Belief Revision in Picnic Planning

Elijah Sumner, Kaitlyn Jesmonth, Franklin Camacho

Introduction

The weather can easily change in an instant, and interpreting forecasts and current observations is often a crucial part in making weather-related decisions. However, this can often pose a difficult task for the general public. In order to model a situation that incorporates belief revision, we decided to use these ideas and simulate a user's decision on whether it was a good or bad day to go on a picnic. We used weather observations, two randomly generated forecasts, and a survey taken from the class to show how people weigh each part of the forecast in terms of importance to display the belief revision process. This project strives to make predictions/decisions similar to a human, and convinces the user of the best possible option, whether to go on a picnic or not.

Background

Weather forecasting as practiced by humans is an example of having to make judgments in the presence of uncertainty. Many human forecasters use approaches based on the science of meteorology to deal with the many challenges they face when forecasting the weather (Doswell, 2004). Probability based forecasting is provided to users as a way of making a decision according to their needs. Many studies have indicated that there was a lack of agreement among the general public and even the weather forecasters themselves on what the probability of a weather event is, showing a variety of interpretations of a statement like this, "70% chance of showers tomorrow" (Elía & Laprise, 2005). Since many people have varying opinions on what the probability of a chance of rain is, they may make different choices regarding their day, like whether to bring out an umbrella, or to go out and have fun for the day.

Most people base their beliefs on what a "chance" of rain actually means on a few different things, that being how long it rains for, how much of an area it will cover, or how many days it will cover. If there was a forecast for a 30% chance of rain, some people might think "It will rain for 30% of the day tomorrow", others might think the rain will cover 30% if the region that they live in, and lastly others might think "It will rain on 30% of days like tomorrow" (Gigerenzer et al., 2005). So it is difficult to pinpoint one general way that humans reason about weather forecasting, as many of us have different opinions and beliefs about what those probabilities really mean. If we were to generalize it, it would use the National Weather Service's definition of what the PoP (Probability of Precipitation) is: the likelihood of occurrence (expressed as a percentage) of a measurable amount of liquid precipitation during a specified period of time at any given point in the forecast area (National Weather Service, n.d.). Basically, a 30% chance of rain tomorrow really means at least some amount of rain will fall the next day in 3 out of 10 cases at some point in the forecast area. With just saying 30% chance of rain, it implies to the general public that there is only a possibility that it will rain tomorrow, where with the time and region definitions, which is the true definition, forecasters mean that it will rain tomorrow for certain in a given place (Gigerenzer et al., 2005).

When looking more specifically at how reasoning and understanding probabilities in humans and as it relates to our project, one study found that humans often analyze things in terms of one or a few samples of different outcomes (Vul et al., 2014). This very idea counteracts some standard assumptions when thinking about optimality. Rather than considering the full distribution, people appear often to make decisions based on a posterior probability distribution, which is the revised probability of an event occurring after taking new information into account (Vul et al., 2014; Hayes, 2020). This means that for example when trying to decide if it might be a good day to go on a picnic like in our project, a person might think of several scenarios based on the weather forecast.

However an important question to ask is, how many samples should one consider in order to optimize their worst or best case reward over several different decisions? The study found that making many quick, but possibly suboptimal decisions based on a few samples over a long period of time is the most optimal strategy (Vul et al., 2014). In particular, the research found when looking at 2, 4, 8, 16, and 32 different discrete alternatives for unidimensional continuous choices, that a decision based on a very small set of samples is about the same as a decision on a full probability distribution (Vul et al., 2014). They also discovered that when the stakes are higher for a given problem, more sample alternatives should be considered (Vul et al., 2014). For our project, coming up with some weighting scheme of the most important weather conditions to a person on a given day should be implemented in the model. Asking the user what they are most comfortable with could also play a role in weighting various different decisions within our model world.

But exactly how much is a person coming up with several sample scenarios going to cost? That is something that needs to be defined for the task at hand. Usually time and effort is an important factor. Another key idea here to keep in mind is that oftentimes human decisions are continuous. These sample scenarios that are brought up in a person's mind might not always have one exact clear answer. Yet, the decisions we make based on the weather can often be very binary. This is why it is imperative that a strategy with the general idea of assessing, "is this going to put me closer to the optimal answer?" is used (Vul et al., 2014). Again, when looking at what our group is trying to model, this could be a very beneficial approach to try to determine the ultimate goal, deciding whether or not it is a good day to have a picnic.

Methods

According to the paper written by Franz Huber, he talks about how an individual should revise their beliefs when they receive new information on that belief. The example they give in the paper is: "Sophia believes many things, among others that it will rain on Tuesday, that it will be sunny on Wednesday, and that weather forecasts are always reliable. Belief revision theory tells Sophia how to revise her beliefs when she learns that the weather forecast for Tuesday and Wednesday predicts rain." (Huber, 2013, p. 3). This information tells us that with her original belief that it will be rainy on Tuesday, and sunny on Wednesday isn't correct anymore, so with her belief that the forecast is always reliable, she keeps her belief that Tuesday is rainy, and gives up her belief that it is sunny on Wednesday. As we can see, even though she had her original belief, the fact that she had the belief that the forecast was 100% reliable made it so she wouldn't doubt her decision to give up her belief that it was sunny on Wednesday.

