CS146 Final Project:
Modeling and Forecasting Atmospheric CO_2 from 1958 Until 2058

Long Le

December 20, 2018

Abstract

In this paper, we employ a quadratic trend model, accounting for seasonal variations and white noises, to predict the CO_2 measurements up to the start of 2058. The model shows that April 1, 2034, is when the CO_2 measurement has a strong probability of reaching the critical level of 450ppm. We also find an unexplained short term pattern in the data set, as well as discuss the model's shortcomings in general.

1 Problem

CO_2 measurements play an important role in global climate change modeling. Our task is to build a statistical model that represents the Mauna Loa data set, and to use it to forecast future CO_2 measurements.

Using the model, we could answer important questions regarding future CO_2 patterns, as well as predict the point in time when dangerous climate change occurs.

To that end, we present two models, with sufficient results and critiques.

2 Assumptions

Both our models rely on the following assumptions:

1. There are three chief components in the model. One, the overarching trend of the data set. Two, the seasonal variations reflecting seasonal behavior patterns of humans and plants. And three, a random noise factor, which justifies the random fluctuations of the measurements. This assumption is based on our observation of the data, as well as our prior understanding of CO_2 cycles on Earth. We would later on explore if there are other hidden factors as well.

2. The data are collected under a standard condition (e.g. with similar tools, techniques, and procedures). This is an important motivation for the use of the aforementioned components of the model. In other words, we assume there is no other factor at work that influences our measurements. The noise will the be i.i.d, thus having low auto-correlation, as well as being normally distributed.

3 Variables & Quantities

The following variables and quantities are to be used through the paper:

1. Time step t and CO_2 measurements y. The time steps are positive integers representing the number of days since the first measurement recorded in the data set, plus 1 (so that the minimum of t is 1). The CO_2 measurement values are positive real numbers. These quantities are observed up to October 27, 2018. Any y values after that date are unobserved and are to be predicted by our models.

2. A vector c of unknown parameters, including the coefficients of the trend function (c_0 and c_1 in the case of linear trend, and c_0, c_1, and c_2 in the case of quadratic trend), the amplitude and the "adjusted" [1] phase of the periodic function, as well as the parameter for noise distribution (in this case - σ, as we assume the noise is normally distributed).

4 Data Preprocessing

Upon obtaining the data from the Scripps CO_2 program, the data undergoes preprocessing to facilitate future analysis efforts:

1. The wordy header of descriptions is removed using Excel. This is so that pandas could easily parse the csv.

2. The date column is converted from *string* to Python's *datetime* type, then further converted to the unit of days since the first measurement is made. The variable t is now a positive integer, which could be fed into Stan.

3. The data is split into a train set (first 70% of the data) and a test set (remaining 30 %). This is so that we could validate a model's robustness on the test set before using it for predictions.

4. There has been an attempt to normalize the data to a $[0, 1]$ scale. However, this idea was discarded as it would have made it harder to interpret the periodic function.

[1] Following the notation of the assignment pdf, c_3 is not the actual phase of the cosine function. $c_3 \times \left(2\pi \frac{1}{365.25}\right)^{-1}$ is, hence "adjusted".

5 Model 1: Linear trend

5.1 Description

The first model is built with the following assumptions of the major components:

1. Overarching trend: the CO_2 measurements follow a linear trend: $c_0 + c_1 \times t$.
2. Seasonal Variations: modeled using a periodic function: $c_2 \times cos(2\pi \frac{1}{365.25} + c_3)$
3. Noise: the noise is drawn from a normal distribution centered at 0, with $\sigma = c_4$.

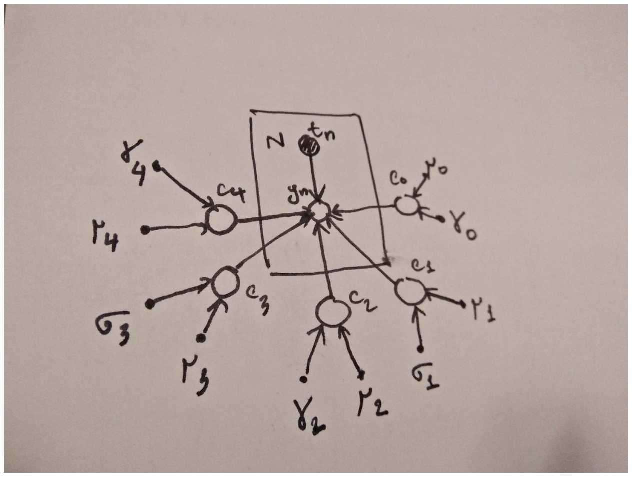
Our goal is to find the posterior of each of the 5 parameters represented by the vector c.

