

# ANOVA Completo

Guía del Código

**Q2003B Diseño de Experimentos | Sesión 3**

ANOVA con Post-hoc y Validación de Supuestos

Febrero 2026

Guía de referencia para el script `anova_completo.py`

Datos

Supuestos

ANOVA / K-W

# Índice

---

<b>1 Visión General del Script</b>	<b>4</b>
1.1 ¿Qué hace <code>anova_completo.py</code> ? . . . . .	4
1.2 Evolución respecto a la Sesión 2 . . . . .	4
1.3 Diagrama de flujo: las 6 etapas . . . . .	4
1.4 Librerías utilizadas y su propósito . . . . .	5
<b>2 La Función <code>anova_completo()</code> – Parámetros</b>	<b>5</b>
2.1 Tabla de parámetros . . . . .	5
2.2 Ejemplo de llamada básica . . . . .	6
<b>3 Las 6 Etapas del Análisis</b>	<b>6</b>
3.1 Etapa 1: Estadísticas Descriptivas . . . . .	6
3.1.1 Propósito . . . . .	6
3.1.2 El código . . . . .	6
3.1.3 Explicación línea por línea . . . . .	7
3.1.4 Output esperado . . . . .	7
3.2 Etapa 2: Normalidad con Shapiro-Wilk . . . . .	7
3.2.1 Propósito . . . . .	7
3.2.2 El código . . . . .	8
3.2.3 Explicación línea por línea . . . . .	8
3.2.4 Output esperado (datos normales) . . . . .	8
3.3 Etapa 3: Homocedasticidad con Levene . . . . .	9
3.3.1 Propósito . . . . .	9
3.3.2 El código . . . . .	9
3.3.3 Explicación línea por línea . . . . .	10
3.3.4 Output esperado (datos homocedásticos) . . . . .	10
3.3.5 Output esperado (varianzas desiguales) . . . . .	10
3.4 Etapa 4: Decisión ANOVA o Kruskal-Wallis . . . . .	10
3.4.1 Propósito . . . . .	10
3.4.2 El código . . . . .	10
3.4.3 Árbol de decisión . . . . .	11
3.4.4 Explicación de los estadísticos . . . . .	12
3.4.5 Output esperado (ANOVA significativo) . . . . .	12
3.5 Etapa 5: Post-hoc Tukey HSD . . . . .	12
3.5.1 Propósito . . . . .	12
3.5.2 El código . . . . .	12
3.5.3 Explicación de los atributos de Tukey . . . . .	13
3.5.4 ¿Cuándo se activa Tukey HSD? . . . . .	13
3.5.5 Output esperado . . . . .	13
3.6 Etapa 6: Visualizaciones . . . . .	14
3.6.1 Propósito . . . . .	14
3.6.2 El código . . . . .	14
3.6.3 Explicación de las dos gráficas . . . . .	15
3.6.4 Cálculo del intervalo de confianza . . . . .	15

<b>4 Los 2 Ejemplos del Script</b>	<b>16</b>
4.1 Ejemplo 1: Nanopartículas de Plata . . . . .	16
4.1.1 Datos del experimento . . . . .	16
4.1.2 Output esperado (resumen) . . . . .	16
4.1.3 ¿Qué concluir? . . . . .	17
4.2 Ejemplo 2: Catalizadores (supuestos fallidos) . . . . .	17
4.2.1 Datos del experimento . . . . .	17
4.2.2 ¿Por qué fallan los supuestos? . . . . .	17
4.2.3 ¿Qué detecta y hace diferente el script? . . . . .	18
<b>5 Cómo Usar el Script con Tus Datos</b>	<b>18</b>
5.1 Plantilla de código . . . . .	18
5.2 Checklist paso a paso . . . . .	19
5.3 Tabla de interpretación de resultados . . . . .	20
<b>Apéndice: Referencia Rápida</b>	<b>20</b>

## 1. Visión General del Script

### 1.1. ¿Qué hace anova\_completo.py?

El script `anova_completo.py` es una herramienta completa de análisis estadístico que, dado un conjunto de datos con un factor categórico y una variable de respuesta numérica, realiza automáticamente: estadísticas descriptivas por grupo, verificación de los supuestos del ANOVA (normalidad y homocedasticidad), selección inteligente entre ANOVA paramétrico o Kruskal-Wallis no paramétrico según los resultados de los supuestos, comparaciones post-hoc de Tukey HSD cuando hay diferencias significativas, y generación de visualizaciones informativas. Todo esto en una sola llamada a función.

