Encuesta_Social_General

June 3, 2024

Encuesta Social General

Distribuciones y sus aplicaciones

Autor: Jesús Antonio Santillán Hernández

[ ]:
```
# En este archivo trabajaremos con distribuciones aplicando los siguiente␣
 ↪conceptos:
#Funcion distribución de probabilidad (PMF), Función de distribución␣
 ↪acumulativa (CDF) y Estimación de densidad de núcleo (KDE).
#Para esto trabajaremos con una base de datos correspondiente a una encuesta␣
 ↪realizada en Estados Unidos, de la cual
#se recabaron los siguientes datos:
#      column(1)    numeric                    YEAR   %20f  "Gss year for this␣
 ↪respondent"                        "
#      column(2)    numeric                      ID   %20f  "Respondent id␣
 ↪number"
#      column(3)    numeric                     AGE   %20f  "Age of respondent"
#      column(4)    numeric                    EDUC   %20f  "Highest year of␣
 ↪school completed"
#      column(5)    numeric                     SEX   %20f  "Respondents sex"
#      column(6)    numeric                  GUNLAW   %20f  "Favor or oppose gun␣
 ↪permits"
#      column(7)    numeric                   GRASS   %20f  "Should marijuana be␣
 ↪made legal"
#      column(8)      float                 REALINC   %20f  "Family␣
 ↪income in constant $"
#El archivo el cual contiene los datos se llama GSS.cvs
```

[ ]:
```
######################LIBRERIAS#########################
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm
import pandas as pd
import seaborn as sns
from empiricaldist import Pmf
from empiricaldist import Cdf
```

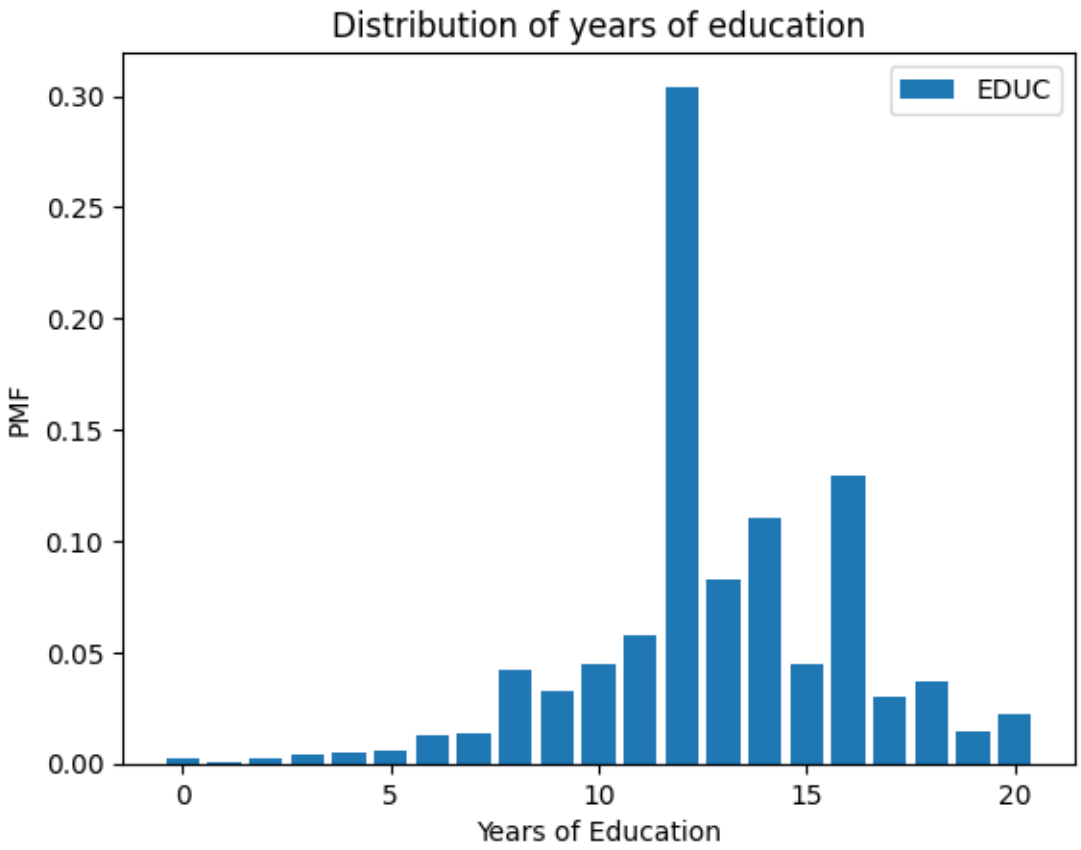
[ ]:
```
#Se realiza la importanción de datos desde nuestro archivo.
gss=pd.read_excel("GSS.xlsx",header=0)
print(type(gss))
print(gss.head())
```

```
<class 'pandas.core.frame.DataFrame'>
   YEAR  ID  AGE  EDUC  SEX  GUNLAW GRASS  REALINC
0  1972   1   23    16    2       1     0  18951.0
1  1972   2   70    10    1       1     0  24366.0
2  1972   3   48    12    2       1     0  24366.0
3  1972   4   27    17    2       1     0  30458.0
4  1972   5   61    12    2       1     0  50763.0
```