5.2 Priors

1. c_0: As the data is not normalized (justified above), c_0 should be large, probably close to $y[1]$ (or the minimum CO_2 value in the series, judging that the series is heading upwards only), thus could be drawn from a Normal$(y[1], \sigma)$ prior. In the Stan model, we let it be drawn from a $Cauchy(0, 1)$ prior, however, so we do not have to assume the σ, or to set a hyperparameter for σ. c_0 is a positive real number.
2. c_1: The linear coefficient is approximately $\frac{max(y)-min(y)}{max(t)-min(t)}$ 0, thus drawn from a standard normal distribution. We clearly observe an uptrend, so c_1 is a positive real number
3. c_2: We do not have an approximation for this parameter. We expect it to be low. But as the data is unnormalized, we do not want to make a guess regarding its range, thus choosing to use a $Cauchy(0, 1)$ prior. c_2 is conditioned to be positive and real, as a negative c_2 simply flips the function around the horizontal axis, which is equivalent to shifting the function by half of the period.
4. c_3: The phase shifts the periodic function to the left $c_3 \times \left(2\pi \frac{1}{365.25}\right)^{-1}$ units. We would like $|phase| \leq \frac{1}{2}(period)$ to avoid its distribution having multiple modes, which is due to the parameter's periodic implications. We know the period is 365.25 days, thus $-\pi \leq c_3 \leq \pi$. Having explicitly specified its range in Pystan, we sample c_3 from Normal$(0, \pi)$.
5. c_4: The standard deviation of the normal distribution from which noise is drawn. c_4, then, is a positive real number. I choose to draw c_4 from a $Cauchy(0, 1)$ prior, for simplicity.

5.3 Graph

The model is represented as a directed graph below:



The day value t_n is observed, while each parameter c_x is a variable that is determined using the two known hyperparameters, a location μ_x, and a scale γ_x (if drawn from Cauchy distribution), or σ_x (if drawn from Gaussian distribution). For example, $\mu_0 = 0$ and $\gamma_0 = 1$, corresponding to the parameters of a Cauchy(0,1) distribution.

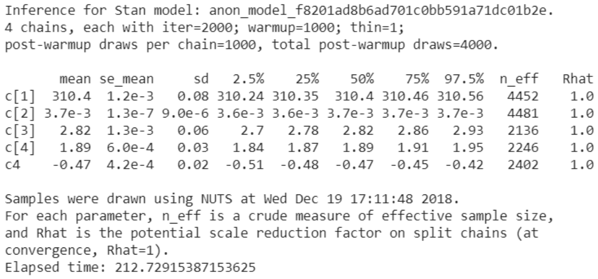
Notice that the graph is for the predictive process only, where t_n is observed but not y_n. Otherwise, in the training model, y_n is also observed, and according Bishop (2006), need to be shaded.

5.4 Evaluation

5.4.1 Convergence

The below results are obtained after running the model on the train set:

The number of effective samples (N$_{eff}$) is high ($\geq$ 2000) and R-hat equals to 1.0. These indicate that the model converges well.



```
Inference for Stan model: anon_model_f8201ad8b6ad701c0bb591a71dc01b2e.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

       mean se_mean     sd   2.5%    25%    50%    75%  97.5%  n_eff   Rhat
c[1]  310.4  1.2e-3   0.08 310.24 310.35  310.4 310.46 310.56   4452    1.0
c[2] 3.7e-3  1.3e-7 9.0e-6 3.6e-3 3.6e-3 3.7e-3 3.7e-3 3.7e-3   4481    1.0
c[3]   2.82  1.3e-3   0.06    2.7   2.78   2.82   2.86   2.93   2136    1.0
c[4]   1.89  6.0e-4   0.03   1.84   1.87   1.89   1.91   1.95   2246    1.0
c4    -0.47  4.2e-4   0.02  -0.51  -0.48  -0.47  -0.45  -0.42   2402    1.0

Samples were drawn using NUTS at Wed Dec 19 17:11:48 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
Elapsed time: 212.72915387153625
```

Below is the pair plot of unknown parameters **c**. The distribution plots show unimodal distributions, which would not incur sampling error to Stan. c_1 looks narrowly distributed and that is understandable, as we expected it to be close to 0, with fairly high uncertainty. Indeed, the 95% confident interval of the parameter is very narrow: $[3.6e-3, 3.7e-3]$.

