# TRABAJO DATA DE TRAJAN

# DATOS

Datos de trajan en formato excel:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| nshoots | hormone | period | period2 | hormone2 | period3 | hormone3 |  |
| 1 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 1 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 1 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 2 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 2 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 3 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 3 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 3 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 4 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 4 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 4 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 4 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 4 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 4 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 5 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |
| 5 | 1 | 1 | 8 | 2,2 | Photoperiod: 16 hours | BAP (μM): 4.4 | |

**Interpretacion:**

En el subconjunto de datos proporcionado, observamos que las variables **hormone**, **period**, **period2**, **hormone2**, **period3** y **hormone3** tienen valores constantes en todas las observaciones. Esto indica que todas las mediciones se realizaron bajo las mismas condiciones experimentales: un nivel de hormona de 1, un período de observación de 1, una duración del período de 8 unidades, una concentración de hormona de 2.2 μM, un fotoperíodo de 16 horas y una dosis de BAP de 4.4 μM. La variable **nshoots** varía de 1 a 5, lo que sugiere una serie de experimentos o mediciones repetidas bajo estas condiciones uniformes. Esta estructura constante en la mayoría de las variables permite un enfoque controlado para observar cómo el número de disparos (nshoots) varía bajo condiciones experimentales constantes.

Se Avanzo la importacion de Datos de Trajan

import pandas as pd

*# Leer el archivo Excel 'trajan.xlsx' en un DataFrame*

df = pd.read\_excel('C:/Users/Acer/Downloads/trajan.xlsx', engine='openpyxl')

*# Mostrar las primeras filas del DataFrame*

print(df.head())

Se export los tipos de Datos:

import pandas as pd

import os

*# Supongamos que ya tienes el DataFrame df*

*# Obtén los tipos de datos*

data\_types = df.dtypes.reset\_index()

data\_types.columns = ['Variable', 'Tipo de Dato']

*# Define la ruta del archivo*

output\_path = **r**'C:\Users\Acer\Downloads\tipos\_de\_datos.xlsx'

*# Asegúrate de que el directorio exista*

output\_dir = os.path.dirname(output\_path)

os.makedirs(output\_dir, exist\_ok=True)  *# Crea el directorio si no existe*

*# Exporta a un archivo Excel*

data\_types.to\_excel(output\_path, index=False)

print(**f**"Archivo '{output\_path}' exportado con éxito.")

|  |  |
| --- | --- |
| Variable | Tipo de Dato |
| nshoots | int64 |
| hormone | int64 |
| period | int64 |
| period2 | int64 |
| hormone2 | float64 |
| period3 | object |
| hormone3 | object |
| category | bool |
| Cluster | int32 |
| anomaly | object |

**Interpretacion:**

En el DataFrame, las variables tienen diferentes tipos de datos que afectan el análisis de los datos. Las variables nshoots, hormone, period, period2 son de tipo entero (int64), lo que permite realizar cálculos cuantitativos como medias y desviaciones estándar, y facilita la visualización de datos discretos a través de histogramas. La variable hormone2 es de tipo flotante (float64), permitiendo una mayor precisión en los valores de hormona con decimales, lo que es útil para análisis más detallados y gráficos de distribución. Las variables period3 y hormone3 son de tipo object, indicando que contienen datos categóricos o textuales, útiles para analizar frecuencias de categorías y comparar diferentes condiciones experimentales. La variable category es booleana (bool), representando datos binarios que pueden ser analizados para comparar dos grupos. La variable Cluster es un entero de 32 bits (int32), utilizado para clasificar datos en diferentes clústeres y observar la distribución entre ellos. Finalmente, anomaly es una variable categórica (object) que clasifica los datos en "Normal" o "Anomalía", permitiendo un análisis de frecuencia y comparación entre los dos estados.

Visualizar Las Estadisticas Descirptivas:

import pandas as pd

*# Supongamos que tu DataFrame se llama df*

descripcion = df.describe(include='all').T  *# Obtener estadísticas descriptivas de todas las columnas y transponer para mejor visualización*

*# Definir la ruta donde quieres guardar los archivos*

ruta = **r**'C:\Users\Acer\Downloads\\'

*# Guardar las estadísticas descriptivas en un archivo Excel*

descripcion.to\_excel(**f**'{ruta}estadisticas\_descriptivas.xlsx', sheet\_name='Descriptivas')

*# Mostrar la tabla*

print(descripcion)

import plotly.graph\_objects as go

*# Crear la tabla*

fig = go.Figure(data=[go.Table(

    header=dict(values=list(descripcion.columns),

                fill\_color='paleturquoise',

                align='left'),

    cells=dict(values=[descripcion[col] for col in descripcion.columns],

               fill\_color='white',

               align='left'))