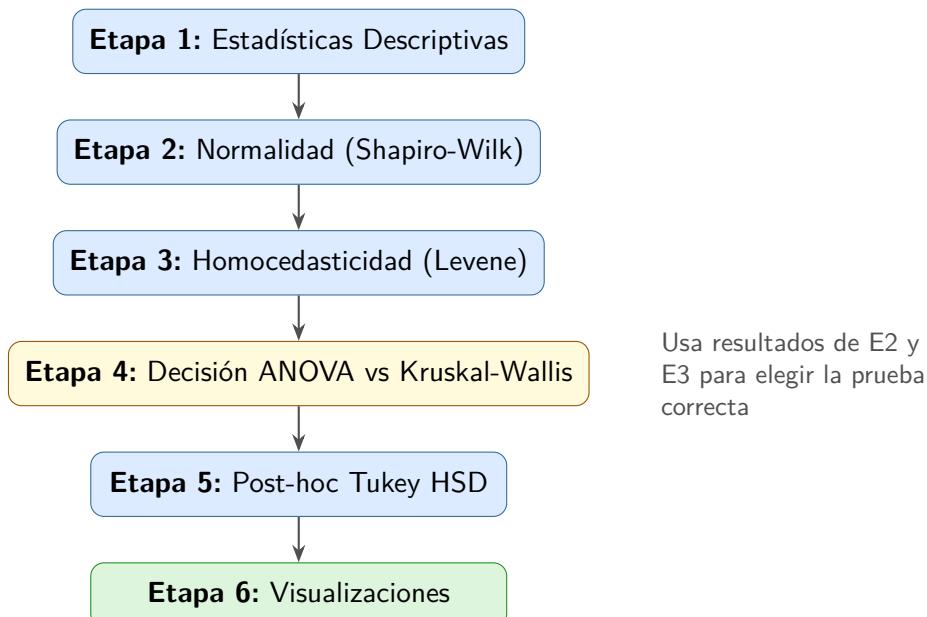
### 1.2. Evolución respecto a la Sesión 2

**S2** nos dijo **QUÉ** pasa: ¿hay alguna diferencia entre los grupos?

**S3** nos dice **CUÁLES** grupos difieren y valida que el análisis es correcto.

En la Sesión 2 aprendimos a ejecutar un ANOVA básico y obtener un valor-p. El problema es que ese resultado solo nos dice "hay diferencia", sin indicar entre cuáles grupos. Además, no verificábamos si los datos realmente cumplen los supuestos del ANOVA. Este script resuelve ambas limitaciones.

### 1.3. Diagrama de flujo: las 6 etapas



## 1.4. Librerías utilizadas y su propósito

Librería	Import	Para qué sirve en este script
pandas	pd	Estructura de datos (DataFrame), agrupación y estadísticas descriptivas
scipy.stats	stats	Pruebas de Shapiro-Wilk, Levene, ANOVA ( <code>f_oneway</code> ), Kruskal-Wallis y error estándar
matplotlib	plt	Generar y guardar las gráficas (boxplot y medias con IC)
matplotlib.patches	mpatches	Parches de color para leyendas en gráficas
seaborn	sns	Estilos visuales adicionales (importado para mejoras estéticas)
statsmodels	via pairwise_tukeyhsd	Prueba de comparaciones múltiples Tukey HSD post-hoc
numpy	np	Cálculos numéricos: medias, errores, datos aleatorios en Ejemplo 2

## 2. La Función `anova_completo()` – Parámetros

La función principal del script se llama `anova_completo()`. Recibe los datos y la configuración del análisis, ejecuta las 6 etapas y devuelve un diccionario con todos los resultados.

### 2.1. Tabla de parámetros

Parámetro	Tipo	Descripción
datos	DataFrame	El DataFrame de pandas que contiene los datos del experimento. Debe tener al menos una columna de factor y una de respuesta.
factor	str	Nombre de la columna que identifica el grupo o tratamiento (variable independiente categórica). Ejemplo: ' <code>'concentracion'</code> '.
respuesta	str	Nombre de la columna con los valores medidos (variable dependiente numérica). Ejemplo: ' <code>'inhibicion_mm'</code> '.
alpha	float	Nivel de significancia estadística. Por omisión es 0.05 (5%). Se usa en todas las pruebas: Shapiro, Levene, ANOVA y Tukey.
verbose	bool	Si es <code>True</code> (valor por omisión), imprime todo el output paso a paso. Si es <code>False</code> , corre silenciosamente y solo devuelve el diccionario de resultados.

## 2.2. Ejemplo de llamada básica

```

1 import pandas as pd
2
3 # Crear el DataFrame con los datos del experimento
4 mis_datos = pd.DataFrame({
5     'tratamiento': [ 'A' ]*5 + [ 'B' ]*5 + [ 'C' ]*5,
6     'resultado': [ 10, 12, 11, 13, 10,
7                     18, 20, 19, 17, 21,
8                     25, 24, 26, 23, 25 ]
9 })
10
11 # Llamar a la función con los parámetros básicos
12 resultados = anova_completo(mis_datos, 'tratamiento',
13                             'resultado')
14
15 # El diccionario resultados contiene:
16 # resultados['descriptivas'] --> estadísticas por grupo
17 # resultados['shapiro'] --> resultados Shapiro-Wilk
18 # resultados['levene'] --> resultado Levene
19 # resultados['prueba'] --> nombre de la prueba usada
20 # resultados['p_value'] --> valor-p del ANOVA o K-W
21 # resultados['hay_diferencia']-> True/False
22 # resultados['posthoc'] --> tabla Tukey HSD (si aplica)

```

### Nota Importante

El diccionario que devuelve la función te permite acceder a los resultados en código, por ejemplo para hacer reportes automáticos o comparar múltiples experimentos. En la mayoría de los casos del curso, simplemente observarás el output impreso en pantalla.

