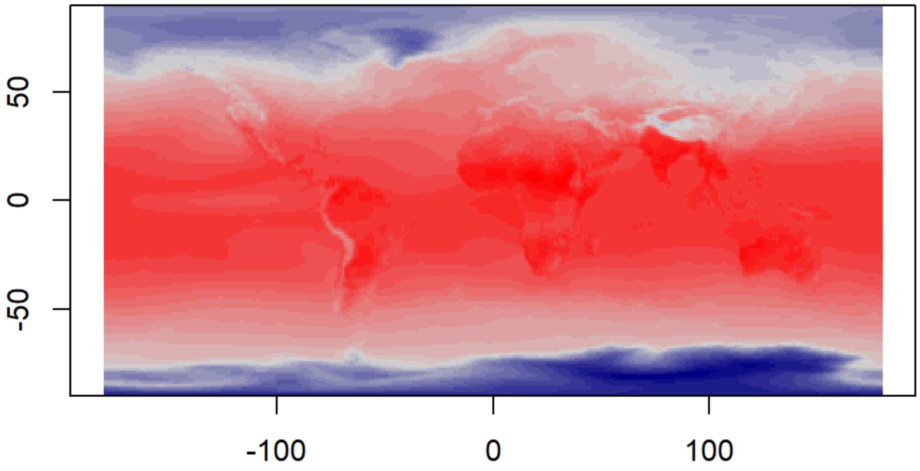
Forecasting Gambier

By: Grant Culbertson

2022



Introduction:

At Kenyon I'll often hear fellow students complaining about how the weather seems to be all over the place here in Ohio. Living in Ohio my whole life, I'm used to it, but I can see how going from 10 degrees one day to 40 the next might seem jarring for someone from California who's enjoyed a near constant 80 degrees every day before coming here for school. Another thing I hear fellow students complain about in regards to Ohio is that there isn't anything to do here, their idea of the whole state is made up by their perception of the immediate radius of the campus, a mall-less wasteland made up of corn fields and pastures for livestock. Considering that the average Kenyon student summarizes Ohio as corn fields and crazy weather, why not bring these two together for my stats project? Using the methods we've learned in this course I'll take it upon myself to try and locally forecast this crazy Ohio weather, and then apply that forecasted weather to the corn growing season.

Data:

For this project I first started by acquiring 5 datasets from the *Open-Meteo Historical API*, which is just an API that houses a historical database of weather. The 5 datasets I pulled from this API were all timeseries of weather here in Gambier, Ohio. The first three datasets were all daily measurements: the first ranged from 1970 to 1975, the second from 2016-2021, and the third from 1971-2021. The final two datasets contained hourly weather data from 1970 to 1975 and 2016 to 2021. Upon getting the datasets they were formatted a bit strangely and I couldn't load them into *R* before adjusting some columns in *wordpad* manually. After reworking the datasets and getting them into *R* I was able to see that every dataset contained 16 columns, but for the purposes of this project I only focused on these four: daily/hourly maximum temperature, daily/hourly minimum temperature, total rain sum in inches, precipitation hours for the day (how many hours it rained during the day), and just from the hourly datasets: hourly rainfall in inches. At this point I wanted to break down all of the datasets to only contain data during the corn growing season here in Ohio. After some research I found that: "the optimal planting period is between April 20th and May10th"[1] and that the harvest is generally between "July 5th and August 31st"[2]. Those are some fairly wide date ranges so I just tried to pick a start and end date that captured most of the season and was easy to filter for. My start/plant date ended up being April 20th and my chosen end/harvest date was August 31st. After filtering the data down to just the corn growing season I did some breakdowns of the data regarding means and it revealed to me that the 1970-1975 and 2016-2021 datasets were actually fairly different, most likely due to climate change.

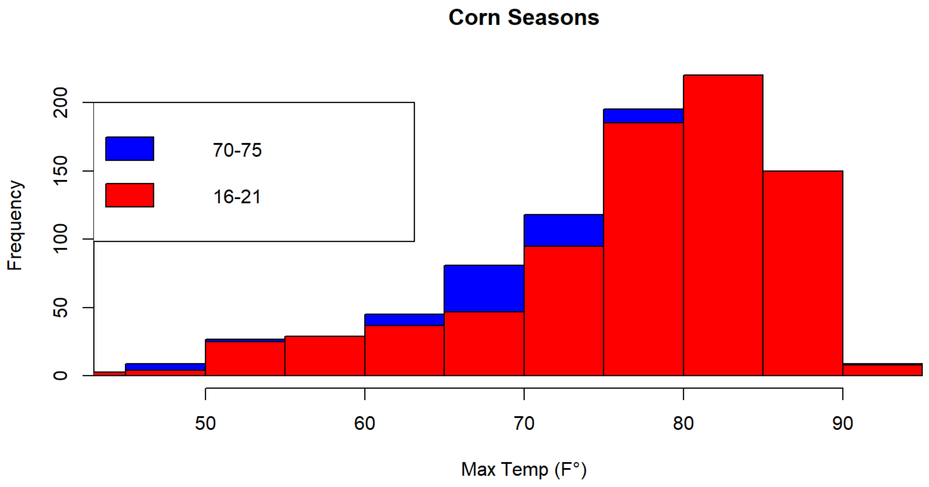


Fig. 1 - Max Temp Comparison

Considering the difference in the datasets it made more sense to move forward with my project focusing primarily on the 2016-2021 dataset because it should give the most realistic modeling

