

```
from google.colab import drive
drive.mount('/content/drive')
```

↗ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv('/content/drive/MyDrive/Custom-Churn-Records.csv')
```

1. ¿Cuál es la distribución de clientes por región?

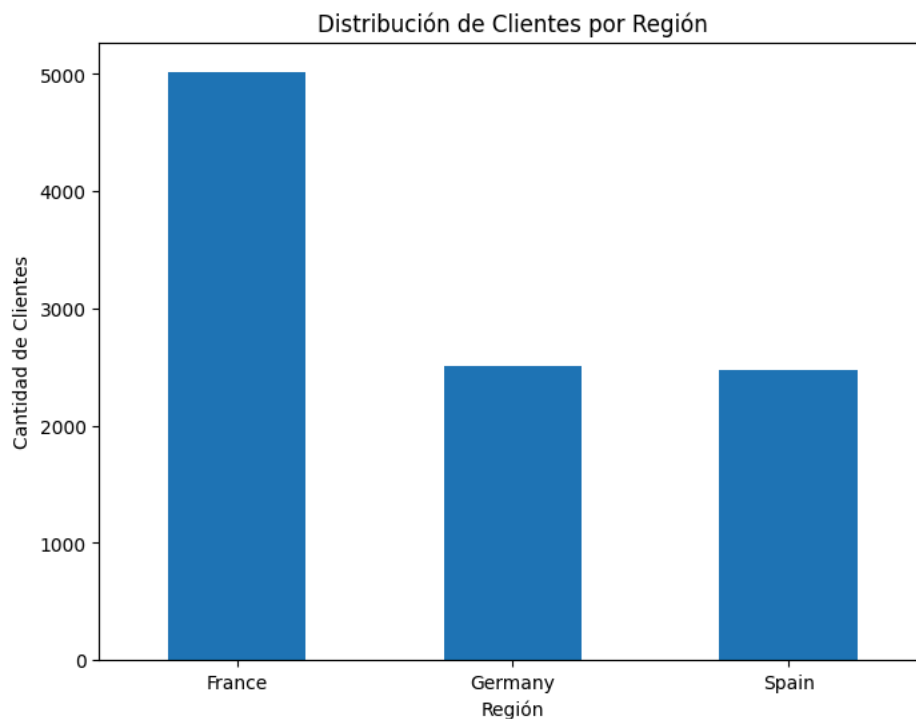
```
# Conteo exacto de clientes por región
region_counts = data['Geography'].value_counts()
print("Cantidad de Clientes por Región:")
print(region_counts)
```

```
# Gráfico de barras
region_counts.plot(kind='bar', figsize=(8, 6))
plt.title("Distribución de Clientes por Región")
plt.xlabel("Región")
plt.ylabel("Cantidad de Clientes")
plt.xticks(rotation=0)
plt.show()
```

↗ Cantidad de Clientes por Región:

Geography	count
France	5014
Germany	2509
Spain	2477

Name: count, dtype: int64



2. ¿Cuál es la proporción de abandono vs. no abandono?

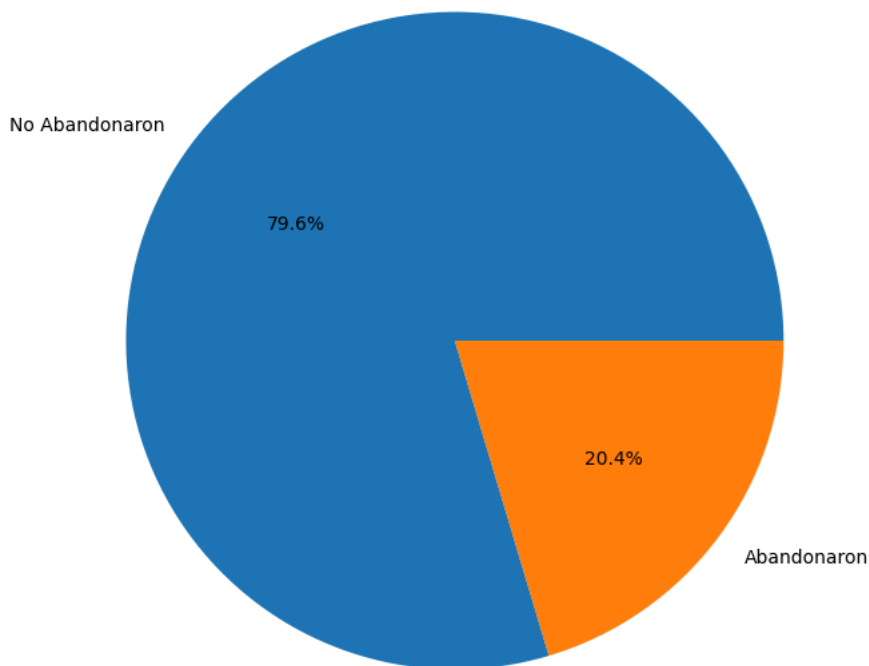
```
# Porcentaje exacto de clientes que abandonaron vs no abandonaron
exit_counts = data['Exited'].value_counts(normalize=True) * 100
print("Porcentaje de Clientes que Abandonaron vs No Abandonaron:")
print(exit_counts)
```