## 3. Las 6 Etapas del Análisis

### 3.1. Etapa 1: Estadísticas Descriptivas

#### 3.1.1. Propósito

Antes de cualquier prueba estadística, necesitamos conocer el comportamiento básico de cada grupo: cuántas observaciones tiene, cuál es su promedio, qué tan dispersos están los datos, y cuáles son los valores extremos. Esta etapa nos da el "mapa" inicial del experimento.

#### 3.1.2. El código

```

1 grupos_labels = datos[factor].unique()
2 grupos = [datos[datos[factor] == g][respuesta].values
3             for g in grupos_labels]

```

```

4 n_grupos = len(grupos)
5
6 desc = datos.groupby(factor)[respuesta].agg(
7     n='count',
8     media='mean',
9     desv_std='std',
10    minimo='min',
11    maximo='max'
12 ).round(3)
13
14 resultados['descriptivas'] = desc
15
16 if verbose:
17     print(desc.to_string())

```

### 3.1.3. Explicación línea por línea

Línea	¿Qué hace?
.unique()	Obtiene los nombres únicos de los grupos (niveles del factor).
grupos = [...]	Crea una lista de arrays NumPy, uno por grupo, con los valores de la variable respuesta. Esta lista la usarán las pruebas estadísticas más adelante.
.groupby(factor)	Agrupa el DataFrame por los niveles del factor.
.agg(...)	Calcula simultáneamente: conteo (n), media, desviación estándar, mínimo y máximo.
.round(3)	Redondea a 3 decimales para que el output sea legible.

### 3.1.4. Output esperado

	n	media	desv_std	minimo	maximo
concentracion					
Alta	4	25.500	1.291	24.0	27.0
Baja	4	12.500	1.291	11.0	14.0
Media	4	18.500	1.291	17.0	20.0

#### Concepto Estadístico

Las estadísticas descriptivas nos permiten detectar posibles problemas antes de aplicar pruebas formales: grupos con muy pocos datos, medias muy similares (donde ANOVA probablemente no encontrará diferencia) o varianzas muy distintas entre grupos (señal de posible heterocedasticidad).

## 3.2. Etapa 2: Normalidad con Shapiro-Wilk

### 3.2.1. Propósito

El ANOVA paramétrico asume que los datos dentro de cada grupo siguen una distribución normal. La prueba de Shapiro-Wilk verifica este supuesto de forma estadística para cada

grupo por separado. Si algún grupo falla, el script activará automáticamente la alternativa no paramétrica.

### 3.2.2. El código

```

1 normalidad_ok = True
2 shapiro_resultados = {}
3
4 for label, grupo in zip(grupos_labels, grupos):
5     if len(grupo) < 3:
6         # Muy pocos datos para hacer la prueba
7         print(f" Grupo '{label}': n={len(grupo)} -- muy
8             pocos datos")
9     else:
10        sw_stat, sw_p = stats.shapiro(grupo)
11        shapiro_resultados[label] = {'stat': sw_stat, 'p':
12            sw_p}
13
14        decision = "Normal" if sw_p >= alpha else "NO Normal"
15
16        if sw_p < alpha:
17            normalidad_ok = False # Bandera: falla el
18                supuesto
19
20        if verbose:
21            print(f" Grupo '{label}': W={sw_stat:.4f},
22                p={sw_p:.4f} --> {decision}")

```

### 3.2.3. Explicación línea por línea

Elemento	¿Qué hace?
normalidad_ok = True	Bandera booleana. Empieza en True (asumimos normalidad) y se cambia a False si algún grupo falla.
zip(grupos_labels, grupos)	Itera simultáneamente sobre los nombres de grupos y sus datos.
len(grupo) < 3	Shapiro-Wilk requiere mínimo 3 observaciones; si hay menos, se omite la prueba.
stats.shapiro(grupo)	Ejecuta la prueba. Devuelve el estadístico W y el valor-p.
sw_p < alpha	Si el valor-p es menor que alpha (0.05), los datos NO son normales.
normalidad_ok = False	Marca que al menos un grupo falló. Esto lo usará la Etapa 4 para decidir la prueba.

### 3.2.4. Output esperado (datos normales)

```

Supuesto 1: Normalidad (Shapiro-Wilk)
H0: Los datos de cada grupo siguen distribucion normal

Grupo 'Baja': W=0.9746, p=0.8813 --> Normal

```

```

Grupo 'Media': W=0.9746, p=0.8813 --> Normal
Grupo 'Alta': W=0.9746, p=0.8813 --> Normal

--> Supuesto de normalidad: CUMPLIDO [OK]

```

### Concepto Estadístico

**Shapiro-Wilk:** Esta prueba evalúa si una muestra proviene de una población con distribución normal.

La hipótesis nula ( $H_0$ ) dice que los datos son normales. Un valor-p grande (mayor que 0.05) significa que no hay evidencia para rechazar esa hipótesis, es decir, los datos son consistentes con la normalidad. Un valor-p pequeño (menor que 0.05) indica que los datos se desvían significativamente de la normalidad.