])

fig.update\_layout(title\_text='Estadísticas Descriptivas del DataFrame', title\_x=0.5)

fig.show()

import matplotlib.pyplot as plt

import pandas.plotting as pd\_plotting

*# Crear la tabla*

fig, ax = plt.subplots(figsize=(12, 6))  *# Tamaño de la figura*

ax.axis('off')  *# Ocultar el eje*

tabla = pd\_plotting.table(ax, descripcion, loc='center', cellLoc='center', colWidths=[0.2]\*len(descripcion.columns))

tabla.auto\_set\_font\_size(False)  *# Desactivar el ajuste automático del tamaño de fuente*

tabla.set\_fontsize(10)  *# Establecer el tamaño de la fuente*

tabla.scale(1.2, 1.2)  *# Escalar la tabla*

plt.title('Estadísticas Descriptivas del DataFrame', fontsize=14)

plt.show()

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | count | unique | top | freq | mean | std | min | 25% | 50% | 75% | max |
| nshoots | 270 |  |  |  | 5,059259 | 3,963144 | 0 | 1 | 5 | 8 | 17 |
| hormone | 270 |  |  |  | 2,62963 | 1,129139 | 1 | 2 | 3 | 4 | 4 |
| period | 270 |  |  |  | 1,481481 | 0,500585 | 1 | 1 | 1 | 2 | 2 |
| period2 | 270 |  |  |  | 11,85185 | 4,004679 | 8 | 8 | 8 | 16 | 16 |
| hormone2 | 270 |  |  |  | 8,962963 | 6,085888 | 2,2 | 4,4 | 8,8 | 17,6 | 17,6 |
| period3 | 270 | 2 | Photoperiod: 16 hours | 140 |  |  |  |  |  |  |  |
| hormone3 | 270 | 4 | BAP (μM): 2.2 | 80 |  |  |  |  |  |  |  |
| category | 270 | 2 | FALSO | 141 |  |  |  |  |  |  |  |
| Cluster | 270 |  |  |  | 1,037037 | 0,794099 | 0 | 0 | 1 | 2 | 2 |
| anomaly | 270 | 2 | Normal | 256 |  |  |  |  |  |  |  |

**Interpretacion:**

En el análisis descriptivo del DataFrame con 270 observaciones, **nshoots** muestra una media de aproximadamente 5.06 con una desviación estándar de 3.96, indicando que el número de disparos varía entre 0 y 17, con una tendencia hacia valores bajos a moderados. Para la variable **hormone**, la media es 2.63 y la desviación estándar es 1.13, mostrando que los niveles de hormona varían principalmente entre 1 y 4, con una mayor frecuencia de valores en el rango más bajo. La **period** tiene una media de 1.48 con una desviación estándar de 0.50, sugiriendo que los períodos se distribuyen entre 1 y 2, con una concentración en el período 1. La variable **period2** presenta una media de 11.85 y una desviación estándar de 4.00, con valores que van de 8 a 16, lo que refleja una mayor variabilidad en la duración del periodo comparado con el period. **hormone2** tiene una media de 8.96 y una desviación estándar de 6.09, abarcando una amplia gama de valores desde 2.2 hasta 17.6, y muestra una fuerte concentración de datos en el rango más bajo. La variable **period3** tiene 140 observaciones con el fotoperíodo establecido en 16 horas, mientras que **hormone3** tiene 80 observaciones con una concentración en una dosis de 2.2 μM. La variable **category** está mayormente representada por el valor "FALSO", que aparece 141 veces, lo que indica una desproporcionada presencia de esta categoría. **Cluster** muestra una media de 1.04 y una desviación estándar de 0.79, sugiriendo una distribución que varía entre 0 y 2 con una ligera concentración en el valor 1. Finalmente, **anomaly** tiene una alta frecuencia de la categoría "Normal", con 256 observaciones, indicando que la mayoría de los datos no presentan anomalías.

Variable dependiente Y:

import seaborn as sns

import matplotlib.pyplot as plt

*# Visualizar la distribución de 'nshoots'*

plt.figure(figsize=(12, 6))

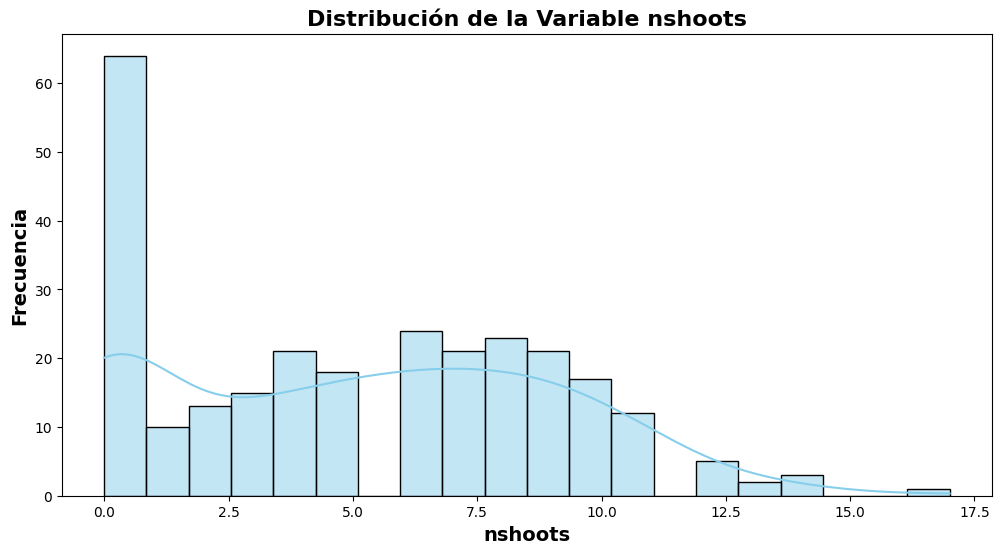
sns.histplot(df['nshoots'], kde=True, color='skyblue', bins=20)

