01_Estadistica_I_Descriptiva

January 7, 2026

Contenido

1 ESTADÍSTICA I PARA DATA SCIENCE: DESCRIPTIVA

1.1 INSTALACIÓN

[ ]:
```
# conda install pandas
# conda install scipy
# conda install seaborn
# conda install -c conda-forge statsmodels
```

1.2 PREPARACION

[2]:
```
#Carga de paquetes y datos
import pandas as pd
import numpy as np
import statistics
import scipy as sp
import seaborn as sns
import random
import math
from statsmodels.stats.proportion import proportions_ztest

df = sns.load_dataset('tips')
```

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   total_bill  244 non-null    float64
 1   tip         244 non-null    float64
 2   sex         244 non-null    category
 3   smoker      244 non-null    category
 4   day         244 non-null    category
 5   time        244 non-null    category
 6   size        244 non-null    int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.4 KB
```

[ ]:
```
df.head()
```

1.3 CONCEPTOS BÁSICOS

1.3.1 Descriptiva Vs Inferencial

Son las 2 grandes ramas de la estadística.

Y por tanto así también vamos a organizar este módulo.

La **DESCRIPTIVA** se centra en describir los datos que estamos analizando. Resumirlos, visualizarlos y sacar conclusiones.

Es la que más se usa en la práctica, y son el tipo de operaciones más básicas e imprescindibles que vamos a estar haciendo en Data Science.

Simplemente usando las técnicas que vamos a aprender en este apartado ya podremos hacer el 80% de los análisis que normalmente se hacen en contextos empresariales.

La **INFERENCIAL** se pregunta "vale, pero ¿esas conclusiones que hemos detectado en la descriptiva realmente son robustas y estables y pueden ser extrapoladas?".

Parte del concepto teórico de que realmente siempre vamos a estar analizando muestras, y no toda la población (más sobre esto en el siguiente apartado).

Por lo que son necesarias técnicas que nos garanticen estadísticamente que esas conclusiones no son fruto del azar o de nuestra muestra en concreto.

Esta parte está muy infrautilizada en la práctica del día a día, ya que pocas veces se usan estas técnicas para contrastar si los resultados de un análisis, estudio o informe son realmente estadísticamente significativos.

Sin embargo como Data Scientist, conocerlas te va a servir tanto para garantizar que la calidad de tu trabajo es la correcta como para poder identificar falsas conclusiones que te intenten "colar".

Ahora bien, debes saber un hecho importante que afecta a todo lo que veamos en la parte inferencial.

Todas las técnicas y herramientas que veremos están basadas en que la muestra ha sido obtenida ALEATORIAMENTE y sin ningún tipo de sesgo.

Para garantizar esto la teoría nos dice que debemos hacer un correcto "diseño de experimento" que en la empresa lógicamente no se puede hacer casi nunca.

Así te corresponde como analista el evaluar el grado de "aleatoridad" y no sesgo que tienen tus datos y hasta qué punto puedes apoyarte en los recursos de la estadística inferencial.

1.3.2 Población Vs Muestra

Población:

Cuando trabajamos con todos los elementos de interés.

Se denota por N.

Los números que la describen se llaman parámetros.

Muestra:

Cuando trabajamos con algunos de los elementos de interés.

Se denota por n.

Los números que la describen se llaman estadísticos o estimadores.

1.3.3 Escalas de medida y tipos de variables

Este es uno de los puntos más importantes que debes conocer y sin embargo pocas veces es tratado en las formaciones de Data Science.

La estadística nos dice que hay 4 grandes escalas de medida:

1. **Nominal**: Tiene la propiedad de identidad (igual/diferente) pero NO la de orden. Ejemplos: sexo, colores, …
2. **Ordinal**: Añade la propiedad de orden, pero NO tiene una igual distancia entre elementos. Ejemplos: clase social, nivel formativo, …
3. **Intervalo (o discreta)**: Añade la propiedad de igualdad de distancias, pero no permite infinitos valores entre sus elementos. Ejemplos: número de hijos, número de siniestros, …

4. **Razón (o continua)**: Puede tener (teóricamente) infinitos valores entre sus elementos. Ejemplos: precio de un producto, distancias, …

A partir de las escalas se derivan los tipos de variables, que pueden ser diferente en cada implementación (por ej en cada lenguaje de programación), pero casi siempre podremos hacer una asignación aproximada a una escala.

Por ejemplo los enteros serían variables discretas, los reales continuas, los objetos categóricas, etc.

A un nivel más elevado podemos agruparlos en dos grupos que son super relevantes, ya que estos dos grupos van a determinar qué técnicas analíticas o qué gráficos podemos aplicar en cada uno de ellos:

- **Categóricas (o cualitativas)**: incluyen nominales y ordinales
- **Numéricas (o cuantitativas)**: incluyen discretas y razón

Como este curso tiene un caracter eminentemente aplicado vamos a usar esta división para ver qué tipo de análisis podemos hacer en cada una.

1.4 ESTADÍSTICA DESCRIPTIVA

Como ya habíamos anticipado la misión de la descriptiva es resumir, graficar y analizar los datos para intentar encontrar patrones de interés.

A grandes rasgos tenemos dos tipos de técnicas a nuestra disposición:

- **Cálculos**: aplicar algún tipo de estadístico o análisis y obtener uno o varios resultados numéricos (indicadores, tablas, porcentajes, etc.)
- **Gráficos**: resumir y mostrar la información de forma gráfica para detectar los patrones más fácilmente

Y también como decíamos más arriba el tipo de cálculos y gráficos que podemos hacer van a depender directamente del tipo de variable. Así que los vamos a dividir entre los de variables categóricas y los de variables cuantitativas.

1.4.1 Análisis de variables categóricas

Con cálculos

Conteos y frecuencias Se entiende por frecuencia el conteo del valor de cada variable.

[4]:
```
df.smoker.value_counts()
```

[4]:
```
smoker
No     151
Yes     93
Name: count, dtype: int64
```

Moda Es el valor más frecuente de una variable.

[5]:
```
#Moda
numeros = np.random.randint(0,11,1000)
```

