Laboratorio de Datos

Primer Cuatrimestre 2024

Práctica N° 2: Estadística descriptiva y visualización de datos.

Data frames

Un data frame es una representación de los datos en formato de tabla en la que cada columna son vectores del mismo tamaño. Como cada columna es un vector, cada columna puede contener datos de un único tipo. Se pueden pensar como variables.

1. Una forma de crear un data frame es utilizando un "diccionario". Todas las variables del diccionario deben ser vectores o listas de la misma longitud. Ejecutar el siguiente código.

```
data = {"nombre": ["Rodrigo", "Sergio", "Cristina", "Diana"], "altura":
    np.array([178, 172, 175, 168]), "peso": np.array([81.2, 76.1, 68.5,
    64.0])}
display(data)

pacientes = pd.DataFrame(data)
display(pacientes)
```

2. ¿Cuál es la clase del objeto `pacientes`? ¿Cuál es la clase de cada uno de los vectores columna? (para saber la clase de un objeto, utilizar el comando `type`, para saber el tipo de datos de un array de numpy, utilizar `np.dtype`)

3. Guardar en una variable nueva el vector columna `altura`. Pueden utilizar `pacientes["altura"]` o `pacientes.altura` (la primera opción es preferible, la segunda puede dar error si el nombre coincide con alguna función ya existente).

4. Cargar la biblioteca `gapminder` utilizando

```
from gapminder import gapminder
```

   Si da error es posible que no esté instalado. En tal caso ejecuten primero

```
pip install gapminder
```

   Esto crea un nuevo objeto `gapminder`. Pueden ver el contenido con el comando `display(gapminder)` o las primeras filas utilizando `gapminder.head()`.

5. ¿De qué clase es el objeto `gapminder`? ¿Qué variables tiene el dataset `gapminder` y de qué clase son?

6. ¿De cuántos países hay datos? Ayuda: averiguar qué hacen la función `unique()` y `nunique()`.

7. Explorar el tamaño del dataset gapminder usando la función `shape()`.

8. ¿Cuáles son las variables? Usar el comando `gapminder.columns.values`.

9. Extraer la información de Argentina, Uruguay y Chile y guardarla en un nuevo data frame `gm.sur`. Sugerencia: `np.isin`.

   ¿Cuántas filas tiene? ¿Cuál es el primero y el último año para el cuál existen datos de Argentina en gapminder?

Estadística descriptiva

10. Dar tres ejemplos de variables categóricas y numéricas.

11. En el dataset `gapminder` del paquete homónimo, una de las variables es el producto bruto per capita de los países (gdpPercap). ¿Es una variable categórica (nominal u ordinal) o numérica (discreta o continua)?

12. Supongamos que definimos una nueva variable que puede tomar los siguientes valores:

$$I.gdp = \begin{cases} 0, & \text{si } \texttt{gdpPercap} < 1600. \\ 1, & \text{si } 1600 \leq \texttt{gdpPercap} < 6600. \\ 2, & \text{en otro caso.} \end{cases}$$

    ¿La nueva variable es categórica (nominal u ordinal) o numérica (discreta o continua)? ¿Cambia la respuesta si la variable toma valores "bajo", "medio" y "alto" en lugar de 0, 1, 2?

13. Filtrar el dataset de `gapminder` para el año 2007. Luego, para ese año, calcular la cantidad de países en cada continente. Explorar las funciones aplicables a DataFrame: `groupBy`, `size`, `nunique`.

14. Con el mismo filtro que el ejercicio anterior (es decir, sólo para el año 2007), crear una variable que valga 1 si `gdpPercap` es mayor que 2000 dólares y 0 si no lo es. Luego crear una tabla de 2 filas y 5 columnas que calcule la cantidad de países donde $I = 0$ o $I = 1$ en cada continente.

15. Definir funciones que calculen la media y mediana de un vector de valores numéricos y la moda de un vector de valores categóricos. ¿Qué tiene que pasar para que existan dos modas?

16. Probar las funciones definidas con las variables numéricas y categóricas del dataset gapminder utilizando solo los datos del año 2007.

17. Graficar el producto bruto interno promedio en América en función del año.

18. Definir desvío estándar. ¿Por qué la diferencia en el numerador está elevada al cuadrado? Escribir una función de Python que calcule el desvío estándar. Comparar el resultado de usar la función `np.std()`.

19. Calcular el mínimo, el máximo y el desvío estandar de la expectativa de vida (`lifeExp`) entre países tomando sólo el dataset `gapminder` para el año 1952.

Archivos de datos

20. La biblioteca `Pandas` nos permite también trabajar con archivos de datos.

    (a) Leer el archivo `casos_coronavirus.csv`.

    (b) Graficar la curva de casos por día.

    (c) Graficar la curva de casos acumulados.

(d) Definir y como el logaritmo de la cantidad de casos acumulados y graficar y en función de la cantidad de días transcurridos.

