# MET3220C & MET6480
# Computational Statistics

## Programming – week #11
## Dr. Mark Bourassa

TA: David Rock, LOV305A, 644-8577
Office Hours: Tue. 1-2PM & Wed. 2:30-3:30
drock@met.fsu.edu
/usr/lib/sendmail drock@met.fsu.edu < AS9_your_last_name.f90

# Assignment #9
# Goodness of Fit

- EITHER
  - 1) Copy your code AS5.f90 or AS6b.f90 to AS9.f90
    - cp AS6b_your_last_name.f90 AS9_your_last_name.f90

  OR
  - 2) Copy a working version that I have provided.
    - cp /u/a/met3220-02/AS9_pseudonym.f90 AS9_your_last_name.f90

# Normalization of the PDF

- Is If using option (1). Check the part of the code that should be used to normalize the histogram.
- The previous (incorrect) wording was ‘Use a DO loop to divide all values in the histogram array by the total number of used observations (i_good).’
- The goal is to make the area under the curve equal to one. This is not the same as making the sum of the values in the bin add to one. That would only work if the bin widths were all one.
- To make the area under the curve equal to unity (a fancy way of saying one), then we must add up the area of the bins, and divide by that total.
- The areas is equal to the sum of ‘the product of the bin height and the bin width.’ Since our bin width is constant, this can be simplified.
- Use a DO loop to divide all values in the histogram array by ‘the total number of used observations times the bin width’ (i.e., / ( REAL(i_good) * bin_width) ).

# The Goodness of Fit Test

- Use the $\chi^2$ test evaluate the goodness of fit, for $\alpha = 1\%$.
  - Test two idealized distributions:
    - Gaussian, and
    - log-normal distribution.
- Use your 120 bins in this comparison. You may compare to the values in Wilks' Table B.3, for $\nu = 100$.
- Right your conclusions as a comment at the bottom of the program. Your results depend on whether or not you used the rain flag to quality control the data. State whether or not you did this in your conclusions.

# The Goodness of Fit Test
# The Calculation

- The formula for the $\chi^2$ test is

$$\chi^2 = n \sum_{bins} \frac{\left( PDF \text{ observed} - \Pr\{\text{data in bin}\} \right)^2}{\Pr\{\text{data in bin}\}}$$

  - In this case, $n$ is equal to the number of data points (i_good or ikeep), and Pr for each bin come from your theoretical pdf.
- You could also write this equation as

$$\chi^2 = n \sum_{bins} \frac{\left( PDF \text{ observed} - \Pr\{\text{data in bin}\} \right)^2}{\Pr\{\text{data in bin}\}}$$

- A note of caution!
  - What do you do if the number of expected observations is zero?

- Due date: **April 4$^{th}$**, before 5:00PM (local time).

# Comments on Numerical Error in Summation

- Computers store data as binary numbers (ones and zeros) in a set amount of computer memory. The amount of space depends on the computer system and settings in the program
  - Example: an integer might have the space for 16 ones and zeros (bits).
- This situation means there is a limit on how big or small a number can be, and be stored in memory. Why?
- For integers, the first binary digit is used as a sign (+ or –).
  - If there are n binary digits, that leaves n – 1 digits for magnitude.
  - Zero is one of allowable numbers, therefore the largest magnitude is equal to one less than the number of possible numbers.
- The formula for the largest integer magnitude is $2^{n-1} - 1$.
- If you are calculating a sum of integer values, this is a key limiting factor. Example: sum of all ECMWF surface pressures, in units of Pa, for each six hour period over 40 years.
  - How can you deal with this problem?

# Comments on Numerical Error in Summation

- One (good) approach is to work with REAL variables, rather than integers.
  - Real numbers are much more complicated, so we will go over the concept, but not the gory details.
  - Real numbers have the same number of ones and zeros (bits) as integers, but they are arranged as a sign, and exponent, and a mantissa.
- REAL numbers have a MUCH wider range of values; however, in many cases they can only approximate base 10 numbers (whole numbers and fractions).
  - Example, this approximation is why we don’t test if a data value is EQUAL to a REAL missing value.
  - Assume that the rounding error for any one addition (in a sum) will be of similar scale to all other rounding errors. Then apply your error propagation formula.
  - The rounding error for $x^2$ is greater than the error for $(x-E[x])^2$.