```
print(pd.Series(numeros).value_counts())
statistics.mode(numeros)
```

```
1     106
8     103
10     93
2      91
6      91
9      88
0      87
5      87
3      87
7      86
4      81
Name: count, dtype: int64
```

[5]:
```
np.int32(1)
```

Tablas cruzadas Consiste en cruzar dos o más variables para analizar la frecuencia.

Suelen usarse o en absoluto o en porcentaje.

[6]:
```
#En absoluto
pd.crosstab(df.sex,df.smoker,margins = True)
```

[6]:

smoker	Yes	No	All
sex			
Male	60	97	157
Female	33	54	87
All	93	151	244

[7]:
```
#En porcentaje
pd.crosstab(df.sex,df.smoker,margins = True,normalize = 'all')
```

[7]:

smoker	Yes	No	All
sex			
Male	0.245902	0.397541	0.643443
Female	0.135246	0.221311	0.356557
All	0.381148	0.618852	1.000000

Chi Cuadrado Al hacer la tabla cruzada hemos visto que hay diferencias en si es fumador o no en base a si es hombre o mujer.

Pero, ¿estas diferencias son suficientes para tomar el resultado como una conclusión?. O quizá están dentro del margen esperable por efecto del muestreo y por tanto no podemos concluir nada.

Es lo que se llama ver si una conclusión es estadísticamente significativa. Y veremos el concepto en profunidad en la parte de estadística inferencial.

Pero en un análisis de tablas cruzadas como este podemos hacerlo con el estadístico chi-cuadrado.

Que parte de una hipótesis nula de que no hay relación entre las variables, es decir que son independientes y por tanto fumar o no, no depende del sexo, y la contrasta contra los datos obtenidos.

Si el pvalor es menor o igual que 0.05 entonces rechaza esa hipótesis nula, y significa que los datos si apoyan que hay diferencias significativas y por tanto podemos concluir que sí hay diferencias en cuanto al fumar entre hombres y mujeres.

[ ]:
```
#Chi-cuadrado
tabla = pd.crosstab(df.sex,df.smoker)
chi, pvalor, gl, experado = sp.stats.chi2_contingency(tabla)
print(chi)
print(pvalor)
#En este caso el valor es mayor que 0.05, por tanto no podemos rechazar la␣
 ↪hipótesis nula,
#y por tanto no hay diferencias significativas entre hombres y mujeres
```

Con gráficos Los gráficos más frecuentes que podremos hacer con variables categóricas son:

- Gráficos de barras: en vertical, horizonal, apilados, etc.
- Gráficos de sectores (o de tarta)

[ ]:
```
#Ejemplo de gráfico de barras
df.smoker.value_counts().plot(kind = 'bar');
```

[ ]:
```
#Ejemplo de gráfico de sectores
df.smoker.value_counts().plot(kind = 'pie');
```

1.4.2 Análisis de variables cuantitativas

Con cálculos Se diferencian dos tipos de medida:

- De **centralización**: que intentan resumir la información de la variable en un sólo dato que pueda ser lo más representativo posible. Aquí están medidas como las medias, mediana o moda.
- De **dispersión**: que representan el grado de variabilidad existente en la variable. Normalmente cuanto más baja sea la medida de dispersión más representativa será la medida de centralización. Aquí entran métricas como la varianza, la desviación típica o el coeficiente de variación.

También veremos las métricas para cuantificar la relación existente entre dos variables cuantitativas: las correlaciones.

Medias La media normal se llama **media aritmétrica**, y será la que usaremos en la mayoría de los casos.

La **media winsorizada** se usa cuando hay valores atípicos que pueden sesgar el valor de la media.

Para hacer la media de porcentajes o tasas hay que usar la **media geométrica**.

Para hacer la media de medias hay que usar la **media armónica**.

[ ]:
```
#Media arimética
statistics.mean([3,4,5])
```

[ ]:
```
#Media winsorizada
numeros = np.array([3,4,5,6,999997])
print(numeros)
print(numeros.mean())

