Forecasting Weekly Weather Conditions Through the Use of Markov Chains

David Holcer

IB Mathematics HL

May 2019

Introduction

Personal Interest and Connection:

Growing up in Los Angeles California, I was exposed to an almost perpetual sunny climate. I often noted that cloudy were more likely to appear on a given day if the previous day was also cloudy. Moving to Vancouver, British Columbia when I was thirteen allowed me to experience a different climate with more volatile weather. Just as the forecast in Los Angeles was somewhat predictable, I observed that rainy days often followed rainy days, but this pattern was a lot less consistent and the predictability of weather in Vancouver decreased significantly. Interested in better understanding these weather patterns, I researched modern weather forecasting techniques. Instead of studying the instruments used in these forecasting methods, such as the Doppler Radar which is able to detect particles, their intensity, and their motion (*How Do Radars Work?*); satellites; dropsondes; or weather stations (Ward), I focused my efforts on the mathematical analyses that are performed on weather data in order to obtain a forecast. I finally narrowed down my focus to a technique that is widely used to predict both forecast and precipitation (Rotondi) known as Markov Chains. A *Markov Chain* is a model that is used to represent the transition of different states based on certain probabilities and is determined solely based on a current event not on a collection of past events. (Fewster)

Focus:

To investigate the weather forecast of a given city, I will analyze historical data over 6 years (2012-2017 inclusive). Probabilities of future conditions will be retrieved from the data and will be utilized in a first and second order Markov Chain to retrieve the probabilities of future forecasts. This paper aims to predict a weekly weather forecast given a week's starting weather condition.

Theory:

A Markov Chain is a model to map the probabilities of transitioning from one state to another state, and is used often in the analysis of literature, information theory, voice recognition, and when searching web pages. (Hilgers et al.)

Markov Chain models focus on time dependent data with a series of discrete random variables. This model is comprised of numerous *states*, or the different values it contains. Markov Chains are known as being *k-order* where k represents the number of values used to determine the probability of the next state (Schoof and Pryor). For this particular study, I will be focusing on first and second order Markov Chains with a total of 7 states ($X_t = n \; \forall \; n \in \mathbb{Z}^+ \leq 7$).

All Markov Chains fulfill the *Markov Property*, which states that only the most recent event affects the probability of the next event occurring. This property can be shown mathematically as follows (Fewster):

$$P(X_{t+1} = s \mid X_t = s_t, X_t = s_{t+1}, \ldots, X_0 = s_0) = P(X_{t+1} = s \mid X_{t+1} = s_t)$$

Where X_t represents the current state, and X_{t+1} the next state, and X_{t-n} former states.
The variables s and s_n represent all possible states, and
the subscript $n, t \in \mathbb{Z}^+$.
Thus, a sequence of discrete random variables $\{X_t, X_{t+1}, X_{t+2}, \ldots\}$ is a Markov Chain if and only if it satisfies the *Markov Property* (Fewster).
A *Transition Matrix,* or *Stochastic Matrix* is a matrix which shows in each position, the probability of a transition between states. Its rows represent the current states X_t and its columns the next state X_{t+1} (Fewster). Elements of this matrix must be real numbers in the closed interval [0,1] (Weisstein) and the elements of each row must add to 1 (Fewster).

Discussion

Data retrieval:

Historical hourly weather data from 2012-2017 was acquired from the website "Kaggle". This data was collected using a Weather Application Program Interface (API) obtained from the "OpenWeatherMap" website. This data contains hourly conditions for 30 US & Canadian Cities and 6 Israeli Cities. The focus of this investigation will be on the Los Angeles and Vancouver "weather conditions" data sets. A brief description of the following cities is summarized below.

Table 1. City attributes

City	Country	Latitude	Longitude
Vancouver	Canada	49.24966	-123.119339
Los Angeles	United States	34.052231	-118.243683

Data manipulation:

In order to output a daily weather forecast, the data must be displayed in daily increments, not hourly ones. A Python code was written in order to parse the original comma separated values (.csv) file into the required format. The unique mode of each day (24 hours) was taken as the daily condition, and if there was no unique mode present, the condition at noon (12:00 pm) was used instead. Weather conditions were mapped onto one of seven states accordingly:

Table 2. Mapped conditions obtained from data conditions

N (state)	**Mapped condition**	**Data conditions**
1	Clear	sky is clear, few clouds
2	Light rain	light rain, light intensity shower rain, proximity shower rain, light intensity drizzle
3	Cloudy	overcast clouds, scattered clouds, broken clouds
4	Foggy	mist, fog
5	Heavy rain	moderate rain, heavy intensity rain, shower rain, very heavy rain
6	Smoky	haze, fog
7	Snowy	light shower snow, light snow, heavy snow, snow

The Vancouver data set contained 44,460 hourly conditions of which 11 were removed so that the data set began at the beginning (12:00 am) of 2012-10-02. The Los Angeles data set contained 45,252 hourly conditions of which 11 were removed so that the data began at the beginning (12:00 am) of 2012-10-02, and an additional 792 data points were removed so that the data set ended on 2017-10-27, the same date as the Vancouver data set. The remaining 44,449 data points were converted to 1,852 daily data points of which the final 365 days were removed for future comparison. This left each data set with 1,487 data points each. The frequency of the mapped conditions for both cities may be found below.