(e) Tomando dos valores, estimar la pendiente de la recta para los datos a partir del dia 30.

Utilicen el siguiente código para leer el archivo.

```
import pandas as pd
datos = pd.read_csv("casos_coronavirus.csv")    # dataFrame
datos
```

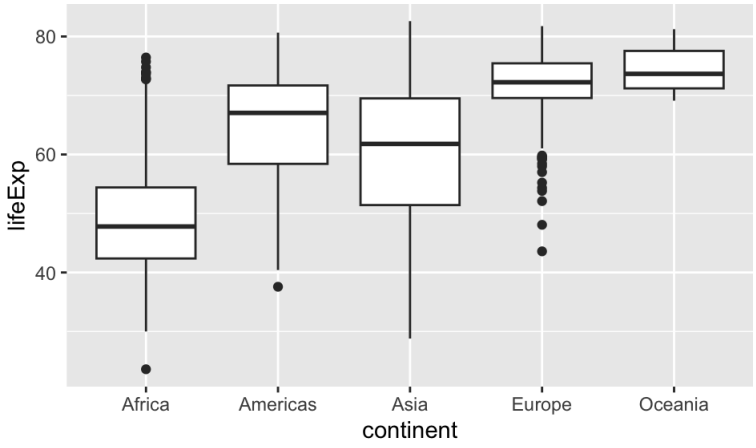
Visualización

21. Tenés datos de una encuesta realizada en distintas provincias de Argentina y querés saber cuántas personas respondieron a la encuesta en cada provincia. ¿Hacés un gráfico de líneas, de dispersión (scatter), histograma o un gráfico de barras (bar plot)? Hacé a mano en tu cuaderno cómo esperás que se vea el gráfico.

22. Estás estudiando la relación entre altura y peso de las personas. Tenés un data-set que tiene como variables la edad, sexo y peso de cada persona. Si querés describir estas variables por separado, ¿qué gráfico harías para cada una? ¿y si querés visualizar la relación entre peso y altura? Hacé a mano en tu cuaderno cómo esperás que se vea el gráfico.

23. Hacé un gráfico de barras que muestre la cantidad de países hay en cada continente según los datos de gapminder (recordar el ejercicio 1.4)

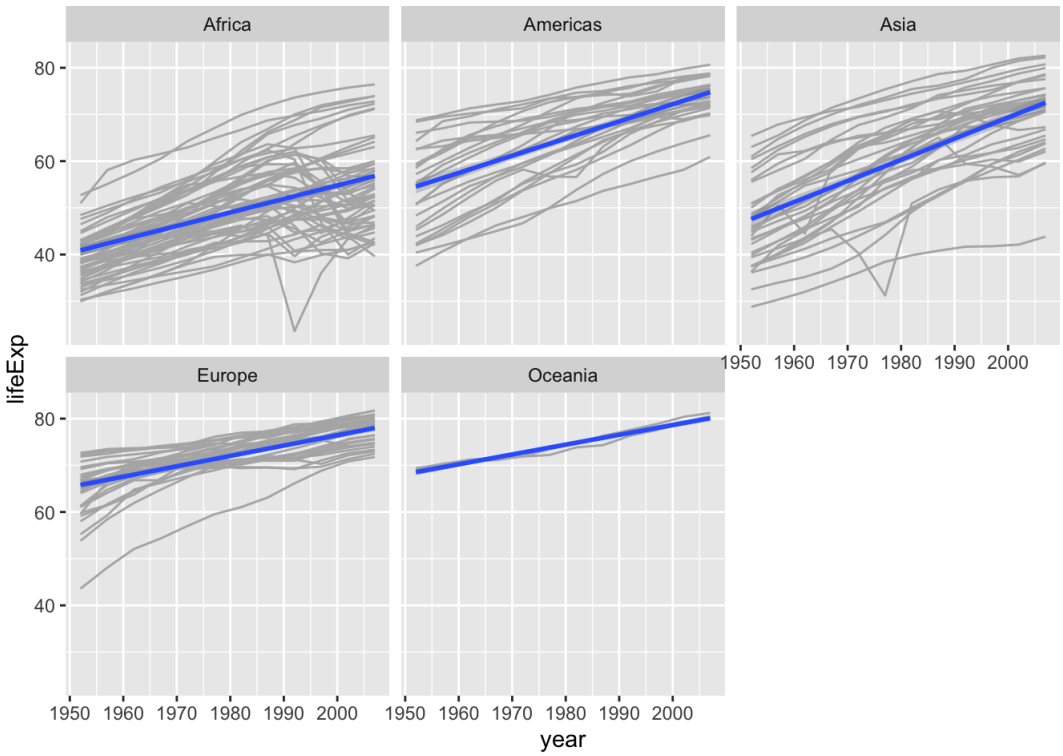
24. Querés investigar cómo varía la expectativa de vida entre los continentes. Para eso necesitás un gráfico como el siguiente:



Reproducí el gráfico de arriba reemplazando adecuadamente lo que falta en el siguiente código:

```
sns.boxplot(gapminder, x=COMPLETAR, y=COMPLETAR, order=sorted(COMPLETAR
    ))
```

25. Reproducir el siguiente gráfico:



26. (De acá en adelante, trabajar con el dataset `penguins` del paquete `palmerpenguins`). ¿Cuántas filas y columnas hay en el dataset penguins?