[1] Reese, Matt. "Corn Planting Date Considerations – Ohio Ag Net | Ohio's Country Journal." *Ohio Ag Net | Ohio's Country Journal*, Ohio Ag Net | Ohio&#039;s Country Journal, 10 Mar. 2020, https://ocj.com/2020/03/corn-planting-date-considerations-4/.

[2] https://www.pickyourown.org/OHharvestcalendar.htm

of what weather in the near future will look like in Gambier. Additionally, I want to note here that while the weather data is for Gambier it can really also be viewed as Mount Vernon weather data as well because Gambier is essentially inside of Mount Vernon.

Forecasting Method:

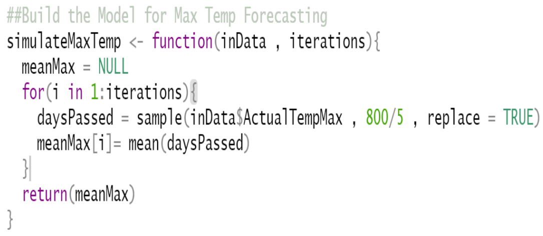
Following the formatting of my datasets the next step was to determine what kind of model or method I wanted to use to forecast future weather data. First off I made multiple types of forecasting models based in regression using *R* packages like *forecast*[3] and *h2o*[4]. While these models were very good at adjusting for changes in seasons and predicting the trends of temperature in future years, they were very bad at predicting extreme weather events, or in other words outliers. This inability to forecast future weather outliers made me reject these kinds of models, because my main interest, and the main interest of corn farmers are those outlier weather events that can affect corn growth such as exceptionally cold or rainy days during the growing season that can cause things like crazy top[5] or just plant death. This is when I settled on using bootstrapping to forecast the future weather. While bootstrapping might not be as accurate at forecasting overall weather trends over the year it is much more accurate than my regression attempts at predicting outlier events, the thing that I'm concerned with. Additionally, bootstrapping's inability to account for seasonal changes in weather isn't as important here because I'm only looking at one season during the year which should offset that shortcoming of this simulation method. Finally here I want to make note of the fact that most all weather modeling works off the assumption that future weather should be very similar to past weather within a similar timeframe, so my bootstrap forecasting works off the assumption that the next

[3] https://michaelminn.net/tutorials/r-weather/index.html
[4] https://www.michaelplazzer.com/weather-forecasting-with-machine-learning-in-r/
[5] A disease which can affect corn that is partially submerged underwater for extended periods following heavy rain

five years of growing seasons should exhibit similar weather patterns/distributions as the last five years of growing seasons. What this essentially means is that all estimations from my bootstrap forecasting can be viewed as a plausible forecast for any growing season within the next five years following 2021.

Creating the Simulation:

After selecting bootstrapping as being my method of simulating future weather for the growing season here, it was time to make the simulation functions in *R*. To begin, I first made a function to bootstrap the daily maximum temperatures from the 2016 to 2021 corn seasons data set, and then get the mean of those values to get an estimated mean max temperature for a future growing season[6].



```
##Build the Model for Max Temp Forecasting
simulateMaxTemp <- function(inData , iterations){
  meanMax = NULL
  for(i in 1:iterations){
    daysPassed = sample(inData$ActualTempMax , 800/5 , replace = TRUE)
    meanMax[i]= mean(daysPassed)
  }
  return(meanMax)
}
```

Fig. 2 - simulateMaxTemp()

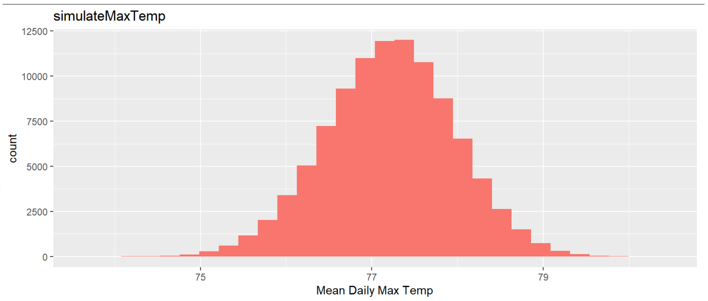


Fig. 3 - simulateMaxTemp() results, 100000 iterations

Within this *simulateMaxTemp()* function I'm just sampling one season's worth of daily maximum temperatures, so 160 days or 160 values. After that sample of daily max temperatures has been taken I get the mean of that data and store it in a vector, which the function will return. So overall the function will return a spread of potential mean daily max temperatures for a future growing season that is the size of *iterations* (see Fig 3.). Following the creation of *simulateMaxTemp()* I made two more essentially identical functions: one to forecast total rainfall for future seasons, and one to forecast mean daily minimum temperature for future seasons.

