Assignment 3
Introduction to Probabilistic Machine Learning

Part 1: TrueSkill Implementation (4 Points)

- Relevant source files: `factors.jl, trueskill.jl`
- Supporting library files from previous assignments: `gaussian.jl, distributionbag.jl`
- Test files: `test*_factors.jl, test*_trueskill.jl`

This exercise consists of implementing some of the TrueSkill models presented in Unit 5.

To do so, please implement the following 4 factors:

- GaussianFactor
- GaussianMeanFactor
- WeightedSumFactor
- GreaterThanFactor

All four of those, follow the general idea of the factors-setup used in the last assignment. That means, that each of those factors has one update_msg_to_ method for every outgoing connection in the factor graph. As before, it's also only supposed to store the outgoing messages, and again use the division of variable and outgoing messages to compute incoming messages wherever needed. The most notable difference here is that instead of using Discrete distributions, we used different continuous distributions. The underlying functions used in the model are all specified along the slides of Unit 5, on the slides mentioned above. The general behavior of the update functions is the same as before. Determine possible required input signals, compute the updated outbound message and the resulting marginal, and update both in the bag.

As we need to possibly normalize resulting distributions, the additionally introduces normalize_log_factor!, normalize_log_var!, and normalize!. The first two functions should be added for every single distribution, to compute the respective normalization factors using the library provided (and the logic discussed earlier). This normalization has to take place after computation in the actual true skill logic. The last factor, due to its very non-gaussian nature ;), needs more complex logic to approximate it and turn it into a valid Gaussian distribution. A detailed explanation of how the approximation is done will follow in the tutorials.

Once the factors are designed and the tests run successfully, you can point your attention to the actual implementation of the true-skill graphs. Please use the graphs from the lecture as a reference, with three overall implementations needed:

- To start it off: Implement the true-skill model to derive the marginal distribution of the skill of two players given a match outcome. Please note that the match outcome is incorporated in the model implicitly (the "left" player wins). See Slide 7 for detailed reference.
- Following that, we can use the factors provided earlier, to implement the two-team model as well. See Slide 8 for reference.
- Last but not least, we use the multi-team match model from Slide 13 as a more complex example.

As previously done, the respective functions should implement the logic required to initialize the graph, run the computation pass through them in the correct order, and return the respective marginal distributions. While the values might be continuous now, the general layer is very similar to the seir example.

Part 2: Derivation of Gaussian-Mean Vector Update (1 Point)

Consider the following two densities of Normal Distributions, i.e., $N(y; x, \beta^2)$ and $N(x; \mu, \sigma^2)$ with $\beta^2, \sigma^2 > 0$, see Unit 5 (slide 15, upper right corner).

Show the following identity using the Gaussian Multiplication Theorem (Unit 2):

$$\int N(y; x, \beta^2) \cdot N(x; \mu, \sigma^2) dx = N(y; \mu, \beta^2 + \sigma^2).$$

Hint 1: Use that $N(y; x, \beta^2) \cdot N(x; \mu, \sigma^2) = N(x; y, \beta^2) \cdot N(x; \mu, \sigma^2)$.

Hint 2: The Multiplication Theorem says that

$$G\left(x; \underbrace{\tau_1}_{\frac{y}{\beta^2}}, \underbrace{\rho_1}_{\frac{1}{\beta^2}}\right) \cdot G\left(x; \underbrace{\tau_2}_{\frac{\mu}{\sigma^2}}, \underbrace{\rho_2}_{\frac{1}{\sigma^2}}\right) = G\left(x; \tau_1 + \tau_2, \rho_1 + \rho_2\right) \cdot N\left(\underbrace{\frac{\tau_1}{\rho_1}}_{\mu_1 = y}; \underbrace{\frac{\tau_2}{\rho_2}}_{\mu_2 = \mu}, \underbrace{\frac{1}{\rho_1}}_{\sigma_1^2 = \beta^2} + \underbrace{\frac{1}{\rho_2}}_{\sigma_2^2 = \sigma^2}\right).$$

Hint 3: Use that any Gaussian density integrates to 1.