final

November 14, 2025

```
packages imported
```

1 Problem Statement

Construct a mathematical model to estimate the **total annual income** of Loafe Café at both the Alumni Centre and Sauder locations as a function of daily temperature, daily precipitation, daily humidity and frequency of customers.

2 Variables and Parameters

Symbol	Description	Type	Dimension	Units
d	day index within a (non-leap) year, $d \in \{1, \dots, 365\}$	independent variable	1	—
T_d, P_d, H_d	temperature, precipitation, humidity on day d	independent variable (weather input)	$T, L, 1$	°C,mm,%
C_d	expected total customers on day d (both locations)	dependent variable (model output)	1	customers/day
s	average spend per customer	parameter	1	CAD/customer
R	total annual income	derived aggregate output	1	CAD/year
T_0, P_0, H_0	reference temperature, precipitation, humidity level	parameter (reference level)	$T, L, 1$	°C,mm,%
C_0	baseline customers at reference weather (T_0, P_0, H_0)	parameter	1	customers/day

Symbol	Description	Type	Dimension	Units
a	rate of change of customers w.r.t temperature (squared)	parameter (sensitivity)	T^{-2}	customers / (°C)^2
b	rate of change of customers w.r.t precipitation	parameter (sensitivity)	L^{-1}	customers / mm
c	rate of change of customers w.r.t humidity (squared)	parameter (sensitivity)	1	customers / (%)^2

3 Assumptions and Constraints:

- The number of customers is non-negative and proportional to temperature squared, precipitation, and humidity squared.
- Annual revenue scales linearly with the number of customers per year.
- Menu prices and the amount of money individual customers spend remain constant year-round.
- On average, customers at the Alumni Centre spend the same amount of money as customers at the Sauder building. This allows us to combine the customer data of the two.
- To evaluate the potential correlation of weather to customer visits, we do not account for external factors such as the school year schedule.
- For usage of this model in current and future years, we assume that weather trends each month remain the same in 2025 and onwards.
- We assume weather trends are different each month, so we will construct separate functions for each to estimate customer visits over the whole year more accurately.
- A non-leap year will be used to model annual revenue.
- Leap years are included in the data we use to build the model, but since we sample from each month's pool of data without consideration for what specific day it is, we can still take 365 samples to make a total customer estimate for a non-leap year.
- Reference values are the mean values taken from the entire dataset, as these are the points from which deviating weather measurements would cause a difference in the expected number of customers.

4 Building the Solution:

We estimate total annual revenue of Loafe by: 1) Loading the provided datasets and doing some light exploratory data analysis.
2) Constructing probability density functions for temperature, precipitation, and humidity (T, P, H) for each month. We achieve this with kernel density estimation over the data set using a gaussian kernel.
3) Estimate the parameters of the daily customers equation from data. - Estimate a, b, c, c_0 using the January/July customer dataset. - Estimate t_0, p_0, h_0 using the entire weather dataset. 4) Simulate a year of revenue by sampling weather conditions for each day (ex. if a day is in March

then we sample (T_d, P_d, H_d) off of each respective March PDF). We then use those weather samples (T_d, P_d, H_d) as inputs to the daily customer function and then multiply the daily customers by an average purchase amount. We do this over all 365 days of the year. 5) Finally, we perform uncertainty analysis via Monte Carlo methods on our yearly revenue calculations to get a range of plausible values.

4.1 Data Processing

In the code cell below we combine both Loafe locations customer data into a January total and a July total respectively.

4.2 Build Monthly PDFs

We split weather data by month and fit Gaussian KDEs for temperature, precipitation, and humidity. We use KDEs to build these monthly weather PDFs (for temperature, precipitation, humidity) because KDEs give a smooth probability density function without having to make any assumptions about what type of distribution the data follows.

Figure 1 overlays **January and July** PDFs for temperature so we can visually confirm that the winter/summer weather patterns that will drive the customer model are different.

[ ]:

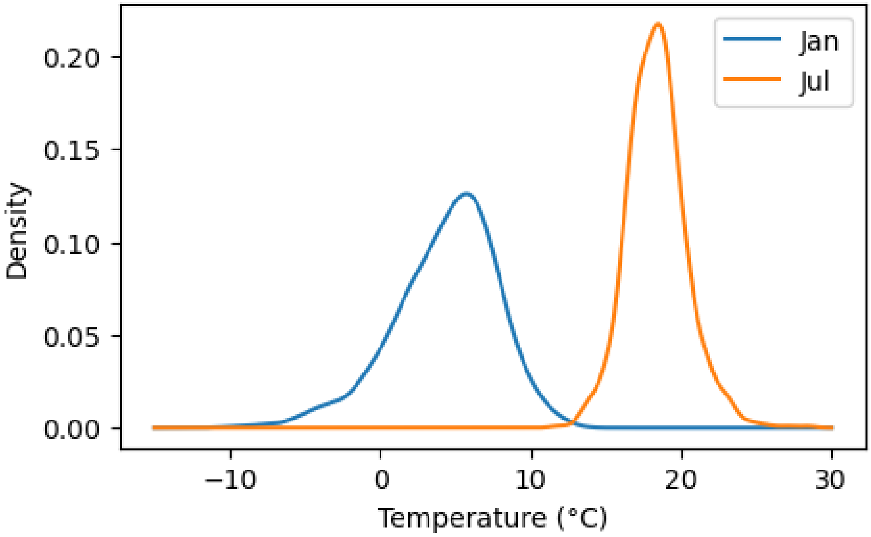
```
# ========= Monthly splits & KDEs =========
# split weather into 12 months
listOfMonthData = []
for i in range(1, 13):
    listOfMonthData.append(weather.loc[weather["Month"] == i, ["temperature",␣
 ↪"precipitation", "humidity"]].dropna())

class Month:
    def __init__(self, index, month_pdf_T, month_pdf_P, month_pdf_H):
        self.index = index # 1 indexed from January
        self.month_pdf_T, self.month_pdf_P, self.month_pdf_H = month_pdf_T,␣
 ↪month_pdf_P, month_pdf_H

# build KDEs per month
listOfMonthPDFs = []
for i in range(12):
    month = listOfMonthData[i]
    pdf_T, pdf_P, pdf_H = stats.gaussian_kde(month["temperature"]), stats.
 ↪gaussian_kde(month["precipitation"]), stats.gaussian_kde(month["humidity"])
    listOfMonthPDFs.append(Month(str(i+1), pdf_T, pdf_P, pdf_H))

# spaces to evaluate estimated PDFs over
gT, gP, gH = np.linspace(-15, 30, 400), np.linspace(0, 60, 400), np.
 ↪linspace(20, 100, 400)
```

Figure 1 — Temperature PDFs (Jan and Jul)



Specify the daily customer amount equation as

$$C_d(T_d, P_d, H_d) = \max\{\, C_0 + a\,(T_d - T_0)^2 - b\,(P_d - P_0) - c\,(H_d - H_0)^2,\ 0\,\}$$

- We chose T_0, P_0, and H_0 to be the mean value for each of the respective weather measurements over the entire dataset. The justification for this is that from the structure of the equation if the inputted values (T_d, P_d, H_d) are exactly equal to each respective parameter (T_0, P_0, and H_0) the amount of customers for that day will just be equal to C_0. Therefore the reference values should represent what typical weather looks like, and the mean value is the statistic that does that.

- We get **Jan/Jul mean weather** from the KDEs (numerical mean via integration on a fine grid).

- We estimate a, b, c using a finite-difference idea based on the **change in average daily customers** from January to July.

- We chose C_0 to be the mean between the average daily number of customers for January and the average daily number of customers for July.

4.2.1 Estimating the coefficients:

We know the customer response model as defined above:

$$C_d(T_d, P_d, H_d) = C_0 + a(T_d - T_0)^2 - b(P_d - P_0) - c(H_d - H_0)^2$$

where C_0 is the baseline value at reference weather (T_0, P_0, H_0), and a, b, c represent sensitivities to temperature, precipitation, and humidity.

Let $\Delta C = C_{Jul} - C_{Jan}$ be the observed change in average daily customers between January and July, let $(T_{Jan}, P_{Jan}, H_{Jan})$ and $(T_{Jul}, P_{Jul}, H_{Jul})$ denote the corresponding mean weather conditions.

Coefficient a: Assume, for the purpose of isolating the effect of temperature, that precipitation and humidity remain at their reference levels while temperature moves from its January mean to its July mean. Then the change predicted by the model is

$$\Delta C \approx a[(T_{Jul} - T_0)^2 - (T_{Jan} - T_0)^2] \implies a = \frac{\Delta C}{(T_{Jul} - T_0)^2 - (T_{Jan} - T_0)^2}$$

Similarly for Coefficient b and c, isolating precipitation and humidity respectively gives us

$$b = \frac{\Delta C}{(P_{Jul} - P_0) - (P_{Jan} - P_0)}, \quad c = \frac{\Delta C}{(H_{Jul} - H_0)^2 - (H_{Jan} - H_0)^2}$$

