2017

Case Study for Guided Project in Stochastic Hydrology

Meghna Babbar-Sebens
Oregon State University, meghna@oregonstate.edu

CE 540/BE 525 STOCHASTIC HYDROLOGY
GUIDED PROJECT
SPRING 2017

ACTIVITY 1: Probabilities and Distributions of Random Variables and Functions

Please perform the various tasks below and then include the results, discussion of the results, and conclusions sections into your project report. Any code you generate should be included in the Appendix of your project report. The final combined report that include material from all activities will be due at the end of the term.

Goal: In this project activity, we will use daily, monthly, and annual data on rainfall and streamflows in a Midwestern U.S. watershed (Eagle Creek Watershed, Indiana) to identify properties of the various probability distributions of these random variables.

Data: The data can be downloaded as zip files (daily.zip, monthly.zip, and annual.zip) from Canvas course webpage. For any one of the time scales (i.e., daily, monthly, and annual), please unzip the file to find a folder with the following files in it:

- Master_Daily.xslx: This file tabulates all the data along with description of the datasets.
- Precipitation data: Based on which GCM-RCM (****_****) combination was chosen for your group, download these corresponding precipitation Matlab data files:
  - Pr_****_****.mat
  - Pr_****_****_header.mat
- Flow data:
  - Q_Observed_ *Daily/Monthly/Annual*_cms.mat
  - Q_Observed_ *Daily/Monthly/Annual*_cms.mat
- Date_Daily.mat: This shows the year, month, and date for the data.

Methodology: For each of the time scales (daily, monthly, and annual), please repeat the following tasks.

1. **Create Histograms**. We will first treat our continuous variables as discrete variables, by dividing the observed values into bins. Load all the data .mat files and then read the various precipitation and flow data for the period 1968-1998 into Matlab arrays. **For example**,

```
Pr_data = Pr_CRCM_CCSM(:,4)
Fl_data = Q_Observed_cms(:,3)
```

   Then identify how many bins you want to divide the data into, e.g. N = 100. Find the minimum and maximum values of your data array, and then uniformly divide the data into bins and create a bins array that contains the bins. **For example**:

```
N = 100
minP = min(Pr_data)
```

```
maxP = max(Pr_data)
bins = [minP : (maxP-minP)/N : maxP]
```

Now, create a histogram of your data array using the `hist` command in Matlab and then plot the bar graph. **For example**,

```
[y_val, x_val] = hist(Pr_data, bins)
y_val = y_val/length(Pr_data) % to get probabilities
bar(x_val, y_val)
```

Now create a cumulative distribution function of all probability mass functions, by adding the probability mass of all values of `x_val` below any value of `x_val`. You can use the `for` loop to calculate cumulative values. **For example**,

```
CDF_y_val = zeros(1,length(y_val))
for i = 1: length(y_val)
 for j = 1:i
      CDF_y_val(1,i) = y_val(1,j)+ CDF_y_val(1,i);
 end
end
figure();
bar(x_val, CDF_y_val)
```

Another way to create the cumulative distribution function from your dataset is via the function `ecdf` in Matlab. Use this function to obtain CDF of all your datasets, and then to compare with the CDF you obtained via the `for` loop approach above by plotting it on the same bar graph. **For example**,

```
hold on      % This retains bar graph from above
[f,x] = ecdf(Pr_data);
plot(x,f, 'g')
```

2. **Calculate Moments.**
   a. Use the formulae provided in the lecture to calculate the sample mean, sample median, sample mode, sample variance, sample skewness, and sample kurtosis for all datasets. Compare them with the values you get from inbuilt functions in Matlab. **For example**,

```
m = mean(Pr_data)
mo = mode(Pr_data)
va = var(Pr_data)
sk = skewness(Pr_data)
k = kurtosis(Pr_data)
```

   b. Calculate covariance and correlation between the flow dataset (random variable X) and the various multiple precipitation datasets (each dataset can be assumed to be random variable Y) to identify any existence of linear relationship between flow and precipitation datasets. Use the formulae provided in the lecture and

then compare the results with inbuilt functions `covar` and `correl`. **For example**,

```
covar = cov(Pr_data, Fl_data)
correl=corrcoef(Pr_data,Fl_data,'rows','complete')
```

Note that the diagonals of the matrix produced by the inbuilt function `covar` has values of variance.

3. **Identify continuous pdfs.** Now that we have created probability distributions of datasets by discretizing them and have calculated the sample moments, we will now try to fit appropriate continuous probability density functions to the data histograms. For this task, we will use a very useful Matlab tool called `dfittool`. **For example**,

```
dfittool(Pr_data)
```

Matlab will open a new window with a histogram of your dataset (you can change the histogram to have more or fewer bins than the number chosen by Matlab). Try out multiple distribution types and fit them to the data. More information on how to use this tool can be found here: http://www.mathworks.com/help/stats/model-data-using-the-distribution-fitting-tool.html. Once you have fitted the distributions, select the best possible distribution via following tests:

- **Probability plots:** This is a plot of empirical cdf from the sample data versus cdf of a particular fitted distribution. This is available as one of the plots in `dfittool` GUI window. The fitted distribution that best matches the empirical cdf indicates the most suitable fitted distribution for the data.
- **Goodness of fits tests:** In order to use any of the standard statistical tests listed below, first use the **Generate Code** option in the **File** menu of the GUI to create a file that saves the code for all the fitted distributions. Save that file as a createFit.m (*or some other name*) file on your computer in order to obtain the fitted probability distributions in the original Matlab workspace outside the `dfittool` GUI. Also, note that while theory on Goodness of fits tests below are easily available in any standard statistics textbook, here is a website for an easy explanation: http://www.mathwave.com/articles/goodness_of_fit.html. Use the tests below to estimate the best fitted distribution for your data.
    1. Chi-square statistic: `chi2gof` function can be used in Matlab to test if the data comes from a normal distribution.
    2. Kolmogorov-Smirnoff (K-S) statistic: `kstest` function in Matlab can be used for this test.
    3. Anderson-Darling (A-D) statistic: `adtest` function in Matlab can be used for this test.