[6] When I say "future growing season" it is any season within the five years following 2021

Next, I created two functions to forecast for future outlier weather events, one function to forecast for days that are cold enough to cause frost[7] which can damage corn, and another function to forecast for days where there is heavy rainfall[8] that can result in crops being submerged in standing water. To forecast future heavy rainfall events I created a function called *simulateFlooding()*. Essentially this function just bootstraps for hourly rainfall from the hourly 2016 to 2021 corn season data set, then checks for values greater than .3 in the bootstrap sample and records the number of values above .3. Finally, the function returns a vector containing forecasted heavy rain days during a future season.

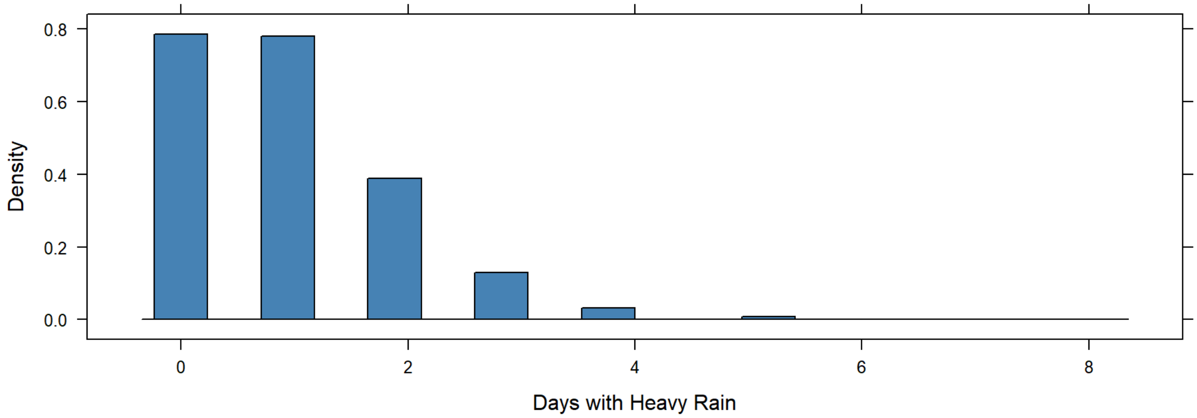


Fig. 4 - simulateFlooding() results, 100000 iterations

Comparing the results of this function against the actual data shows that my function is pretty spot on. The actual data has an average of 1 heavy rain event per season and my function returns an average of .997 heavy rain events per season[9]. Next I put together a function to forecast for frost. My function to forecast for days with frost works essentially the same way to *simulateFlooding()* except that I'm back to using the non-hourly dataset and bootstrapping daily

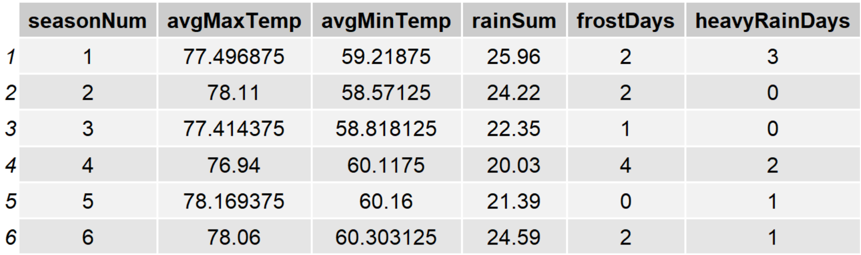
[7] Daily minimum below 32 degrees F°

[8] Heavy Rainfall is characterized as more than .3 inches of rain per hour (https://www.weathershack.com/static/ed-rain-measurement.html)

[9] Interestingly enough the comparison of my frost forecasting function to the actual data was the exact same

minimum temperature and then checking if it's below 32 degrees F°. After I created all of these functions to forecast for individual weather conditions I put them all together in one ultimate function called *simulateTheHarvest()* which returns a dataframe that's a forecast for overall weather for future growing seasons.