[ ]:
```
########################(PMF)##############################
#La función de masa de probabilidad, es una función que proporciona la␣
 ↪probabilidad de que una variable discreta tome un valor específico
#Para aplicar la función de masa de probabilidad se utilizará la columna de de␣
 ↪EDUC o educación.
#En este caso se tienen valores de 1 a 20 (variables aleatorias discretas)
#los cuales representan los grados o niveles de educación.
#Sin embargo, tambien hay valores de cantidad 98 y 99, los cuales significan no␣
 ↪lo sé y sin respuesta.
```

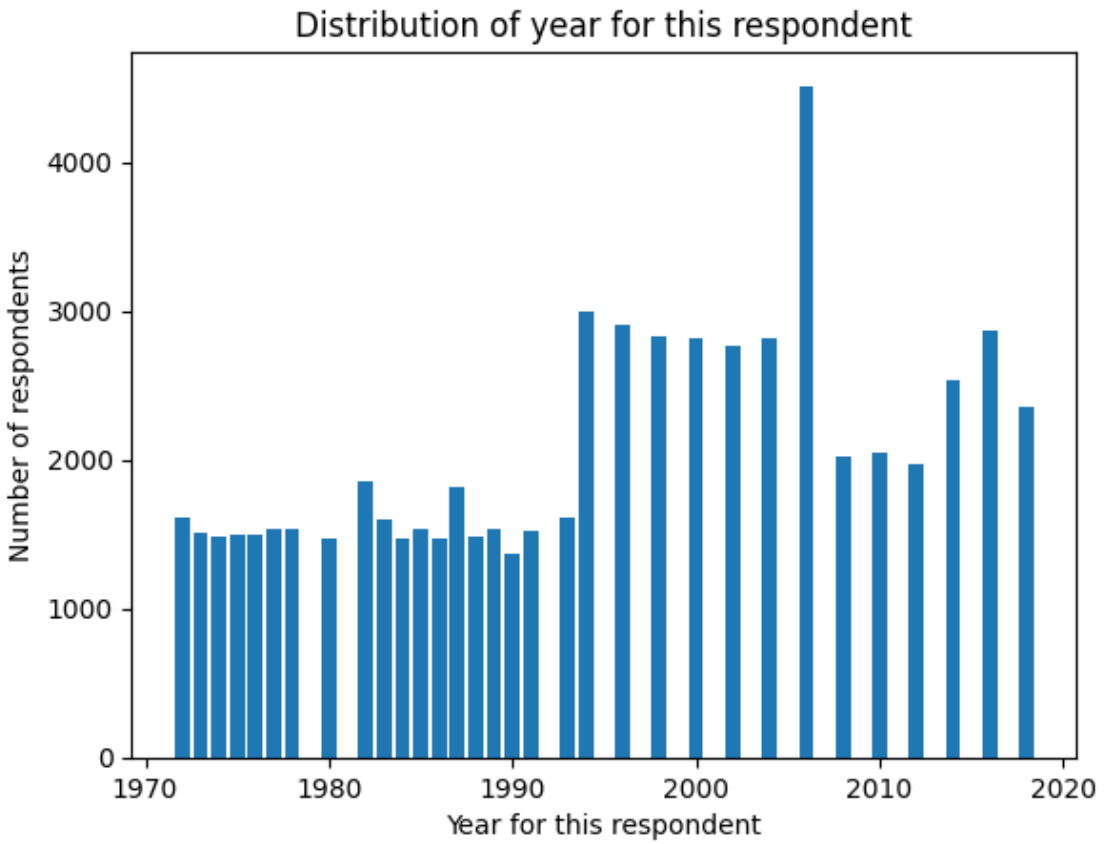
[ ]:
```
# Para comenzar el procesamiento de datos primero hay que quitar el tipo de␣
 ↪valores 98 y 99 ya que no aportan información.
#Como resultado tenemos un nuevo dataframe en Serie (Una sola columna)
educ=gss["EDUC"].replace([98,99],np.nan)
```