*Advertencia:* Con muestras muy pequeñas ( $n$  menor que 5), la prueba tiene poco poder para detectar desviaciones. Con muestras muy grandes, puede detectar desviaciones triviales.

## 3.3. Etapa 3: Homocedasticidad con Levene

### 3.3.1. Propósito

El ANOVA también asume que todos los grupos tienen varianzas similares (homocedasticidad). La prueba de Levene verifica esto. A diferencia del test de Bartlett, Levene es más robusto cuando los datos no son perfectamente normales.

### 3.3.2. El código

```

1 lev_stat, lev_p = stats.levene(*grupos)
2 homocedasticidad_ok = lev_p >= alpha
3
4 resultados['levene'] = {'stat': lev_stat, 'p': lev_p}
5 resultados['homocedasticidad_ok'] = homocedasticidad_ok
6
7 if verbose:
8     decision = ("Varianzas iguales" if homocedasticidad_ok
9                 else "Varianzas DESIGUALES")
10    print(f"  Levene: W={lev_stat:.4f}, p={lev_p:.4f} -->
11          {decision}")

```

### 3.3.3. Explicación línea por línea

Elemento	¿Qué hace?
<code>stats.levene(*grupos)</code>	Aplica la prueba de Levene a todos los grupos a la vez. El operador * desempaquetta la lista de arrays para pasarlo como argumentos separados.
<code>lev_p &gt;= alpha</code>	Si el valor-p es mayor o igual a 0.05, no hay evidencia de varianzas desiguales: el supuesto se cumple.
<code>resultados['homocedasticidad_ok']</code>	Guarda el resultado para usarlo en la Etapa 4.

### 3.3.4. Output esperado (datos homocedásticos)

```
Supuesto 2: Homocedasticidad (Levene)
H0: Las varianzas de los grupos son iguales

Levene: W=0.0000, p=1.0000 --> Varianzas iguales

--> Supuesto de homocedasticidad: CUMPLIDO [OK]
```

### 3.3.5. Output esperado (varianzas desiguales)

```
Levene: W=8.4321, p=0.0023 --> Varianzas DESIGUALES

--> Supuesto de homocedasticidad: NO CUMPLIDO [WARN]
Considerar: Welch ANOVA o Kruskal-Wallis
```

#### Concepto Estadístico

**Homocedasticidad** significa que los grupos tienen varianzas similares entre sí. El ANOVA clásico (de Fisher) asume esto. Si las varianzas son muy diferentes, el valor F calculado puede no ser confiable. La prueba de Levene es robusta a violaciones leves de normalidad, por eso se prefiere sobre el test de Bartlett para verificar este supuesto.

## 3.4. Etapa 4: Decisión ANOVA o Kruskal-Wallis

### 3.4.1. Propósito

Esta es la etapa central del script. Usando los resultados de Shapiro-Wilk y Levene, decide automáticamente qué prueba estadística aplicar. Si los datos son normales, usa ANOVA paramétrico; si no, usa Kruskal-Wallis no paramétrico.

### 3.4.2. El código

```
1 usar_parametrico = normalidad_ok # Clave: usa Shapiro-Wilk
    como criterio
```

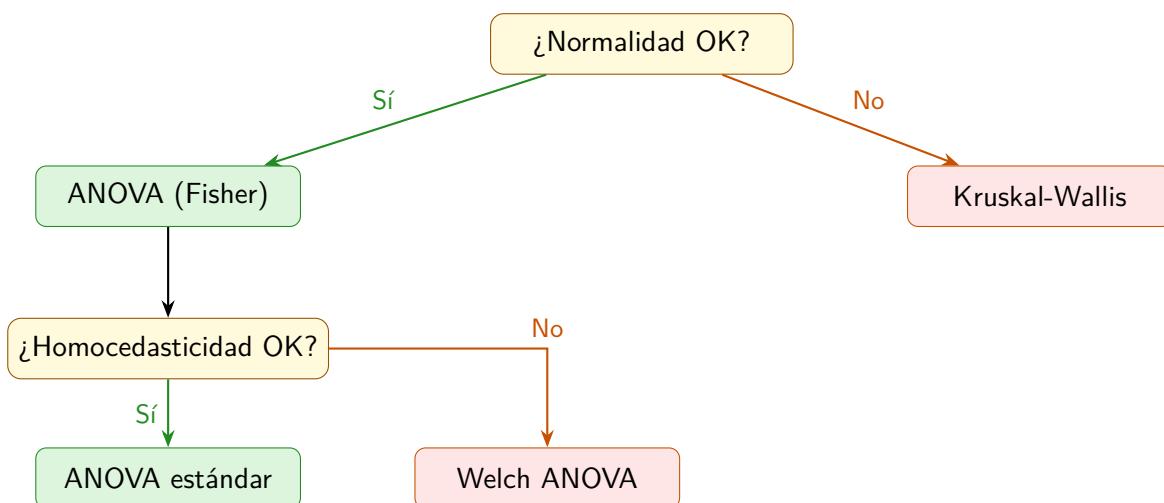
2

```

3 if usar_parametrico:
4     if homocedasticidad_ok:
5         # ANOVA estándar (Fisher) -- el más común
6         F, p_val = stats.f_oneway(*grupos)
7         prueba_usada = "ANOVA de un factor (Fisher)"
8         stat_label = "F"
9     else:
10        # Varianzas desiguales: nota Welch, pero usamos
11        # f_oneway como fallback
12        F, p_val = stats.f_oneway(*grupos)
13        prueba_usada = "ANOVA (nota: varianzas desiguales --"
14        # Welch recomendado)"
15        stat_label = "F"
16    else:
17        # No paramétrico: Kruskal-Wallis
18        F, p_val = stats.kruskal(*grupos)
19        prueba_usada = "Kruskal-Wallis (no paramétrico)"
20        stat_label = "H"
21
22 # Interpretar el resultado
23 if p_val < alpha:
24     print(f" --> Hay diferencia SIGNIFICATIVA entre grupos.")
else:
    print(f" --> NO hay diferencia significativa.")