Figure 1. Frequency of Vancouver's daily weather conditions

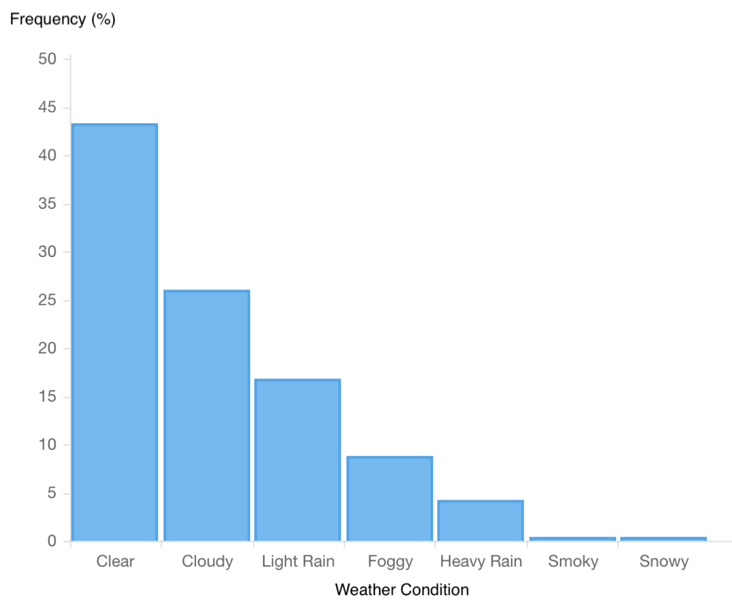
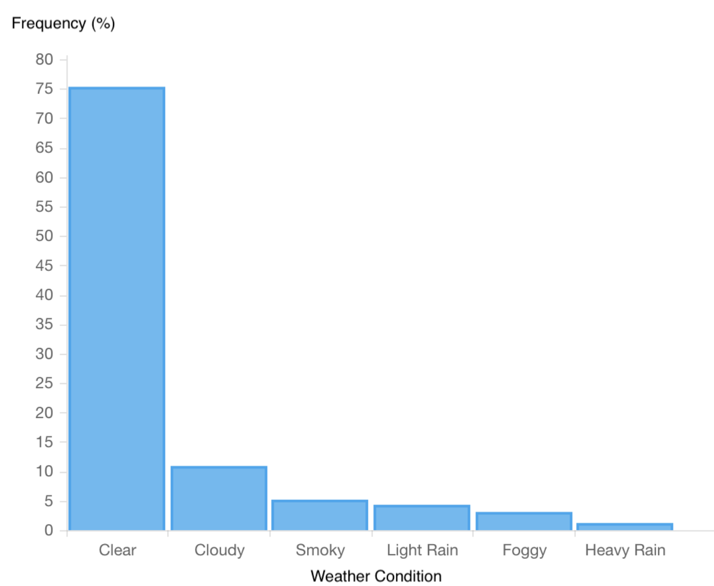


Figure 2. Frequency of Los Angeles' daily weather conditions



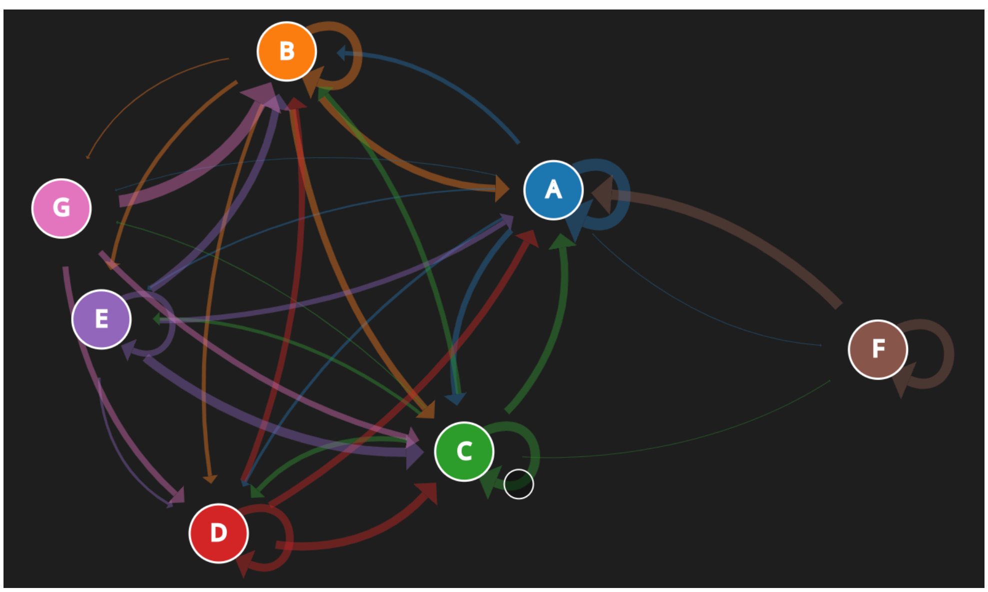
Data Analysis:

To begin the data analysis, first-order transition matrices were obtained from the manipulated data sets. To obtain these matrices, the frequencies of weather conditions that came exactly one day after initial weather conditions were determined through the use of Python code for every initial weather condition. For example, if the initial weather condition was 1 (Clear), then the frequencies of conditions 1-7 that appeared exactly one day after condition 1 were analyzed. The following matrix is the transition matrix of the Vancouver data set obtained from the analysis detailed above. Position p_{ij} represents the probability (to 3 significant figures) of transitioning from state i to state j where $i, j \in \mathbb{Z}^+ \leq 7$.

$$\boldsymbol{P} = \begin{bmatrix} 0.691 & 0.0870 & 0.163 & 0.0373 & 0.0155 & 0.00310 & 0.00310 \\ 0.240 & 0.384 & 0.212 & 0.0760 & 0.0800 & 0.000 & 0.00800 \\ 0.256 & 0.140 & 0.419 & 0.121 & 0.0568 & 0.00250 & 0.00470 \\ 0.221 & 0.176 & 0.313 & 0.290 & 0.000 & 0.000 & 0.000 \\ 0.129 & 0.274 & 0.387 & 0.0320 & 0.178 & 0.000 & 0.000 \\ 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.500 & 0.000 \\ 0.000 & 0.667 & 0.166 & 0.167 & 0.000 & 0.000 & 0.000 \end{bmatrix}$$

This transition matrix may be visualized with the aid of technology (Powell) to obtain the following image in which the $\boldsymbol{i}^{\text{th}}$ row of $\boldsymbol{P}$ is shown by the $\boldsymbol{i}^{\text{th}}$ letter of the English alphabet, and the thickness of the connecting lines is directly proportional to the probability of transitioning states.