According to another paper written by Fabio Paglieri, he writes that individuals also store information that they do not currently believe, since in the future that information could then become believed if new evidence is gathered (Paglieri, 2004). Along with this, beliefs that are accepted currently may be refused later on, either because they were shown to be incorrect, or because more plausible information was provided from a different source (Paglieri, 2004). This ties into our forecast problem, since someone could observe sunny skies and a warm day, but after an hour they watch the news, and the forecast states that there is a storm about to roll into the area. Now this person uses this new evidence to refuse their original belief that it would be sunny, and then believe that it will rain in the future instead. An individual also stores information from two contradictory sources (say 2 different news sources like BBC and CNN), one claiming that there will be a perfectly sunny day tomorrow, and the other stating that there will be showers throughout the day tomorrow. The individual is more inclined to believe the one that they have more trust in (or whichever has the better reputation) (Paglieri, 2004). If a third independent source came in and also stated that tomorrow will be a perfectly sunny day, the individual will use this information to update their original beliefs, choosing to believe that it will be sunny since two sources (assuming the third independent source is reliable), said that it would be sunny in their forecast.

For our problem that we have proposed, which is "is today a good day to go have a picnic?", we have developed a process of belief revision that we will be modeling in this project represented in Figure 1. The way that we will represent the original beliefs in our problem will mainly be based on present observations, which was set to ideal conditions in this project. A forecast is then randomly generated, and it is further determined, using a weighted point system, how ideal the forecast is for a picnic. If the forecast is expected to be ideal, the modeled person will continue with their plans for a picnic, while they will cancel their plans if the forecast is not ideal. Drawing off some points presented earlier, they will not expect this decision to change or go back and check the forecast again before they leave for their picnic. While the current scope of the forecast in the project does not have much constraint in realism, adding this in later would even better help the belief revision model real-world scenarios. This could even possibly be accomplished by allowing the program to ingest real current weather observations and forecasts to more so eliminate randomness and inconsistencies in the forecast outputs. For example, it would not be expected that the temperature would drop

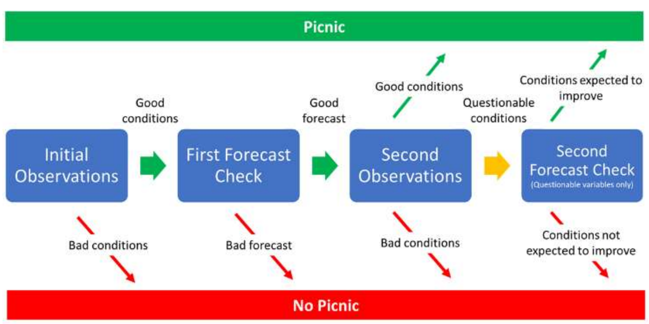

***Figure 1**: Graphical representation of the belief revision process of the model.*

from 70 °F to 40 °F, but the scope of the program does currently allow this scenario without much restriction.

It is also important to note how each forecast variable is weighted in the model's eyes. This was done using a form sent out to the class, which asked them to rank in terms of importance temperature, dewpoint, wind, and the current weather conditions, and these results were reported in Figure 2 and translated into the weight values in Figure 3. Based on the forecast, the weights determined by the average importance of the class, and what's displayed in the forecast, the model will make their original prediction as to whether or not it is a good day to have a picnic.

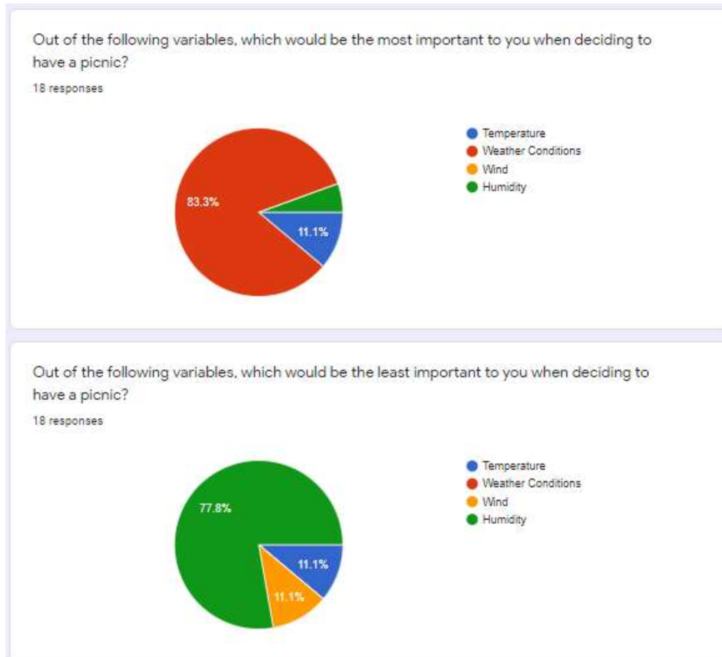

***Figure 2**: Condition importance survey results.*

Once it is time for the picnic, the model will once again check the current observational conditions. This is not intended as an intention check of the conditions, since their belief has already been set, but designed as a response to a person walking outside to leave and subconsciously re-observing the weather. If the model determines that the conditions are now questionable for a picnic, they will check the forecast only for the variables in question, since their belief has only changed for those. If the model determines that one of the variables is too bad for a picnic, they will cancel it without