[ ]:
```
#Se procede a aplicar la normalización de los datos del dataframe:
PMF_educ_norm=Pmf.from_seq(educ,normalize=True)
PMF_educ_norm.bar(label="EDUC")
plt.xlabel("Years of Education")
plt.ylabel("PMF")
plt.title("Distribution of years of education")
plt.legend()
plt.show()
```



[ ]:
```
#En este caso los encuestados desmostrarón tener una educación en su mayoría␣
 ↪hasta el 12vo grado, teniendo picos tambien en
# 14 y 16 lo cuales representan dos y tres años de universidad, esto nos␣
 ↪permite e permite tener una visión clara de cómo se distribuyen estos␣
 ↪niveles dentro de la población estudiada.
#Esta información es crucial para entender la estructura educativa de la␣
 ↪población y para hacer comparaciones y predicciones sobre tendencias futuras.
```

[ ]:
```
#Ahora se realizará el mismo procedimiento pero para los años en los cuales se␣
 ↪respondió la encuesta:
Year=gss["YEAR"]
PMF_Year=Pmf.from_seq(Year,normalize=False)
PMF_Year.bar(label=False)
plt.xlabel("Year for this respondent")
plt.ylabel("Number of respondents")
plt.title("Distribution of year for this respondent")
plt.show()
```



[ ]:
```
#En este caso se representan los años en los cuales se fueron obteniendo los␣
 ↪datos, siendo el 2006 el año en el cual hubo
#mayor cantidad de personas que participaron en la encuesta, esto en␣
 ↪comparación con el nivel educativo nos permite tener una idea sobre como se␣
 ↪distribuye la educación
#para ciertos años en particular
```

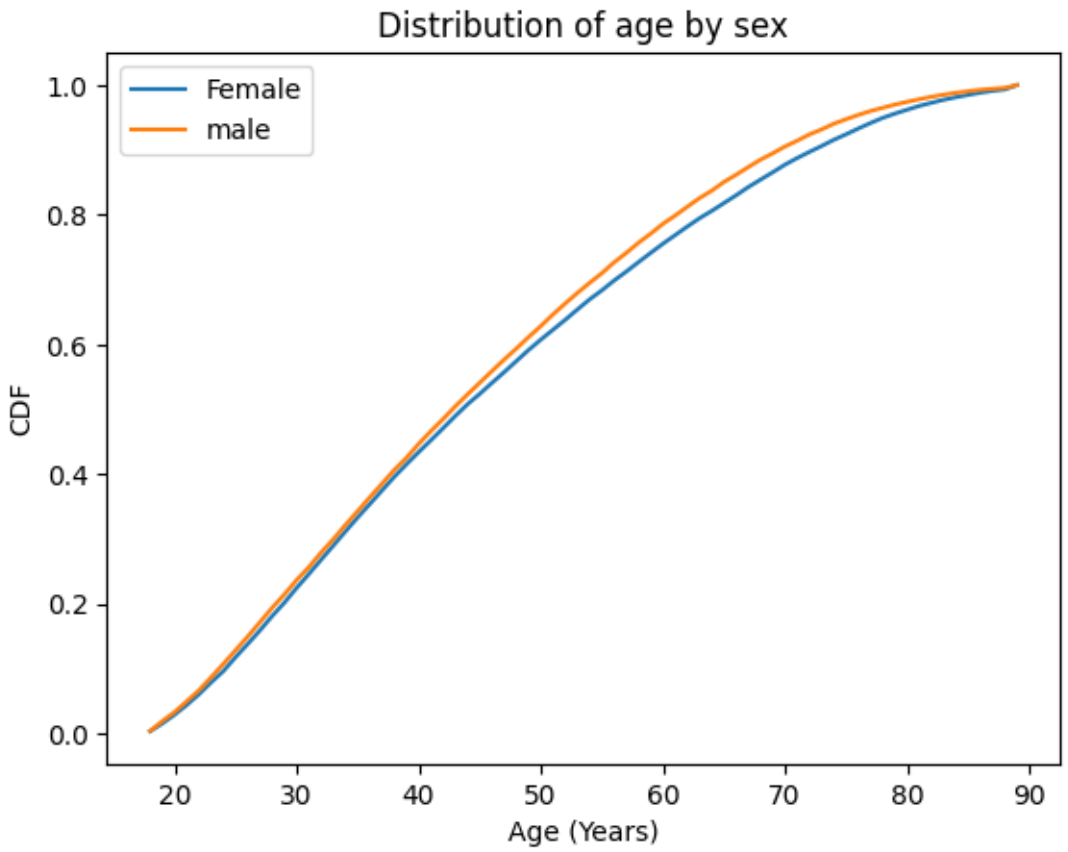
[ ]:
```
########################(CDF)##############################
#En este apartado se utilizaran las propiedades de la Función de distribución␣
 ↪acumulada.
#La cual es una función que describe la probabilidad de que una variable␣
 ↪aleatoria sea menor o igual que un valor específico.
#En este caso realizaremos la distribución de edad de mujeres y hombres, que␣
 ↪participaron en la encuenta:
```