Fig. 5 - simulateTheHarvest() results, head of 1000000 iterations



	seasonNum	avgMaxTemp	avgMinTemp	rainSum	frostDays	heavyRainDays
1	1	77.496875	59.21875	25.96	2	3
2	2	78.11	58.57125	24.22	2	0
3	3	77.414375	58.818125	22.35	1	0
4	4	76.94	60.1175	20.03	4	2
5	5	78.169375	60.16	21.39	0	1
6	6	78.06	60.303125	24.59	2	1

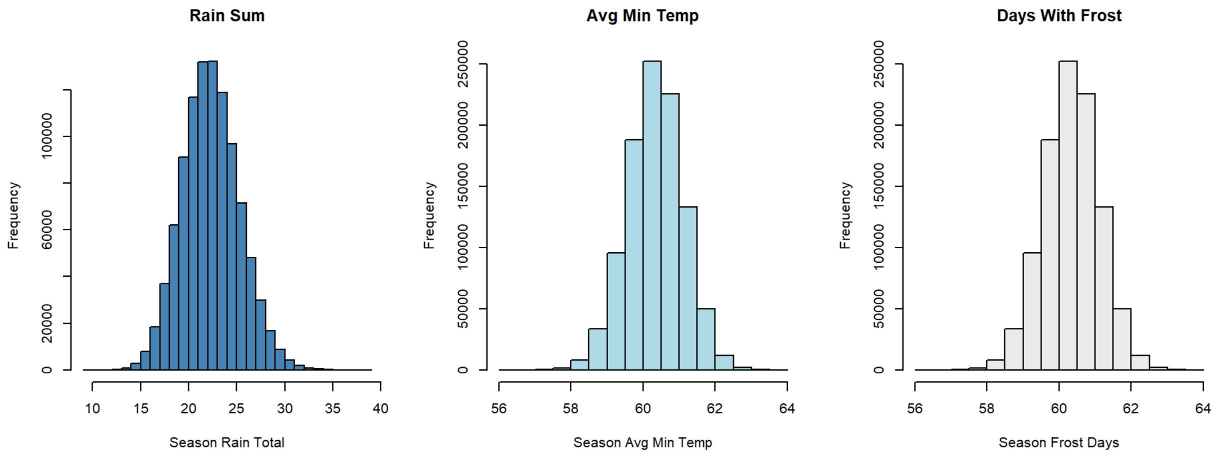
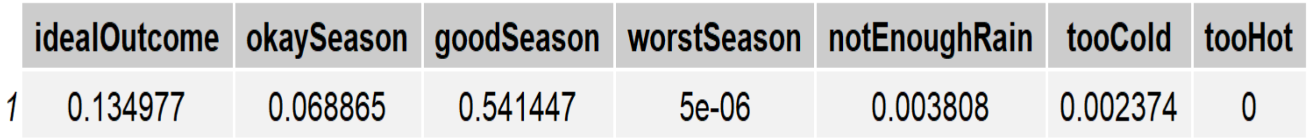


Fig. 6 - selected simulateTheHarvest() results, 1000000 iterations

Breaking down the Forecasts Results:

After running *simulateTheHarvest()* for one million iterations I have a huge dataset of forecasts for future growing seasons, but at this point it still isn't especially helpful yet for farmers, so I wrote another function called *breakDownTheHarvest()* to get the data to a more useful/interpretable state. Essentially this function will break down the forecast results into seven different outcomes based on the parameters for a good corn growing season: “idealOutcome” , “okaySeason” , “goodSeason” , “worstSeason” , “notEnoughRain” , “tooCold” , and “tooHot”. Obviously the next question is what are the parameters for a good corn growing season? My research on corn growing led me to this outline for a good season[10]: the average daily maximum temperature should be between 75 and 86 degrees F° , there should be more than 15 inches of rain over the course of the season, and then obviously there should be limited negative extreme weather events like frost and heavy rain. From here I set definitions for all those outcomes of the *breakDownTheHarvest()* function: “idealOutcome” is where every condition is met and there are no extreme weather events, “okaySeason” is where the weather and rain condition is met but above average heavy rain and frost days are observed, “goodSeason” is the same as “okaySeason” but average or below average heavy rain and frost days are observed, “worstSeason” is where the temperature and rain conditions aren’t met and above average heavy rain and frost days are observed, and the rest of the outcomes are self-explanatory[11]. After running the now created *breakDownTheHarvest()* function on the results from the *simulateTheHarvest()* function I get this data table[12].



	idealOutcome	okaySeason	goodSeason	worstSeason	notEnoughRain	tooCold	tooHot
1	0.134977	0.068865	0.541447	5e-06	0.003808	0.002374	0

Fig. 7 - breakDownTheHarvest() results on 1000000 iterations of *simulateTheHarvest()*

[10] Of course more than this is important for growing something, i.e cloud cover and sunlight etc., but this is the most baseline definition for quality weather conditions for corn growing.