Variable Weights	
Variable	Weight
Temp	4.4
Condition	8.3
Humidity	0.1
Wind	3.7

***Figure 3**: Individual variable weights.*

checking the forecast. If the model determines that the new conditions are still within a set comfort level, then it will proceed with the picnic as planned. However, if the new conditions do not fall within this comfort level, the model will call off the picnic. In the scope of the model, there is a 1 in 8 chance that any individual variable will change from the forecast, leading to about a 1 in 2 chance of any part of the conditions deviating from the forecast. The conditions for what is considered good, questionable, or bad were set by Elijah and Kaitlyn using their meteorology background and their knowledge on what is reasonable in the meteorology realm, and reported in Figure 4.

Questionable Observations	
Observation	Forecast to Check
Overcast w/ light clouds	Condition
Partly cloudy w/ dark clouds	Condition
Precipitation w/ visible clearing	Condition
Warm (80s °F)	Temperature
Cold (50s °F)	Temperature
Humid (60s °F)	Dewpoint
Breezy (5-14 MPH)	Wind

Bad Conditions	
Observed	Forecasted
Overcast	Overcast w/ dark clouds
Lightning	Precipitation w/ no visible clearing
Tornado	Very warm (temp >= 90 °F)
Snow	Very cold (temp < 50 °F)
Graupel	Very humid (dewpoint >= 70 °F)
Sleet	Windy (wind >= 15 MPH)

Figure 4: *Bad and questionable condition criteria.*

Results:

We were able to create several scenarios with our model successfully within the scope of our picnic problem. All of the scenarios started out with good conditions in order to keep things simple. However, the differences came in once the randomly generated forecast was created and when the second observations were checked right before actually going on the picnic. For example, in one of the most basic scenarios the conditions started out ideal, the initial forecast had good conditions, and the second observations right before going on the picnic was also ideal.

This is where the model was able to show that the weather turned out great for a picnic.

```
?- picnic.
Walking outside, the current weather looks good for a picnic later...
It is currently sunny with mild temperatures, no humidity, and calm winds

The current forecast looks good for a picnic later...
The forecast has a temperature of 73° with sunny conditions, 48° dewpoints, and 4 MPH winds with a 22% chance of precipitation.

It is now time for the picnic...

Walking outside, the weather still looks good for a picnic...
It is currently sunny with mild temperatures, no humidity, and calm winds

The weather turned out to be great for the picnic, so you decide to go and have one!
true
```

One the other end of the spectrum, another scenario that was established was when the initial forecast looked so bad that the user just decided right away not to go on a picnic.

```
?- picnic.
Walking outside, the current weather looks good for a picnic later...
It is currently sunny with mild temperatures, no humidity, and calm winds

The current forecast does not look good for a picnic later...
The forecast has a temperature of 95° with overcast conditions, 64° dewpoints, and 4 MPH winds with a 10% chance of precipitation.

The weather did not turn out so great for the picnic, so you decided to just stay home
true
```

Finally, two other situations arose. One of which was when there was a good forecast to start, but then the conditions became questionable and ended up staying good in the end. This was replicating a situation where the weather might have changed over the course of a couple of hours and a person might have considered not going on the picnic for just a quick second, before still resorting to his or her initial belief. Each of the questionable conditions within that second forecast were also displayed.

```
?- picnic.
Walking outside, the current weather looks good for a picnic later...
It is currently sunny with mild temperatures, no humidity, and calm winds

The current forecast looks good for a picnic later...
The forecast has a temperature of 67° with sunny conditions, 41° dewpoints, and 12 MPH winds with a 16% chance of precipitation.

It is now time for the picnic...

Walking outside, the weather looks worse than forecasted...
It is currently sunny with mild temperatures, moderate humidity, and breezy winds

The current humidity feels questionable...
The dewpoint is now forecasted to be 64°, which is good for a picnic

The current wind is questionable...
The wind is now forecasted to be 4MPH, which is good for a picnic

The weather turned out to be great for the picnic, so you decide to go and have one!
true
```

This final example really displayed our belief revision capabilities within the model. In this scenario, an originally favorable forecast for a picnic ends up not panning out and the second observations don't look as ideal anymore because of quickly changing weather conditions. The model ends up re-checking the forecast a second time, and finds that conditions are not expected to improve and displays that it was no longer a good day to go on a picnic.

```
?- picnic.
Walking outside, the current weather looks good for a picnic later...
It is currently sunny with mild temperatures, no humidity, and calm winds

The current forecast looks good for a picnic later...
The forecast has a temperature of 66° with partly cloudy conditions, 44° dewpoints, and 11 MPH winds with a 9% chance of precipitation.

It is now time for the picnic...

Walking outside, the weather looks worse than forecasted...
It is currently precipitating with distant clearing with mild temperatures, no humidity, and breezy winds

The current weather conditions are questionable...
The weather conditions are now forecasted to be rain, which is not good for a picnic

The current wind is questionable...
The wind is now forecasted to be 14MPH, which is good for a picnic

The weather did not turn out so great for the picnic, so you decided to just stay home
true
```

Frequently this can happen in real life situations when it is supposed to be hot out in the summer and some afternoon thunderstorms pop up which could impact a person's outdoor plans and their decisions regarding those plans.

