Análisis estadístico de datos

Trabajo práctico: Test de hipótesis

Nombre: Alvaro Concha

Fecha: 18 de noviembre de 2021

La diabetes se diagnostica en base al nivel de glucosa en sangre. Según la base de datos de indios Pima la glucosa en sangre de personas sanas tiene una media $\mu_0 = 110.6\ \mathrm{mg/dl}$ y una desviación estándar $\sigma_0 = 24.8\ \mathrm{mg/dl}$. Para personas con diabetes la media es $\mu_1 = 142.3\ \mathrm{mg/dl}$ y la desviación estándar $\sigma_1 = 29.6\ \mathrm{mg/dl}$. Asumir que la glucosa en sangre para personas sanas y con diabetes sigue una distribución normal. Siguiendo las recomendaciones de la Organización Mundial de la Salud, a una persona se le diagnostica diabetes si su nivel de glucosa en sangre excede $126\ \mathrm{mg/dl}$. Graficar las distribuciones para la hipótesis nula y la alternativa. Indicar en la figura el valor crítico de la glucosa en sangre. Calcular la precisión del test de glucosa en sangre, la probabilidad que un paciente sano sea diagnosticado como diabético (falso positivo) y la probabilidad que un paciente diabético sea diagnosticado como sano (falso negativo). Si la prevalencia de la diabetes es del 35%, calcular aplicando el teorema de Bayes, la probabilidad que una persona con un test positivo efectivamente tenga diabetes.

Nota: Entregar informe del TP en formato PDF + código + figuras

In [ ]:
```
"""Instalar librerias necesarias"""
%pip install numpy scipy seaborn
```

```
Requirement already satisfied: numpy in /home/alvaro/.local/lib/python3.8/sit
e-packages (1.17.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-package
s (1.5.3)
Requirement already satisfied: seaborn in /usr/local/lib/python3.8/dist-packa
ges (0.11.0)
Requirement already satisfied: pandas>=0.23 in /home/alvaro/.local/lib/python
3.8/site-packages (from seaborn) (1.1.3)
Requirement already satisfied: matplotlib>=2.2 in /home/alvaro/.local/lib/pyt
hon3.8/site-packages (from seaborn) (3.4.3)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/lib/python3/dis
t-packages (from pandas>=0.23->seaborn) (2.7.3)
Requirement already satisfied: pytz>=2017.2 in /usr/lib/python3/dist-packages
(from pandas>=0.23->seaborn) (2019.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/
dist-packages (from matplotlib>=2.2->seaborn) (1.2.0)
Requirement already satisfied: pillow>=6.2.0 in /home/alvaro/.local/lib/pytho
n3.8/site-packages (from matplotlib>=2.2->seaborn) (8.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-
packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in /usr/local/lib/python3.8/d
ist-packages (from matplotlib>=2.2->seaborn) (2.4.7)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from cy
cler>=0.10->matplotlib>=2.2->seaborn) (1.14.0)
Note: you may need to restart the kernel to use updated packages.
```

Solución

Consideremos al nivel de azúcar en sangre de una persona como a una variable aleatoria X continua.

Consideremos además que una persona puede ser clasificada, según su estado de salud Θ, como sin diabetes (θ_0) o con diabetes (θ_1).

Es decir, el estado de salud de una persona adopta los valores

$$\Theta \in \{\theta_0, \theta_1\}.$$

Por su parte, la distribución de probabilidad del nivel de glucosa en sangre X, dado que una persona tiene diabetes, es

$$P(X = x|\Theta = \theta_1) = f_1(x),$$

y dado que no tiene diabetes es

$$P(X = x|\Theta = \theta_0) = f_0(x),$$

con f_i gaussiana de media μ_i y varianza σ_i^2, para $i \in \{0, 1\}$.