[11] “tooHot” is avgMaxTemp > 86 , “tooCold” is avgMaxTemp < 75 , “notEnoughRain” is rainSum < 15 inches

[12] See Figure 7

Looking at the data table you start to understand why so much corn is grown here. The data table shows the environment here to be very agreeable to corn with 54% of forecasts having a “goodSeason” outcome, 13.4% having an “idealOutcome”, and maybe the best stat of all: there is essentially a 0% chance for a nightmare “worstSeason”. Putting all that together we can see that within the next five years[13] farmers have a 67.4% chance of having a very quality growing season[14]. Additionally, making a confidence interval for “idealOutcome” I can conclude with 95% confidence that the true percentage of “idealOutcome” outcomes is between 13.42% and 13.55%, so farmers have fairly decent odds of getting a perfect growing season within the next 5 years. While my *simulateTheHarvest()* and *breakDownTheHarvest()* functions are useful for predicting corn season weather for the next 5 years, they aren’t really trustworthy past that time frame… especially with the thought that weather in the far future could be drastically different due to global warming. Following that thought, I wonder if I’d be able to adjust my function so it can account for global warming and maybe even predict the weather for future corn seasons here in Gambier. How will those columns that have zeroes now like “tooHot”, “tooCold”, and

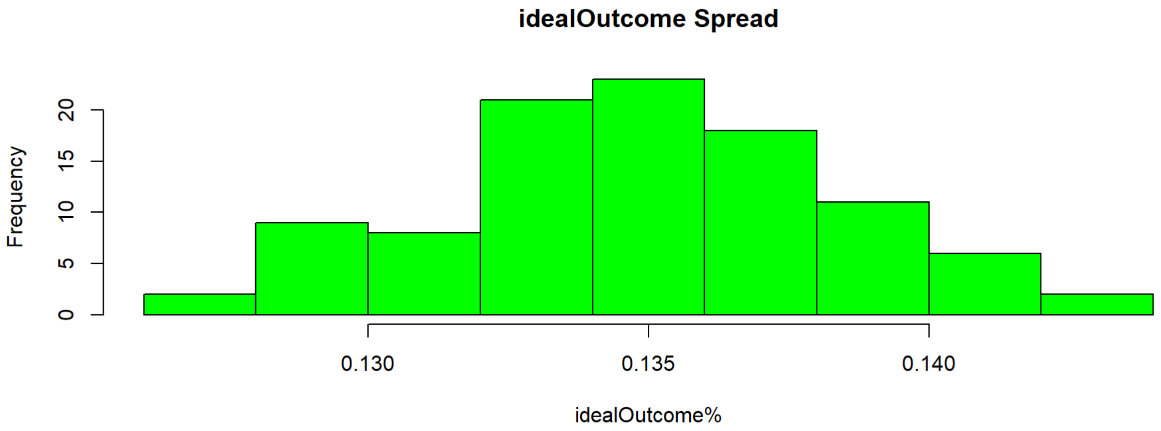


Fig. 7 - result of bootstrapping “idealOutcome” for the confidence interval

[13] Remember that *simulateTheHarvest()* results can be seen as a forecast for any one of the five years following 2021, with the assumption that the five years following 2021 will be similar to the 2016-2021 weather.

[14] “idealOutcome” percentage + “goodSeason” percentage

“notEnoughRain” change over the next 50, or even next 100 years? Will Gambier stay a utopia for corn? Let’s find out.

Adjusting for Global Warming:

The first step of adjusting my forecasting functions is figuring out generally how much the weather here has changed year over year, so I can add that change into my new adjusted function *simulateFutureHarvest()*. Using average daily max temperature as an example, I first found the difference in mean daily max temperature between the 2016-2021 and 1970-1975 datasets, as well as the difference in the standard deviation of daily max temperature between the two datasets. This difference turned out to be 1.65 F° for the means, and .37 F° for the standard deviations. From here I just divided both values by 50[15] to get what should be the year over year change for both values, which were .03 F° and .007 F° respectively. From analyzing[16] the results of my previous forecasting function *simulateTheHarvest()*, I know I should be able to use *rnorm(n , mean , sd)* to simulate daily maximum temperatures for the growing season.

