Problem Set 4

Q1. 0.0167

The probability is given by the multinomial formula:

Probability = (15! / (6! × 4! × 2! × 3!)) × (0.4^6) × (0.3^4) × (0.1^2) × (0.2^3)

= 6,306,300 × (0.4^6) × (0.3^4) × (0.1^2) × (0.2^3)

= 0.0167

Q2.

a) 0.1378

Let X ~ Poisson(lambda = 7.5).

Probability = 1 - sum{k=0}^{10} e^{-7.5} (7.5^k / k!).

= 0.1378

b) 0.2187

Quickgram 2-minute rate = 5.5 × 2 = 11, so Y ~ Poisson(lambda = 11).

Probability = 1 - sum{k=0}^{13} e^{-11} (11^k / k!).

=0.2187

c) 0.003942

Two-minute rates: Lookbook = 7.5 × 2 = 15, Quickgram = 5.5 × 2 = 11. Combined T ~ Poisson(lambda = 15 + 11 = 26).

Probability = 1 - sum{k=0}^{40} e^{-26} (26^k / k!).

= 0.003942

Q3. 0.5207

We have two possible scenarios:

- Phone A (drop probability = 0.2) with prior probability 0.5.
- Phone B (drop probability = 0.3) with prior probability 0.5.

In the first 3 calls, exactly 1 is dropped.

For Phone A: P(1 drop in 3 calls) = C(3,1) × (0.2)^1 × (0.8)^2 = 3 × 0.2 × 0.64 = 0.384.

For Phone B: P(1 drop in 3 calls) = C(3,1) × (0.3)^1 × (0.7)^2 = 3 × 0.3 × 0.49 = 0.441.

Posterior for Phone A = (0.5 × 0.384) / (0.5 × 0.384 + 0.5 × 0.441)

= 0.384 / (0.384 + 0.441)

= 0.4655.

Posterior for Phone B = 0.441 / (0.384 + 0.441)

= 0.5345.

If Phone A: expected drops = 6 × 0.2 = 1.2.

If Phone B: expected drops = 6 × 0.3 = 1.8.

So unconditional expected total drops = 0.4655 × 1.2 + 0.5345 × 1.8

≈ 0.5586 + 0.9621 = 1.5207.

Since we already observed 1 drop in the first 3 calls, expected future drops in the remaining 3 calls = 1.5207 – 1 = 0.5207.

Q4. 0.013333

With probability 0.99 the person is immune. In that case, return value = 0.

With probability 0.01 the person is *not* immune. In that case, they become infected themselves and may infect others.

Expected number of contacts infected (k):

k ~ Binomial(n=100, p=0.25) → expected k = 100 × 0.25 = 25.

So on average, each infected person makes 25 attempts to spread.

Recurrence for expectation:

Let E = expected return value of num_infected().

- With probability 0.99: result = 0.
- With probability 0.01: result = 1 (this person) + expected infections from k others.

So:

E = 0.99 × 0 + 0.01 × (1 + E[k] × E).

= 0.01 × (1 + 25E)

E = 0.01 / 0.75 = 0.013333

Q5.

a)

$P(X = x, Y = y) = 1/(5*x)$ for $1 \le y \le x \le 5$, otherwise 0.

b)

$P(X = x \mid Y = i) = 0$ if $x < i$, and $P(X = x \mid Y = i) = (1/x) / \sum_{t=i}^{5} (1/t)$ if $x \ge i$

Examples:

- Y = 1: P(X = 1)=60/137, P(X = 2)=30/137, P(X = 3)=20/137, P(X = 4)=15/137, P(X = 5)=12/137
- Y = 2: P(X = 2)=30/77, P(X = 3)=20/77, P(X = 4)=15/77, P(X = 5)=12/77
- Y = 3: P(X = 3)=20/47, P(X = 4)=15/47, P(X = 5)=12/47
- Y = 4: P(X = 4)=5/9, P(X = 5)=4/9
- Y = 5: P(X = 5)=1

c)

X and Y are **not independent** because Y cannot exceed X. For example, $P(X = 1, Y = 1) \neq P(X = 1) \cdot P(Y = 1)$, so the joint probability is not the product of the marginals.

Q6.

$Cov(Y_n, Y_{n+j}) = Cov(X_n + X_{n+1}, X_{n+j} + X_{n+j+1})$

Use the facts that $Cov(X_a, X_b) = 0$ if $a \neq b$ and $Cov(X_a, X_a) = Var(X_a) = \sigma^2$.

a) $j = 0$:

$Cov(Y_n, Y_n) = Var(X_n + X_{n+1}) = Var(X_n) + Var(X_{n+1}) = \sigma^2 + \sigma^2 = 2\sigma^2$

b) $j = 1$:

$Cov(Y_n, Y_{n+1}) = Cov(X_n + X_{n+1}, X_{n+1} + X_{n+2})$

Only $Cov(X_{n+1}, X_{n+1})$ is nonzero, so $Cov(Y_n, Y_{n+1}) = \sigma^2$

c) $j = 2$:

$Cov(Y_n, Y_{n+2}) = Cov(X_n + X_{n+1}, X_{n+2} + X_{n+3})$

All indices are distinct, so every covariance term is 0, hence $Cov(Y_n, Y_{n+2}) = 0$

Q7. code

Q8. Not Done

Q9.

a) code
b) The EPSILON term is necessary to avoid returning a probability of zero for words that do not appear in the training data. Without it, taking the product of word probabilities (as in the multinomial model) would result in zero for any document containing an unseen word, which would make Bayesian predictions impossible. EPSILON ensures that unseen words contribute a very small probability instead of zero, allowing the model to handle new words gracefully.

Q10.

Arithmetic underflow occurs because the product of many small probabilities becomes extremely small. In this example, each word probability (p_i or q_i) is around 0.03–0.094, and the document contains multiple words. When we compute the product (p_i^c_i) or (q_i^c_i), we are multiplying several numbers much smaller than 1, which quickly produces a number smaller than the smallest representable double-precision floating-point number in Python (approximately 5×10^{-324}). Once the result is below this threshold, Python rounds it to zero, causing underflow.

Q11. Not Done

Q12.

a) code
b) code