Siendo los valores de los parámetros

$$(\mu_0, \sigma_0) = (110.6, 24.8)\ \mathrm{mg/dl},$$

y

$$(\mu_1, \sigma_1) = (142.3, 29.6)\ \mathrm{mg/dl}.$$

Estas distribuciones de probabilidad se interpretan como las likelihoods de los parámetros

$$\mathcal{L}(\mu_i, \sigma_i; x) = f_i(x),$$

con $i \in \{0, 1\}$.

Experimentalmente, podemos medir el nivel de azúcar en sangre x_{obs} de una persona, y estimar su estado de salud $\hat{\theta}$ a partir de esa medición.

Ante una observación x_{obs} del nivel de azúcar en sangre de una persona, podemos considerar dos hipótesis mutuamente excluyentes.

Por un lado, la hipótesis nula (persona sin diabetes)

$$H_0 : x_{\mathrm{obs}} \text{ viene de } f_0 \implies \hat{\theta} = \theta_0,$$

y por otro lado, la alternativa (persona con diabetes)

$$H_1 : x_{\mathrm{obs}} \text{ viene de } f_1 \implies \hat{\theta} = \theta_1.$$

En principio, podemos calcular un valor crítico para el nivel de azúcar en sangre x_c, y usarlo como umbral para clasificar el estado de salud de una persona.

En este caso, clasificamos el estado de salud $\hat{\theta}$ de una persona como

$$\hat{\theta} = \begin{cases} \theta_0 & \text{si } x_{\text{obs}} < x_{\text{umbral}} \\ \theta_1 & \text{si } x_{\text{obs}} > x_{\text{umbral}} \end{cases},$$

con x_{umbral} algún valor umbral, y se puede asignar al caso $x_{\text{obs}} = x_{\text{umbral}}$ el estado $\hat{\theta} = \theta_1$, o utilizar otro criterio.

Podríamos utilizar como umbral al valor crítico crossover error rate (CER)

$$\begin{aligned} x_c &= \frac{\sigma_1}{\sigma_0 + \sigma_1}\mu_0 + \frac{\sigma_0}{\sigma_0 + \sigma_1}\mu_1 \\ &= 125.05 \text{ mg/dl}, \end{aligned}$$

que es el punto donde se iguala el error por falsos positivos (α_c, error tipo I) con el error por falsos negativos (β_c, error tipo II).

Este valor crítico x_c es un promedio, pesado por las desviaciones cruzadas, de las medias de las distribuciones.

El nivel de azúcar en sangre x_c no es necesariamente el valor donde se cruzan las curvas f_0 y f_1.

Luego, usando los parámetros del ejercicio, la probabilidad de falso positivo es

$$\begin{aligned} \alpha_c &= P(X > x_c | \Theta = \theta_0) \\ &= \int_{x_c}^{\infty} f_0(x)dx \\ &= \int_{\frac{x_c - \mu_0}{\sigma_0}}^{\infty} f(z)dz \\ &= 1 - \Phi\left(\frac{\mu_1 - \mu_0}{\sigma_0 + \sigma_1}\right) \\ &= \Phi\left(\frac{\mu_0 - \mu_1}{\sigma_0 + \sigma_1}\right) \\ &= 0.28, \end{aligned}$$

haciendo un cambio a una variable normal estándar

$$z = \frac{x - \mu_0}{\sigma_0},$$

siendo $f(z)$ una distribución normal estándar, y usando que

$$\frac{x_c - \mu_0}{\sigma_0} = \frac{\mu_1 - \mu_0}{\sigma_0 + \sigma_1},$$

y además

$$1 - \Phi(z) = \Phi(-z).$$

Haciendo un cálculo similar, la probabilidad de falso negativo es

$$\begin{aligned}\beta_c &= P(X < x_c | \Theta = \theta_1) \\ &= \int_{-\infty}^{x_c} f_1(x)dx \\ &= \int_{-\infty}^{\frac{x_c-\mu_1}{\sigma_1}} f(z)dz \\ &= \Phi\left(\frac{\mu_0-\mu_1}{\sigma_0+\sigma_1}\right) \\ &= 0.28,\end{aligned}$$

