# Applied Probability I: Group Assignment

STU22004-202526

Team Members:
Denys Keleshohlu
Adam Zielinski
Pavit Gogia
Aran Banner

December 2025

## 1 Introduction

Dublin bikes is a public transportation network comprising 116 stations and over 1,500 bicycles across Dublin city centre.
Using Python simulations and various mathematical models, we wanted to research the following research questions:

**RQ 1**: What is the probability distribution of bike availability at Dublin Bikes stations?

**RQ 2**: Is station availability independent of temporal factors?
Specifically, P(Empty — Hour) and P(Empty — Hour & Day)

**RQ 3**: Is the availability of bikes correlated with the availability of nearby stations?

We believe these questions are interesting as they require precise definitions of random variables, involve both marginal and conditional probability estimation and testing using chi-squared tests.
Beyond that, these questions have direct practical implications for commuters and system operators.
Lastly, using Python enables us to demonstrate our findings in an intuitive and visual way.

### Note on Figures

All figures in this report are hosted externally to reduce file size. Clickable links are provided throughout the document (indicated by ↗). If viewing in an environment where links are not accessible, the figures can be downloaded from here: https://github.com/CoffeeMan1ac/dublinbikesanalysis

## 2 Key Data Assumptions

- Each station–timestamp observation is treated as an independent draw from the underlying distribution of availability.
- Station capacities are fixed and correctly reported.
- Observations with faulty data, closed stations, or unrealistic capacities are removed.
- Probability estimates approximate the true long-run frequencies due to the extremely large sample size.

## 3 Our Approach

The analysis is conducted in two complementary parts:

1. **Mathematical formulation**: all random variables, events, and probability relationships are formally defined.
2. **Python implementation**: these definitions are applied to real Dublin Bikes data to compute probabilities to estimate or empirically verify figures (i.e PMF & Convergence)

## 4 Question 1

### 4.1 Objective

To estimate the probability distribution of bike availability at Dublin Bikes stations, quantify key event probabilities (Empty, Full, Usable, Low, High), and demonstrate the Law of Large Numbers by showing empirical convergence of these probabilities using over 32 million observations.

### 4.2 Mathematical

#### 4.2.1 System & Events

For each station $s$ and timestamp $t$, the random variable

$X_{s,t} \in \{0, 1, 2, \ldots, C_s\}$

represents the number of bikes available, where $C_s$ is the station capacity. Events:

$Empty...E = \mathbf{1}\{X = 0\}$
$Full...F = \mathbf{1}\{X = C_s\}$
$Usable...U = \mathbf{1}\{X \geq 1\}$
$Low...L = \mathbf{1}\{X/C_s < 0.2\}$
$High...H = \mathbf{1}\{X/C_s > 0.8\}$

## 4.3 Python Implementation

### 4.3.1 Monte Carlo Simulations

See Figure 1: LLN Convergence for Empty Probability ↗

### 4.3.2 P(Empty) by Station

See Figure 2: Empty Probability by Station ↗

### 4.3.3 PMF of Bike Availability

See Figure 3: Overall PMF of Bike Availability ↗

## 4.4 Observations

### 4.4.1 Overall Probability Mass Function

The PMF shows clear right skew. Empty states dominate the distribution peak with $P(X = 0) \approx 9.6\%$. Most probability mass lies between 4–15 bikes, with $E[X] = 11.5$. High-capacity stations create a long upper tail.

### 4.4.2 Station-Level Variability

Large variation across stations. Worst performers exceed 20% emptiness:

- Clarendon Row: 25.48%
- Molesworth Street: 25.16%
- Parnell Square North: 23.47%

Network mean is 9.87%, meaning several stations are consistently unreliable.

### 4.4.3 Law of Large Numbers

The running estimate of $P(E = 1)$ stabilises rapidly. After $10^5$ observations, variance is negligible. Final converged value:

$$P(E = 1) = 0.0966.$$

# 5 Question 2: QUESTION 2

## 5.1 Objective

Determine whether bike usage correlates with temporal factors and weather. We will be looking for an event of $E$(station empty) and test whether $P(\text{Empty} \mid \text{Hour} = h) = P(\text{Empty})$ for all hours of $h$ and days of the week.

## 5.2 Approach

We will compute conditional probabilities $P(E \mid \text{hour})$, $P(E \mid \text{day})$, and joint conditional $P(E \mid \text{Hour}, \text{Day})$. To test for independence, we will be using the chi-squared test, comparing observed frequencies against expected frequencies under the null hypothesis of independence.