#Winsorizar sustituye los valores fuera de los límites por el último valor
winsorizados = sp.stats.mstats.winsorize(numeros, limits = [0, 0.2])
print(winsorizados)
print(winsorizados.mean())
```

[ ]:
```
#Media geométrica (para porcentajes)
statistics.geometric_mean([0.3,0.4,0.5])
```

[ ]:
```
#Comparamos con lo que hubiera salido en una media arimética
statistics.mean([0.3,0.4,0.5])
```

[ ]:
```
#Media armónica (para media de medias)
#Suponemos que los datos de esta lista son medias, ej velocidad media de coches␣
 ↪en 3 calles
statistics.harmonic_mean([30.3,40.8,50.1])
```

[ ]:
```
#Comparamos con lo que hubiera salido en una media arimética
statistics.mean([30.3,40.8,50.1])
```

Mediana Si se ordenan todos los valores de la variable en orden ascendente o descendente la medianta es el valor de la variable correpondiente al elemento que ocupa la posición central, es decir, el que está en el 50%.

La mediana es una medida de centralización más recomendable que la media cuando tenemos distribuciones que no son normales, o cuando tenemos atípicos.

[ ]:
```
#Mediana
numeros = np.random.randint(0,11,11)
numeros_ord = np.sort(numeros)
print(numeros_ord)
print(statistics.median(numeros_ord))
```

Varianza La varianza es el resultado de restar la media a cada valor de la variable, elevarlo al cuadrado (para evitar los negativos), sumarlo todo, y dividir el resultado por el número de datos.

Es una medida de dispersión, porque será mayor cuanto más lejos estén el global de los valores con respecto a la media.

[ ]:
```
#Vamos a calcularla manualmente para entenderla
var1 = np.random.randint(0,11,20)
```