haciendo un cambio a una variable normal estándar

$$z = \frac{x-\mu_1}{\sigma_1},$$

siendo $f(z)$ nuevamente una distribución normal estándar, y usando que

$$\frac{x_c-\mu_1}{\sigma_1} = \frac{\mu_0-\mu_1}{\sigma_0+\sigma_1}.$$

Y efectivamente, usando x_c como umbral se igualan las probabilidades de error tipo I y tipo II

$$\alpha_c = \beta_c.$$

Por su parte, la precisión de un test estadístico con umbral de decisión x_c es

$$\begin{aligned}\text{Precisión}_c &= 1-\alpha_c \\ &= 0.72.\end{aligned}$$

Sin embargo, en este ejercicio nos piden utilizar como umbral al valor provisto por la OMS

$$x_{\text{OMS}} = 126 \text{ mg/dl}.$$

Usando x_{OMS} resulta una probabilidad de falso positivo

$$\begin{aligned}\alpha_{\text{OMS}} &= P(X > x_{\text{OMS}} | \Theta = \theta_0) \\ &= \int_{x_{\text{OMS}}}^{\infty} f_0(x)dx \\ &= \int_{\frac{x_{\text{OMS}}-\mu_0}{\sigma_0}}^{\infty} f(z)dz \\ &= 1-\Phi\left(\frac{x_{\text{OMS}}-\mu_0}{\sigma_0}\right) \\ &= \Phi\left(\frac{\mu_0-x_{\text{OMS}}}{\sigma_0}\right) \\ &= 0.27,\end{aligned}$$

y falso negativo

$$\begin{aligned}\beta_{\text{OMS}} &= P(X < x_{\text{OMS}}|\Theta = \theta_1) \\ &= \int_{-\infty}^{x_{\text{OMS}}} f_1(x)dx \\ &= \int_{-\infty}^{\frac{x_{\text{OMS}}-\mu_1}{\sigma_1}} f(z)dz \\ &= \Phi\left(\frac{x_{\text{OMS}}-\mu_1}{\sigma_1}\right) \\ &= 0.29,\end{aligned}$$

en este caso, el cociente entre falsos positivos y negativos es

$$\frac{\alpha_{\text{OMS}}}{\beta_{\text{OMS}}} = 0.92,$$

por lo que se diagnostican un poco menos casos falsos positivos que falsos negativos con este umbral, con una diferencia relativa

$$\frac{|\alpha_{\text{OMS}} - \beta_{\text{OMS}}|}{\frac{1}{2}(\alpha_{\text{OMS}} + \beta_{\text{OMS}})} = 0.08.$$

Por último, la precisión de este test es

$$\begin{aligned}\text{Precisión}_{\text{OMS}} &= 1 - \alpha_{\text{OMS}} \\ &= 0.73.\end{aligned}$$

Finalmente, usando el Teorema de Bayes, la probabilidad de que una persona con test positivo tenga diabetes es

$$P(\Theta = \theta_1|X > x_{\text{OMS}}) = \frac{P(X > x_{\text{OMS}}|\Theta = \theta_1)P(\Theta = \theta_1)}{P(X > x_{\text{OMS}})}.$$

reemplazando el valor de la likelihood

$$P(X > x_{\text{OMS}}|\Theta = \theta_1) = 1 - \beta_{\text{OMS}},$$

la prior

$$P(\Theta = \theta_1) = 0.35,$$

y la marginal

$$\begin{aligned}P(X > x_{\text{OMS}}) &= P(X > x_{\text{OMS}}|\Theta = \theta_0)P(\Theta = \theta_0) \\ &\quad + P(X > x_{\text{OMS}}|\Theta = \theta_1)P(\Theta = \theta_1) \\ &= \alpha_{\text{OMS}}(1 - P(\Theta = \theta_1)) \\ &\quad + (1 - \beta_{\text{OMS}})P(\Theta = \theta_1),\end{aligned}$$

resulta el posterior

$$P(\Theta = \theta_1|X > x_{\text{OMS}}) = 0.59,$$

por lo que la probabilidad de que una persona con test positivo tenga diabetes es 59%.