```
# Gráfico de pastel
data['Exited'].value_counts(normalize=True).plot(
    kind='pie', autopct='%1.1f%%', labels=["No Abandonaron", "Abandonaron"], figsize=(8, 8))
```

```
plt.title("Proporción de Clientes que Abandonaron")
plt.ylabel("") # Elimina la etiqueta del eje y
plt.show()
```

```
↳ Porcentaje de Clientes que Abandonaron vs No Abandonaron:
Exited
0    79.62
1    20.38
Name: proportion, dtype: float64
```

Proporción de Clientes que Abandonaron



3. ¿Cuál es el puntaje crediticio promedio por región?

```
# Puntaje crediticio promedio por región
credit_score_by_region = data.groupby('Geography')['CreditScore'].mean()
print(credit_score_by_region)
```

```
↳ Geography
France    649.668329
Germany   651.453567
Spain     651.333872
Name: CreditScore, dtype: float64
```

4. ¿Cuál es la región con mayor porcentaje de abandono?

```
# Región con mayor porcentaje de abandono
abandon_rate_by_region = data.groupby('Geography')['Exited'].mean() * 100
print(abandon_rate_by_region.sort_values(ascending=False))
```

```
↳ Geography
Germany    32.443204
Spain      16.673395
France     16.174711
Name: Exited, dtype: float64
```

5. ¿Cuál es el promedio de edad de los clientes que abandonaron el banco?

```
# Promedio de edad de los clientes que abandonaron
average_age_churn = data[data['Exited'] == 1]['Age'].mean()
print(f"Promedio de edad de los clientes que abandonaron: {average_age_churn}")
```

➞ Promedio de edad de los clientes que abandonaron: 44.83562315996075

6. ¿Cuál es la probabilidad de que un cliente tenga saldo nulo (Balance = 0)?

```
# Probabilidad de saldo nulo
p_saldo_nulo = (data['Balance'] == 0).mean()
print(f"Probabilidad de que un cliente tenga saldo nulo: {p_saldo_nulo:.2f}")
```

➞ Probabilidad de que un cliente tenga saldo nulo: 0.36

7. ¿Cuál es la probabilidad conjunta de que un cliente haya abandonado y tenga más de 2 productos bancarios?

```
# Probabilidad conjunta de abandono y más de 2 productos
p_abandono_y_mas_productos = ((data['Exited'] == 1) & (data['NumOfProducts'] > 2)).mean()
print(f"Probabilidad conjunta de abandono y más de 2 productos: {p_abandono_y_mas_productos:.2f}")
```

➞ Probabilidad conjunta de abandono y más de 2 productos: 0.03

8. ¿Cuál es la probabilidad condicional de que un cliente haya abandonado dado que tiene más de 2 productos?

```
# Probabilidad condicional de abandono dado que tiene más de 2 productos
p_abandono_dado_mas_productos = ((data['Exited'] == 1) & (data['NumOfProducts'] > 2)).sum() / (data['NumOfProducts'] > 2).sum()
print(f"Probabilidad condicional de abandono dado que tiene más de 2 productos: {p_abandono_dado_mas_productos:.2f}")
```

➞ Probabilidad condicional de abandono dado que tiene más de 2 productos: 0.86

9. ¿Qué proporción de gente tiene cada tipo de tarjeta?