```
print(var1)
media = var1.mean()
print(media)
suma_cuadrados = sum((var1 - media) ** 2)
print(suma_cuadrados)
varianza = suma_cuadrados / (len(var1))
print(varianza)
```

[ ]:
```
#En la práctica usaremos una función para calcularla directamente
var1.var()
```

Desviación típica El problema con la varianza es que hemos tenido que elevar al cuadrado para quitarnos los negativos.

Entonces el dato obtenido ya no está en la misma escala que la media por ejemplo para poder compararlos.

Como solución se hace la raiz cuadrada a la varianza para volver a traerla a la escala, y es lo que se llama desviación típica.

[ ]:
```
#Desviación típica
var1.std()
```

Coeficiente de variación Como decíamos más arriba cuando más baja sea la dispersión con respecto a la media será mejor porque nos indica que esa media es más representativa del total de los datos de la variable.

Ahora que ya tenemos una medida, la desviación típica, que sí está en la misma escala que la media, podemos compararlas.

Es lo que se llama el coeficiente de variación, y consiste en dividir la desviación típica por la media. Normalmente se pone en porcentaje.

Permite comparar entre variables de diferentes escalas, por ejemplo: ¿en qué somos más variables las personas: en la altura o en el peso?

[ ]:
```
#Coeficiente de variación
var1.std() / var1.mean() * 100
```

Correlación de Pearson Cuando la gente habla de correlación para medir la relación entre dos variables se está refiriendo a Pearson.

Sólo sería técnicamente correcto usar Pearson con variables contínuas y linealmente relacionadas, y si estas tienen una distribución normal.

Pero si las variables no son contínuas o normalmente distribuídas, o su relación es no lineal usar Pearson no es lo más correcto. Podríamos usar Spearman como veremos más abajo.

Además la correlación de Pearson es una medida lineal, y por tanto no válida para medir relaciones no lineales.

Ello significa que si dos variables aparecen como correlacionadas según Pearson efectivamente tendrán relación. Pero lo contrario no es siempre cierto, ya que podría haber relación pero ser no lineal.

La forma más fácil para identificar correlaciones no lineales es usar un gráfico de dispersión como veremos más adelante.

La correlación de Pearson se interpreta así:

- +1 o -1: es una correlación perfecta (siempre que sube una la otra también o siempre que sube una la otra baja)
- 0: no hay relación entre las variables
- Si el pvalor está por debajo de 0.05 la relación sí es significativa (se recomienda considerarlo sólo para más de 500 datos)

[ ]:
```
#Correlación de Pearson
#Esta implementación de scipy devuelve la correlación como primer valor y el␣
 ↪pvalor como el segundo
var1 = np.random.randint(1,21,1000)
var2 = np.random.randint(1,21,1000)

sp.stats.pearsonr(var1,var2)
```

R cuadrado R cuadrado es simplemente el cuadrado de la correlación de Pearson.

Pero sin embargo es una métrica muy útil en modelización, ya que se puede entender como el porcentaje de una variable que podemos explicar a partir de otra (o combinación de otras como en los modelos lineales).

Por tanto será frecuente verla en modelos como las regresiones, donde el software nos reportará por ejemplo un R cuadrado de 0.6 y eso significa que con nuestro modelo estamos siendo capaces de explicar el 60% de la variable objetivo, y por tanto nos queda un 40% que no sabemos explicar.

Al igual que la correlación de Pearson es una medida lineal, por tanto si cuando estamos modelizando con una técnica lineal nos sale un R cuadrado bajo puede ser interesante probar una técnica no lineal para ver si hay relaciones no lineales que no estaban siendo capturadas.

Veremos todo esto en la parte de modelización del programa.

[ ]:
```
#R cuadrado
#Esta implementación de scipy devuelve la correlación como primer valor y el␣
 ↪pvalor como el segundo
(sp.stats.pearsonr(var1,var2)[0] ** 2) * 100
```

Correlación de Spearman La mayoría de la gente conoce la correlación de Pearson para medir la relación entre dos variables.

Pero realmente sólo es correcto usar Pearson con variables contínuas, linealmente relacionadas y si estas tienen una distribución normal.

Pero si las variables no son contínuas o no son normalmente distribuídas usar Pearson no es lo más correcto.

La **Rho de Spearman** proporciona una solución para hacer la correlación cuando:

- las variables son contínuas pero no se distribuyen según la normal
- las variables son rankings
- la relación es no lineal
- las variables son discretas pero con menos de 30 valores distintos

Se interpreta así:

- +1 o -1: es una relación perfectamente monotónica (siempre que sube una la otra también o al revés)
- 0: no hay relación entre las variables
- Si el pvalor está por debajo de 0.05% la relación sí es significativa (se recomienda considerarlo sólo para más de 500 datos)

[ ]:
```
#Correlación de Spearman
var1 = np.random.randint(1,21,1000)
var2 = np.random.randint(1,21,1000)