In [ ]:

```
"""Ejercicio 5"""
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
import seaborn as sns


def ej5():
    """Ejercicio 5."""
    mu_0 = 110.6
    sigma_0 = 24.8
    mu_1 = 142.3
    sigma_1 = 29.6
    x_lims = (0, 250)
    x = np.linspace(*x_lims, 10000)
    f_0 = norm(loc=mu_0, scale=sigma_0).pdf
    f_1 = norm(loc=mu_1, scale=sigma_1).pdf
    phi = norm.cdf
    x_c = (sigma_1 * mu_0 + sigma_0 * mu_1) / (sigma_0 + sigma_1)
    alpha_c = phi((mu_0 - x_c) / sigma_0)
    beta_c = phi((x_c - mu_1) / sigma_1)
    precision_c = 1 - alpha_c
    x_OMS = 126.0
    alpha_OMS = phi((mu_0 - x_OMS) / sigma_0)
    beta_OMS = phi((x_OMS - mu_1) / sigma_1)
    cociente_OMS = alpha_OMS / beta_OMS
    dif_rel_OMS = (
        np.abs(alpha_OMS - beta_OMS)
        / (0.5 * (alpha_OMS + beta_OMS))
    )
    precision_OMS = 1 - alpha_OMS
    prior = 0.35
    likelihood = 1 - beta_OMS
    marginal = alpha_OMS * (1 - prior) + (1 - beta_OMS) * prior
    posterior = likelihood * prior / marginal

    print(f"Valor critico:\tx_c = {x_c:.2f} mg/dl")
    print(f"Error tipo I:\talpha_c = {alpha_c:.2f}")
    print(f"Error tipo II:\tbeta_c = {beta_c:.2f}")
    print(f"Precision:\tPrecision_c = {precision_c:.2f}")
    print()
    print(f"Valor OMS:\tx_OMS = {x_OMS:.2f} mg/dl")
    print(f"Error tipo I:\talpha_OMS = {alpha_OMS:.2f}")
    print(f"Error tipo II:\tbeta_OMS = {beta_OMS:.2f}")
    print(f"Cociente:\talpha_OMS / beta_OMS = {cociente_OMS:.2f}")
    print(f"Dif. relativa:\t{dif_rel_OMS:.2f}")
    print(f"Precision:\tPrecision_OMS = {precision_OMS:.2f}")
    print(f"Posterior:\t{posterior:.2f}")

    plt.figure(figsize=(8, 8))
    sns.set_context("paper", font_scale=1.5)
    plt.plot(x, f_0(x), label=r"$f_0$")
    plt.plot(x, f_1(x), label=r"$f_1$")
    y_lims = plt.gca().get_ylim()
    plt.vlines(x_c, *y_lims, ls="-", colors="k", label=r"$x_c$")
    plt.vlines(
        x_OMS, *y_lims, ls="--", colors="k",
        label=r"$x_{\mathrm{OMS}}$"
    )
```

```
    plt.legend()
    plt.title("Umbral de decisión")
    plt.xlabel(r"$x$")
    plt.ylabel(r"Densidad de probabilidad")
    plt.savefig("tp6_ej5.pdf", bbox_inches="tight")
    plt.show()
    plt.close()


if __name__ == "__main__":
    ej5()
```

```
Valor critico:  x_c = 125.05 mg/dl
Error tipo I:   alpha_c = 0.28
Error tipo II:  beta_c = 0.28
Precision:      Precision_c = 0.72

Valor OMS:      x_OMS = 126.00 mg/dl
Error tipo I:   alpha_OMS = 0.27
Error tipo II:  beta_OMS = 0.29
Cociente:       alpha_OMS / beta_OMS = 0.92
Dif. relativa:  0.08
Precision:      Precision_OMS = 0.73
Posterior:      0.59
```