```
# Calcular la proporción de clientes por tipo de tarjeta
card_type_proportion = data['Card Type'].value_counts(normalize=True) * 100

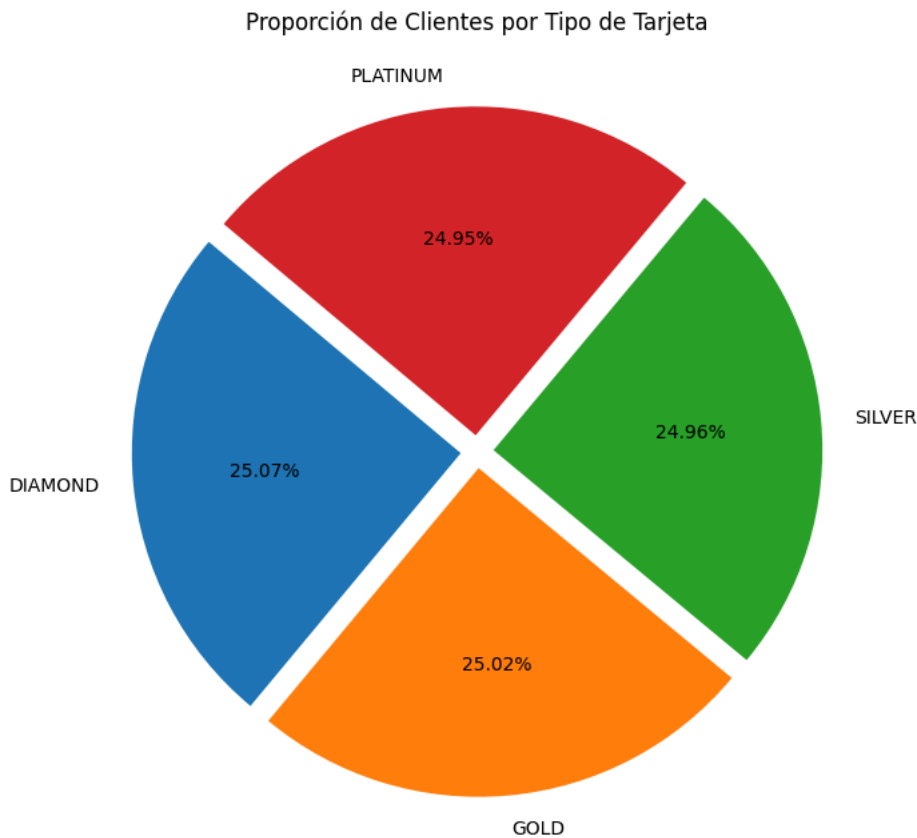
# Mostrar resultados
print("Proporción de clientes por tipo de tarjeta:")
print(card_type_proportion)

# Crear un gráfico de pastel
plt.figure(figsize=(8, 8))
plt.pie(
    card_type_proportion,
    labels=card_type_proportion.index,
    autopct='%1.2f%%',
    startangle=140,
    explode=[0.05] * len(card_type_proportion) # Separar las secciones ligeramente
)
plt.title("Proporción de Clientes por Tipo de Tarjeta")
plt.show()
```

```

Proporción de clientes por tipo de tarjeta:
Card Type
DIAMOND      25.07
GOLD         25.02
SILVER       24.96
PLATINUM     24.95
Name: proportion, dtype: float64

```



10. ¿Qué tan activo está cada género en promedio?

```

# Actividad promedio por género
activity_by_gender = data.groupby('Gender')['IsActiveMember'].mean()*100
print(activity_by_gender)

```

```

Gender
Female    50.275149
Male      52.538025
Name: IsActiveMember, dtype: float64

```

11. ¿Cuál es el salario estimado promedio de los clientes satisfechos?

```

# Salario estimado promedio de clientes satisfechos
satisfied_salary_avg = data[data['Satisfaction Score'] > 3]['EstimatedSalary'].mean()
print(f"Salario promedio de clientes satisfechos: {satisfied_salary_avg}")

```

```

Salario promedio de clientes satisfechos: 99344.21097208375

```

12. ¿Cuál es la diferencia en el balance promedio entre clientes que abandonaron y los que no?

```

# Diferencia en balance promedio
balance_churn = data[data['Exited'] == 1]['Balance'].mean()
balance_no_churn = data[data['Exited'] == 0]['Balance'].mean()
difference = balance_churn - balance_no_churn
print(f"Promedio de los que abandonaron: {balance_churn}")
print(f"Promedio de los que no abandonaron: {balance_no_churn}")
print(f"Diferencia en balance promedio: {difference}")

```

```

Promedio de los que abandonaron: 91109.47600588812
Promedio de los que no abandonaron: 72742.75066314996
Diferencia en balance promedio: 18366.725342738166

```

13. ¿Cuál es el valor esperado del saldo (Balance) considerando la segmentación por grupos de edad?