Figure 3. A visual representation of the transition matrix P



If we let the probability of the initial state be represented by a 1 x 7 matrix $\boldsymbol{S_0}$, then the weather 1 day after the initial state can be determined by: $\boldsymbol{S_1} = \boldsymbol{S_0 P}$. It is noted that $\boldsymbol{S_1}$ is a 1 x 7 matrix as a k x m matrix multiplied by an m x n matrix results in a matrix of size k x m.

This property can be generalized to find the weather n days after the initial state to obtain Theorem 1.

Theorem 1.

The weather *n* days after the initial state equation is represented by $\boldsymbol{S_n} = \boldsymbol{S_0 P^n}$.

Theorem 1. is proved by induction as follows formatted according to this reference (Mauch):

Base Step:

Let $\boldsymbol{Q(n)}$ be the proposition that $\boldsymbol{S_n} = \boldsymbol{S_0 P^n}\ \forall\, n \in \mathbb{Z}^+$.

Given that $\boldsymbol{S_1} = \boldsymbol{S_0 P}$,

then for n=2, $\boldsymbol{S_2} = \boldsymbol{S_1 P} = (\boldsymbol{S_0 P})\boldsymbol{P} = \boldsymbol{S_0 P^2}$.

Inductive Hypothesis:

Given that $\boldsymbol{S_n} = \boldsymbol{S_0 P^n}$, then assume that $\boldsymbol{Q(n)}$ holds for numbers greater than $\boldsymbol{n}$ such that

$\boldsymbol{S_{n+1}} = \boldsymbol{S_0 P^{n+1}}$

Inductive Step:

If the inductive hypothesis is true then**,**

$\boldsymbol{S_{n+1}} = (\boldsymbol{S_n})\boldsymbol{P} = (\boldsymbol{S_0 P^n})\boldsymbol{P} = \boldsymbol{S_0 P^{n+1}}$

thus $\boldsymbol{Q(n)}$ holds $\forall\, n \in \mathbb{Z}^+$ ■

For the sake of convenience, we can define the following axiom:

Axiom 1.

Let each element in $\boldsymbol{S_n}$ be denoted as $\boldsymbol{p_{1j}}$ where $\boldsymbol{j}$ represents the column of $\boldsymbol{S_n}$. If we let

$\boldsymbol{p_{1r}} = \boldsymbol{max}(\boldsymbol{p_{1j}})\ \forall\, j \in \boldsymbol{S_n}$, then the forecast for a future day $\boldsymbol{n}$ is the state $\boldsymbol{r}$.

From Theorem 1. a weekly forecast may be defined as the set containing a group of 7 daily forecasts. Mathematically, we let a weekly forecast $\boldsymbol{W} = \{\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}, \dots, \boldsymbol{r_7}\}$ where $\boldsymbol{r_d}$ corresponds to $\boldsymbol{S_d}$.

From the transition state $\boldsymbol{P}$ based on Vancouver's forecast data, a weekly forecast may be obtained from the initial condition by setting the probability of the column corresponding with the initial condition to 1 and the probability of all other columns to 0 within the initial vector (one dimensional array) $\boldsymbol{S_0}$.

To calculate the daily forecast given a clear (state=1) initial forecast, let

$$\boldsymbol{S_0} = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\boldsymbol{S_1} = \boldsymbol{S_0 P}$$

$$\boldsymbol{S_1} = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]\begin{bmatrix} 0.691 & 0.0870 & 0.163 & 0.0373 & 0.0155 & 0.00310 & 0.00310 \\ 0.240 & 0.384 & 0.212 & 0.0760 & 0.0800 & 0.000 & 0.00800 \\ 0.256 & 0.140 & 0.419 & 0.121 & 0.0568 & 0.00250 & 0.00470 \\ 0.221 & 0.176 & 0.313 & 0.290 & 0.000 & 0.000 & 0.000 \\ 0.129 & 0.274 & 0.387 & 0.0320 & 0.178 & 0.000 & 0.000 \\ 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.500 & 0.000 \\ 0.000 & 0.667 & 0.166 & 0.167 & 0.000 & 0.000 & 0.000 \end{bmatrix}$$

$$\boldsymbol{S_1} = [0.691 \quad 0.0870 \quad 0.163 \quad 0.0373 \quad 0.0155 \quad 0.00310 \quad 0.00310]$$

(Intermediate calculations obviated as they are considered trivial.)

This result should seem intuitive as it agrees with the definition of a Markov Chain, that any given row $\boldsymbol{i}$ represents the transition probabilities from state $\mathbf{i}$ to state $\boldsymbol{j}$.

By Axiom 1. Since state 1 has the greatest probability of occurring 1 day after the initial condition is state 1, we can forecast the condition for the first day as $F_1 = 1$, or Clear.

Let the frequencies of conditions $\boldsymbol{n}$ days after the initial condition obtained through the analysis of the 365 future daily values be $\boldsymbol{S_{nF}}$. It follows that:

$$\boldsymbol{S_{1F}} = [0.671 \quad 0.123 \quad 0.0644 \quad 0.103 \quad 0.0129 \quad 0.00640 \quad 0.0193]$$

If we take the values in $\boldsymbol{S_1}$ to be the experimental values and the values in $\boldsymbol{S_{1F}}$ to be the accepted values, we can find the percent error for each value as follows:

$$\%\ error = \frac{|experimental\ value - accepted\ value|}{accepted\ value} \text{ x } 100\%$$

From which the set of percent errors may be obtained for any $\boldsymbol{n}^{\text{th}}$ day and set to $\boldsymbol{S_{nE}}$.