From creating this model, we were easily able to output several different types of scenarios that can in fact occur in the real world, and observe how the model responded to those different scenarios. We were even able to have some fun with it and create a 1 in 1000 chance that if it was decided that the picnic would happen, several orangutans would escape from the zoo and ruin the picnic.

```
?- picnic.
Walking outside, the current weather looks good for a picnic later...
It is currently sunny with mild temperatures, no humidity, and calm winds

The current forecast looks good for a picnic later...
The forecast has a temperature of 73° with sunny conditions, 48° dewpoints, and 4 MPH winds with a 22% chance of precipitation.

It is now time for the picnic...

Walking outside, the weather still looks good for a picnic...
It is currently sunny with mild temperatures, no humidity, and calm winds

The weather turned out to be great for the picnic, so you decide to go and have one!

However...
You arrive to find the park swarming with several
orangutans that escaped from the zoo, so you can't have your picnic :(
true
```

All of these examples really display the versatility of our system while still modeling belief revision and the human thought process.

Discussion

Overall, the model performed well and did what we set out to do. However, because of the limited time constraint, our version could still be improved in several manners. To make things simpler for the time given for this project, we were only able to have set weights to determine the most important and least important weather information for determining whether or not someone should go and have a picnic. This was accomplished by sending out a Google Form survey to the entire COG/CSC 366 class and asking our fellow classmates what their opinions were for this given situation. Based on their responses, that is how we determined the weights. So in the future, if there was something that we would do differently, it would be to probably try to either sample a bigger population to determine the weights, or try to make the weights dynamic in the sense that they could change depending on a given situation. For our purposes just setting the weights worked just fine, but it would be a nice added bonus to be able to weigh certain weather phenomena more heavily depending on a given forecast for example.

Another success of this model was that we were able to create a random forecast that was different from the current observations. Additionally, this forecast is more sophisticated than the observations. It gives a little bit more information that you can expect than just reciting back the current observations which is a success of our model. Right now, the random forecast generated by the model doesn't have any constraints on it. So meteorologically speaking, the outputs for the forecast don't always make complete sense. For example, if a temperature of 99 degrees is forecasted with a dewpoint of 48 degrees and there is 90% chance of rain in the forecast this doesn't make true sense. In order to have precipitation from a meteorological standpoint, the dewpoint and the temperature need to be close together because when this is the case, it means the air is saturated and can produce precipitation. Improving on this forecast and making it a little bit more realistic is something that should be worked on moving forward as that will make our model even more practical when comparing it to the real world.

However, even though we were not able to set all of the constraints for our model, we were able to set some minimal constraints on what the model can produce. For example, the model output dewpoint must be lower than the temperature. This makes sense because the meteorological definition for the dewpoint is the temperature which the air needs to be cooled to reach saturation. Dewpoint has to be lower than the temperature. Therefore, we made sure to reflect this accurately in our model. Additionally, another added success in our model is the fact that we were able to get the probability of precipitation or PoP to follow the weather condition. For example, if the conditions were cloudier then your PoP value might be higher than on a day when it is perfectly clear and sunny. These basic constraints that we set were just some of the first things that needed to make sense and be considered for our model, so we felt it was best to focus on these elements first.

Setting our weights based on the survey and using our meteorological knowledge to set some constraints for our model to make sense was generally a good approach for this project. While it would be nice to expand on some functionalities of this project, this model is still a useful cognitive tool. Given that meteorological information can be very hard to digest as an end user, this model helps give a scenario of what the weather could look like on a given day. It displays some of the thought processes behind making a decision to go on a picnic or not, and possibly revising that belief, something many of us do daily when it comes to making weather-related decisions.

Conclusion

In conclusion, our project turned out to be a success, and our Prolog code worked as intended. While there are some limitations, our model was able to take in the randomly generated forecasts, and then verify based on observations if the conditions were favorable for a picnic. The belief revision came in when the forecast changed leading up to the picnic and a new decision needed to be made using the weights determined by the class survey. Overall, knowledge from this course about Prolog coding as well as belief revision was very beneficial in this project and accomplishing our goal of modeling decision making based on the weather.

References

Doswell, C. A. (2004). Weather Forecasting by Humans—Heuristics and Decision Making. *Weather and Forecasting*, *19*(6), 1115–1126. https://doi.org/10.1175/waf-821.1

Elía, R. de, & Laprise, R. (2005). Diversity in Interpretations of Probability: Implications for Weather Forecasting. *Monthly Weather Review*, *133*(5), 1129–1143. https://doi.org/10.1175/MWR2913.1

Gigerenzer, G., Hertwig, R., van den Broek, E., Fasolo, B., & Katsikopoulos, K. V. (2005). "A 30% Chance of Rain Tomorrow": How Does the Public Understand Probabilistic Weather Forecasts?. *Risk Analysis*, *25*(3), 623–629. https://doi.org/10.1111/j.1539-6924.2005.00608.x

Hayes, A. (2020). *Understanding Posterior Probability*. Investopedia. https://www.investopedia.com/terms/p/posterior-probability.asp

Huber, F. (2013). Belief Revision I: The AGM Theory. *Philosophy Compass*, *8*(7), 604–612. https://doi.org/10.1111/phc3.12048.