```

# Crear rangos de edad
bins = [18, 25, 35, 45, 55, 65, 100] # Rango de edad
labels = ['18-25', '26-35', '36-45', '46-55', '56-65', '65+']
data['AgeGroup'] = pd.cut(data['Age'], bins=bins, labels=labels, right=False)

```

```

# Calcular el valor esperado del saldo por grupo de edad
balance_by_age_group = data.groupby('AgeGroup')['Balance'].mean()
print(balance_by_age_group)

```

```

AgeGroup
18-25    74022.029869
26-35    73843.462381
36-45    76726.893243
46-55    81272.310480
56-65    81576.547950
65+      71689.609894
Name: Balance, dtype: float64
<ipython-input-35-66b338567bde>:7: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future ve
balance_by_age_group = data.groupby('AgeGroup')['Balance'].mean()

```

14. ¿Cuál es la media, mediana y desviación estándar del puntaje crediticio?

```

# Estadísticas descriptivas del puntaje crediticio
credit_mean = data['CreditScore'].mean()
credit_median = data['CreditScore'].median()
credit_std = data['CreditScore'].std()

print(f"Media del Puntaje Crediticio: {credit_mean}")
print(f"Mediana del Puntaje Crediticio: {credit_median}")
print(f"Desviación Estándar del Puntaje Crediticio: {credit_std}")

```

```

Media del Puntaje Crediticio: 650.5288
Mediana del Puntaje Crediticio: 652.0
Desviación Estándar del Puntaje Crediticio: 96.65329873613061

```

15. ¿Cómo se comparan las distribuciones de saldo entre clientes activos e inactivos?

```

# Histograma del saldo por actividad
sns.histplot(data, x='Balance', hue='IsActiveMember', bins=20, kde=True)
plt.title("Distribución del Saldo por Actividad")
plt.xlabel("Saldo")
plt.ylabel("Frecuencia")
plt.legend(title="Actividad", labels=["Inactivo", "Activo"])
plt.show()

```



Distribución del Saldo por Actividad



16. ¿La proporción de abandono es la misma en las diferentes regiones? Hipótesis nula (H_0): La proporción de abandono es igual en todas las regiones. Hipótesis alternativa (H_1): La proporción de abandono difiere entre regiones.



```
from scipy.stats import chi2_contingency

# Tabulación cruzada de región y abandono
region_exited = pd.crosstab(data['Geography'], data['Exited'])

# Prueba de chi-cuadrado
chi2, p_value, dof, expected = chi2_contingency(region_exited)

print(f"Estadístico chi-cuadrado: {chi2}")
print(f"p-valor: {p_value}")
if p_value < 0.05:
    print("Rechazamos la hipótesis nula: La proporción de abandono difiere entre regiones.")
else:
    print("No se rechaza la hipótesis nula: La proporción de abandono es la misma en todas las regiones.")
```



```
Estadístico chi-cuadrado: 300.6264011211942
p-valor: 5.245736109572763e-66
Rechazamos la hipótesis nula: La proporción de abandono difiere entre regiones.
```

17. ¿Hay una diferencia significativa en el saldo promedio entre hombres y mujeres? Prueba de hipótesis: H_0 : La media del saldo es igual entre hombres y mujeres. H_1 : La media del saldo es diferente entre hombres y mujeres.

```
from scipy.stats import ttest_ind

# Dividir datos por género
balance_male = data[data['Gender'] == 'Male']['Balance']
balance_female = data[data['Gender'] == 'Female']['Balance']

# Prueba t para medias independientes
t_stat, p_value = ttest_ind(balance_male, balance_female)

print(f"Estadístico t: {t_stat}")
print(f"Valor p: {p_value}")
if p_value < 0.05:
    print("Rechazamos la hipótesis nula: Hay una diferencia significativa en el saldo promedio entre hombres y mujeres.")
else:
    print("No se rechaza la hipótesis nula: No hay evidencia suficiente para una diferencia significativa.")
```



```
Estadístico t: 1.2086242759422712
Valor p: 0.22683580909200307
No se rechaza la hipótesis nula: No hay evidencia suficiente para una diferencia significativa.
```

Haz doble clic (o ingresa) para editar