27. Dar una estadística descriptiva de la variable `bill_depth_mm`.

28. Hacer un `scatterplot` de `bill_depth_mm` (en el eje y) vs. `bill_length_mm` (en el eje x).

29. ¿Cuál sería un buen geom para ver la relación entre `species` y `bill_depth_mm`?

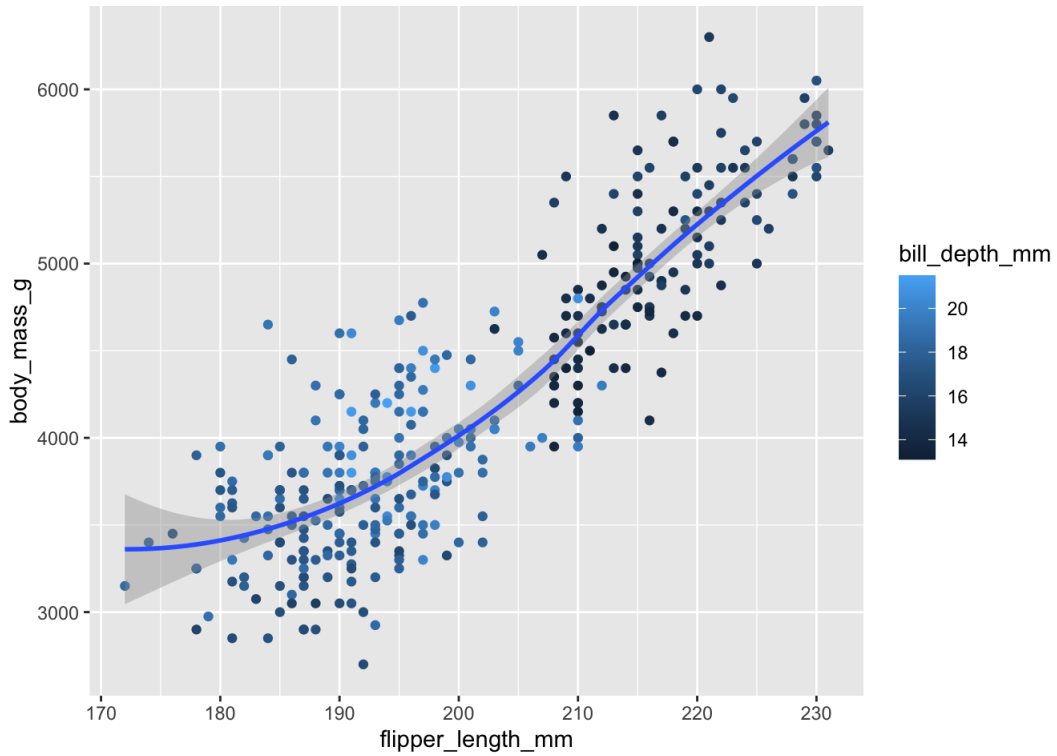
30. Corregir el siguiente código:

```
ggplot(data = penguins) +
  geom_point()
```

31. ¿Qué significa el argumento `na.rm` en `geom_point()`? Usando el dataset de `palmerpenguins` crear un gráfico donde se requiera usar ese argumento como `TRUE`.

32. Agregar un "caption" al gráfico de arriba. Ayuda: Mirar la documentación de `labs()`.

33. Recrear la siguiente visualización. ¿A qué aes debería mapearse `bill_depth_mm`? ¿El mapeo debe ser global o local?



34. Sin correr el código, predecir qué gráfico produce.

```
ggplot(data = penguins,
       mapping = aes(x = flipper_length_mm, y = body_mass_g, color =
   island) ) +
  geom_point() +
  geom_smooth(se = FALSE)
```

35. Sin correr el código, ¿estos dos gráficos van a ser iguales o diferentes? ¿Por qué?

```
# grafico 1
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point() +
  geom_smooth()

# grafico 2
ggplot() +
  geom_point(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  ) +
  geom_smooth(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  )
```

36. Sin correr el código, predecir qué gráfico produce.

```
ggplot(data = penguins,
       mapping = aes(x = flipper_length_mm, y = body_mass_g, color =
   island) ) +
  geom_point() +
  geom_smooth(se = FALSE)
```

37. Sin correr el código, ¿estos dos gráficos van a ser iguales o diferentes? ¿Por qué?

```
# grafico 1
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point() +
  geom_smooth()

# grafico 2
ggplot() +
  geom_point(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  ) +
  geom_smooth(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  )
```