National Weather Service. (n.d.). *FAQ - What is the Meaning of PoP*. Www.weather.gov. Retrieved November 2, 2021, from https://www.weather.gov/ffc/pop#:~:text=Forecasts%20issued%20by%20the%20National%20Weather%20Service%20routinely

Paglieri, F. (2004). Data-oriented belief revision: Towards a unified theory of epistemic processing. In Proceedings of STAIRS (pp. 179-190).

Vul, E., Goodman, N., Griffiths, T. L., & Tenenbaum, J. B. (2014). One and Done? Optimal Decisions From Very Few Samples. *Cognitive Science*, *38*(4), 599–637. https://doi.org/10.1111/cogs.12101

Appendix A: Prolog Code

```
%%%%%%%%%%%
%Knowledge Base%
%%%%%%%%%%%

%Weighting Points
temp_pts(4.4).
cond_pts(8.3).
humid_pts(0.1).
wind_pts(3.7).
max_pts(50).

%Ideal Conditions
temp_ideal(70).
dewpoint_ideal(60).
wind_ideal(0).

%Forecast condition points
weather_pts(condition(sunny), 0).
weather_pts(condition(partly_cloudy), 0).
weather_pts(condition(overcast), 1).
weather_pts(condition(weather(rain)), 5).
weather_pts(condition(weather(snow)), 5).
weather_pts(condition(weather(sleet)), 5).
weather_pts(condition(weather(graupel)), 5).
weather_pts(condition(weather(lightning)), 7).
weather_pts(condition(weather(tornado)), 5).
weather_pts(_, 0).


%%%%%%%%%%%%%%%%%%%%
%Forecast & Observation Variables%
%%%%%%%%%%%%%%%%%%%%

%Initial Observations
obs_cond(obs(sunny)).
obs_temp(obs(mild)).
obs_humid(obs(nhumid)).
obs_wind(obs(calm)).

%Second Observations
obs2_cond(_, NCond):-
   random(1, 8, 1),
   get_cond(GCond),
   cond_to_obs(GCond, NCond).
```

```
obs2_cond(Cond, NCond):-
   cond_to_obs(Cond, NCond).

obs2_temp(_, NTemp, GTemp):-
         random(1, 8, 1),
   temp(GTemp),
   temp_to_obs(GTemp, NTemp).
obs2_temp(Temp, NTemp, Temp):-
   temp_to_obs(Temp, NTemp).

obs2_humid(_, Temp, NDew):-
   random(1, 8, 1),
   dewpoint(GDew, Temp),
   dew_to_obs(GDew, NDew).
obs2_humid(Dew, _, NDew):-
   dew_to_obs(Dew, NDew).

obs2_wind(_, NWind):-
   random(1, 8, 1),
   wind(GWind),
   wind_to_obs(GWind, NWind).
obs2_wind(Wind, NWind):-
   wind_to_obs(Wind, NWind).

%Variables to observations
cond_to_obs(condition(sunny), obs(sunny)).
cond_to_obs(condition(partly_cloudy), obs(cloudy(partly(dark)))):-
   random(1, 4, 1).
cond_to_obs(condition(partly_cloudy), obs(cloudy(partly(light)))).
cond_to_obs(condition(overcast), obs(cloudy(overcast(dark)))):-
   random(1, 4, 1).
cond_to_obs(condition(overcast), obs(cloudy(overcast(light)))).
cond_to_obs(condition(weather(_)), obs(precip(nclearing))):-
   random(1, 4, 1).
cond_to_obs(condition(weather(_)), obs(precip(clearing))).

temp_to_obs(Temp, obs(vcold)):-
   Temp < 50.
temp_to_obs(Temp, obs(cold)):-
   Temp < 60,
   Temp >= 50.
temp_to_obs(Temp, obs(mild)):-
   Temp < 80,
   Temp >= 60.
temp_to_obs(Temp, obs(warm)):-
   Temp < 90,
```

```
    Temp >= 80.
temp_to_obs(Temp, obs(vwarm)):-
    Temp >= 90.

dew_to_obs(Dew, obs(nhumid)):-
    Dew < 60.
dew_to_obs(Dew, obs(humid)):-
    Dew >= 60,
    Dew < 70.
dew_to_obs(Dew, obs(vhumid)):-
    Dew >= 70.

wind_to_obs(Wind, obs(calm)):-
    Wind < 5.
wind_to_obs(Wind, obs(breezy)):-
    Wind >= 5,
    Wind < 15.
wind_to_obs(Wind, obs(windy)):-
    Wind >= 15.

%Values to conditions
get_cond(Output):-
    random(0, 100, Random),
    get_cond(Random, Output).
get_cond(Random, Output):-
    Random < 30,
    Output = condition(sunny).
get_cond(Random, Output):-
    Random < 75,
    Output = condition(partly_cloudy).
get_cond(Random, Output):-
    Random < 82,
    Output = condition(overcast).
get_cond(Random, Output):-
    Random < 85,
    Output = condition(weather(rain)).
get_cond(Random, Output):-
    Random < 90,
    Output = condition(weather(lightning)).
get_cond(Random, Output):-
    Random < 95,
    Output = condition(weather(snow)).
get_cond(Random, Output):-
    Random < 96,
    Output = condition(weather(graupel)).
get_cond(Random, Output):-
```

```
    Random < 100,
    Output = condition(weather(sleet)).
get_cond(Random, Output):-
    Random < 101,
    Output = condition(weather(tornado)).


%Forecast
cond(Output) :-
    get_cond(Output).
temp(Temp):-
    random(50, 100, Temp).
dewpoint(Dew, Temp):-
    random(35, Temp, Dew).
wind(Wind):-
    random(0, 20, Wind).
pop(Pop, condition(weather(_))):-
    random(30, 100, Pop).
pop(Pop, _):-
    random(0, 29, Pop).

%%%%%%%%%%
%Variable Checks%
%%%%%%%%%%

%Does a forecast needs to be checked?