[ ]:
```
#Primero realizaremos una limpieza rapida de los datos, eliminando las edades␣
 ↪correspondientes a 98 y 99 años
#los cuales son códigos que indican 98=no lo sé , 99=sin respuesta:
age=gss["AGE"].replace([98,99],np.nan)
```

[ ]:
```
#Ahora creamos los valores para mujeres y hombres:
female=(gss["SEX"]==2)
male=(gss["SEX"]==1)
```

[ ]:
```
#Ahora selecionamos las edades para cada sexo:
female_age=age[female]
male_age=age[male]
```

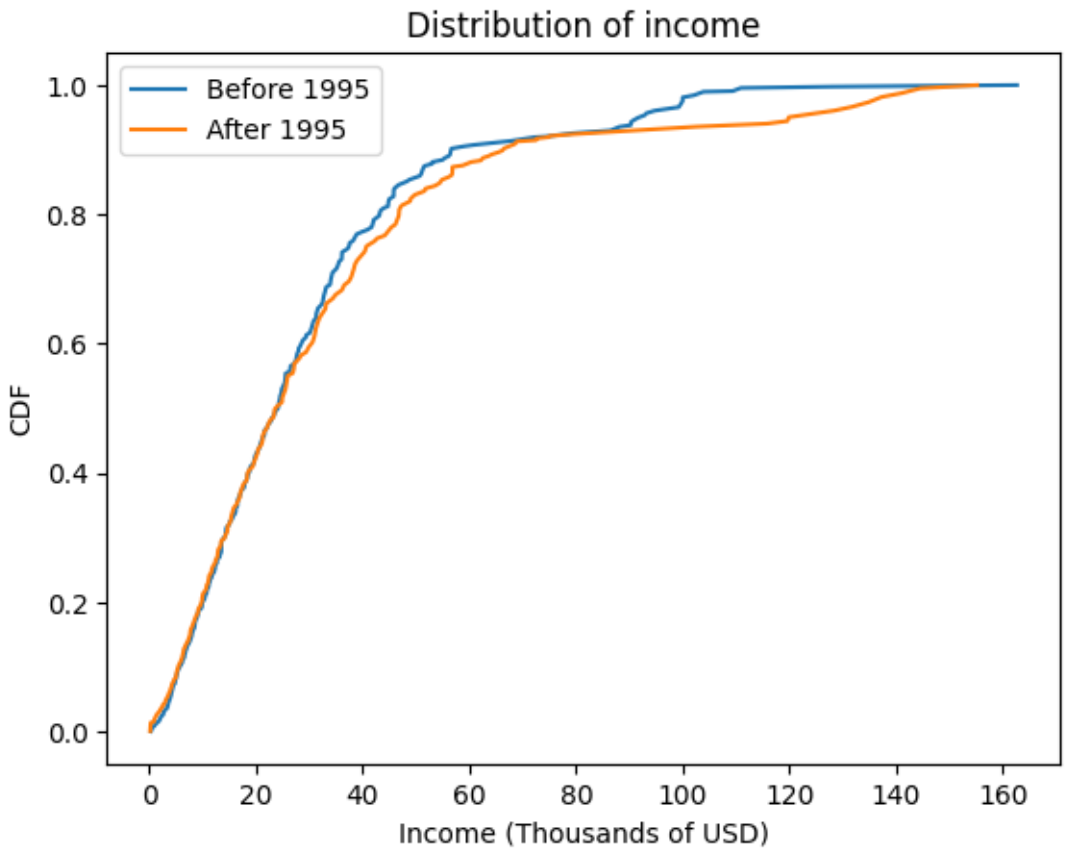
[ ]:
```
#Se procede a realizar la gráfica utilizando el método de CDF:
cdf_female_age=Cdf.from_seq(female_age)
cdf_female_age.plot(label='Female')

cdf_male_age=Cdf.from_seq(male_age)
cdf_male_age.plot(label='male')

plt.xlabel("Age (Years)")
plt.ylabel("CDF")
plt.title("Distribution of age by sex")
plt.legend()
plt.show()
```