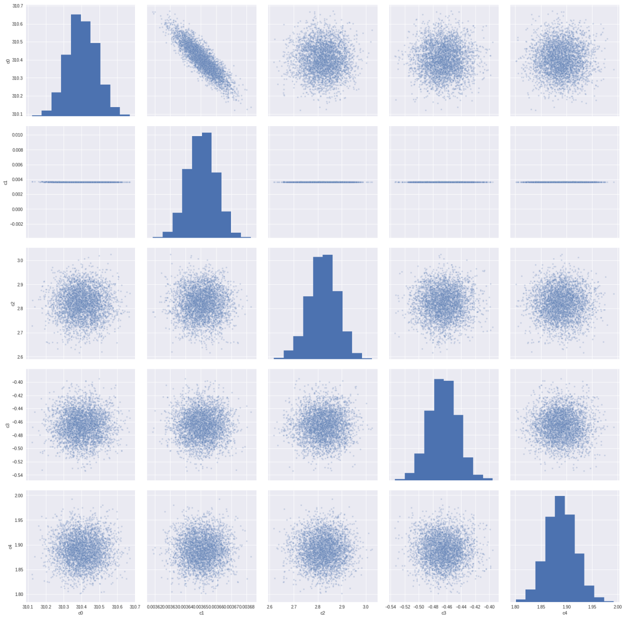
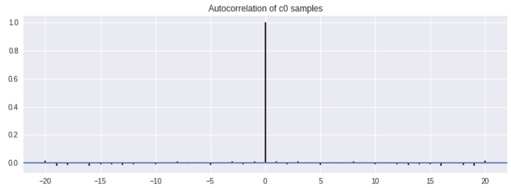
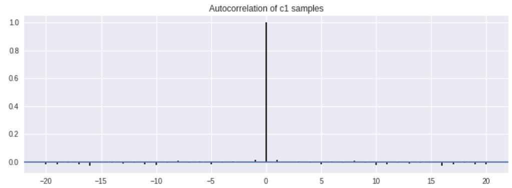
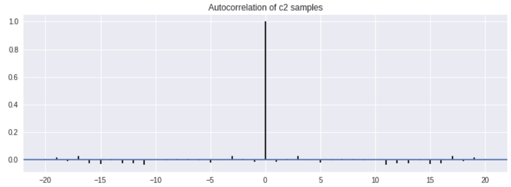


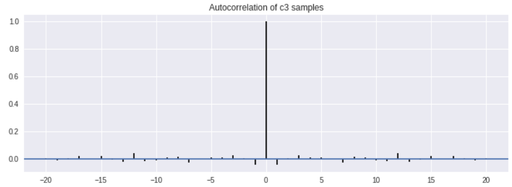
Fig 1: Pair plot of unknown parameters **c**.

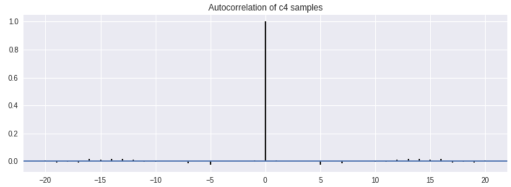
For each parameter from c_0 to c_5, the autocorrelation is low, for all lag values (except for 0, as all series correlate with themselves), indicating that samples generated for each parameter are indeed independent:





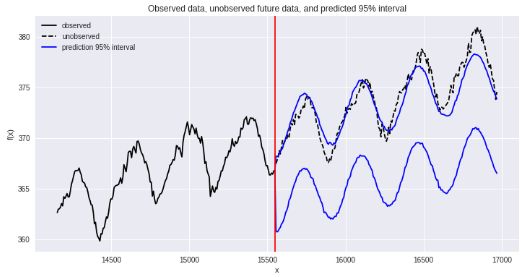




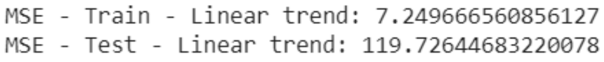


5.4.2 Robustness

We test the model on the test set. The results are discussed below:

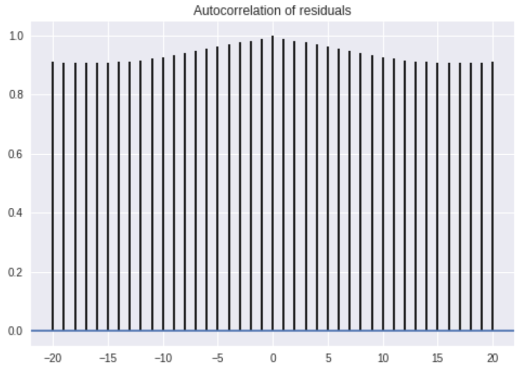


Inspecting the model's predictions on the test set, we notice large errors even in the beginning. In fact, the mean squared error of the test set is very high - at up to 119.7, while the mean squared error of the train set is low - at only 7.25. This signalling that the model is overfitted to the train set, thus having very limited predictive power.

