# Begum Zubeda: Stats Exam

**#1. From the pack of 52 cards, three cards are drawn randomly without replacement then what is the probability that one card is a diamond, one card is a heart and one is spade?**

cards = 52

heart = 13

spade = 13

diamond = 13

prob\_diamond = 13/52

prob\_heart = 13/51

prob\_spade = 13/50

req\_prob = prob\_diamond \* prob\_heart \* prob\_spade

req\_prob

**#2. The information gained by a survey done by a production company: 42 % of the respondents said that they like action movies, 54 % like comedy movies, 36 % like drama movies, and 12 % like horror movies. If a person is selected at random, find the probability that his or her,**

**#a. Favorite movie type is either action or drama.**

**#b. Favorite movie type is either comedy or horror.**

action = 0.42

comedy = 0.54

drama = 0.36

horror = 0.12

prob\_action\_or\_drama = action + drama

prob\_comedy\_or\_horror = comedy + horror

print("Probability of his/her Favorite movie type is either action or drama is ", prob\_action\_or\_drama)

print("Probability of his/her Favorite movie type is either comedy or horror ", prob\_comedy\_or\_horror)

**#3. A bag A contains 3 red and 5 black balls and bag B contains 4 white and 7 black balls. A bag is selected randomly and a ball is drawn from it. A drawn ball is observed to be black. Find the probability that bag ‘B ’ was selected.**

prob\_A = prob\_B = 1 / 2

prob\_black\_A = 5 / 8

prob\_black\_B = 7 / 11

prob\_B\_black = (prob\_B \* prob\_black\_B) / ((prob\_A \* prob\_black\_A) + (prob\_B \* prob\_black\_B))

print("The probability of B selected and black ball drawn is ", prob\_B\_black)

**#4. A MNC company receives 450 applications from applicants in one hour. Find the probability of:**

**from scipy import stats**

**#a. Receiving 10 applications in 1 minute.**

avg1 = 450 / 60

x1 = 10

print("Probability of Receiving 10 applications in 1 minute is ", stats.poisson.pmf(x1, avg1))

#b. Receiving at least 17 applications in 2 minutes.

avg2 = 450 / (60 \* 2)

x2 = 17

print("Probability of Receiving at least 17 applications in 2 minutes is ", stats.poisson.cdf(x2, avg2))

**#5. The government of state union has declared a free medical insurance for below poverty line population by using following assumptions:**

**#a. In every year, there can be at most one patient who needs medical insurance in a family.**

**#b. In every year, the probability of a medical emergency is 0.05.**

**#c. The number of patients in every year is independent.**

**#Using the assumptions, calculate the probability that there are fewer than 3 patients in a 10 years period in one family.**

stats.poisson.cdf(3, 0.01 \* 0.05)

**#6. Suppose prices of new flats in Manchester are normally distributed with an average of $350870 and a standard deviation of $12405. Find the 75th percentile of prices among new homes in Manchester.**

avg = 350870

sd = 12405

p50 = avg

p75 = 0

for i in range(1,16):

x = avg + sd

p75 = p75 + x

print("75% Percentile: ", p75)

**#7. The Department of Transportation of California claims that the average number of road accidents that are occurred in two cities - Union City and Mountain View - during the span of the last 12 months. Assuming that population variances are equal**

**#Compute the value of F-statistic by using the relation F = t2**

import numpy as np

union = np.array([44, 21, 36, 34, 77, 32, 29, 39, 92, 67, 45, 37])

mview = np.array([47, 81, 41, 79, 38, 20, 33, 83, 41, 40, 69, 36])

def f\_test(x, y):

f = np.var(x, ddof=1)/np.var(y, ddof=1) #calculate F test statistic

dfn = x.size - 1 #define degrees of freedom numerator

dfd = y.size - 1 #define degrees of freedom denominator

p = 1 - stats.f.cdf(f, dfn, dfd) #find p-value of F test statistic

return f, p

#perform F-test

f\_test(union, mview)

**#8. Annual project of university done by three groups of students with equal sample sizes. Each group was given a different task. After the final presentation students get the marks. The summary statistics are given below:**

group1 = {'mean': 44.5, 'sd': 5, 'size': 6}

group2 = {'mean': 42, 'sd': 5, 'size': 6}

group3 = {'mean': 46.5, 'sd': 8, 'size': 6}

**#9. The number of pizzas sold per day by a food zone "Pazzi per Pizza" follows a poisson distribution at a rate of 76 pizzas per day. What is the probability that the number of pizza sales exceeds 80 in a day? Write Python code to calculate the probability.**

avg = 76

x = 80

1 - stats.poisson.cdf(x, avg)

**# 10. In the population, the mean weight is 82. A team of dietitians wants to test a new protein supplier to see if it has either a positive or negative effect on diet, or no effect at all. A sample of 25 participants who have taken the protein supplier has a mean of 95 with standard deviation is 20. Did the protein supplier affect diet? Where a = 0.05**

rvs1 = stats.norm.rvs(loc=82, scale=20, size=25)

rvs2 = stats.norm.rvs(loc=95, scale=20, size=25)

stats.ttest\_rel(rvs1, rvs2)

**#11. The number of customers that arrive on different days in a week at the Hotel Taj is given below:**

**# Test the claim that the different days of the week have the same frequency of customers. Use the significance level as 0.1.**

import pandas as pd

df = pd.DataFrame({'Mon': [1419], 'Tues': [1320], 'Wed': [1526], 'Thurs': [1289], 'Fri': [1620], 'Sat': [2067], 'Sun': [2055]})

from scipy.stats import chi2\_contingency

#Check for independence of frequence

stat, p, dof, expected = chi2\_contingency(df)

stat, p

**#12. The number of deaths in the 7 metro cities of the US during the last month due to Covid-19 is recorded by the government, and that time, experts predicted that upto next month death rate will be increased by 30%. So the number of deaths in current month is also recorded in 7 cities in US are given below:**

City = ["New York", "New Jersey", "Michigan", "California", "Florida", "Massachusetts", "Texas"]

No\_of\_deaths\_in\_last\_month = [3406, 1469, 662, 583, 582, 526, 461]

No\_of\_deaths\_in\_current\_month = [4398, 1846, 1288, 382, 879, 430, 321]

df = pd.DataFrame({"City": City, "No\_of\_deaths\_in\_last\_month":No\_of\_deaths\_in\_last\_month, "No\_of\_deaths\_in\_current\_month": No\_of\_deaths\_in\_current\_month})

#Is the death distribution of the current month the same as the expert’s prediction? Use the level of significance is 0.1.

stats.ttest\_rel(df['No\_of\_deaths\_in\_last\_month'], df['No\_of\_deaths\_in\_current\_month'])