In this manner, $\boldsymbol{S_{1E}} = \{2.98\%, 29.3\%, 153\%, 63.8\%, 20.2\%, 51.6\%, 83.9\%\}$

Continuing this procedure,

$$\boldsymbol{S_2} = \boldsymbol{S_1 P}$$

$$\boldsymbol{S_2} = [0.691 \quad 0.0870 \quad 0.163 \quad 0.0373 \quad 0.0155 \quad 0.00310 \quad 0.00310]\begin{bmatrix} 0.691 & 0.0870 & 0.163 & 0.0373 & 0.0155 & 0.00310 & 0.00310 \\ 0.240 & 0.384 & 0.212 & 0.0760 & 0.0800 & 0.000 & 0.00800 \\ 0.256 & 0.140 & 0.419 & 0.121 & 0.0568 & 0.00250 & 0.00470 \\ 0.221 & 0.176 & 0.313 & 0.290 & 0.000 & 0.000 & 0.000 \\ 0.129 & 0.274 & 0.387 & 0.0320 & 0.178 & 0.000 & 0.000 \\ 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.500 & 0.000 \\ 0.000 & 0.667 & 0.166 & 0.167 & 0.000 & 0.000 & 0.000 \end{bmatrix}$$

$$\boldsymbol{S_2} = [0.552 \quad 0.129 \quad 0.218 \quad 0.0638 \quad 0.0297 \quad 0.00400 \quad 0.00350]$$

Sample Calculation for p_{11} follow:

p_{11} = 0.691 x 0.691 + 0.087 x 0.24 + 0.163 x 0.256 + 0.0373 x 0.221 + 0.0155 x 0.129 + 0.0031 x 0.5 + 0.0031 x 0 = 0.5518818 ≈ 0.552

By Axiom 1. Since state 1 has the greatest probability of occurring 2 day after the initial condition is state 1, we can forecast the condition for the second day $\boldsymbol{F_2} = 1$, or Clear.

The data obtained through the analysis of the 365 future daily values follows:

$$\boldsymbol{S_{2F}} = [0.561 \quad 0.142 \quad 0.0903 \quad 0.116 \quad 0.0258 \quad 0.0259 \quad 0.039]$$

$$\boldsymbol{S_{2E}} = \{1.6\%, 9.15\%, 141\%, 45.0\%, 15.1\%, 84.6\%, 91.0\%\}$$

In order to speed up calculations, the property of Theorem 1. Was used in order to write a Mathematica code that could raise a matrix to a certain power and multiply it by the initial vector. (See Appendix A.)

$$\boldsymbol{S_3} = \boldsymbol{S_2 P} = \boldsymbol{S_0 P^3}$$

$$\boldsymbol{S_3} = [0.488 \quad 0.150 \quad 0.241 \quad 0.0768 \quad 0.0364 \quad 0.00410 \quad 0.00370]$$

$\boldsymbol{F_3} = 1$ = Clear

$$\boldsymbol{S_{3F}} = [0.536 \quad 0.129 \quad 0.103 \quad 0.136 \quad 0.0192 \quad 0.0384 \quad 0.0384]$$

$$\boldsymbol{S_{3E}} = \{8.96\%, 16.3\%, 134\%, 43.5\%, 89.6\%, 89.3\%, 90.4\%\}$$

$$\boldsymbol{S_4} = \boldsymbol{S_3 P} = \boldsymbol{S_0 P^4}$$

$$\boldsymbol{S_4} = [0.459 \quad 0.160 \quad 0.251 \quad 0.0823 \quad 0.0397 \quad 0.00420 \quad 0.00380]$$

$\boldsymbol{F_4} = 1$ = Clear

$$\boldsymbol{S_{4F}} = [0.468 \quad 0.169 \quad 0.110 \quad 0.143 \quad 0.0194 \quad 0.0384 \quad 0.0453]$$

$$\boldsymbol{S_{4E}} = \{1.92\%, 5.33\%, 128\%, 42.4\%, 105\%, 90.7\%, 91.6\%\}$$

$$\boldsymbol{S_5} = \boldsymbol{S_4 P} = \boldsymbol{S_0 P^5}$$

$$\boldsymbol{S_{5F}} = [0.445 \quad 0.164 \quad 0.256 \quad 0.0855 \quad 0.0413 \quad 0.00420 \quad 0.00400]$$

$\boldsymbol{F_5} = 1$ = Clear

$$\boldsymbol{S_{5F}} = [0.468 \quad 0.175 \quad 0.117 \quad 0.149 \quad 0.0130 \quad 0.0520 \quad 0.0260]$$

$$\boldsymbol{S_{5E}} = \{4.91\%, 6.29\%, 119\%, 42.6\%, 218\%, 91.9\%, 84.6\%\}$$

$$\boldsymbol{S_6} = \boldsymbol{S_5 P} = \boldsymbol{S_0 P^6}$$

$$\boldsymbol{S_6} = [0.439 \quad 0.167 \quad 0.258 \quad 0.0861 \quad 0.0419 \quad 0.00410 \quad 0.00390]$$

$$\boldsymbol{F_6} = 1 = \text{Clear}$$

$$\boldsymbol{S_{6F}} = [0.500 \quad 0.150 \quad 0.130 \quad 0.149 \quad 0.0130 \quad 0.0520 \quad 0.00600]$$

$$\boldsymbol{S_{6E}} = \{12.2\%, 11.3\%, 98.5\%, 42.2\%, 222\%, 92.1\%, 35.0\%\}$$

$$\boldsymbol{S_7} = \boldsymbol{S_6 P} = \boldsymbol{S_0 P^6}$$

$$\boldsymbol{S_7} = [0.436 \quad 0.168 \quad 0.259 \quad 0.0870 \quad 0.0420 \quad 0.00410 \quad 0.00390]$$

$$\boldsymbol{F_7} = 1 = \text{Clear}$$

$$\boldsymbol{S_{7F}} = [0.500 \quad 0.149 \quad 0.110 \quad 0.162 \quad 0.0130 \quad 0.0520 \quad 0.0140]$$

$$\boldsymbol{S_{7E}} = \{12.8\%, 12.8\%, 135\%, 46.3\%, 223\%, 92.1\%, 72.1\%\}$$

