**Visualización Aplicada: Mostrando una Variable Categórica**

Nos centramos ahora en visualizar una única variable (tanto para análisis como presentación). Dividimos en dos grupos categóricas y numéricas. En esta sesión trataremos las categóricas.

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

**import** seaborn **as** sns

sns.set(color\_codes**=True**)

df\_titanic **=** pd.read\_csv("./data/titanic.csv")

df\_seguros **=** pd.read\_csv("./data/Marketing-Customer-Analysis.csv")

**Gráficas y consideraciones**

Veremos las siguientes:

* Diagrama de barras para frecuencias
* Diagrama de esferas/círculos para frecuencias
* Quesos, donuts y otras cosas de comer: lolipops

**Consideraciones generales:**

* **Cuándo**: Análisis univariante de frecuencias y presentación de distribución de valores cuando esto aporte algo (en general para introducciones y dar contexto). Los diagramas de tarta y queso funcionan mejor con frecuencias relativas que con absolutas. Los lolipops no son de mi gusto, pero para que tengas otros.
* **Cuándo no**: Incluso en la situación de que sea necesario por dar contexto hablar de los valores de una variable categórica, si la cardinalidad es mayor de 5 (más o menos) reducirla mostrando 4-5 valores como mucho colapsando los no interesantes en un grupo "otros" (ojo, los no interesantes no son los de menor frecuencia son los que no aporten nada a los mensajes a transmitir )

**Barras**

*### Creamos figura y axes*

fig,axs **=** plt.subplots(nrows**=**1,ncols**=**2, figsize**=**(10,4))

fig.suptitle("Barras en Seaborn");

Gráfico

Descripción generada automáticamente

*### Countplot, nos permite frecuencias absolutas*

sns.countplot(x**=**"embark\_town",data**=**df\_titanic, ax**=**axs[0],hue**=**"embark\_town",legend**=False**)

axs[0].set\_title("Frecuencias absolutas embark\_town")

fig

Gráfico, Gráfico de barras

Descripción generada automáticamente

*# Para frecuencias relativas... hay que calcularlas previamente y puedes usar el barplot*

valores **=** df\_titanic.embark\_town.value\_counts(normalize **=** **True**) **\*** 100

sns.barplot(x **=** valores.index, y **=** valores.values, hue **=** valores.index, ax **=** axs[1])

axs[1].set\_title("Frecuencias relativas embark\_town")

axs[1].set\_ylabel("%")

fig

Gráfico, Gráfico de barras

Descripción generada automáticamente

**Circulos**

*### Creamos figura y axes*

fig,axs**=**plt.subplots(nrows**=**1,ncols**=**1,figsize**=**(14,4))

fig.suptitle("Círculos en Seaborn");

Tabla

Descripción generada automáticamente

*### Frecuencias absolutas y relativas*

frecuencias **=** df\_seguros["state"].value\_counts()

df\_frecuencias **=** df\_seguros["state"].value\_counts().reset\_index()

df\_frecuencias.columns **=** ["categorias","frecuencias"]

sns.scatterplot(x **=** "categorias", y **=** [1]**\***len(frecuencias), hue **=** "categorias", data **=** df\_frecuencias, size **=** "frecuencias", legend **=** **False**, ax **=** axs, sizes **=** (500,5000))

axs.set\_xlabel("") *#hue = "categorias" --> automaticamente coge un color diferente para cada estado. Sino sería el mismo para todos.*

**for** estado,valor **in** frecuencias.items():

axs.text(estado,1.01,f"{valor}({round(valor**\***100**/**frecuencias.sum())}%)") *#1.01 --> indica la altura a la que se posiciona el número*

axs.set\_facecolor("none")

axs.yaxis.set\_ticks([])

fig

[7]:

Gráfico, Gráfico de burbujas

Descripción generada automáticamente

**Donuts, tartas, quesos**

Seaborn no viene con ellos, así que usamos matplotlib, empezamos con la tarta/queso:

data **=** df\_titanic["class"].value\_counts()

fig,ax **=** plt.subplots(1,1,figsize **=** (4,4))

ax.pie(data.values,

labels**=**data.index,

autopct**=**'%.2f%%', startangle**=** 90); *# autopct='%.2f%%' --> 2 decimales*

Gráfico, Gráfico circular

Descripción generada automáticamente

Ahora tipo donut:

my\_circle**=**plt.Circle((0,0),0.5, color**=**"white") *#(0,0)--> indica que el donut se crea en el medio, el 0.5 --> grosor del donut*

ax.add\_artist(my\_circle)

fig

Gráfico, Gráfico de proyección solar

Descripción generada automáticamente

**Lolipops**

Este tampoco lo tenemos en seaborn:

conteo **=** df\_seguros['vehicle\_class'].value\_counts(ascending**=False**)

plt.figure(figsize**=**(10,5))

plt.hlines(y**=**conteo.index,

xmin**=** 50,

xmax**=**conteo,

color**=**'skyblue')

p**=**plt.gcf()

p.gca().set\_facecolor("none")

plt.plot(conteo, conteo.index, "o");

Gráfico, Gráfico de líneas, Gráfico de dispersión

Descripción generada automáticamente