

31119

Monday, September 16, 2019

Course #6

Notes:

1. Review of datestr and datenum

In Matlab, dates can be converted into numbers using the datenum command, i.e.,

```
>> date_as_a_number=datenum('Jan_2_2010');
```

This will create a new variable, date_as_a_number, and it will have a numeric value. What the value actually is is not important, what is important is that you can view the number as a date, using the command datestr, i.e.,

```
>> datestr(date_as_a_number)
```

Will display: "Jan_2_2010"

The numerical values of dates in the datenum format are sorted in chronologic order, so a recent date will have a larger value than an older date.

2. Review of plotting

See the additional handout called: 31119_notes_plotting

3. Averaging using the nanmean.m command on any dimension

nanmean

The nanmean command calculates an average while ignoring any nans.

nan is short for "not a number". It is Matlab's way of keeping track of missing data.

nanmean() can be applied to any dimension of a matrix, by entering the dimension in the 2nd column. If no dimension is listed, the default is the 1st.

squeeze

>>squeeze(matrix) becomes useful when you are doing mathematical analysis that reduces the dimensionality of a matrix

For instance, assume that I have daily temperature for all latitudes and longitudes on a 2.5 degree grid::

T2001 : a matrix with dimensions (days, lat,lon): (365,90,144)

Then I take an average over time:

```
Tmean=nanmean(T2001);
```

Tmean has dimensions (1,90,144);

If I want to access data in Tmean, have to keep track of the first column, even though it has dimension 1, this is annoying.

For instance, if I want the 25th lat entry at the the 90th lon entry, then I would need to type:

```
tnow=Tmean(1,25,90);
```

Whereas, if I use squeeze() after I used nanmean, I would then type:

```
tnow=Tmean(25,90);
```

This might not seem like much, but it saves hassle in the long run.