The null hypothesis , which is independent, is:

$$P(E \text{ and Hour} = h) = P(E) \times P(\text{Hour} = h)$$

The alternative hypothesis, which indicates that the weather phenomenon influence availability:

$$P(E \text{ and Hour} = h) \neq P(E) \times P(\text{Hour} = h)$$

### 5.2.1 Data Preparation

Conditional probability by hour and day of week

```
hourly_counts = df.groupby('hour')['E'].agg(['sum', 'count'])
hourly_counts['P_empty'] = hourly_counts['sum'] / hourly_counts['count']
```

```
daily_counts = df.groupby('day_of_week')['E'].agg(['sum', 'count'])
daily_counts['P_empty'] = daily_counts['sum'] / daily_counts['count']
```

For weather analysis, daily precipitation was categorized and merged into the data set. We first compute the conditional probability by hour and day of the week, using

$$P(\text{empty} \mid \text{hour} — \text{day}) = \frac{\#\text{empty events}}{\#\text{total events}}.$$

Joint conditional probability (Hour $\times$ Day)

```
heatmap_data = df.groupby(['day_of_week', 'hour'])['E'].mean().unstack()
```

We merged the data by day and hour, took the mean of $E$ (the probability of the empty event), and created a heat map of this joint distribution.

Chi-squared test for independence (Empty vs Hour)

```
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(df['E'], df['hour'])
chi2, p_value, dof, expected = chi2_contingency(contingency_table)
```

To test whether $E$ is independent of the hour of the day, we perform a chi-squared test by constructing a contingency table of $E$ vs. hour and comparing observed versus expected frequencies.

Weather categories

```
def categorise_rain(mm):
    if mm == 0:
        return 'Dry'
    elif mm <= 2:
        return 'Light'
    elif mm <= 10:
        return 'Moderate'
    else:
        return 'Heavy'
```

For the weather component, rainfall (in mm) is categorised as:

$$\text{Dry},\ \text{Light},\ \text{Moderate},\ \text{or Heavy}.$$

```
weather_df['rain_category'] = weather_df['rain'].apply(categorise_rain)
df = df.merge(weather_df[['date', 'rain_category']], on='date')
```

We then merge the weather dataset with the bike-usage dataset to determine whether station emptiness depends on both rainfall category and time of day.

### 5.2.2 Visualization

See Figure 4: Conditional Probability Heatmap (Hour × Day) ↗

The heatmap displays $P(\text{Empty} \mid \text{Hour}, \text{Day})$ across all 168 hour–day combinations. The darker cells indicate a higher chance that a station is empty. This graph shows a clear pattern emerges of weekday mornings (especially Monday to Thursday, 8:00 to 11:00) of elevated empty rates, meanwhile weekends and early morning on weekend shows the lowest amount of emptiness.

See Figure 5: Weekday vs Weekend Comparison ↗

The comparison of weekday to weekend reveals distinct temporal patterns. Weekdays show a peak of emptiness at rush hours, 7:00 AM to 10:00 AM and 5:00 PM to 8:00 PM, with gradual recoveries in early morning and afternoon. Weekends have a more even and flat level of emptiness, as fewer people have to commute and use it for errands and leisure.

See Figure 6: Weather Impact on Hourly Usage ↗

The difference in use due to weather and we see light rain has the highest usage, this could be that it is faster to travel on bike in light rain then taking public transport. We got our data for the weather form Met Éireann ( Met Éireann Merrion Square data)

### 5.3 Observation

The graphs clearly show that bike availably is affected by temporal factors; therefore, we reject the null hypotheses and accept the alternative hypothesis, For the following reason:

Fig 4 shows that 9:00 am on Monday has an emptiness of 17.5% meanwhile 8:00pm on Sunday has an emptiness of 5.6%. This is clearly demonstrates the differences of time of day and day of the week have a significate impact on bike availability.

Fig 5 shows that weekdays has a 1.5 times more emptiness than weekends, indicates that bikes are more available on weekend. For example at 9:00 am on weekday emptiness exceed 14%, meanwhile weekends has 7% at the same time. This further support the rejection of the null hypothesis.

Fig 6 shows that weather condition also effects the availability of bike. Surprisingly light rain has the highest emptiness at 13.3% compare to 12.1% in dry weather at similar times. This highlights the difference of weather on bike usage. One possibly explanation for light rain having a higher bike usage to dry weather dry is that public transport often slows down significantly in wet weather, leading people to cycle and risk of getting wet compared to being late.