forecast(cond, [obs(cloudy(overcast(light))), obs((cloudy(partly(dark)))), obs(precip(clearing))]).
forecast(temp, [obs(warm), obs(cold)]).
forecast(dewpoint, [obs(humid)]).
forecast(wind, [obs(breezy)]).
forecast(pop, [cond(weather(_))]).

%Is a certain variable good or bad for a picnic?
bad([condition(overcast), condition(weather(lightning)), condition(weather(tornado)),
        condition(weather(snow)), condition(weather(graupel)),
        condition(weather(rain)), condition(weather(sleet)), obs(cloudy(overcast(dark))),
        obs(precip(nclearing)), obs(vwarm), obs(vcold), obs(vhumid), obs(windy)]).

is_good(Variable) :-
    bad(List),
        \+ member(Variable, List).

%Check temperature forecast
check(temp, Obs, _, good):-
    forecast(temp, TempList),
    \+ member(Obs, TempList).
```

```
check(temp, Obs, Forecast, good) :-
        forecast(temp, TempList),
    member(Obs, TempList),
    Forecast >= 50,
    Forecast < 90,
    write("The current temperature is questionable..."), nl,
    write("The temperature is now forecasted to be "), write(Forecast),
        write("°, which is good for a picnic"), nl, nl.
check(temp, _, Forecast, bad):-
     write("The current temperature is questionable..."), nl,
    write("The temperature is now forecasted to be "), write(Forecast),
        write("°, which is not good for a picnic"), nl, nl.

%Check dewpoint forecast
check(dew, Obs, _, good):-
        forecast(dewpoint, DewList),
    \+ member(Obs, DewList).
check(dew, _, Forecast, good):-
        Forecast < 70,
    write("The current humidity feels questionable..."), nl,
    write("The dewpoint is now forecasted to be "), write(Forecast),
        write("°, which is good for a picnic"), nl, nl.
check(dew, _, Forecast, bad):-
     write("The current humidity feels questionable..."), nl,
    write("The dewpoint is now forecasted to be "), write(Forecast),
        write("°, which is notgood for a picnic"), nl, nl.

%Check wind forecast
check(wind, Obs, _, good):-
        forecast(wind, WindList),
    \+ member(Obs, WindList).
check(wind, _, Forecast, good):-
        Forecast < 15,
        write("The current wind is questionable..."), nl,
    write("The wind is now forecasted to be "), write(Forecast),
        write("MPH, which is good for a picnic"), nl, nl.
check(wind, _, Forecast, bad):-
        write("The current wind is questionable..."), nl,
    write("The wind is now forecasted to be "), write(Forecast),
        write("MPH, which is not good for a picnic"), nl, nl.
%Check conditions forecast
check(cond, Obs, _, _, good):-
    is_good(Obs),
        forecast(cond, CondList),
    \+ member(Obs, CondList).
check(cond, Obs, Forecast, Pop, good):-
```

```
    is_good(Obs),
    check(pop, Pop, good),
    parse_obs(cond, Forecast, NForecast),
    write("The current weather conditions are questionable..."), nl,
    write("The weather conditions are now forecasted to be "), write(NForecast),
            write(", which is good for a picnic"), nl, nl.
check(cond, _, Forecast, _, bad):-
    parse_obs(cond, Forecast, NForecast),
    write("The current weather conditions are questionable..."), nl,
    write("The weather conditions are now forecasted to be "), write(NForecast),
            write(", which is not good for a picnic"), nl, nl.

%Check dewpoint forecast
check(pop, Pop, good):-
            Pop < 50.
check(pop, Pop, bad):-
            Pop >= 50.



%%%%%%%%%%%
%Parsing variables%
%%%%%%%%%%%

%Clouds
parse_clouds(Input, Output):-
            Input = overcast(dark),
            Output = "overcast with dark clouds".
parse_clouds(Input, Output):-
            Input = overcast(light),
            Output = "overcast with light clouds".
parse_clouds(Input, Output):-
            Input = partly(dark),
            Output = "partly cloudy with dark clouds".
parse_clouds(Input, Output):-
            Input = partly(light),
            Output = "partly cloudy with light clouds".

%Precip
parse_precip(Input, Output):-
            Input = clearing,
            Output = "precipitating with distant clearing".
parse_precip(Input, Output):-
            Input = nclearing,
            Output = "precipitating with no distant clearing".
```

```
%Temperatures
parse_obs(temp, Input, Output):-
   Input = obs(Temp),
         parse_obs(temp, Temp, Output).
parse_obs(temp, Input, Output):-
         Input = vwarm,
         Output = "very warm".
parse_obs(temp, Input, Output):-
         Input = warm,
         Output = "warm".