[ ]:
```
#En este caso, las líneas están muy juntas hasta los 40 años; después de eso,␣
 ↪el CDF es más alto para los hombres que para las mujeres.
#Una forma de interpretar la diferencia es que la fracción de hombres menores␣
 ↪de una edad determinada es generalmente mayor que la fracción de mujeres␣
 ↪menores de la misma edad.
#Por ejemplo, alrededor del 79% de los hombres tienen 60 años o menos, en␣
 ↪comparación con el 76% de las mujeres.
```

[ ]:
```
#Tambien se puede aplicar este tipo de distribución para observar el ingreso de␣
 ↪los hogares y realizar una comparación de estos antes y despues de 1995.
#primero realizamos una serie booleana para seleccionar a los encuestados␣
 ↪entrevistados  antes y después de 1995:
pre95=(gss["YEAR"]<1995)
post95=(gss["YEAR"]>1995)
income=gss['REALINC'].replace(0,np.nan)/1000
```

[ ]:
```
#Ahora graficaremos el CDF:
Cdf.from_seq(income[pre95]).plot(label="Before 1995")
Cdf.from_seq(income[post95]).plot(label="After 1995")
plt.xlabel("Income (Thousands of USD)")
plt.ylabel("CDF")
plt.title("Distribution of income")
plt.legend()
plt.show()
```