sp.stats.spearmanr(var1,var2)
```

Correlación Vs Causalidad Este es uno de los errores más frecuentes en el uso cotidiano de la estadística.

Se tiende a pensar que si dos variables están correlacionadas entonces se puede establecer una relación causa - efecto entre ellas.

Realmente, para poder establacer una relación causa - efecto se tiene que cumplir que:

1. Exista correlación
2. La causa preceda en el tiempo a la consecuencia
3. Se puedan descartar explicaciones alternativas

Aquí entran los conceptos de correlación espúrea y correlación parcial.

Correlación espúrea es cuando dos variables parece que están relacionadas (pueden correlacionar de hecho), pero es realmente por el efecto de otras terceras variables no consideradas.

Correlación parcial es la correlación real que queda entre las 2 primeras variables una vez que se elimina el efecto de la tercera o terceras.

Por ejemplo, está demostrado que existe alta correlación entre el tamaño del pie y el cociente de inteligencia.

Sin embargo esa correlación está marcada realmente por una tercera variable, la edad. Si se controla la edad y se elimina su efecto entonces ya no existe correlación entre el tamaño del pie y el cociente de inteligencia.

En contextos empresariales está plagado de este tipo de efectos y es conveniente conocer este concepto y aplicar siempre la visión crítica antes de obtener conclusiones.

Ejemplos:

- No suben las ventas cuando hay promociones: ¿se han controlado las promociones de la competencia?

- Los empleados que viajan dejan más la empresa: ¿se ha controlado la edad?
- Los clientes que más ganan sin embargo tienen menos ahorros: ¿se ha diferenciado entre los que tienen hipoteca y los que no?

Con gráficos Los gráficos más frecuentes que podremos hacer con variables cuantitativas son:

- Histogramas
- Gráficos de densidad
- Gráficos de dispersión (2 variables)

[ ]:
```
df.info()
```

[ ]:
```
#Ejemplo de histograma
df.total_bill.plot(kind = 'hist');
```

[ ]:
```
#Ejemplo de densidad
df.total_bill.plot(kind = 'density');
```

[ ]:
```
#Ejemplo de gráfico de dispersión
df.plot.scatter('total_bill','tip');
```

1.4.3 ¿Para qué usamos todo esto en Data Science?

Para todo :-)

Cuando estés haciendo análisis te pasarás gran parte del tiempo haciendo este tipo de análisis y gráficos para las fases de un proyecto de Data Science de:

- Calidad de datos
- Corrección de errores
- Análisis exploratorio
- Transformación y creación de variables

Además la correlación es una de las técnicas de preselección de variables en modelización predictiva, así que la usaremos potencialmente de dos formas:

- Identificar qué variables NO están correlacionadas con la variable a predecir, y por tanto no invertir tiempo en trabajar sobre ellas ni incluirlas en la modelización
- Identficiar qué variables están correlacionadas entre sí y por tanto no es conveniente usarlas simultáneamente en los modelos

Realmente esta parte descriptiva será mucho más usada en la práctica que la inferencial. Así que debes conocerla y entenderla bien.

1.5 EJERCICIOS DE REPASO

1.5.1 A partir del dataset tips, muestra la proporción de fumadores y no fumadores mediante una tabla cruzada normalizada

[18]:
```
import pandas as pd
import seaborn as sns

df = sns.load_dataset('tips')
pd.crosstab(df.smoker, columns ='count', normalize = 'all', margins=True) * 100
```

[18]:
```
col_0        count         All
smoker
Yes      38.114754   38.114754
No       61.885246   61.885246
All     100.000000  100.000000
```

1.5.2 Genera un array de enteros aleatorios entre 1 y 20, y calcula tanto su media armónica como su media geométrica

[21]:
```
import numpy as np
import statistics

datos = np.random.randint(1,20,10)
print(statistics.harmonic_mean(datos))
print(statistics.geometric_mean(datos))
```

```
2.791779759596743
4.068999377922609
```

1.5.3 Crea dos vectores de enteros aleatorios entre 1 y 20 y calcula la correlación de Spearman, indicando si es estadísticamente significativa

[24]:
```
import scipy as sp

x = np.random.randint(1,20,1000)
y = np.random.randint(1,20,1000)

