بارش سالانه در نظر گرفته می‌شود، با استفاده از تابع `sum` و با در نظر گرفتن بعد دوم (عدد 2 به عنوان عدد بعد سطرها)، به راحتی اطلاعات مقادیر بارش سالانه محاسبه می‌شوند. البته از آنجایی که 5 ماه اول سال اول `NAN` هستند، بدیهی است مقدار بارش سالانه سال اول نیز برابر با `NAN` باشد. دقت نمایید که در اینجا نبایستی از فرمان `nansum` استفاده نماییم. چرا که نبود داده را در 5 ماه اول سال 1985 را نادیده می‌گیرد و لذا خروجی که ارائه می‌نماید صحیح نمی‌باشد.