## 6 Question 3

To determine whether availability states at nearby stations are independent. Formally, for two stations A and B, we test whether:

$$P(A = \emptyset \text{ and } B = \emptyset) = P(A = \emptyset) * P(B = \emptyset).$$

where $\emptyset$ refers to the empty set.

If the above equality test fails, knowing one station is empty provides information about neighbouring stations, with practical implications for users and operators.

## 6.1 Approach

We compute the probability $P(\text{A} \cap \text{B})$ for all pairs of stations by looking at observations at identical timestamps. The dependence ratio quantifies co-occurrence, avoidance or independence between the two probabilities and is defined as

$$\text{Dependence Ratio} = \frac{P(A \cap B)}{P(A)\,P(B)}.$$

A ratio of 1 indicates independence. A ratio greater than 1 indicates positive dependence (or co-occurrence), meaning stations tend to be empty together more often than chance would predict. A ratio below 1 indicates negative dependence (or avoidance), meaning that one station tends to be empty, while the other does not. To examine how dependence varies with distance, we computed the Haversine distance between all pair of stations and aggregated the results by distance bins. The Haversine distance is a method of distance calculation between two points on Earth, calculated from their latitude and longitude. It provides an accurate measure of the straight-line geographic distance between two geographic points (in this case, Dublin Bikes stations).

## 6.2 Python Implementation

```
import numpy as np

def haversineDistance(latA, lonA, latB, lonB):
    r = 6371
    a1, o1, a2, o2 = map(np.radians, [latA, lonA, latB,
        lonB])
    x = np.sin((a2 - a1)/2)**2
    y = np.sin((o2 - o1)/2)**2 * np.cos(a1) * np.cos(a2)
    return 2 * r * np.arcsin(np.sqrt(x + y))

df['timeRounded'] = df['time'].dt.round('5min')
emptyMatrix = df.pivot_table(index='timeRounded',
    columns='station_id', values='E', aggfunc='first')

results = []
stationList = list(emptyMatrix.columns)

for i in range(len(stationList)):
    for j in range(i+1, len(stationList)):
        sA, sB = stationList[i], stationList[j]
        a, b = emptyMatrix[sA], emptyMatrix[sB]
        m = a.notna() & b.notna()
        a, b = a[m], b[m]
        pA = float(a.mean())
        pB = float(b.mean())
        pAB = float(((a == 1) & (b == 1)).mean())
        dep = pAB / (pA * pB) if pA * pB > 0 else np.nan
```

```
        pBgA = ((a == 1) & (b == 1)).sum() / (a == 1).sum()
            if (a == 1).sum() > 0 else np.nan
        d = haversineDistance(lat_a, lon_a, lat_b, lon_b)
        results.append({'distanceKm': d, 'pA': pA, 'pB':
            pB, 'pAB': pAB, 'pBgA': pBgA, 'dep': dep})

resultsDf = pd.DataFrame(results)

resultsDf['distanceBin'] = pd.cut(
    resultsDf['distanceKm'],
    bins=[0,0.3,0.5,0.75,1.0,1.5,2.0,3.0,5.0],
    labels=['0-0.3','0.3-0.5','0.5-0.75','0.75-1','1-1.5','1.5-2','2-3','3-5']
)

binned = resultsDf.groupby('distanceBin')
.agg({'dep':['mean','std','count'],'pBgA':'mean','pB':'mean'})
```

## 6.3 Visualization

See Figure 7: Spatial Dependence Decay with Distance ↗

The data suggests that the dependence ratio keeps decreasing with distance. Nearby stations (less than 300 meters apart) show a ratio of 3.86, indicating they are nearly four times more likely to be empty together than independence would predict. The ratio decreases to 1.36 for stations 3-5 km apart, approaching but not reaching independence.

See Figure 8: Conditional Lift by Distance ↗

This figure above compares the marginal probability $P(B_{\text{empty}})$ against the conditional probability $P(B_{\text{empty}} \mid A_{\text{empty}})$ for each distance bin. The gap between bars represents the "lift" from conditioning on station $A$ being empty. For nearby stations, knowing $A$ is empty increases the probability that $B$ is empty from 11 percent to 37 percent, a 3.3-fold increase.