corr, p_valor = sp.stats.spearmanr(x,y)
print(corr)
print(p_valor)
```

```
0.012809116481808434
0.6857937539301057
```

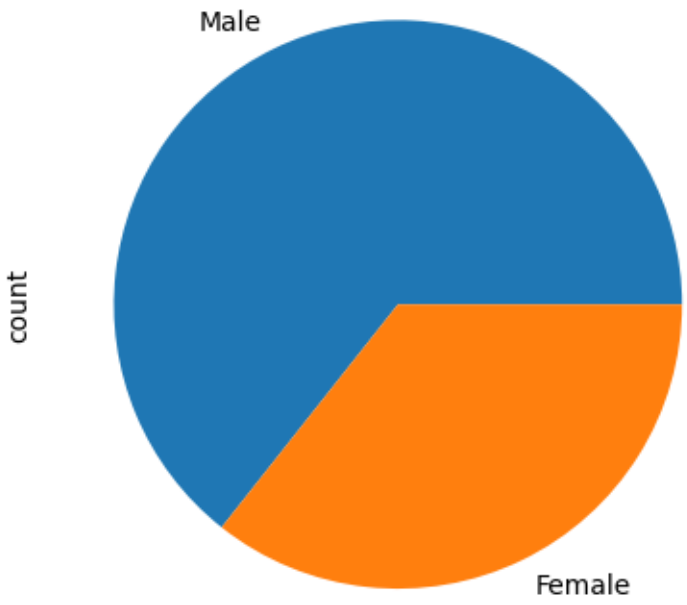
1.5.4 Simula un array con valores extremos, aplica winsorización y compara la media antes y después del tratamiento

[26]:
```
x = np.array([3,4,5,6,999])
print(x.mean())
x_win = sp.stats.mstats.winsorize(x,limits=[0,0.2])
print(x_win.mean())
```

```
203.4
4.8
```

1.5.5 Visualiza la distribución de la variable sex del dataset tips mediante un gráfico de sectores

[30]:
```
df.sex.value_counts().plot(kind='pie');
```



1.5.6 Genera un vector numérico aleatorio, ordénalo y calcula su mediana usando el módulo statistics

[33]:
```
numeros = np.random.randint(0,11,11)
numeros_ord = np.sort(numeros)
print(numeros_ord)
print(statistics.median(numeros_ord))
```