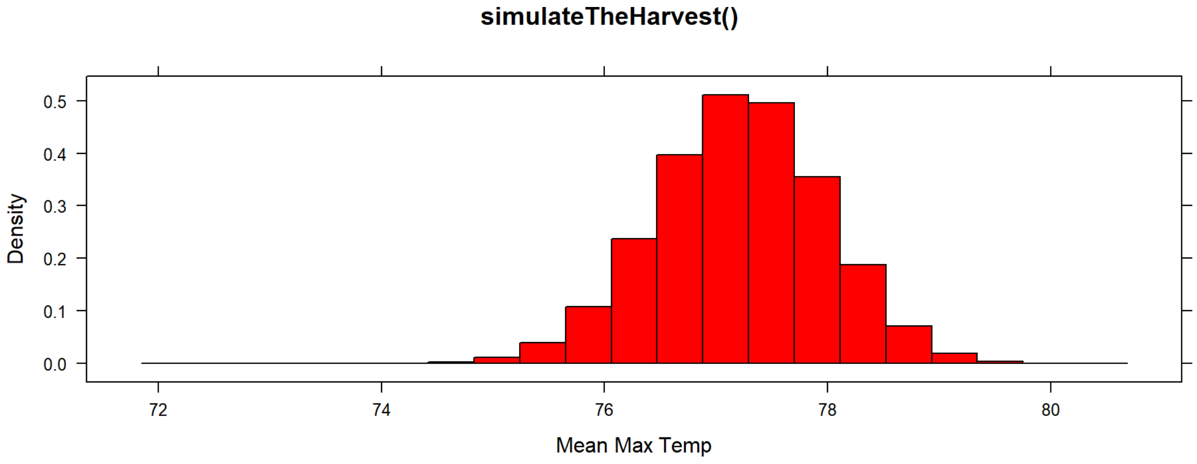


Fig. 8 - spread of mean daily maximum temperatures from simulateTheHarvest()

Within the *simulateFutureHarvest()* function a targeted year in the future will be passed, and using that I can adjust the maximum temperature estimation *rnorm()* by adding the targeted year

[15] There are 50 years between the two datasets
[16] See Figure 8

times the year over year change in mean daily max temperature to the mean of the *simulateFutureHarvest()* mean daily max temperature results, and vice versa for the standard deviation.

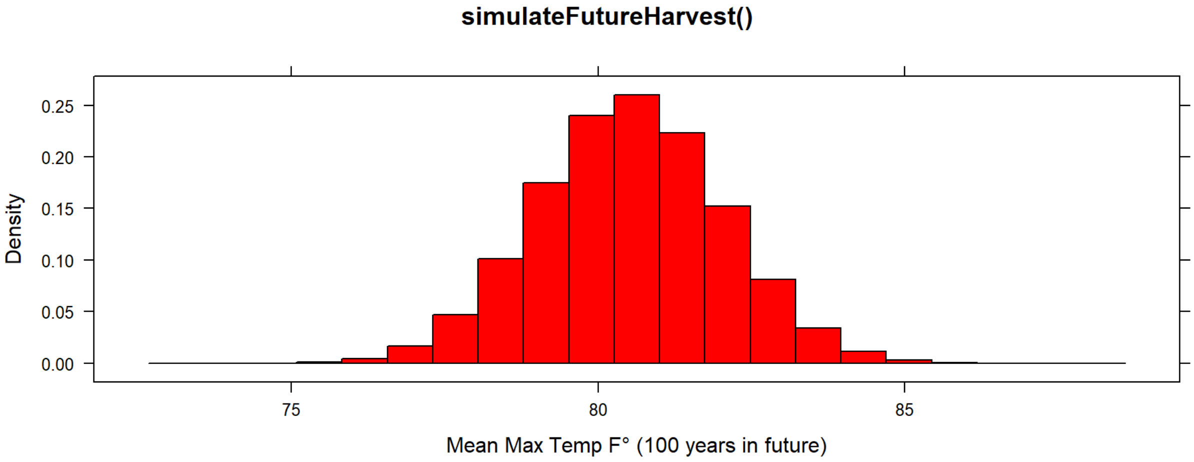


Fig. 9 - spread of possible mean daily maximum temperatures 100 years in the future

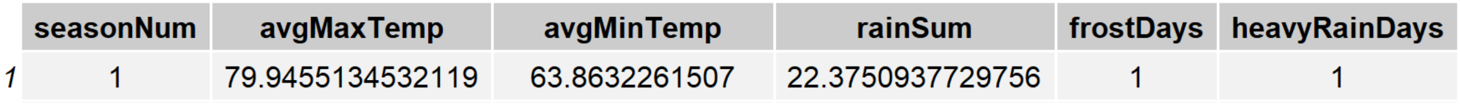
I used an identical process for putting together estimations for “rainSum” and “avgMinTemp”, but when I got to adjusting my forecasting for “heavyRainDays” and “frostDays” I ran into some problems. Unlike every other variable here I couldn’t use *rnorm()* to simulate the spread of “heavyRainDays” or “frostDays”, instead I had to switch to using *rexp()*[17]. Having to switch from *rnorm()* gave me some major headaches to be honest, but I was able to adjust my forecast for both variables in essentially the same way that I adjusted for “avgMaxTemp”. Overall using *rnorm()* and *rexp()* still resulted in very accurate forecasting of recent years when compared to the results of forecasting with bootstrapping[18], so honestly they could replace the bootstrapping if I really wanted to do that. Once I had all of these individual weather forecasting functions adjusted I put them all together into one function, *simulateFutureHarvest()*, which returns an

[17] See figure 4, or *simulateFloodingFuture()* or *simulateFrostFuture()* in my code.