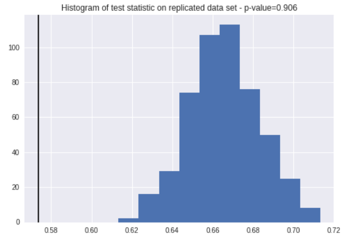


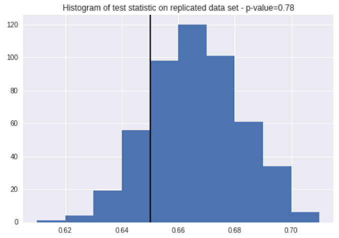
```
MSE - Train - Linear trend: 7.249666560856127
MSE - Test - Linear trend: 119.72644683220078
```

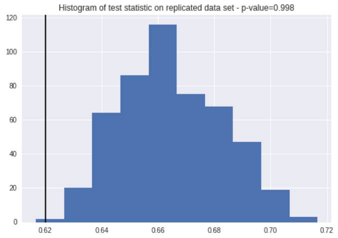
Indeed, by plotting the autocorrelation of the residuals on the test set, we notice very high autocorrelation, which signals that there are patterns in the set not accounted for by the model.



Last but not least, we inspect a test statistic - the proportion of data in the range $[\mu - \sigma; \mu + \sigma]$. However, we do not calculate the test statistic for the whole time series, due to the time and computational power restrictions.Instead, we resort to calculating the test quantity for the first 300 data points, middle 300 data points, and the last 300 data points, of the train set. This method effectively covers almost half of the train set, while giving us insights into the performance of the model in various parts of the time series.







The very high p-values (0.90 and 0.98) for the first 300 and the last 300 data points mean that the replicated data are statistically significantly different from the real data. The model is not capable of explaining the distribution of the real data well in those periods. The model seems to work better for the group of data points in the middle of the time series ($p - value = 0.78$). This is understandable as we could see from the time series plot that the middle is the densest, thus it is likely that the regression line $c_0 + c_1 \times t$ runs through this part.

Many indicators do not favor this model. We conclude that this model could not represent the data well and, thus, unfit for predictive purposes.

6 Model 2: Quadratic trend

6.1 Description

The second model is built with the following assumption of the major components:

1. Overarching trend: The CO$_2$ measurements follow a quadratic trend: $c_0 + c_1 \times t + c_2 \times t^2$.

2. Seasonal Variations: modeled using a periodic function: $c_3 \times cos(2\pi \frac{1}{365.25} + c_4)$

3. Noise: the noise is drawn from a normal distribution centered at 0, with $\sigma = c_5$.

In other words, we start with a simple modification from the first model - shifting from a linear trend to a quadratic trend. We would later on explore if there are alternative methods to model the seasonal variations and the noise.

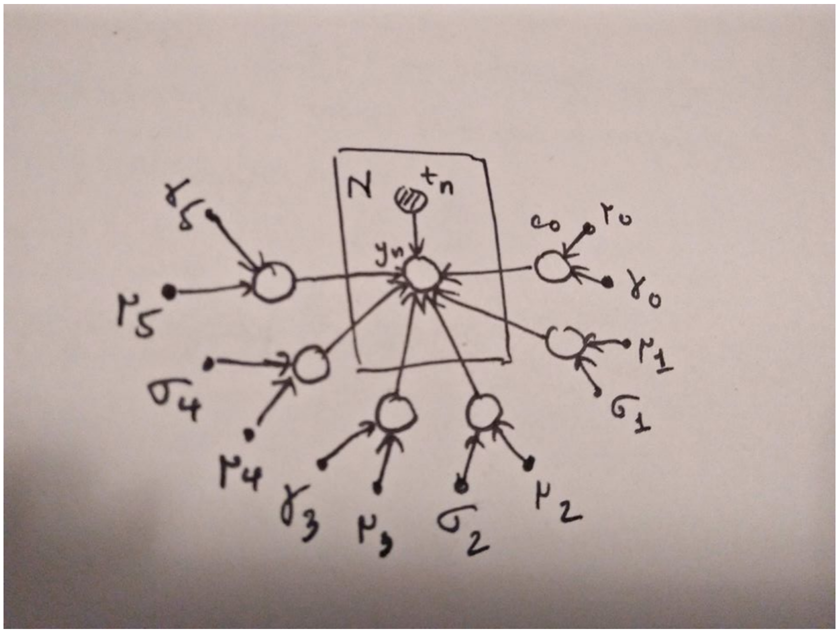
6.2 Priors

There are 6 parameters in this model. Apart for c_2, which is the quadratic coefficient of the trend function, other parameters are similar to the first model, thus having similar constraints and prior distribution.

We could guess that c_2 is close to 0 as the rate of growth of the parabola trend is very slow. Thus, c_2 is drawn from a standard normal distribution. The trend is upward, so c_2 is positive.