Justification: Because we only have customer data for two months, we cannot reliably fit all three weather effects at the same time. Instead, we use a simple one-variable-at-a-time approach: we treat the change in customers from January to July as if it were caused by the change in just one weather variable (temperature, precipitation, or humidity), keeping the others fixed at their typical values. Solving the model under these simplified comparisons gives the formulas for a, b, and c. This produces reasonable, interpretable estimates given the limited data.

```
Reference mean averages: T0=10.60°C, P0=3.19mm, H0=78.10%
Jan vs Jul daily customers: 350.4 → 175.4
a=-8.121, b=36.557, c=-14.826, C0=262.9
```

4.3 Yearly Simulation

We simulate a **typical** (non-leap) year by: - mapping day $d \in \{1..365\}$ to its **month**,
- drawing a single (T, P, H) sample from that month's PDFs,
- applying $C_d(T_d, P_d, H_d)$, and
- multiplying by a constant $\$\ s$ **spend per customer**.

Note: At the Sauder Loafe location, the average price for a food item is \$13.73 and the average price for a drink is \$5.74. If we assume that 25% of customers buy only food, 50% buy only a drink, and 25% both food and a drink, we get that the average customer would spend

$$s = 0.25 * (\$13.73) + 0.5 * (\$5.74) + 0.25 * (\$13.73 + \$5.74) = \$11.17.$$

This produces a **sample** R of annual revenue that follows the equation

$$R = \sum_{d=1}^{365} C(T_d, P_d, H_d) \times s.$$

[15]:

```python
# Used for determining which month to sample weather data from
days_per_month = np.array([31,28,31,30,31,30,31,31,30,31,30,31])
month_edges = np.cumsum(days_per_month)

# Used for determining which month to sample from
```

```
def dayToMonth(day):
    return int(np.searchsorted(month_edges, day))

# Daily revenue function definition
def dailyRev(month, spend_per_customer):
    T_hat = listOfMonthPDFs[month].month_pdf_T.resample(1).ravel()[0]
    P_hat = listOfMonthPDFs[month].month_pdf_P.resample(1).ravel()[0]
    H_hat = listOfMonthPDFs[month].month_pdf_H.resample(1).ravel()[0]
    customersDay = cFunc(T_hat, P_hat, H_hat)
    return customersDay * spend_per_customer

# Yearly revenue function definition
def yearlyRev(spend_per_customer):
    total = 0.0
    for i in range(1, 365+1):
        m = dayToMonth(i)
        total += dailyRev(m, spend_per_customer=spend_per_customer)
    return total

s=11.17
yearRev = yearlyRev(s)
print(f"Sample revenue/year (s=${s:.2f}): ${yearRev:,.0f}")
```

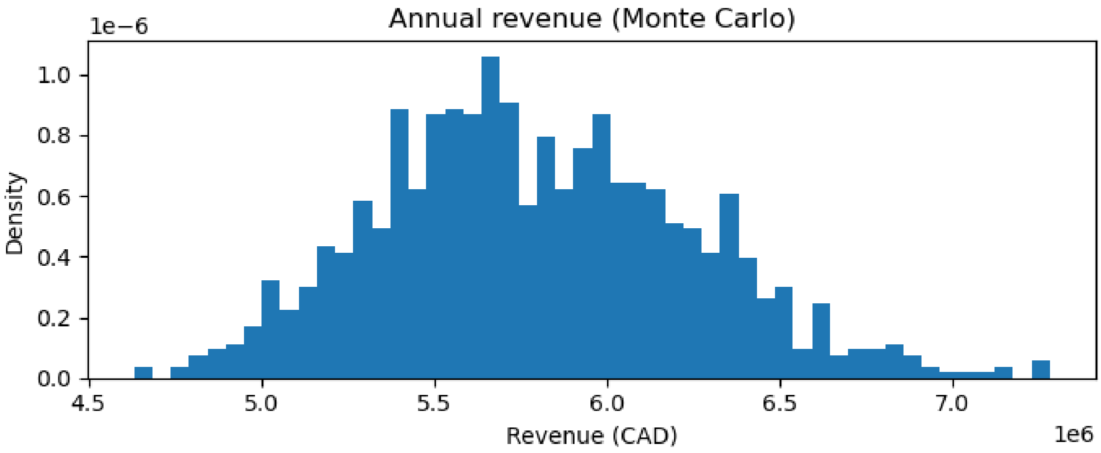
```
Sample revenue/year (s=$11.17): $5,365,225
```

4.4 Monte Carlo Estimation

Below is an unperturbed Monte Carlo simulation of our yearly revunue:

[6]:

```
N = 1000
yearlyRevenues = [yearlyRev(s) for _ in range(N)] # list unpacking python␣
  ↪notation
```



```
Mean annual revenue: $5,792,563 |Median annual revenue: $5,749,088 | 90% CI:
[$5,076,921, $6,558,142]
```