As $\boldsymbol{n}$ increases, it seems that $\boldsymbol{S_n}$ approaches a fixed state, known as a steady state vector. This led me to wonder how I can determine this increase of $\boldsymbol{n}$ asymptotically. Essentially, we are trying to solve for $\lim_{n\to\infty} \boldsymbol{S_n} = \boldsymbol{S_0 P^{\infty}}$. This can be found through the use of the Mathematica code which can take a matrix P and output that matrix to a certain power. It was found that the initial vector $\boldsymbol{S_0}$ has little to no effect on the asymptotic matrix $\lim_{n\to\infty} \boldsymbol{P^n}$ thus $\lim_{n\to\infty} \boldsymbol{S_n} = \boldsymbol{S_0 P^n} \approx \lim_{n\to\infty} \boldsymbol{P^n}$.

$$\lim_{n\to\infty} \boldsymbol{S_n} \approx [0.433 \quad 0.169 \quad 0.260 \quad 0.0880 \quad 0.0420 \quad 0.00399 \quad 0.00401]$$

Essentially this limiting vector shows the probability of obtaining all states when the initial state is any state. This is almost equivalent to the frequency of states of the original data set which is:

$$Freq = [0.433 \quad 0.261 \quad 0.168 \quad 0.0881 \quad 0.0420 \quad 0.00395 \quad 0.00395]$$

From these two vectors, a percentage error may be calculated:

$$\%\ error = \{0.00\%, 35.2\%, 54.8\%, 0.114\%, 0.00\%, 1.01\%, 1.52\%\}$$

The aforementioned process was repeated for all initial states to obtain the following. The observed frequency of the first $\boldsymbol{n}$ days of the weekly forecast was obtained from the future data set by comparing the percentage of days matching the weekly forecast to the total number of days beginning with the initial state. When the probability of a given day was found to be equal for multiple weather conditions, the weather condition with the greatest overall frequency (least state number) was taken.

Table 3. Summarized forecast for all initial states based on a first order Markov Chain.

Initial State	**Weekly forecast**	**Frequency of the First *n* Forecasted Days from Future Data Set**
1 (Clear)	{1, 1, 1, 1, 1, 1, 1}	{0.409, 0.214, 0.149, 0.0779, 0.0584, 0.0390, 0.0325}
2 (Light Rain)	{2, 2, 1, 1, 1, 1, 1}	{0.254, 0.0746, 0.0299, 0.0149, 0.0149, 0.0149, 0.0149}
3 (Cloudy)	{3, 3, 3, 3, 3, 3, 3}	{0.189, 0.0541, 0.00, 0.00, 0.00, 0.00, 0.00}
4 (Foggy)	{3, 3, 3, 1, 1, 1, 3}	{0.127, 0.0282, 0.0141, 0.0141, 0.0141, 0.0141, 0.00}
5 (Heavy Rain)	{3, 3, 1, 3, 1, 1, 1}	{0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00}
6 (Smoky)	{1, 1, 1, 1, 2, 2, 2}	{0.0769, 0.0769, 0.0769, 0.00, 0.00, 0.00, 0.00}
7 (Snowy)	{2, 2, 2, 1, 1, 2, 1}	{0.0833, 0.0833, 0.00, 0.00, 0.00, 0.00, 0.00}

To improve accuracy of a weekly prediction, a second order Markov Chain was created through an analysis of the original data, by taking the probabilities of transitioning to a certain state based on a combination of two current states, which resulted in a 49 x 7 transition matrix. To find the probability of weather on subsequent days, the initial state is used to determine a first order probability vector, and then from there the state with the highest probability is appended to the most recent predicted state. The current state is now comprised of two weather conditions, and the process is continued by using the second order Markov Chain transition matrix.

Table 4. Summarized forecast for all initial states based on a second order Markov Chain.

Initial State	**Weekly forecast**	**Frequency of the First *n* Forecasted Days from Future Data Set**
1 (Clear)	{1, 1, 1, 1, 1, 1, 1}	{0.409, 0.214, 0.149, 0.0779, 0.0584, 0.0390, 0.0325}
2 (Light Rain)	{2, 2, 2, 1, 1, 1, 1}	{0.254, 0.0746, 0.0149, 0.00, 0.00, 0.00, 0.0149}
3 (Cloudy)	{3, 3, 3, 3, 3, 3, 3}	{0.189, 0.0541, 0.00, 0.00, 0.00 0.00, 0.00}
4 (Foggy)	{3, 3, 3, 3, 3, 3, 3}	{0.127, 0.0282, 0.0141, 0.00, 0.00, 0.00, 0.00}
5 (Heavy Rain)	{3, 3, 3, 3, 3, 3, 3}	{0.00, 0.00, 0.00, 0.00, 0.00 0.00, 0.00}
6 (Smoky)	{1, 1, 1, 1, 1, 1, 1}	{0.0769, 0.0769, 0.0769, 0.0769, 0.0769, 0.0769, 0.0769}
7 (Snowy)	{2, 3, 1, 1, 1, 1, 1}	{0.0833, 0.00, 0.00, 0.00 0.00, 0.00, 0.00}

The second order Markov Chain was supposed to increase the accuracy of the weekly forecasts, but as shown by Table 3. and Table 4., there was little to no significant improvement in the frequency of observed predictions in the future data set. This could be due to the limited size of the future sample data set which contains only 365 data points as opposed to the 1,487 data points used to create the weekly forecast, or it could be due to an overly simplistic model of how to choose the next day's predicted weather. Nevertheless, it was shown that the Markov Chain model is slightly reliable for predicting weather conditions one day into the future, and continually less reliable for prediction weather conditions $\boldsymbol{n}$ days into the future as $\boldsymbol{n}$ increases.