## 6.4 Results

Table 1: Spatial dependence by distance (6,660 station pairs analysed)

| Distance (km) | Pairs | Dep. Ratio | $P(B_{\text{empty}} \mid A_{\text{empty}})$ | $P(B_{\text{empty}})$ | Lift |
|---|---|---|---|---|---|
| 0–0.3 | 123 | 3.86 | 37.15% | 11.17% | 3.33× |
| 0.3–0.5 | 262 | 3.19 | 29.90% | 10.48% | 2.85× |
| 0.5–0.75 | 452 | 3.02 | 28.11% | 10.34% | 2.72× |
| 0.75–1 | 550 | 2.55 | 25.01% | 10.56% | 2.37× |
| 1–1.5 | 1,259 | 2.15 | 20.72% | 10.22% | 2.03× |
| 1.5–2 | 1,245 | 1.76 | 15.84% | 9.66% | 1.64× |
| 2–3 | 1,961 | 1.47 | 12.57% | 9.37% | 1.34× |
| 3–5 | 800 | 1.36 | 11.12% | 9.44% | 1.18× |

The dependence ratio decreases with distance, exhibiting a correlation of $r = -0.325$ ($p < 0.001$). The effect decreases from a ratio of 3.86 at close range to 1.36 at 3–5 km, representing a 65% reduction with a marginal increase in distance. However, even distant stations show positive dependence where the dependence ratio is $> 1$, indicating that stations in the system can empty simultaneously due to temporal demand patterns in the city (e.g., morning rush hour, evening reverse-commute time, holidays etc.).

To isolate the true spatial effect, we subtract the baseline dependence observed at far distances from the strong dependence observed at close distances:

$$\text{Spatial contribution} = \text{Nearby ratio} - \text{Far ratio} = 3.86 - 1.36 = 2.50.$$

Thus, spatial proximity contributes an additional dependence 2.5x beyond the temporal correlation of the city.

### 6.5 Observations

- The results confirm the existence of "bike deserts" in the Dublin Bikes system. When one station becomes empty, nearby stations are much more likely to also be empty. This occurs for three reasons:
  - Nearby stations share the same demand pool. Commuters in a neighbourhood choose whichever station is closest that is not empty. If one empties, demand shifts to the next nearest, which then also empties.
  - Cascading failures can occur as a user finding station A empty walks to station B and grabs a bike from there. This behaviour propagates emptiness through a concentrated local area.
  - Due to the design of rebalancing efforts, a truck with a load of bikes can target specific areas, not individual stations. If a load has not reached an area, it suffers from emptiness.
- For stations less than 300 metres apart, the conditional probability $P(B_{\text{empty}} \mid A_{\text{empty}})$ equals 37.15%, compared to the marginal $P(B_{\text{empty}})$ of 11.17%. This 3.3-fold increase has practical implications: if a user finds their nearest station empty, walking to a nearby alternative offers only modest improvement in expected availability. People online have complained about availability in closeby station on Reddit, and it can be looked at here.
- The monotonic decay of dependence with distance is consistent with spatial autocorrelation observed in urban systems. The approximately exponential decay pattern suggests a characteristic length scale of roughly 1–2 km, beyond which stations behave more independently.

# 7 Conclusions

## 7.1 Model Effectiveness

The combination of formal probability modelling and Python computation worked effectively for analysing the Dublin Bikes system. Clear definitions of random variables and access to a large dataset enabled stable estimation of probabilities, temporal patterns, and spatial dependence.

## 7.2 Key Findings

- **RQ1:** Bike availability is right-skewed with mean $E[X] = 11.5$. The emptiness probability is $P(E = 1) = 0.0966$, with strong variation across stations (1%–25%).
- **RQ2:** Emptiness depends heavily on hour and day. Weekday rush hours show more than double the emptiness rate of weekends. Chi-squared tests reject independence for both temporal and weather factors.
- **RQ3:** Nearby stations show strong positive dependence. For stations within 300 m, the dependence ratio is 3.86, and $P(B_{\text{empty}} \mid A_{\text{empty}})$ increases to 37%. Dependence decreases with distance but remains above 1 due to system-wide temporal effects.

## 7.3 Implications

- Empty stations cluster hence nearby stations cannot be treated as independent backups.
- Rebalancing efforts should target local level clusters, not isolated stations.
- Temporal patterns dominate usage, suggesting increased redistribution efforts during morning and evening peaks.

## 7.4 Limitations

- Weather data is aggregated daily, limiting precision and correlation with temporal factors.
- "Birds Eye Distance" measures do not account for real travel routes, traffic or accessibility factors.

# References

[1] Dublin City Council. (2025). *Dublin Bikes API (2018–2022).* https://data.smartdublin.ie/dataset/dublinbikes-api

[2] Met Éireann. (2025). *Merrion Square Dublin: Daily precipitation data (2018–2022)*. https://www.met.ie/climate/available-data/historical-data

[3] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

[4] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature, 585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2