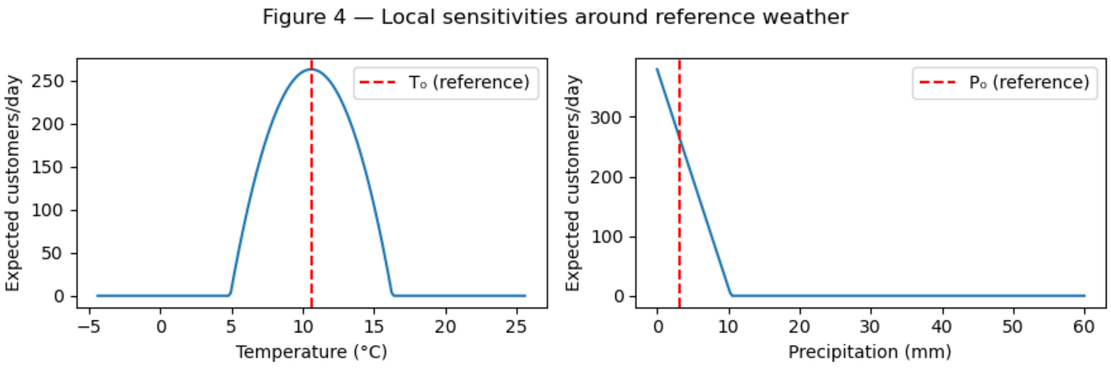
Below is the perturbed Monte Carlo simulation:

```
Uncertainty / Error Scales Used in Monte Carlo
--------------------------------------------------
T0 error ( _T0): 5.840  °C
P0 error ( _P0): 6.473  mm
H0 error ( _H0): 9.464  %
C0 error ( _C0): 435.738  customers/day

Corresponding variances ( ²):
--------------------------------------------------
Var(T0): 34.105
Var(P0): 41.896
Var(H0): 89.572
Var(C0): 189867.972
```

Figure 4: Simple one-dimensional sensitivities of expected customers against each weather variable around the reference levels (helps interpret a, b, c).
We also print the median and 90% interval of revenue.

```
MC median: $10,696,469 | 90% CI: [$9,602,895, $11,795,932]
```



5 Analysis and Assessment:

From our Monte Carlo simulation of total annual revenue, we have that the mean annual revenue for Loafe across all locations is $\$5,792,563$ per year with a median value of $\$5,749,088$ and a 90% confidence interval of $[\$5,076,921, \$6,558,142]$. Since the number of customers each day is dependent on random weather variables, our use of Monte Carlo analysis is justified in order to construct a range of plausible estimates. These values for the yearly revenue and confidence interval range seem very reasonable for Loafe's total revenue. Doing a quick sanity check using the average value of revenue this would imply that Loafe makes around $\$15,870$ in revenue daily which further implies that they sell around 1356 goods at a price of 11.70 daily. These values make sense in the context of the real world because Loafe operates two locations in high foot traffic areas like Sauder and the Alumni Center, and a daily revenue of $\$15,870$ is very much in line with what a small cafe shop would make.

From the sensitivity plots, we see that for variables that get squared in the daily customers equation like temperature or humidity, the number of customers changes in the same direction around the reference value as weather values deviate from it. We see that for precipitation, which does not get squared, the number of customers changes linearly in opposite directions around the reference value. The equation doesn't take into consideration how certain combinations of weather values might affect the number of customers, as these weather conditions often correlate together and don't act independently as the model might suggest.

As for the assumptions we made, the model doesn't account for the school year schedule and seasonal patterns. The number of people on campus is usually lower during the summer months, and directly tying revenue to the number of customers means that it would be greatly influenced by the number of people around in the first place rather than weather. With access to more accurate data on customer spending habits, we would also be able to create a better estimate for the average amount of revenue gained from each customer instead of assuming customer spending behaviour. This value could even be non constant, potentially varying depending on the time of year if spending trends change throughout different seasons.

The model could be improved if the customer data includes the number of customers per day or if the monthly data spanned multiple years. These changes would allow us to determine the distribution of the number of daily customers, which could let us find the distribution of proportionality parameters a, b, and c.

We isolated the effects of the parameters' uncertainty on total revenue. The standard deviation of total revenue is $\$217,494.37, \$221,409.57$, and $\$222,519.60$ when temperature error, precipitation error, and humidity error are included, respectively, with all other errors set to zero. This implies that the model is not particularly sensitive to each parameter. The cafe may be interested in measuring humidity more precisely than other parameters, as it can mean the difference of thousands of dollars.