6.3 Graph

The predictive model is represented as a directed graph below:



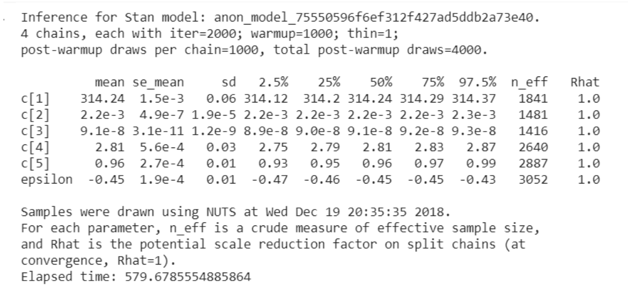
The graph is very similar to that of the first model. The only difference is that we have added one more parameter, corresponding to the one additional coefficient as we change from linear trend model to quadratic trend model.

6.4 Evaluation

6.4.1 Convergence

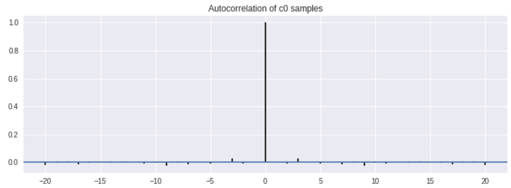
The below results are obtained after running the model on the train set:

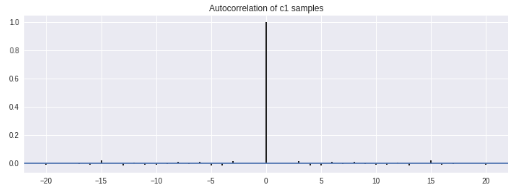
The number of effective samples (N_{eff}) is high (≥ 1400) and R-hat equals to 1.0. These indicate that the model converges well.

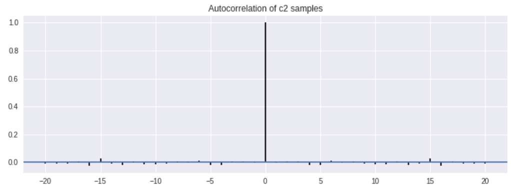


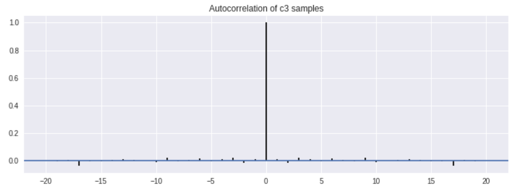
```
Inference for Stan model: anon_model_75550596f6ef312f427ad5ddb2a73e40.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean     sd   2.5%    25%    50%    75%  97.5%  n_eff   Rhat
c[1]    314.24  1.5e-3   0.06 314.12  314.2 314.24 314.29 314.37   1841    1.0
c[2]    2.2e-3  4.9e-7 1.9e-5 2.2e-3 2.2e-3 2.2e-3 2.2e-3 2.3e-3   1481    1.0
c[3]    9.1e-8 3.1e-11 1.2e-9 8.9e-8 9.0e-8 9.1e-8 9.2e-8 9.3e-8   1416    1.0
c[4]      2.81  5.6e-4   0.03   2.75   2.79   2.81   2.83   2.87   2640    1.0
c[5]      0.96  2.7e-4   0.01   0.93   0.95   0.96   0.97   0.99   2887    1.0
epsilon  -0.45  1.9e-4   0.01  -0.47  -0.46  -0.45  -0.45  -0.43   3052    1.0

Samples were drawn using NUTS at Wed Dec 19 20:35:35 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
Elapsed time: 579.6785554885864
```

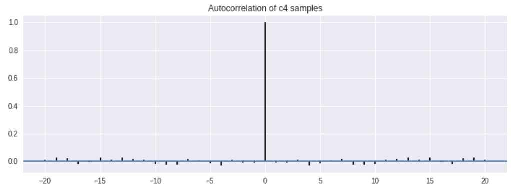
Indeed, autocorrelation is low for all parameters (displayed below) at all lag values apart from 0:

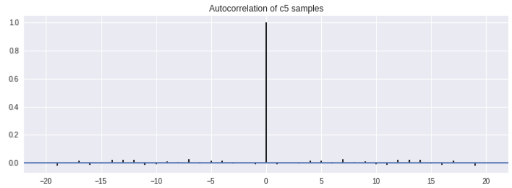


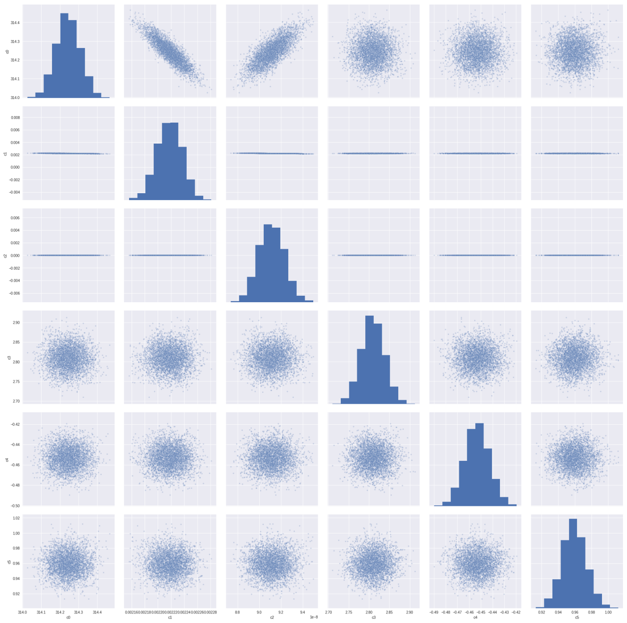










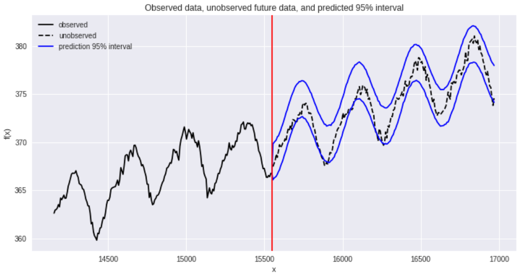


The pair plot shows unimodal distributions, which is also one reason why the model does not run into errors.

Again, c_1 and c_2 seems narrowly distributed, and that is understandable given that these variables are distributed densely around 0 only.

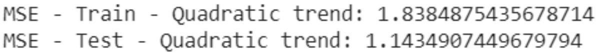
6.4.2 Robustness

Again, we test the model on the test set. The results are discussed below:



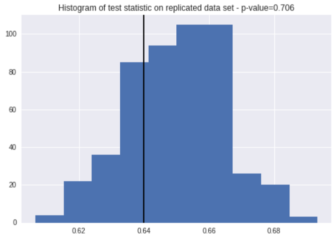
This time, we notice that the prediction seems to closely follows the real data, at least in the first 200 data points in the test set. The confidence interval also seems pretty narrow.

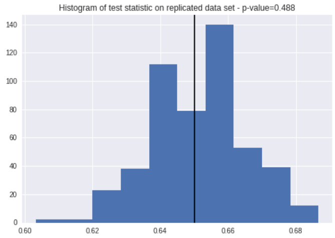
Indeed, we show that the MSE for both the train set and the test set are both low, indicating the model could predict sufficiently well the test set.

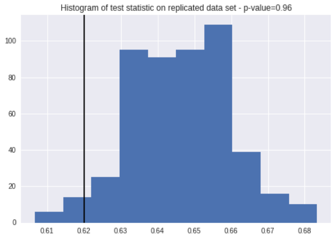


```
MSE - Train - Quadratic trend: 1.8384875435678714
MSE - Test - Quadratic trend: 1.1434907449679794
```

Test statistics are also not ideal, but visibly superior than those of the first model. The first 300 and the middle 300 points represent the model well, with p-value close to 0.5, thus much better than the first model. The last 300 points are somewhat concerning, for the p-value is very high (0.96). However, notice that the distribution of the test statistic is narrow, with the min value being 0.61 and the max value being 0.68. Most mass concentrates at around 0.645. At the same time, the test statistic distribution of the first model for the last 300 data points ranges widely, from 0.62 to 0.72, with the most mass around 0.665. In other words, the first model does not perform nearly as consistently close to the real data as this model.





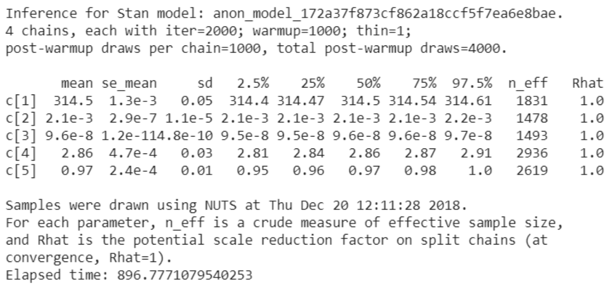


The test statistics hint that this model is superior to the first model, as well as being capable of representing the data to an acceptable extent. It probably is not overfitted either, as 1/ the MSE is also low for the test set and 2/ its the descriptive effect is not perfect, as seen in the test statistics performance. Thus, we decide to use this model for predictions. We would later reflect the model's shortcomings and propose approach to improve it.

6.4.3 Predictions

We reuse the model from above to make predictions up to the start of 2058. The model is first fit on the whole data set of observed values (i.e. including both the train and test set in previous sections). The results are discussed below:

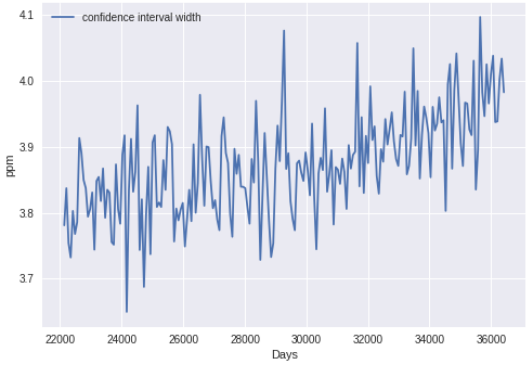
First, the model converges well, achieving high N$_{eff}$ and R-hat, thus well-suited for making predictions.