parse_obs(temp, Input, Output):-
         Input = mild,
         Output = "mild".
parse_obs(temp, Input, Output):-
         Input = cold,
         Output = "cold".
parse_obs(temp, Input, Output):-
         Input = vcold,
         Output = "very cold".

%Humidity
parse_obs(humid, Input, Output):-
         Input = obs(Dew),
   parse_obs(humid, Dew, Output).
parse_obs(humid, Input, Output):-
         Input = vhumid,
         Output = "high".
parse_obs(humid, Input, Output):-
         Input = humid,
         Output = "moderate".
parse_obs(humid, Input, Output):-
         Input = nhumid,
         Output = "low".

%Wind
parse_obs(wind, Input, Output):-
         Input = obs(Wind),
   parse_obs(wind, Wind, Output).
parse_obs(wind, Input, Output):-
         Input = windy,
         Output = "windy".
parse_obs(wind, Input, Output):-
         Input = breezy,
         Output = "breezy".
parse_obs(wind, Input, Output):-
         Input = calm,
```

```
        Output = "calm".

%Conditions
parse_obs(cond, Input, Output):-
   Input = obs(Cond),
   parse_obs(cond, Cond, Output).
parse_obs(cond, Input, Output):-
        Input = condition(precip(Precip)),
        parse_precip(Precip, Output).
parse_obs(cond, Input, Output):-
        Input = precip(Precip),
        parse_precip(Precip, Output).
parse_obs(cond, Input, Output):-
        Input = cloudy(Cloud),
        parse_clouds(Cloud, Output).
parse_obs(cond, Input, Output):-
   Input = condition(weather(Output)).
parse_obs(cond, Input, Output):-
        Input = condition(partly_cloudy),
   Output = "partly cloudy".
parse_obs(cond, Input, Output):-
   Input = condition(Output).
parse_obs(cond, Input, Output):-
   Input = Output.

evaluate_score(Score, good):-
   max_pts(Max),
        Score =< Max.
evaluate_score(Score, bad):-
   max_pts(Max),
   Score > Max.

dew_pts(Forecast, Ideal, 0):-
   Forecast =< Ideal.
dew_pts(Forecast, Ideal, Output):-
   Output is (abs(Ideal - Forecast)/2).

evaluate_pts(Temp, Dew, Cond, Pop, Wind, Result):-
        %Get ideal variables
   temp_ideal(Itemp),
   dewpoint_ideal(Idew),
   wind_ideal(Iwind),

   %Get variable weights
   temp_pts(Ptemp),
   cond_pts(Pcond),
```

```
    humid_pts(Pdew),
    wind_pts(Pwind),

    %Get variable differences
    Dtemp is (abs(Itemp - Temp)/2),
    Dwind is (abs(Iwind - Wind)/2),
    dew_pts(Dew, Idew, Ddew),
    weather_pts(Cond, Dcond),

    %Determine variable points
    Stemp is (Dtemp * Ptemp),
    Sdew is (Ddew * Pdew),
    Swind is (Dwind * Pwind),
        Scond is ((Dcond * (Pop / 100)) * Pcond),

        Sum is (Stemp + Sdew + Swind + Scond),
    evaluate_score(Sum, Result).

picnic:-
        %Current observations from knowledge base
    obs_cond(obs(Cond)),
        obs_temp(obs(Temp)),
        obs_humid(obs(Dew)),
        obs_wind(obs(Wind)),

    %Current forecast from knowledge base
    temp(FTemp),
    cond(FCond),
    dewpoint(FDew, FTemp),
    wind(FWind),
    pop(FPop, FCond),

    %Second observations
    obs2_temp(FTemp, NTemp, DTemp),
    obs2_cond(FCond, NCond),
    obs2_humid(FDew, DTemp, NDew),
    obs2_wind(FWind, NWind),

    %Second forecast
    temp(NFTemp),
    cond(NFCond),
    dewpoint(NFDew, NFTemp),
    wind(NFWind),
    pop(NFPop, NFCond),

    %That edge case
```

```
    %Execute scenario
    initial_observations(Cond, Temp, Dew, Wind, Result),
    initial_forecast(FCond, FTemp, FDew, FWind, FPop, Result, Result2),
        second_observations(NCond, NTemp, NDew, NWind, Result2, Result3),
        second_forecast(NCond, NFCond, NTemp, NFTemp, NDew, NFDew, NWind, NFWind,
        NFPop, Result3, FinalResult),
    end(FinalResult).

initial_observations(Cond, Temp, Dew, Wind, good):-
    is_good(Cond),
        parse_obs(cond, Cond, CondOut),
        parse_obs(temp, Temp, TempOut),
        parse_obs(humid, Dew, DewOut),
        parse_obs(wind, Wind, WindOut),
    write("Walking outside, the current weather looks good for a picnic later..."), nl,
    write("It is currently "), write(CondOut), write(" with "), write(TempOut),
            write(" temperatures, "), write(DewOut), write(" humidity, and "),
        write(WindOut), write(" winds"), nl, nl.
initial_observations(Cond, _, _, _, bad):-
    write(Cond), nl,
    write("Walking outside, the current weather does not look good for a picnic later...").

initial_forecast(Cond, Temp, Dew, Wind, Pop, good, good):-
    evaluate_pts(Temp, Dew, Cond, Wind, Pop, good),