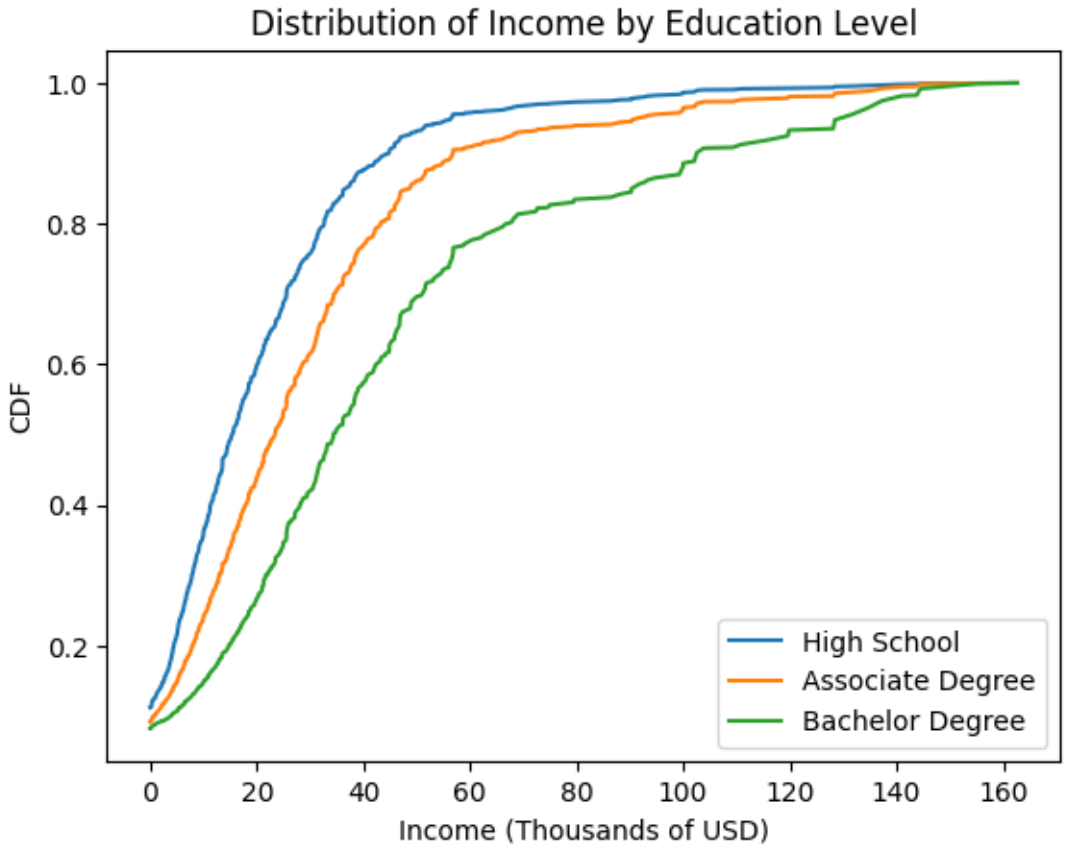


[ ]:
```
#Por debajo de 30.000 dólares, los CDF son casi idénticos;
#encima de eso, podemos ver que la distribución posterior a 1995 se desplaza␣
 ↪hacia la derecha.
#En otras palabras, la fracción de personas con ingresos altos es␣
 ↪aproximadamente la misma, pero los ingresos de las personas con ingresos␣
 ↪altos han aumentado.
```

[ ]:
```
#Por último compararemos los ingresos para diferentes niveles de educación:
#Primero establecemos las series booleanas para los niveles educativos:
high=(gss["EDUC"]<=12)
assc=(gss["EDUC"]>12) & (gss["EDUC"]<16 )
bach=(16<=gss["EDUC"])
```

[ ]:
```
#Seleccionamos los ingresos de cada nivel educativo:
high_income=gss.loc[high,"REALINC"]/1000
assc_income=gss.loc[assc,"REALINC"]/1000
bach_income=gss.loc[bach,"REALINC"]/1000
```

[ ]:
```
#Ahora graficaremos el CDF:
Cdf.from_seq(high_income).plot(label="High School")
Cdf.from_seq(assc_income).plot(label="Associate Degree")
Cdf.from_seq(bach_income).plot(label="Bachelor Degree")

plt.xlabel("Income (Thousands of USD)")
plt.ylabel("CDF")
plt.title("Distribution of Income by Education Level")
plt.legend()
plt.show()
```



[ ]:
```
#Inicialmente las tres curvas comienzan con una pendiente muy pronunciada, lo␣
 ↪que significa que la probabilidad acumulativa
#aumenta rápidamente a medida que avanzamos a lo largo del eje x.
#Esto indica que una proporción relativamente grande de los datos está␣
 ↪concentrada en donde la pendiente es más grande.
#Para el nivel "High School" lo datos se encuentran concentrados en valores de␣
 ↪0 a 40 mil dolares.
```

```
#Para el nivel "Associate Degree" lo datos se encuentran concentrados en␣
 ↪valores de 0 a 70 mil dolares.
#Para el nivel "Bachelor Degree" lo datos se encuentran concentrados en valores␣
 ↪de 0 a 120 mil dolares.
```