If we define a "good prediction" to be one that has a probability greater than 0.5 of occurring, then we note that 0 out of 49 predictions are good predictions.

If we allow a "suitable prediction" to be defined as a prediction that has a higher probability of occurring than a randomly chosen weather condition (1/7≈0.143), but a lesser probability of occurring than 0.5, then we note that a total of 5 out of 49 predictions, or a mere 10.2% are "suitable predictions". If we define a "poor prediction" as a prediction that has a lesser probability of occurring than a randomly chosen weather condition (1/7≈0.143), then we note that the remaining 44 out of 49 predictions, or 89.8% of predictions are "poor predictions". This information can be summarized in the following table:

Table 5. Classification of Vancouver predictions and their corresponding frequencies

Prediction class	**Frequency (%)**
Good	0.00%
Suitable	10.2%
Poor	89.8%

This entire procedure was repeated with the Los Angeles data set obtaining the following information:

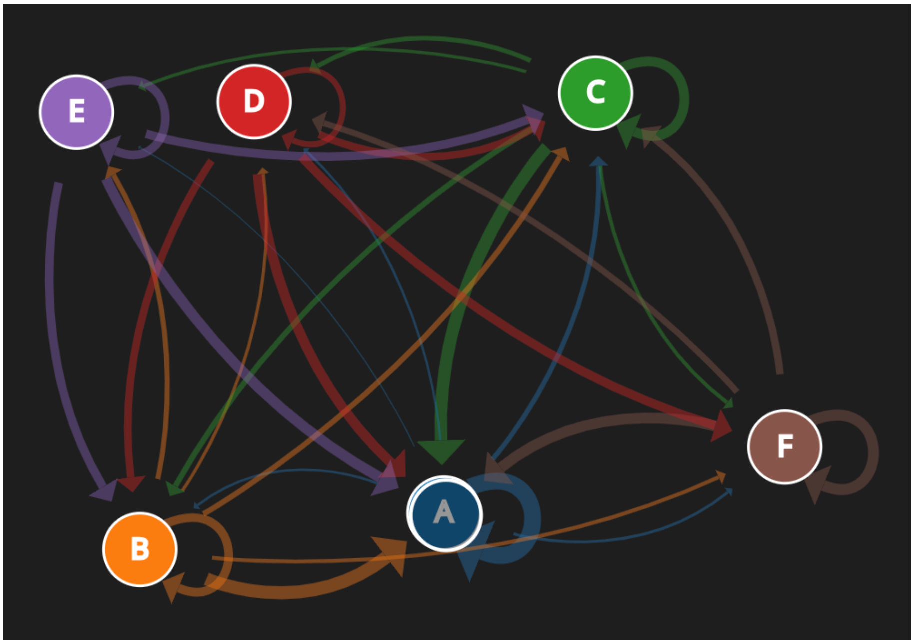
$$P = \begin{bmatrix} 0.861 & 0.0187 & 0.0750 & 0.0169 & 0.00340 & 0.0250 & 0.00 \\ 0.516 & 0.234 & 0.0937 & 0.0313 & 0.0781 & 0.0469 & 0.00 \\ 0.466 & 0.0994 & 0.292 & 0.0683 & 0.0308 & 0.0435 & 0.00 \\ 0.304 & 0.174 & 0.174 & 0.109 & 0.00 & 0.239 & 0.00 \\ 0.334 & 0.222 & 0.222 & 0.00 & 0.222 & 0.00 & 0.00 \\ 0.351 & 0.00 & 0.169 & 0.117 & 0.00 & 0.363 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

The matrix shown above does not meet the definition of a stochastic matrix as not every row adds to 1. Thus, the matrix must be modified to obtain the following.

$$P = \begin{bmatrix} 0.861 & 0.0187 & 0.0750 & 0.0169 & 0.00340 & 0.0250 \\ 0.516 & 0.234 & 0.0937 & 0.0313 & 0.0781 & 0.0469 \\ 0.466 & 0.0994 & 0.292 & 0.0683 & 0.0308 & 0.0435 \\ 0.304 & 0.174 & 0.174 & 0.109 & 0.00 & 0.239 \\ 0.334 & 0.222 & 0.222 & 0.00 & 0.222 & 0.363 \\ 0.351 & 0.00 & 0.169 & 0.117 & 0.363 & 0.00 \end{bmatrix}$$

This transition matrix may be visualized with the aid of technology (Powell) to obtain the following image in which the $\boldsymbol{i}^{\text{th}}$ row of $\boldsymbol{P}$ is shown by the $\boldsymbol{i}^{\text{th}}$ letter of the English alphabet, and the thickness of the connecting lines is directly proportional to the probability of transitioning states.

Figure 4. A visual representation of the transition matrix P



The steady state vector or $\lim_{n\to\infty} \boldsymbol{S_n}$ may be found as follows:

$$\lim_{n\to\infty} \boldsymbol{S_n} = \boldsymbol{S_0}\boldsymbol{P}^{\infty} \approx \boldsymbol{P}^{\infty}$$

$$\lim_{n\to\infty} \boldsymbol{S_n} \approx [0.753 \quad 0.0430 \quad 0.109 \quad 0.0310 \quad 0.0120 \quad 0.0520]$$

This can be compared to the frequency of states of the original data set as was done for the Vancouver data set. The following vector contains the frequencies of the original Los Angeles Data set.

$$Freq = [0.753 \quad 0.109 \quad 0.0518 \quad 0.0432 \quad 0.0309 \quad 0.0121]$$

From these two vectors, a percentage error may be calculated:

$$\%\ error = \{0.00\%, 60.6\%, 110.\%, 28.2\%, 61.2\%, 330.\%\}$$

For the Los Angeles data set, the first and second order Markov Chains both produced the same weekly forecasts for all initial states. They are jointly summarized in the following table:

Table 6. Summarized forecast for all initial states based on a first and second order Markov Chains.