```
[0 0 3 4 4 5 6 6 7 7 9]
5
```

1.5.7 Calcula el R cuadrado entre dos vectores de enteros aleatorios generados entre 1 y 20

[37]:
```
x = np.random.randint(1,21,1000)
y = np.random.randint(1,21,1000)
(sp.stats.pearsonr(x,y)[0] ** 2) * 100
```

[37]:
```
np.float64(0.3131245564853046)
```

1.5.8 Simula una variable con enteros entre 10 y 100, y calcula su coeficiente de variación en porcentaje

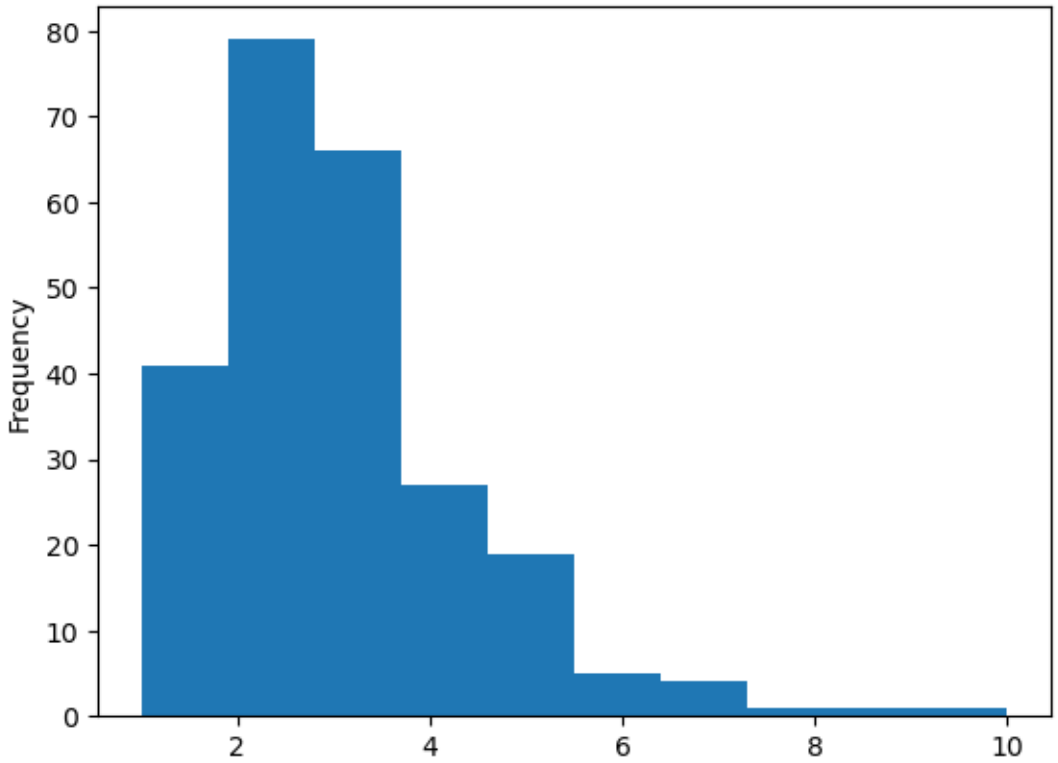
[38]:
```
x = np.random.randint(10,101,1000)
media = x.mean()
std = x.std()
print(std / media *100)
```

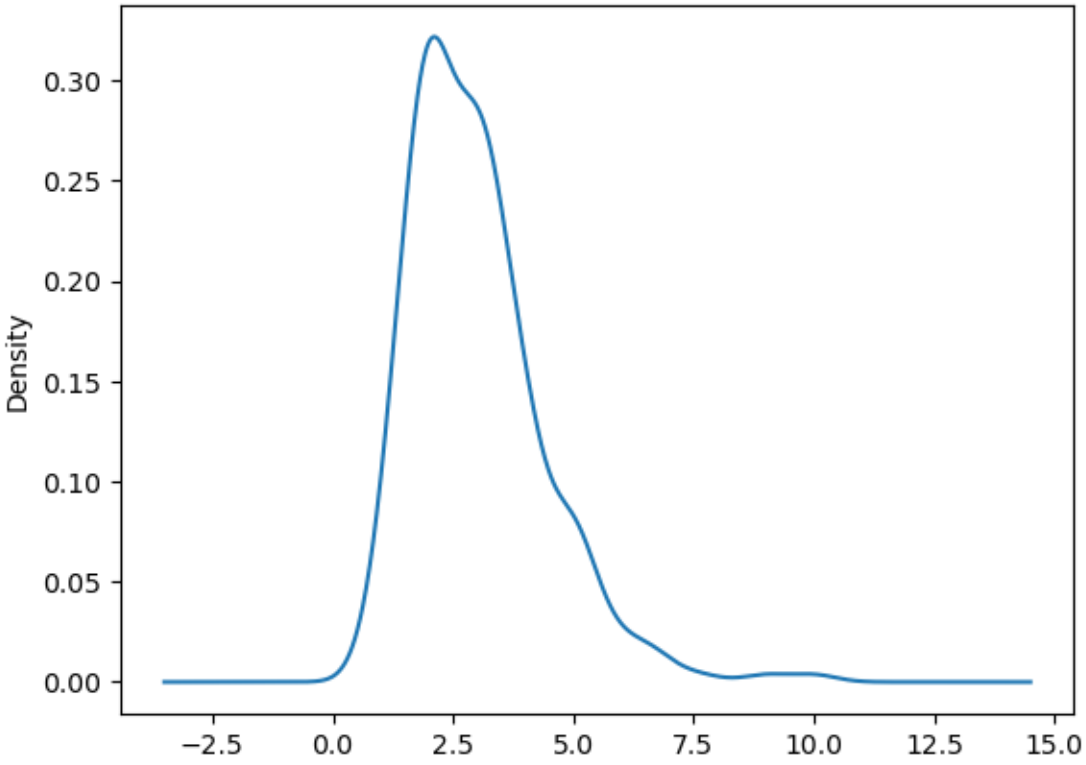
```
48.31016129118503
```

1.5.9 Representa la distribución de la variable tip del dataset tips mediante un histograma y una curva de densidad

[41]:
```
import matplotlib.pyplot as plt

df.tip.plot(kind='hist')
plt.show()
df.tip.plot(kind='density')
plt.show()
```


Frequency
0
10
20
30
40
50
60
70
80
2
4
6
8
10


Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

1.5.10 Realiza un test chi-cuadrado entre las variables sex y day del dataset tips e imprime el valor p del test

[45]:
```
tabla = pd.crosstab(df.sex,df.day)
chi2, pvalor, gl, esperados = sp.stats.chi2_contingency(tabla)
print(pvalor)
```

```
0.004180302092822257
```

[47]:
```
tabla
```

[47]:

day	Thur	Fri	Sat	Sun
sex				
Male	30	10	59	58
Female	32	9	28	18

[ ]:
```

```