    parse_obs(cond, Cond, NewCond),
    write("The current forecast looks good for a picnic later..."), nl,
    write("The forecast has a temperature of "), write(Temp), write("° with "),
        write(NewCond), write(" conditions, "), write(Dew), write("° dewpoints, and "),
      write(Wind), write(" MPH winds with a "), write(Pop), write("% chance of
        precipitation."),nl, nl.
initial_forecast(Cond, Temp, Dew, Wind, Pop, good, bad):-
    parse_obs(cond, Cond, NewCond),
    write("The current forecast does not look good for a picnic later..."), nl,
    write("The forecast has a temperature of "), write(Temp), write("° with "),
        write(NewCond), write(" conditions, "), write(Dew), write("° dewpoints, and "),
      write(Wind), write(" MPH winds with a "), write(Pop), write("% chance of
        precipitation."),nl, nl.
initial_forecast(_, _, _, _, _, _, bad).

second_observations(Cond, Temp, Dew, Wind, good, none):-
    is_good(Cond),
    is_good(Temp),
    is_good(Dew),
    is_good(Wind),
    forecast(cond, CondList),
```

```
    forecast(temp, TempList),
    forecast(dewpoint, DewList),
    forecast(wind, WindList),
    (member(Cond, CondList); member(Temp, TempList); member(Dew, DewList);
        member(Wind, WindList)),
    parse_obs(cond, Cond, CondOut),
        parse_obs(temp, Temp, TempOut),
        parse_obs(humid, Dew, DewOut),
        parse_obs(wind, Wind, WindOut),
    write("It is now time for the picnic..."), nl, nl,
    write("Walking outside, the weather looks worse than forecasted..."), nl,
    write("It is currently "), write(CondOut), write(" with "), write(TempOut),
            write(" temperatures, "), write(DewOut), write(" humidity, and "),
        write(WindOut), write(" winds"), nl, nl.
second_observations(Cond, Temp, Dew, Wind, good, good):-
    is_good(Cond),
    is_good(Temp),
    is_good(Dew),
    is_good(Wind),
        parse_obs(cond, Cond, CondOut),
        parse_obs(temp, Temp, TempOut),
        parse_obs(humid, Dew, DewOut),
        parse_obs(wind, Wind, WindOut),
    write("It is now time for the picnic..."), nl, nl,
    write("Walking outside, the weather still looks good for a picnic..."), nl,
    write("It is currently "), write(CondOut), write(" with "), write(TempOut),
            write(" temperatures, "), write(DewOut), write(" humidity, and "),
        write(WindOut), write(" winds"), nl, nl.
second_observations(Cond, Temp, Dew, Wind, good, bad):-
        parse_obs(cond, Cond, CondOut),
        parse_obs(temp, Temp, TempOut),
        parse_obs(humid, Dew, DewOut),
        parse_obs(wind, Wind, WindOut),
    write("It is now time for the picnic..."), nl, nl,
    write("Walking outside, the weather does not look good for a picnic anymore..."), nl,
    write("It is currently "), write(CondOut), write(" with "), write(TempOut),
            write(" temperatures, "), write(DewOut), write(" humidity, and "),
        write(WindOut), write(" winds"), nl, nl.
second_observations(_, _, _, _, _, bad).

second_forecast(Cond, FCond, Temp, FTemp, Dew, FDew, Wind,
          FWind, FPop, none, good):-
        check(cond, Cond, FCond, FPop, good),
    check(temp, Temp, FTemp, good),
    check(dew, Dew, FDew, good),
    check(wind, Wind, FWind, good).
```

```
second_forecast(Cond, FCond, Temp, FTemp, Dew, FDew, Wind,
        FWind, FPop, none, bad):-
      check(cond, Cond, FCond, FPop, _),
  check(temp, Temp, FTemp, _),
  check(dew, Dew, FDew, _),
  check(wind, Wind, FWind, _).
second_forecast(_, _, _, _, _, _, _, _, _, good, good).
second_forecast(_, _, _, _, _, _, _, _, _, bad, bad).

end(good):-
  random(1, 1000, 1),
  write("The weather turned out to be great for the picnic, so you decide
     to go and have one!"), nl, nl,
  write("However..."), nl,
  write("You arrive to find the park destroyed by a tornado, and can no
     longer have your picnic :(").
end(good):-
  random(1, 1000, 2),
  write("The weather turned out to be great for the picnic, so you decide
     to go and have one!"), nl, nl,
  write("However..."), nl,
  write("You arrive to find the park swarming with several
     orangutans that escaped from the zoo, so you can't have your picnic :(").
end(good):-
  write("The weather turned out to be great for the picnic, so you decide
     to go and have one!").
end(bad):-
  write("The weather did not turn out so great for the picnic, so you decided
     to just stay home").
```