Initial State	**Weekly forecast**	**Frequency of First n days of Forecast from Future Data Set**
1 (Clear)	{1, 1, 1, 1, 1, 1, 1}	{0.437, 0.241, 0.161, 0.111, 0.0704, 0.0452, 0.0302}
2 (Light Rain)	{1, 1, 1, 1, 1, 1, 1}	{0.571, 0.286, 0.286, 0.143, 0.143, 0.143, 0.143}
3 (Cloudy)	{1, 1, 1, 1, 1, 1, 1}	{0.522, 0.348, 0.217, 0.174, 0.130, 0.0870, 0.435}
4 (Foggy)	{1, 1, 1, 1, 1, 1, 1}	{0.300, 0.214, 0.171, 0.157, 0.129, 0.0857, 0.0571}
5 (Heavy Rain)	{1, 1, 1, 1, 1, 1, 1}	{0.500, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00}
6 (Smoky)	{6, 1, 1, 1, 1, 1, 1}	{0.281, 0.228, 0.175, 0.157, 0.105, 0.0702, 0.0351}

Both the first and second order Markov Chains for the Los Angeles data set gave a much higher level of accuracy than that exhibited in the Vancouver data set (Table 3. and Table 4.) as demonstrated by the frequency of observed predictions in the future data set. As stated in my introduction, this increased accuracy could perhaps be due to the decrease in volatility of Los Angeles' climate. For the Los Angeles data set it was shown that the Markov Chain model is quite reliable for predicting weather conditions one day into the future, and continually less reliable for prediction weather conditions $\boldsymbol{n}$ days into the future as $\boldsymbol{n}$ increases. If, we take the same definitions for the qualification of predictions as defined in the Vancouver data analysis, then 3 out of 36 predictions are good, 13 out of 36 predictions are suitable, and the remaining 20 out of 36 predictions are poor.

This information may be summarized in the following table as follows:

Table 7. Classification of Los Angeles predictions and their corresponding frequencies

Prediction class	**Frequency (% to 3 s.f. truncated to add to 100%)**
Good	8.30%
Suitable	36.1%
Poor	55.6%

Conclusion

Through the use of first and second order Markov Chains, daily and weekly weather condition forecasts were obtained from 1,487 original data values. This was achieved through the creation of transition matrices and matrix multiplication. After obtaining the corresponding matrices, the state with the highest transition probability was taken as the future day's forecast. From the repetition of this process, a weekly forecast was produced for each data set (Vancouver and Los Angeles). Daily and weekly forecasts were then compared to a set of 365 daily condition values and the frequency with which these forecasts appeared was noted. These forecasts were categorized into "good", "suitable" and "poor" predictions based on their frequencies. Vancouver's data set resulted in a fewer frequency of "good" and "suitable" conditions when compared to the Los Angeles data set. As suggested by my introduction, this could be a result of a more volatile climate.

Implications:

The results of this investigation suggest that weather prediction should consider many more factors than simply historically observed weather conditions. Although there are various limitations of my research, the results could put into question the reliability of Markov Chain algorithms when used on weather prediction. Although these results likely have little impact on the day to day layman, experts should strongly consider expanding their methodology when utilizing Markov Chains in algorithmic predictions.

Pitfalls:

This research is limited in city scope to solely two cities on similar geographical longitudes and on the same hemisphere. Thus, weather conditions do not show a wide range of possible options. Furthermore, the data sets used are from 2012-2017, and although this gives a suitable data set size, it may be affected by abnormal yearly conditions and a larger data set would provide more information for a more accurate model. This data set comes from a legitimate weather website indirectly, and directly from an open source platform thus the data could have been altered in the process. Throughout the use of this investigation, first and second order Markov Chains were studied, but Markov Chains with order $n \in \mathbb{Z}^+ > 2$ could have altered the model to yield more accurate results. A key factor in the outcome of this study is the categorization of weather conditions. This arbitrary selection which I performed could easily be changed and these changes could affect the outcome of the forecasts.

Future Work:

The focus of this study could be expanded in breadth to better represent a global perspective of weather. This aim can be achieved by analyzing a larger quantity of cities while diversifying the geospatial location of a given city. This would subsequently allow for more diverse data sets which in turn could yield more results. Furthermore, expanding the size of the data sets to include at least 50 years of historical data could produce more accurate results. As weather depends on a multitude of different factors, the incorporation of a greater number of observations in this model so as to indirectly categorize weather conditions would likely provide a more accurate prediction. By combining this model with alternate Bayesian probabilistic models such as Bayes' Theorem or Monte Carlo Based Ensemble Forecasting, more accurate results may be obtained.

Works Cited

Fewster, Rachel. “Markov Chains”. STATS 325. The University of Auckland, New Zealand. 20 June 2014.

Hilgers, Von, et al. "The five greatest applications of Markov chains." Proceedings of the Markov Anniversary Meeting, Boston Press, Boston, MA. 2006.

How Do Radars Work? | Earth Observing Laboratory. https://www.eol.ucar.edu/content/how-do-radars-work. Accessed 19 Nov. 2018.

Mauch, T. *Proof by Induction*. 26 Sept. 2003, http://alumni.cs.ucr.edu/~tmauch/old_web/cs141/cs141_pages/proof_by_induction.html.

Powell, V. *Markov Chains Explained Visually*. Explained Visually, 19 Nov. 2018, http://setosa.io/ev/markov-chains/.

Rotondi, M. A. *To Ski or Not to Ski: Estimating Transition Matrices to Predict Tomorrow’s Snowfall Using Real Data*. Vol. 18, no. 3, Journal of Statistics Education, Jan. 2010.