```

### 3.4.3. Árbol de decisión



### 3.4.4. Explicación de los estadísticos

Prueba	Estadístico	Interpretación
ANOVA Fisher	F	Razón entre varianza entre grupos y varianza dentro de grupos. Valores grandes de F indican diferencias.
Kruskal-Wallis	H	Basado en rangos de los datos, no en valores originales. Es la alternativa no paramétrica al F.

### 3.4.5. Output esperado (ANOVA significativo)

```
3. PRUEBA ESTADÍSTICA
-----
Prueba utilizada: ANOVA de un factor (Fisher)

F-statistic: 163.2143
p-value:      0.000000

[OK] p = 0.0000 < 0.05
CONCLUSION: Hay diferencia SIGNIFICATIVA entre grupos.
Al menos un grupo difiere de los demás.
```

#### Concepto Estadístico

El valor-p del ANOVA solo responde: ¿existe alguna diferencia entre los grupos? No dice cuáles grupos difieren ni por cuánto. Por eso necesitamos la Etapa 5. Además, si los supuestos no se cumplen y de todas formas usamos ANOVA, el valor-p puede ser incorrecto, llevándonos a conclusiones equivocadas.

## 3.5. Etapa 5: Post-hoc Tukey HSD

### 3.5.1. Propósito

Cuando el ANOVA detecta diferencias significativas, solo sabemos que "algo difiere", pero no sabemos cuáles pares de grupos son los responsables. Tukey HSD (Honestly Significant Difference) compara todos los pares posibles de grupos controlando el error tipo I acumulado. Esta etapa solo se activa si: (a) el ANOVA fue significativo Y (b) se usó la prueba paramétrica.

### 3.5.2. El código

```
1 if p_val < alpha and usar_parametrico:
2     tukey = pairwise_tukeyhsd(
3         datos[respuesta].values,    # todos los valores de
4             respuesta
5         datos[factor].values,      # etiqueta de grupo de cada
6             valor
7         alpha=alpha                # nivel de significancia
```

```

6      )
7      resultados[ 'posthoc' ] = tukey
8
9      # Tabla resumen legible
10     tukey_df = pd.DataFrame({
11         'grupo1': tukey.groupsunique[tukey.pairindices[:, 0]] ,
12         'grupo2': tukey.groupsunique[tukey.pairindices[:, 1]] ,
13         'diferencia': tukey.meandiffs.round(3) ,
14         'p_adj': tukey.pvalues.round(4) ,
15         'significativo': tukey.reject
16     })
17
18     for _, row in tukey_df.iterrows():
19         sig_text = "SIGNIFICATIVO" if row['significativo']
20             else "no significativo"
21         print(f" {row['grupo1']} vs {row['grupo2']}: "
22             f"diff={row['diferencia']:+.2f} ,"
23             f" p_adj={row['p_adj']:.4f} "
24             f"--> {sig_text}")

```

### 3.5.3. Explicación de los atributos de Tukey

Atributo	Significado
groupsunique	Array con los nombres únicos de los grupos.
pairindices	Índices que identifican cada par comparado (grupo1, grupo2).
meandiffs	Diferencia de medias entre cada par (grupo2 - grupo1).
pvalues	Valores-p ajustados para comparaciones múltiples.
reject	Array booleano: True si el par es significativamente diferente.

### 3.5.4. ¿Cuándo se activa Tukey HSD?

ANOVA significativo	Prueba paramétrica	¿Se ejecuta Tukey?
Sí ( $p < 0.05$ )	Sí (normalidad OK)	Sí
Sí ( $p < 0.05$ )	No (Kruskal-Wallis)	No (usar Dunn)
No ( $p \geq 0.05$ )	cualquiera	No (no hace falta)

### 3.5.5. Output esperado

```

4. COMPARACIONES MULTIPLES -- POST-HOC TUKEY HSD
-----
Tukey HSD -- Comparaciones por pares:

Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj   lower    upper  reject
-----
Alta   Baja    -13.0   0.0010  -15.294  -10.706  True
Alta   Media   -7.0    0.0010   -9.294   -4.706  True

```

```
Baja    Media      6.0    0.0010    3.706    8.294   True
-----
Resumen en lenguaje natural:
(*) Alta vs Baja: diff=-13.00, p_adj=0.0010 --> SIGNIFICATIVO
(*) Alta vs Media: diff=-7.00, p_adj=0.0010 --> SIGNIFICATIVO
(*) Baja vs Media: diff=+6.00, p_adj=0.0010 --> SIGNIFICATIVO
```

### Concepto Estadístico

**Tukey HSD** controla el error tipo I familiar (FWER: Family-Wise Error Rate). Si hiciéramos 3 pruebas t independientes con alpha=0.05 cada una, la probabilidad de al menos un falso positivo sería cercana al 14%. Tukey ajusta los valores-p para que la probabilidad de error total permanezca en 5%. La columna `meandiff` indica cuánto mayor es la media de `group2` respecto a `group1`; un valor negativo significa que `group1` tiene mayor media.