You can include commands for these inbuilt test functions in the code you obtained from createFit.m file. Test the various fits (pd*) created by the code for the assessing which fit is best suited for your dataset.

4. **Calculate joint probabilities mass functions.** Now, we will work with flow and precipitation data as discrete bivariate random variables. We will combine the flow data with the various precipitation datasets available to us, to create multiple realizations of (Flow, Precip). Use `hist3` function in Matlab to create a bivariate histogram between flow dataset and precipitation dataset for each realization. **For example**,

```
biv_data = zeros(length(Pr_data),2);
biv_data(:,1)=Pr_data(:);
biv_data(:,2)= Fl_data(:);
N = 100
Pr_data_x = min(Pr_data): (max(Pr_data)- min(Pr_data))/N
: max(Pr_data);
Fl_data_y = min(Fl_data): (max(Fl_data)- min(Fl_data))/N
: max(Fl_data);

pmf = hist3(biv_data, {Pr_data_x Fl_data_y})
pmf_normalize = pmf./length(Pr_data);
figure()
surf(Pr_data_x,Fl_data_y,pmf_normalize)

% This should give the CDF which should tend to 1 for
smaller bins
integralOverHistogramPlot = sum(trapz(pmf_normalize))
```

5. **Calculate marginal probabilities.** Once you have obtained the joint pmf in Task 4, use the `For` loop in Matlab to write a code that lets you obtain marginal pmf of flow and rainfall datasets. How do these marginal pmf compare to the single random variable pmf you obtained in Task 1?

6. **Calculate conditional probabilities.** Write a code to use the Bayes rule formula we discussed in the class to calculate the **P[0 cms< flow < 50 cms | precip < 6 mm]**, for all the combinations of bivariate random variables of flow dataset and precipitation datasets.

Results and Discussion: Summarize and discuss the findings from each of the tasks, for all the datasets at all time scales.

Conclusion: Discuss the overall conclusions in your own words. For example, you can include topics such as,

- What were your overall findings from the various tasks?
- Were there any similarities or dissimilarities between different precipitation datasets (including observed precipitation pdf), and across the different time scales?
- Were there any similarities between flow pdfs across different time scales?
- What can you say about independence between the observed flow and precipitation random variables?
- Which GCM_RCM precipitation model data performs the best?

CE 540/BE 525 STOCHASTIC HYDROLOGY
GUIDED PROJECT
SPRING 2017

ACTIVITY 2: Time Series Analysis

Please perform the various tasks below and then include the results, discussion of the results, and conclusions sections into your project report. Any code you generate should be included in the Appendix of your project report. The final combined report that include material from all activities will be due at the end of the term.

Goal: In this project activity, we will use either daily, monthly, or annual data on streamflows in the Midwestern U.S. watershed (Eagle Creek Watershed, Indiana) to develop various time series models of the flow random variable.

Data: We will continue to use flow data from Activity 1 for this activity. Please choose one of the 3 time scales to work with.

Methodology: For one of the time scales (daily, monthly, and annual), please repeat the following tasks.

1. Plot the time series.
2. **Correlogram:** Calculate auto-correlation function for various lags in your stream flow data, and plot a correlogram. What does it tell you about the lags? Plot the Correlogram.
3. **Periodogram:** Create a periodogram using the Matlab function `periodogram` in the signal processing toolbox. While you should first set the WINDOW option to a Tukey window option (type `help tukeywin` in Matlab), feel free to explore other window options and see what kind of results you get. Plot the periodogram. You can also plot a 2-D map that shows the changes of the power spectrum with time by using the `spectrogram` function in Matlab.
4. **Partial Autocorrelations:** Now calculate the partial auto-correlations in Matlab. Plot the partial autocorrelations. What does it tell you about the dependencies and possible order of the auto-regressive terms?
5. **Stationarity and Nonstationarity:** Based on the figures above, what can you say about the stationarity or nonstationarity in your data? If nonstationarity exists, then create secondary time series data by either differencing or standardizing or both if the first one does not work. Do either of the differencing or standardizing methods remove nonstationarity (check by repeating steps 1, 2, 3, and 4)? If yes, which method worked for your data?
6. **Markov Model:** Based on the assessment in steps above, will either first order stationary Markov model or first order nonstationary Markov model work for your original time series data (i.e., data before you removed nonstationarity by differencing or standardizing)? If you think it is possible to use one of the Markov models, then

divide the time series into two parts. The first series (e.g., 80% of the data) can be used for model development, and the remainder of the time series (e.g., 20% of the data) can be used for model validation. Before you write a Matlab code for the Markov model, make sure you test for the assumptions of normality or lognormality (which is an important assumption). You can use `qqplot` to test for normality.

7. **ARIMA models:** Finally, take the stationary data obtained in Step 5 and create various ARIMA models using the 80% of the initial data (i.e., training data). Once developed, for each ARIMA model, predict time series values for the remainder of the dates (dates of the remaining 20% of the data – i.e., testing data) and identify which model best fits the training and testing data. Please explore using the System Identification Toolbox: http://www.mathworks.com/help/ident/index.html. You can launch this toolbox by typing **`ident`** in the Matlab Command Window. You can then use the **`armax`** model in the Time Series Model Identification tool of the System identification Toolbox: http://www.mathworks.com/help/ident/time-series-model-identification.html. There are videos also available for you to learn from for the project: http://www.mathworks.com/videos/estimating-state-space-and-polynomial-models-68898.html.