Schoof, J. T., and S. C. Pryor. *On the Proper Order of Markov Chain Model for Daily Precipitation Occurrence in the Contiguous United States*. Vol. 47, no. 9, J. Appl. Meteor. Climatol., Jan. 2008, pp. 2477-86.

Ward, D. *Weather Forecasting*. Copyright 2003, UCAR/NCAR, https://www.eo.ucar.edu/basics/wx_4.html. Accessed 19 Nov. 2018.

Weisstein, Eric W. "Stochastic Matrix." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/StochasticMatrix.html

Appendix A.

Code 1. Used to obtain S_n for the Vancouver data set.

```
initial = {1, 0, 0, 0, 0, 0, 0};
transition = {{0.691, 0.087, 0.163, 0.0373, 0.0155, 0.0031,
   0.0031}, {0.24, 0.384, 0.212, 0.076, 0.08, 0, 0.008}, {0.256,
   0.14, 0.419, 0.121, 0.0568, 0.0025, 0.0047}, {0.221, 0.176, 0.313,
    0.29, 0, 0, 0}, {0.129, 0.274, 0.387, 0.032, 0.178, 0, 0}, {0.5,
   0, 0, 0, 0, 0.5, 0}, {0, 0.667, 0.166, 0.167, 0, 0, 0}};
Do[Print[initial.MatrixPower[transition, n]], {n, 7}]
Print[initial.MatrixPower[transition, 1000000000]]
```

Code 2. Used to obtain S_n for the Los Angeles data set.

```
initial = {1, 0, 0, 0, 0, 0, 0};
transition = {{0.861, 0.0187, 0.075, 0.0169, 0.0034, 0.025}, {0.516,
   0.234, 0.0937, 0.0313, 0.0781, 0.0469}, {0.466, 0.0994, 0.292,
   0.0683, 0.0308, 0.0435}, {0.304, 0.174, 0.174, 0.109, 0,
   0.239}, {0.334, 0.222, 0.222, 0, 0.222, 0}, {0.351, 0, 0.169,
   0.117, 0, 0.363}};
Do[Print[initial.MatrixPower[transition, n]], {n, 7}]
Print[initial.MatrixPower[transition, 1000000000]]
```

Appendix B.

Code 3. Used to parse the initial data and obtain frequency for first and second order Markov Chains.

```
import csv
from statistics import mode
import math
def round_sigfigs(num, sig_figs):
 #http://code.activestate.com/recipes/57rows114-round-number-to-specified-number-of-significant-di/
  if num != 0:
    return round(num, -int(math.floor(math.log10(abs(num))) - (sig_figs - 1)))
  else:
    return 0  # Can't take the log of 0
hours=[]
with open('la.csv', 'rt', newline='', encoding='utf8') as csvfile:
 spamreader = csv.reader(csvfile, delimiter=',', quotechar='|')
 for row in spamreader:
   hours.append(row[0])
del hours[0]
day_avg=[]
day=[]
c=0
while c<(len(hours)):
 #print(hours[c])
 day.append(hours[c])
 if len(day)%24==0:
```

```
    try:
      day_avg.append(mode(day))
    except:
      day_avg.append(day[11])
    day=[]
  c+=1
pos_days={'null':0,'sky is clear':1, 'few clouds': 1, 'light rain':2, 'light intensity shower rain':2, 'proximity shower rain':2, 'light intensity drizzle':2, 'light intensity drizzle rain':2, 'overcast clouds':3, 'scattered clouds':3, 'broken clouds':3, 'mist':4, 'fog':4, 'moderate rain':5, 'heavy intensity rain':5, 'shower rain':5, 'very heavy rain':5, 'haze':6, 'smoke':6,'dust':6, 'light shower snow':7, 'light snow':7, 'heavy snow':7, 'snow':7}
d=0
while d<len(day_avg):
  day_avg[d]=(pos_days[day_avg[d]])
  d+=1
#print (len(day_avg))
last_year=day_avg[-365:]
del day_avg[-365:]
with open('days.txt', 'w') as f:
    for each in day_avg:
        f.write("%s\n" % each)
s_ly=''
for i in last_year:
    s_ly+=str(i)
print(s_ly)
'''
prob1=float(s_ly.count('11111111'))/(s_ly[:-7].count('1'))
prob2=float(s_ly.count('22211111'))/(s_ly[:-7].count('2'))
prob3=float(s_ly.count('33333333'))/(s_ly[:-7].count('3'))
prob4=float(s_ly.count('43331113'))/(s_ly[:-7].count('4'))
prob5=float(s_ly.count('53313111'))/(s_ly[:-7].count('5'))
prob6=float(s_ly.count('61111222'))/(s_ly[:-7].count('6'))
prob7=float(s_ly.count('72221121'))/(s_ly[:-7].count('7'))
print (prob3)
'''
#future or nah
future=0
probs=[]
#actual rows
rows=6
#modified for loops
rows+=1
for i in range(1,8):
    #n days after initial
    days=i
    print ('\n\nN=%s'%days)
    if future==0:
        for i in range(1,rows):
          after=[]
          e=0
```

```
        while e<len(day_avg):
          if day_avg[e]==i:
            try:
              after.append(day_avg[e+days])
            except: pass
          e+=1
        for p in range(1,rows):
          try:
              prob=round_sigfigs(float(after.count(p))/len(after),4)
              print (prob)
              probs.append(prob)
          except:
              print (0)
              probs.append(0)
        print()
        probs.append('\n')
    else:
        for i in range(1,rows):
          after=[]
          e=0
          while e<len(last_year):
            if last_year[e]==i:
              try:
                after.append(last_year[e+days])
              except: pass
            e+=1
          for p in range(1,rows):
            try:
                prob=round_sigfigs(float(after.count(p))/len(after),4)
                print (prob)
                probs.append(prob)
            except:
                print (0)
                probs.append(0)
          print()
          probs.append('\n')
with open('probs.txt', 'w') as f:
    for each in probs:
        f.write("%s\n" % each)
```