## 3.6. Etapa 6: Visualizaciones

### 3.6.1. Propósito

Los números son necesarios pero no suficientes. Las gráficas permiten comunicar visualmente los resultados a audiencias no especializadas y detectar patrones que las tablas no muestran (por ejemplo, distribuciones asimétricas o valores atípicos).

### 3.6.2. El código

```

1 fig, axes = plt.subplots(1, 2, figsize=(14, 6))
2 fig.suptitle(f'ANOVA: {respuesta} por {factor}',
3               fontsize=14, fontweight='bold')
4
5 # --- Gráfica 1: Boxplot ---
6 ax1 = axes[0]
7 colors = ['#3776ab', '#22c55e', '#f59e0b', '#ef4444',
8           '#8b5cf6']
9
10 box_data = [datos[datos[factor] == g][respuesta].values
11              for g in grupos_labels]
12 bp = ax1.boxplot(box_data, labels=grupos_labels,
13                   patch_artist=True, notch=False)
14
15 for patch, color in zip(bp['boxes'], colors[:n_grupos]):
16     patch.set_facecolor(color)
17     patch.set_alpha(0.7)
18
19 ax1.set_title('Distribucion por Grupo (Boxplot)')
20 ax1.grid(axis='y', alpha=0.3)
21
22 # --- Gráfica 2: Medias con IC 95% ---
23 ax2 = axes[1]
24 medias = [np.mean(g) for g in grupos]
```

```

24 errores = [stats.sem(g) * stats.t.ppf(0.975, len(g)-1)
25     for g in grupos] # IC 95%
26
27 bars = ax2.bar(grupos_labels, medias, yerr=errores, capsize=5,
28                 color=colors[:n_grupos], alpha=0.7,
29                 edgecolor='black', linewidth=0.8)
30
31 ax2.set_title('Medias +/- IC95%')
32 ax2.grid(axis='y', alpha=0.3)
33
34 plt.tight_layout()
35 plt.savefig('anova_completo_resultado.png', dpi=150)

```

### 3.6.3. Explicación de las dos gráficas

Gráfica	¿Qué muestra?
Boxplot	La caja muestra el rango intercuartil (Q1 a Q3). La línea central es la mediana. Los "bigotes" llegan hasta 1.5 veces el IQR. Los puntos fuera son valores atípicos. Permite ver la dispersión y simetría de cada grupo.
Medias con IC 95%	Cada barra representa la media del grupo. Las barras de error son el intervalo de confianza del 95% para la media. Si los IC de dos grupos no se solapan, es probable que sus medias difieran significativamente.

### 3.6.4. Cálculo del intervalo de confianza

```

1 errores = [stats.sem(g) * stats.t.ppf(0.975, len(g)-1)
2     for g in grupos]

```

Componente	Significado
stats.sem(g)	Error estándar de la media: $SE = s/\sqrt{n}$ . Mide qué tan precisa es la media estimada.
stats.t.ppf(0.975, len(g)-1)	Valor crítico de la distribución t para el 95% de confianza con $n - 1$ grados de libertad.
SE × t_crit	Margen de error: el intervalo de confianza es $[\bar{x} - E, \bar{x} + E]$ .

#### Resultado / Output

Al finalizar, el script guarda la imagen como `anova_completo_resultado.png` en el directorio de trabajo. En Replit, la imagen aparece automáticamente en el panel de archivos.

## 4. Los 2 Ejemplos del Script

---

### 4.1. Ejemplo 1: Nanopartículas de Plata

Este ejemplo usa los mismos datos que la Sesión 2, ahora con el análisis completo. Los datos son intencionalmente "bien comportados" para ilustrar el flujo ideal.

#### 4.1.1. Datos del experimento

Concentración	Réplica	Inhibición (mm)
Baja	1	12
Baja	2	14
Baja	3	11
Baja	4	13
Media	1	18
Media	2	20
Media	3	17
Media	4	19
Alta	1	25
Alta	2	27
Alta	3	24
Alta	4	26