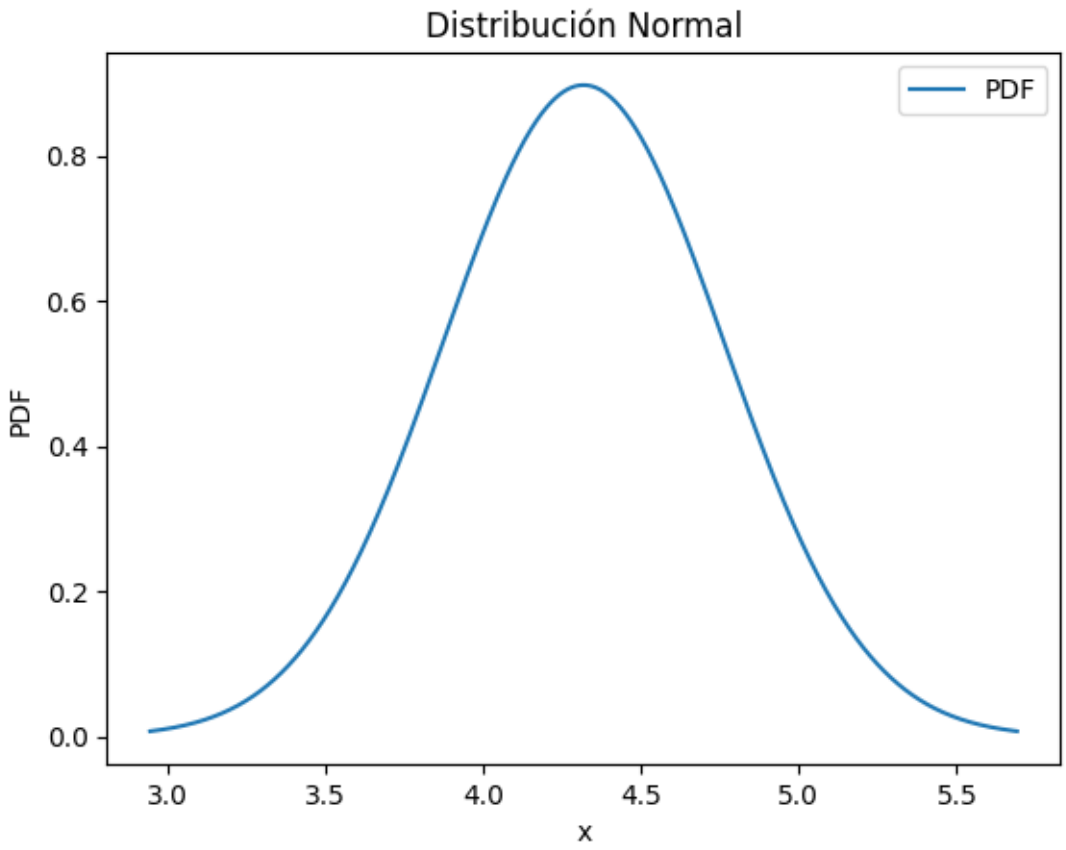
[ ]:
```
########################(KDE)##############################
#KDE es una técnica no paramétrica para estimar la función de densidad de␣
 ↪probabilidad de una variable aleatoria continua.
#Para esto trabajaremos con los ingresos de los participantes de la encuesta:
realinc=gss['REALINC'].replace(0,np.nan)
log_realinc=np.log10(realinc)
```

[ ]:
```
#Obtenemos la media y la desviación estandar de la serie de datos "log_realinc":
media_realinclog=np.mean(log_realinc)
std_realinclog=np.std(log_realinc)
print("Media de los ingresos transformados logaritmicamente:", media_realinclog)
print("Desviación estandar de los ingresos transformados logaritmicamente:",␣
 ↪std_realinclog)
```

```
Media de los ingresos transformados logaritmicamente: 4.3175817560161915
Desviación estandar de los ingresos transformados logaritmicamente:
0.44443518133262117
```

[ ]:
```
#Se crea una distribucción normal con estos datos:
norm_distribution=norm(media_realinclog,std_realinclog)
x = np.linspace(norm_distribution.ppf(0.001), norm_distribution.ppf(0.999),␣
 ↪1000)
```

[ ]:
```
# Se calculan los valores de la función de densidad de probabilidad (PDF) para␣
 ↪los valores de x
pdf = norm_distribution.pdf(x)
plt.plot(x, pdf, label='PDF')
plt.xlabel('x')
plt.ylabel('PDF')
plt.title('Distribución Normal')
plt.legend()
plt.show()
```



[ ]:
```
#Al aplicar KDE a los ingresos de los participantes en la encuesta se obtene␣
↪una visión más clara y detallada de cómo se distribuyen estos ingresos en la␣
↪población.
#Siendo los ingresos más altos aquellos que van de 4 a 4.5 mil dolares.
```