**Aula 5 – Trabalho em Grupo**

**GABRIELLA PINHEIRO**

**GUILHERME ROSSAN**

**GABRIEL COUTINHO**

**LIZ BORGES**

Um poderoso grupo financeiro deseja investir pesadamente no negócio de animais de estimação, com a marca **XYZ**, abrindo milhares de pequenas lojas de conveniência para animais nas grandes capitais do mundo. Seus profissionais de Marketing querem entender como uma pessoa escolhe a aparência de um cachorro para tê-lo como animal de estimação. Há vários estudos que tentam verificar se a aparência dos cães lembra a de seus donos (e vice-versa!). Por exemplo, Roy & Christenfeld publicaram em 2004 o artigo “[*Do dogs resemble their owners?*](http://www.ncbi.nlm.nih.gov/pubmed/15102149)”, concluindo que “[…] *when people pick a pet, they seek one that, at some level, resembles them […]*.”

A **XYZ** fez o seguinte experimento. Selecionou, de forma aleatória, oito de seus clientes e respectivos cachorros (só foram pegos clientes que possuíam apenas um cachorro). Convidou uma banca de juízes (estudantes universitários que não possuíam cachorros e que não conheciam os donos ou os cachorros).

Mostrou aos juízes então a foto do primeiro entre os oito clientes, juntamente com a foto de seu cachorro e mais uma foto de outro cachorro (sorteada entre os demais sete cachorros). Os juízes tinham que dizer qual dos dois cachorros de fato pertencia ao dono. Se a banca de juízes acertasse, seria considerado que houve acerto na determinação do cachorro apenas pela aparência. Se a banca de juízes errasse, seria considerado que houve erro na determinação do cachorro apenas pela aparência. O processo foi então repetido para cada um dos demais clientes.

O resultado foi: para sete dos oito clientes, houve acerto na determinação do cachorro correto pela banca de juízes. **XYZ** concluiu então que, por alguma razão, há razoável semelhança entre a aparência dos cachorros e dos seus respectivos donos. O porquê dessa razão é outra questão: as pessoas podem escolher cachorros parecidos com elas mesmas ao decidirem comprar um animal de estimação ou este último acaba ficando parecido pelo convívio.

### Roteiro

1. Admitindo que a banca de juízes está meramente adivinhando (“chutando”), informem se vocês achariam “provável” ou “improvável” cada um dos resultados abaixo (**usem apenas a intuição de vocês**):
   * Acertar 0 entre os 8 cachorros: 🗆 provável 🗆 improvável
   * Acertar 1 entre os 8 cachorros: 🗆 provável 🗆 improvável
   * Acertar 2 entre os 8 cachorros: 🗆 provável 🗆 improvável
   * Acertar 3 entre os 8 cachorros: 🗆 provável 🗆 improvável
   * Acertar 4 entre os 8 cachorros: 🗆 provável 🗆 improvável
   * Acertar 5 entre os 8 cachorros: 🗆 provável 🗆 improvável
   * Acertar 6 entre os 8 cachorros: 🗆 provável 🗆 improvável
   * Acertar 7 entre os 8 cachorros (este foi o resultado obtido por **XYZ**!): 🗆 provável 🗆 improvável
   * Acertar 8 entre os 8 cachorros: 🗆 provável 🗆 improvável
2. O resultado obtido por **XYZ** nos oferece uma evidência que dá força à ideia de que os juízes estão apenas adivinhando (“chutando”) ou à ideia de que de fato eles conseguem determinar os cachorros pela aparência? Explique.

O resultado **não apoia a ideia de que os juízes estavam apenas chutando**.

Pelo contrário, ele oferece **forte evidência de que os juízes conseguiram determinar os cachorros pela aparência.**

Isso porque, ao simularmos 100.000 vezes o experimento com a hipótese de chute (probabilidade de 50% por tentativa), a chance de se obter **7 ou mais acertos em 8 tentativas** foi de aproximadamente: **3,5%**

1. Usem simulação. Determinem a probabilidade de, no caso de a banca de juízes estar meramente adivinhando (“chutando”), acertar-se o cachorro que pertence ao respectivo dono em pelo menos sete entre os oito casos.

# Importação de bibliotecas

import numpy as np # para gerar números aleatórios e cálculos vetoriais

import pandas as pd # para tabular e organizar os resultados

import matplotlib.pyplot as plt # para gerar os gráficos

import seaborn as sns # para tornar os gráficos mais bonitos

np.random.choice([0, 1])

tentativas = np.random.choice([0, 1], size=8)

acertos = tentativas.sum()

print(f"Acertos em 8 tentativas: {acertos}")

n\_simulacoes = 8

resultados = np.random.binomial(n=8, p=0.5, size=n\_simulacoes)

resultados

frequencia = pd.Series(resultados).value\_counts().sort\_index()

tabela = pd.DataFrame({

'Acertos': frequencia.index,

'Frequência absoluta': frequencia.values,

'Frequência relativa (%)': (frequencia.values / n\_simulacoes \* 100).round(2)

})

tabela

plt.figure(figsize=(12,6))

sns.barplot(data=tabela, x='Acertos', y='Frequência relativa (%)', color='skyblue')

plt.title('Distribuição dos acertos de Pavel em 25 tentativas')

plt.ylabel('Frequência relativa (%)')

plt.xlabel('Número de acertos')

plt.xticks(rotation=0)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()

---

import numpy as np

n\_simulacoes = 200000

n\_testes = 8

p\_acerto = 0.5

resultados = np.random.binomial(n=n\_testes, p=p\_acerto, size=n\_simulacoes)

sucessos = np.sum(resultados >= 7)

probabilidade = sucessos / n\_simulacoes

print(probabilidade\*100, "%")