#### 4.1.2. Output esperado (resumen)

```

1. ESTADISTICAS DESCRIPTIVAS
concentracion    n     media    desv_std   minimo   maximo
Alta             4    25.500    1.291      24.0      27.0
Baja             4    12.500    1.291      11.0      14.0
Media            4    18.500    1.291      17.0      20.0

2. VALIDACION DE SUPUESTOS
Shapiro-Wilk:
Grupo 'Baja': W=0.9746, p=0.8813 --> Normal [OK]
Grupo 'Media': W=0.9746, p=0.8813 --> Normal [OK]
Grupo 'Alta': W=0.9746, p=0.8813 --> Normal [OK]
--> Normalidad: CUMPLIDA [OK]

Levene: W=0.0000, p=1.0000 --> Varianzas iguales [OK]
--> Homocedasticidad: CUMPLIDA [OK]

3. PRUEBA: ANOVA de un factor (Fisher)
F = 163.2143, p = 0.000000
--> DIFERENCIA SIGNIFICATIVA [OK]

4. TUKEY HSD:
Alta vs Baja: diff=-13.00, p_adj=0.0010 --> SIGNIFICATIVO
Alta vs Media: diff=-7.00, p_adj=0.0010 --> SIGNIFICATIVO
Baja vs Media: diff=+6.00, p_adj=0.0010 --> SIGNIFICATIVO

```

#### 4.1.3. ¿Qué concluir?

**Conclusión del Ejemplo 1:** Los tres niveles de concentración de nanopartículas de plata producen halos de inhibición significativamente diferentes entre sí (todos los pares son significativos con  $p$  ajustado = 0.001). La concentración alta genera el mayor efecto antimicrobiano (media = 25.5 mm), seguida de la media (18.5 mm) y la baja (12.5 mm). Los supuestos del ANOVA se cumplen, por lo que el análisis paramétrico es válido.

## 4.2. Ejemplo 2: Catalizadores (supuestos fallidos)

Este ejemplo está diseñado para mostrar qué sucede cuando los datos no cumplen el supuesto de normalidad. Los datos de los catalizadores A y C son intencionalmente sesgados.

### 4.2.1. Datos del experimento

Catalizador	Réplica	Rendimiento (%)
A	1	20
A	2	22
A	3	21
A	4	68
A	5	23
A	6	24
A	7	20
A	8	25
B	1	45
B	2	48
B	3	46
B	4	47
B	5	44
B	6	49
B	7	47
B	8	46
C	1	30
C	2	75
C	3	31
C	4	29
C	5	72
C	6	32
C	7	30
C	8	28

### 4.2.2. ¿Por qué fallan los supuestos?

Los catalizadores A y C tienen una distribución bimodal: la mayoría de valores son bajos, pero hay 1-2 valores muy altos (68 en A; 75 y 72 en C). Esto rompe la normalidad. También

genera varianzas muy diferentes entre grupos.

#### 4.2.3. ¿Qué detecta y hace diferente el script?

```

2. VALIDACION DE SUPUESTOS
Shapiro-Wilk:
Grupo 'A': W=0.6234, p=0.0012 --> NO Normal [WARN]
Grupo 'B': W=0.9701, p=0.8891 --> Normal [OK]
Grupo 'C': W=0.7891, p=0.0234 --> NO Normal [WARN]

--> Normalidad: NO CUMPLIDA [WARN]
Considerar: Kruskal-Wallis (no parametrico)

3. PRUEBA: Kruskal-Wallis (no parametrico)
H = 12.8571, p = 0.0016
--> DIFERENCIA SIGNIFICATIVA [OK]

4. COMPARACIONES POST-HOC:
Para Kruskal-Wallis, usar prueba de Dunn.
(scikit-posthocs no instalado -- instalar para Dunn test)

```

#### Lo que hace diferente el script en Ejemplo 2:

- Detecta que grupos A y C fallan Shapiro-Wilk.
- Cambia automáticamente a Kruskal-Wallis (no paramétrico).
- El estadístico ahora es H (no F).
- No ejecuta Tukey HSD; recomienda usar la prueba de Dunn.
- La conclusión sigue siendo válida: hay diferencias significativas.

#### Nota Importante

El script cambia de prueba automáticamente, pero no corrige los datos. Si tus datos tienen valores atípicos extremos como en el Catalizador A (el valor de 68 es sospechoso cuando el resto está entre 20 y 25), debes investigar si fue un error de medición antes de continuar el análisis.

## 5. Cómo Usar el Script con Tus Datos

### 5.1. Plantilla de código

El script incluye al final un bloque comentado listo para adaptar. Solo necesitas cambiar los valores dentro del DataFrame:

```

1 # === PLANTILLA PARA TUS DATOS ===
2
3 import pandas as pd
4 # (asegúrate de haber copiado la función anova_completo al
5 # inicio)

```

```

6 MIS_DATOS = pd.DataFrame({
7     'nombre_factor': [ 'Nivel1' ]*N + [ 'Nivel2' ]*N +
8         [ 'Nivel3' ]*N,
9     'nombre_respuesta': [
10        # Aquí van tus datos del Nivel1
11        dato1, dato2, dato3, ...,
12        # Aquí van tus datos del Nivel2
13        dato1, dato2, dato3, ...,
14        # Aquí van tus datos del Nivel3
15        dato1, dato2, dato3, ...
16    ]
17 })
18 # Ejecutar el análisis completo
19 resultado = anova_completo(
20     MIS_DATOS,
21     'nombre_factor',      # columna del tratamiento
22     'nombre_respuesta',   # columna de la medición
23     alpha=0.05            # nivel de significancia
24 )

```

## 5.2. Checklist paso a paso

1. **Organiza tus datos:** Necesitas dos columnas: una con el nombre del tratamiento/grupo y otra con el valor medido. Cada fila es una observación.
2. **Reemplaza en la plantilla:** Cambia nombre\_factor, nombre\_respuesta, los niveles (Nivel1, Nivel2...) y tus datos numéricos.
3. **Corre el script** en Replit. Lee el output de las etapas 2 y 3 para saber si los supuestos se cumplen.
4. **Interpreta el p-value:** Si p menor que 0.05, hay diferencia significativa. Lee el output de Tukey para saber entre cuáles grupos.
5. **Reporta en tu informe:** Incluye la prueba usada, el estadístico, el p-value, y las comparaciones post-hoc si aplica.