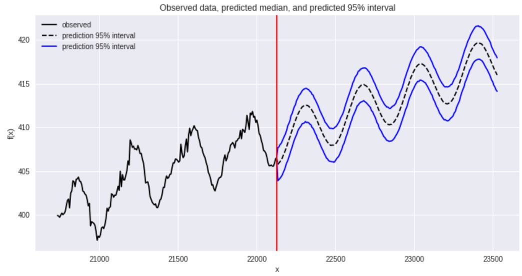
```
Inference for Stan model: anon_model_172a37f873cf862a18ccf5f7ea6e8bae.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

       mean se_mean     sd   2.5%    25%    50%    75%  97.5%  n_eff   Rhat
c[1]  314.5  1.3e-3   0.05  314.4 314.47  314.5 314.54 314.61   1831    1.0
c[2] 2.1e-3  2.9e-7 1.1e-5 2.1e-3 2.1e-3 2.1e-3 2.1e-3 2.2e-3   1478    1.0
c[3] 9.6e-8 1.2e-114.8e-10 9.5e-8 9.5e-8 9.6e-8 9.6e-8 9.7e-8   1493    1.0
c[4]   2.86  4.7e-4   0.03   2.81   2.84   2.86   2.87   2.91   2936    1.0
c[5]   0.97  2.4e-4   0.01   0.95   0.96   0.97   0.98    1.0   2619    1.0

Samples were drawn using NUTS at Thu Dec 20 12:11:28 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
Elapsed time: 896.7771079540253
```

The model achieves high certainty, the standard deviations of all parameters are below 0.05. Hence, we expect that the confidence intervals would be narrow in predictions.

Indeed, we observe a fairly stably low confidence interval width (i.e. the 97.5th-percentile subtracted by the 2.5th-percentile) - at 3 to 4 ppm. There is a clear increasing trend, however, signaling that the model is performing as expected, with increasing uncertainty over time.



The plot below displays the predictions of the model in the next 200 weeks from the last observed data point, the dotted line represent the 50th-percentile values of the predictions, and the two blue lines denote the confidence interval. There is a somewhat abrupt drop from the last observed data point to the first predicted data point. The difference is not significant, however, and should not indicate the quality of the predictions as a whole.



We then use the model to predict when the CO_2 level reaches the critical value. According to the model, it takes 27762 days from the first observed data point until the 50th-percentile CO_2 level reaches the critical threshold of 450 ppm, which corresponds to April 1st, 2034. The prediction corresponds well to that of Climate Central (2016), which forecasts that we could cross the threshold in 2030 (the difference could be due to the usage of different models, or of different thresholds).

We choose to base our inference on the 50th percentile as that signifies that half of our predicted values for one particular point in the time series is above a certain number (in this case, the threshold of 450 ppm).

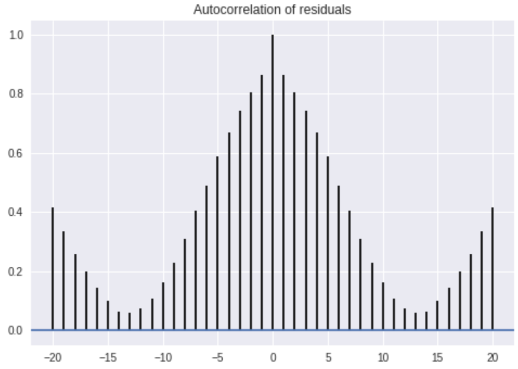
In other words, as the model predicts the CO_2 values at each point in time in the future not as a single value, but as a distribution of values to reflect the uncertainty, the 50the percentile here means that half of the data points in the generated distribution are equal or greater than 450ppm.

We believe that 50% chance of occurring is high enough to be wary of, taken into account that the model might be slightly optimistic, given that our CO_2 emission might increase even faster in the next decade due to the rise of developing countries and the failure to enforce treaties and protocols to regulate this matter.

6.4.4 Hidden patterns

Even though we have justified that the model above could describe the data pretty well, at least the distribution aspect of the data, we also need to acknowledge that there are patterns in the data that our basic assumptions did not account for.

The residuals of the quadratic model is still autocorrelated, especially with low lag values (see graph below). This indicates that there is some pattern in short time frames that our model neglects.