[18] A breakdown of *simulateFutureHarvest()* for one year in the future was nearly identical to the results of *simulateTheHarvest(), some values were different but by a very small margin, such as “worstSeason” and “notEnoughRain” going from already very small values to zero on the results of simulateFutureHarvest()*

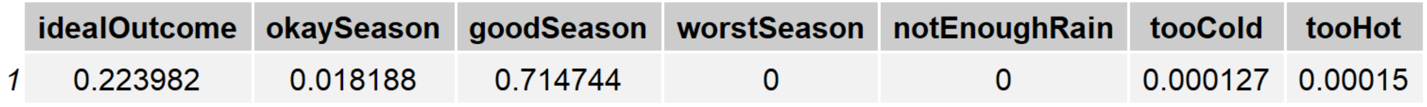
identical table to *simulateTheHarvest()*, except that it's forecasting for a year that is x years in the future. Next, I ran *simulateFutureHarvest()* for one

million intervals for a season 100 years in the future, and then put those results through my *breakDownTheHarvest()* function.



	seasonNum	avgMaxTemp	avgMinTemp	rainSum	frostDays	heavyRainDays
1	1	79.9455134532119	63.8632261507	22.3750937729756	1	1

Fig. 10 - result of simulateFutureHarvest() , 100 years in the future



	idealOutcome	okaySeason	goodSeason	worstSeason	notEnoughRain	tooCold	tooHot
1	0.223982	0.018188	0.714744	0	0	0.000127	0.00015

Fig. 11 - breakDownHarvest() results for 1000000 iterations of a season 100 years in the future

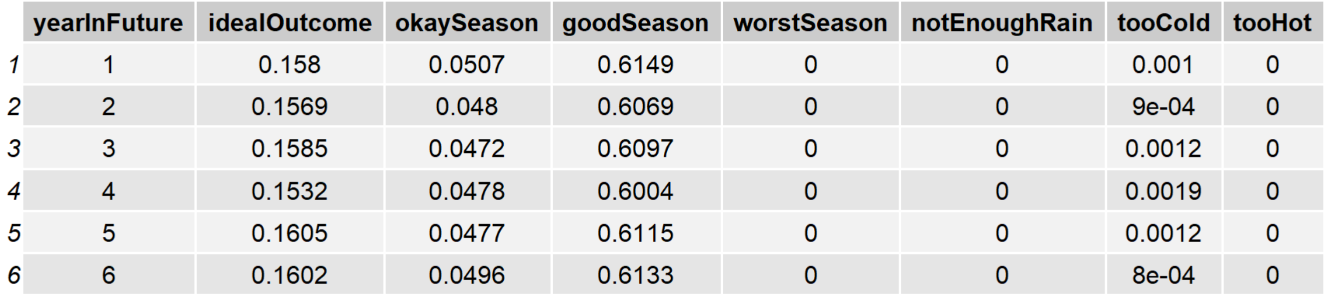
As we can see, the breakdown of the season 100 years in the future is a fairly different picture[19] than the earlier breakdown[20] of the one million forecasts for a season in the near future. What's most interesting to me is that 100 years in the future we can see that the "idealOutcome" is now a bit more likely, increasing by 8 percentage points when compared against the table seen in figure seven. Similarly, we can see that "goodSeason" is now occurring at a much higher rate as well, about 20 percentage points greater than we see in figure seven. Additionally, we can see that "tooHot" is now occurring at a very low rate, an increase from the zero occurrences seen in figure seven. While being able to look at the projected outcomes for seasons in a future year is interesting, I'd like to be able to see exactly how we can expect each of these outcomes to change values as time goes on. Because *simulateFutureHarvest()* is only good at estimating one

[19] You might notice all the percentages across this table add up to a value greater than 1 or 100%, that is because some of the conditions for these outcomes overlap a little bit.
[20] See figure 7

year in the future, I built a new function called *simulateManyYears()* that will return a dataframe essentially identically to the one seen in figure eleven, but it's expanded and goes year by year.

Future Weather Trends:

After running *simulateManyYears()* to cover 500 years[21] ,we can see some pretty interesting results.



	yearInFuture	idealOutcome	okaySeason	goodSeason	worstSeason	notEnoughRain	tooCold	tooHot
1	1	0.158	0.0507	0.6149	0	0	0.001	0
2	2	0.1569	0.048	0.6069	0	0	9e-04	0
3	3	0.1585	0.0472	0.6097	0	0	0.0012	0
4	4	0.1532	0.0478	0.6004	0	0	0.0019	0
5	5	0.1605	0.0477	0.6115	0	0	0.0012	0
6	6	0.1602	0.0496	0.6133	0	0	8e-04	0