### 5.3. Tabla de interpretación de resultados

Resultado que ves	Qué significa	Qué hacer
Shapiro p menor 0.05 en algún grupo	Ese grupo no es normal	El script usará Kruskal-Wallis; revisa si hay valores atípicos
Levene p menor 0.05	Varianzas desiguales entre grupos	El script nota Welch; con grupos balanceados ANOVA sigue siendo robusto
ANOVA/K-W p menor 0.05	Hay diferencia entre al menos un par	Leer Tukey HSD para saber cuáles
ANOVA/K-W p mayor 0.05	No hay evidencia de diferencias	No rechazar $H_0$ ; Tukey no se ejecuta
Tukey: <code>reject=True</code>	Ese par de grupos difiere significativamente	Reportar la diferencia y su intervalo de confianza
Tukey: <code>reject=False</code>	Ese par no difiere significativamente	No hay evidencia de que esos dos grupos sean distintos

### Apéndice: Referencia Rápida

## Tabla resumen de pruebas

Prueba	Hipótesis nula (H0)	Cuándo usar	Interpretar
Shapiro-Wilk	Los datos del grupo siguen distribución normal	Siempre, antes del ANOVA	p mayor 0.05: datos normales; p menor 0.05: no normales
Levene	Las varianzas de todos los grupos son iguales	Siempre, antes del ANOVA	p mayor 0.05: varianzas iguales; p menor 0.05: varianzas distintas
ANOVA Fisher	Las medias de todos los grupos son iguales	Cuando datos son normales	p menor alpha: diferencia significativa
Kruskal-Wallis	Las distribuciones de los grupos son iguales	Cuando datos no son normales	p menor alpha: distribuciones distintas
Tukey HSD	Las medias del par son iguales	Después de ANOVA significativo	reject=True: par difiere; meandiff da la magnitud
Dunn	Medianas del par son iguales	Después de K-W significativo	Requiere scikit-posthocs

## Glosario de términos clave

### Shapiro-Wilk

Prueba estadística que evalúa si una muestra de datos proviene de una población con distribución normal. El estadístico W varía entre 0 y 1; valores cercanos a 1 indican mayor normalidad. Un valor-p menor que 0.05 indica desviación significativa de la normalidad.

### Levene

Prueba para verificar la igualdad de varianzas entre dos o más grupos (homocedasticidad). Es más robusta que el test de Bartlett cuando los datos no son perfectamente normales. Un valor-p menor que 0.05 indica varianzas significativamente diferentes.

### Tukey HSD

(Honestly Significant Difference) Prueba post-hoc que compara todos los pares posibles de grupos después de un ANOVA significativo. Controla el error tipo I acumulado (FWER) ajustando los valores-p. Devuelve la diferencia de medias, un intervalo de confianza y una decisión de rechazo para cada par.

### Kruskal-Wallis

Alternativa no paramétrica al ANOVA de un factor. En lugar de trabajar con los

valores originales, usa los rangos de los datos. No requiere normalidad pero sí que las distribuciones de los grupos tengan forma similar (solo cambia la locación). El estadístico H sigue aproximadamente una distribución chi-cuadrado.

### **Valor-p (p-value)**

Probabilidad de observar un resultado tan extremo como el obtenido, asumiendo que la hipótesis nula es verdadera. Un valor-p pequeño (menor que alpha, usualmente 0.05) indica que los datos son poco compatibles con  $H_0$ , lo que lleva a rechazarla. No es la probabilidad de que  $H_0$  sea verdadera.

### **FWER (Family-Wise Error Rate)**

Probabilidad de cometer al menos un error tipo I (falso positivo) en una familia de comparaciones. Tukey HSD controla el FWER al nivel alpha, independientemente del número de pares comparados.

### **Intervalo de Confianza al 95%**

Rango de valores dentro del cual se espera que caiga el parámetro poblacional (por ejemplo, la media verdadera) en el 95% de los experimentos repetidos con el mismo diseño. Un IC estrecho indica mayor precisión en la estimación.

### **Homocedasticidad**

Supuesto estadístico que indica que todos los grupos tienen la misma varianza poblacional. El ANOVA es razonablemente robusto a violaciones leves de este supuesto, especialmente cuando los tamaños de grupo son iguales.

## **Q2003B – Diseño de Experimentos**

Sesión 3: ANOVA Completo con Post-hoc y Validación de Supuestos

Script: `anova_completo.py` | Febrero 2026