One possible explanation for such patterns is that we have not modeled the seasonal variations well. Inspecting the true seasonal variations (residuals after subtracting the data y_n by the result of the quadratic function $c_0 + c_1 \times t_n + c_2 \times t_n^2$ green line below), we could see that our periodic function does not explain the data, especially the extremely high values at the peak of the seasons, well enough. This suggests that we could find a better way to model the amplitude of the periodic function, such as modeling it as a function of time.

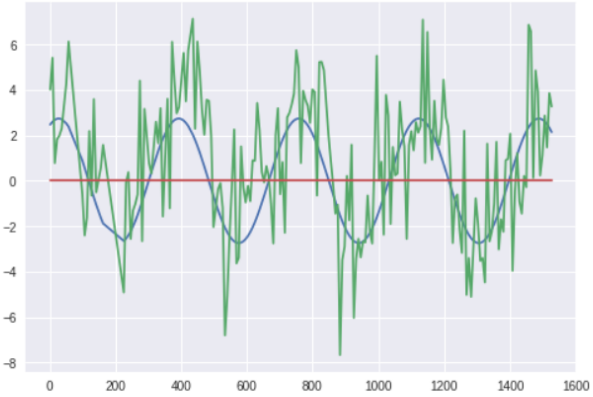


Figure 1: Seasonal variation - first 200 data points

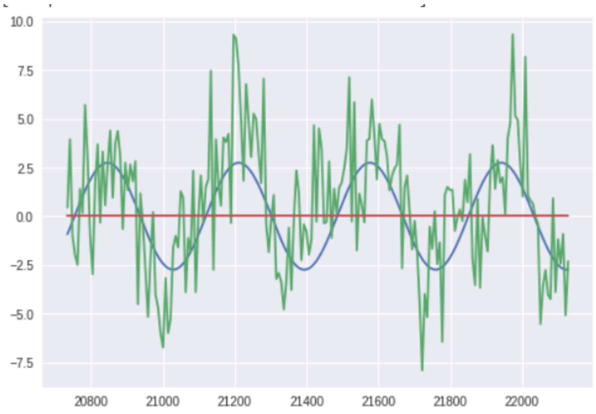


Figure 2: Seasonal variation - last 200 data points

This pattern is not of major impact, as we have seen that our model performs well on the test set, but it should be looked into to improve our accuracy and certainty, as well as our understanding of the CO_2 emission pattern.

7 Closing Remarks

We dedicate the closing remark to the second model, as it is built upon the first model and is shown to be more advanced in both descriptive and predictive aspects.

7.1 Strengths

1. The model could moderately describe the real data. We notice significant overlaps between the replicated data and the real data. Primary factors such as the overarching trend and the seasonal variations are very noticeable and correspond well to reality. The test statistics are not optimal (i.e. exactly 0.5), but still prove that the model is representative of the real data to an acceptable extent.

2. The confidence interval rises over time, which accurately reflects the fact that predictions get less certain the further it is in time. At the same time, the confidence interval growth is slow. Thus, we maintain a fairly stably high certainly (narrow confidence interval) level throughout our validation set. In other words, we probably have struck the right balance between being realistic and being reliable.

7.2 Weaknesses

1. Our replicated data seems much more fuzzy that the real data. In fact, the residuals still shows some autocorrelation! This means that there is still some hidden patterns that are not fully explained by our model.

2. Our test statistic is not perfect. This might relate to the first weakness, but worth being inspected on its own. The point above pertains to our assumption of white noise alone, while we could alter our models for all the three major components (trend, seasonal variations, and noise) to improve our test statistic. It should be worth noticing that as the test statistic of the real data is lower than the replicated data, the distribution of the replicated data has fewer extreme values (that do not lie in between the $[\mu - \sigma; \mu + \sigma]$ interval). Extreme points likely appear at the peaks and the troughs of the periodic function, thus we could hypothesize that there could be a better way to model the amplitude of our periodic function.

3. Practicality-wise, the run time is slow. Some models could take up to 8 minutes for sampling alone.

7.3 Suggested Improvements

1. The ”noise” turns out to be not quite random. We could model it using a Gaussian process. The effort might not be quite worth it, for the quadratic trend model is fairly good with the data. But it would be interesting to find out what this pattern actually corresponds to in reality.

2. Another suggestion is to model the amplitude c_3 not as a fixed value, but as a function that varies over time. This might allows for the model to account for the difference.

3. We could combine various models to enhance the predictive effect, as each model might be excelled at explaining an aspect of the data.

4. We might also want to experiment with narrower priors to optimize run time for practicality. Cauchy distributions are too broad and thus would take more time for sampling.

8 References

1. Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Retrieved from https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/Bishop-PRML-sample.pdf

2. Climate Central. (2016). *Flirting with the* 1.5 *C threshold.* Retrieved from http://www.climatecentral.org/news/world-flirts-with-1.5C-threshold-20260

9 Appendix

The codes are too lengthy to be included. But it could be found here.