Fig. 12 - result of simulateManyYears(), head

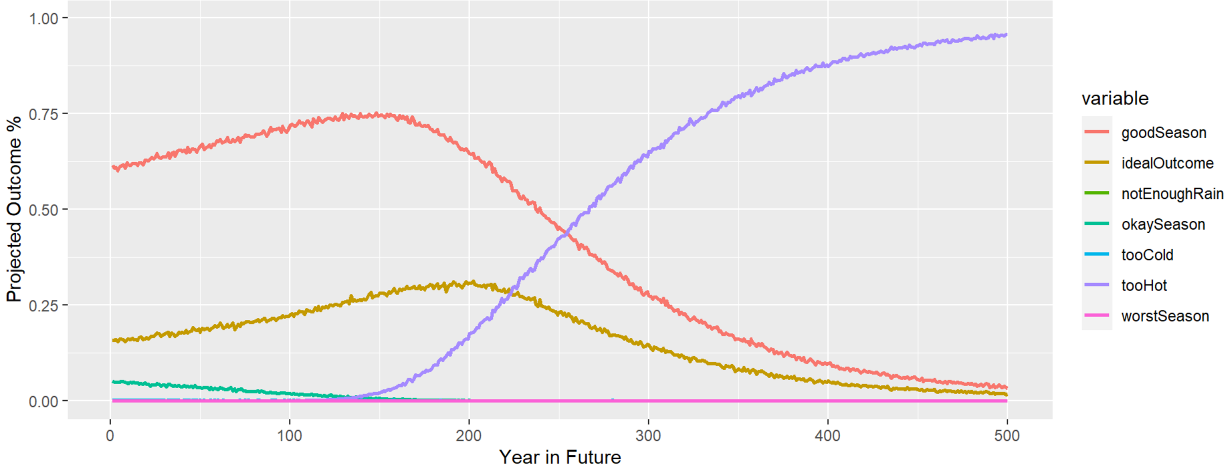


Fig. 13 - plot of simulateManyYears() results

Breaking down the line chart here we can see interestingly enough that farmers here in Mount Vernon should actually benefit somewhat from global warming for a time. 200 years[22] in the future "idealOutcome" has peaked, and "goodSeason" has just started to decline after peaking.

[21] Every year shows the results of 10000 simulated seasons for that year

[22] As you move along the x-axis the reliability of this probably decreases exponentially, but the trends seen do seem very logical when you think about global warming.

Additionally, we can see that the chance of the season outcome being "tooCold" is non-existent at that point, we can also see that there is now a fairly decent chance of the "tooHot" outcome. So while odds are best for a good or perfect growing season around 200 years in the future, these increased odds are offset somewhat by the growing chance of the season being too hot, ruining the crop altogether. At 400 years in the future it seems that corn growing in Mount Vernon, and probably Ohio as a whole will be a lost tradition as the projected outcome for the season being too hot is fairly constant around 95% after the year 400. While heat will be a problem going into the future, we can be assured that lack of rain won't be, with the "notEnoughRain" outcome bottoming out at 0% as we move into the future. This makes sense though considering that when comparing the 2016-2021 and 1970-1975 weather datasets there was an increase[23] in rain from 1975 to 2021, and the model works off of that. While corn growers may benefit in the short run[24] from global warming, in the long run it seems that global warming will wipe out their careers completely, with corn being ungrowable in Mount Vernon and probably the whole state by the year 2422. While 2422 seems far away, it's best if we can act now to try and slow global warming before reaching this scorched earth outcome where Mount Vernon has the climate of Florida.

Conclusions & Further Exploration:

Overall I wasn't too surprised by what my models found, I don't think it should take this project to convince someone that global warming will destroy us within the next 400 years. One thing I was very surprised by though was that global warming will actually benefit corn growers here in Mount Vernon for some time. I'd have to imagine that these benefits are offset by decreased crop growth due to global warming somewhere else in the United States or world

[23] This was an extraordinary small number, so by the year 400 the sum of rain through the growing season shouldn't be too different from what it is now according to my forecasting models

[24] Next 200 years

though. All in all I suppose that if global warming isn’t slowed down within our lifetime we can at least be grateful that we get to live through years of more bountiful corn harvests, and not live through the part of global warming where we get cooked like a steak at Peirce[25]

If I had more time to work on this project I would have liked to delve deeper into some of the individual aspects of my forecasting. For example, I would have liked to forecast daily weather max temperatures in the future adjusted for global warming, and then see how often we would be hitting new record high temperatures as we moved into the future. This is something that I basically did 50% of within the functions in my project, I just didn’t have the time to put it all together after wasting a good bit of time with this project messing around with regression models for forecasting that ended up being useless for the most part. Additionally, it would have been interesting to actually forecast corn bushels(?)[26] harvested based on the future weather forecasts. Sadly, I lack the expertise in the field of corn, no pun intended, to forecast such a measurement. If I could do that though, I could see it being very helpful for corn farmers who grow in Mount Vernon.

[25] Peirce overcooks all of their meat, except for chicken… when done wrong it’s severely undercooked, never overdone.

[26] As you can see, I know so much about corn that there is a question mark after bushel