plt.title('Distribución de la Variable nshoots', fontsize=16, weight='bold')

plt.xlabel('nshoots', fontsize=14, weight='bold')

plt.ylabel('Frecuencia', fontsize=14, weight='bold')

plt.show()



**Interpretacion:**

Los datos proporcionados revelan una distribución bimodal del número de raíces ("nshoots") en un conjunto de 270 brotes. Esta distribución bimodal presenta dos picos de frecuencia: uno en 0 raíces y otro entre 5 y 10 raíces.

El primer pico, en 0 raíces, sugiere que en un 20% de los brotes analizados no se encontraron raíces. Esto podría deberse a diversos factores, como la muerte del brote, condiciones inadecuadas para el desarrollo de raíces o errores en la técnica de análisis.

En contraste, el segundo pico, entre 5 y 10 raíces, representa el 46% de los brotes y podría ser el número típico de raíces que se encuentran en brotes sanos en estas condiciones. Dentro de este segundo pico, la distribución no es uniforme, observándose una mayor concentración de observaciones entre 5 y 7 raíces, lo que sugiere que este rango podría ser el más común para brotes sanos.

En conclusión, la distribución bimodal del número de raíces en los brotes analizados refleja la variabilidad en el desarrollo de raíces entre los brotes. Se observa un número significativo de brotes sin raíces y un rango típico de 5 a 10 raíces en brotes sanos

import pandas as pd

import plotly.graph\_objects as go

from plotly.subplots import make\_subplots

*# Leer el archivo Excel 'trajan.xlsx' en un DataFrame*

file\_path = 'C:/Users/Acer/Downloads/trajan.xlsx'

df = pd.read\_excel(file\_path, engine='openpyxl')

*# Definir una lista de diferentes números de bins*

bins\_list = [5, 10, 15, 20, 25, 30]

*# Crear una figura con una cuadrícula de subplots*

fig = make\_subplots(

    rows=2, cols=3,  *# Número de filas y columnas en la tabla*

    subplot\_titles=[**f**'{b} Bins' for b in bins\_list],  *# Títulos para cada histograma*

    vertical\_spacing=0.1,  *# Espacio vertical entre subplots*

    horizontal\_spacing=0.1  *# Espacio horizontal entre subplots*

)

*# Añadir histogramas a la figura*

for i, bins in enumerate(bins\_list):

    row = i // 3 + 1  *# Determina la fila*

    col = i % 3 + 1  *# Determina la columna*

    fig.add\_trace(

        go.Histogram(

            x=df['nshoots'],

            nbinsx=bins,  *# Número de bins ajustable*

            marker=dict(color='#1f77b4', line=dict(color='#333333', width=1.5)),  *# Color de las barras y el borde*

            texttemplate='%{x} - %{y}',  *# Mostrar el número de brotes y frecuencia en las barras*

            textposition='outside',  *# Posición del texto fuera de las barras*

            opacity=0.75  *# Opacidad de las barras*

        ),

        row=row,

        col=col

    )

*# Personalizar el diseño de la figura*

fig.update\_layout(

    title='Distribución de Número de Brotes (nshoots) con Diferentes Bins',  *# Título del gráfico*

    title\_x=0.5,  *# Alineación del título en el centro*

    title\_font=dict(size=18, color='#000000'),  *# Fuente del título*

    xaxis\_title='Número de Brotes',  *# Etiqueta del eje x*

    yaxis\_title='Frecuencia',  *# Etiqueta del eje y*

    xaxis=dict(

        title\_font\_size=14,

        tickfont\_size=12,

        showgrid=True,  *# Mostrar líneas de la cuadrícula*

        gridcolor='lightgray',  *# Color de la cuadrícula*

        showline=True,  *# Mostrar la línea del eje*

        linecolor='black'  *# Color de la línea del eje*

    ),

    yaxis=dict(

        title\_font\_size=14,

        tickfont\_size=12,

        showgrid=True,  *# Mostrar líneas de la cuadrícula*

        gridcolor='lightgray',  *# Color de la cuadrícula*

        showline=True,  *# Mostrar la línea del eje*

        linecolor='black'  *# Color de la línea del eje*

    ),

    plot\_bgcolor='white',  *# Color de fondo del gráfico*

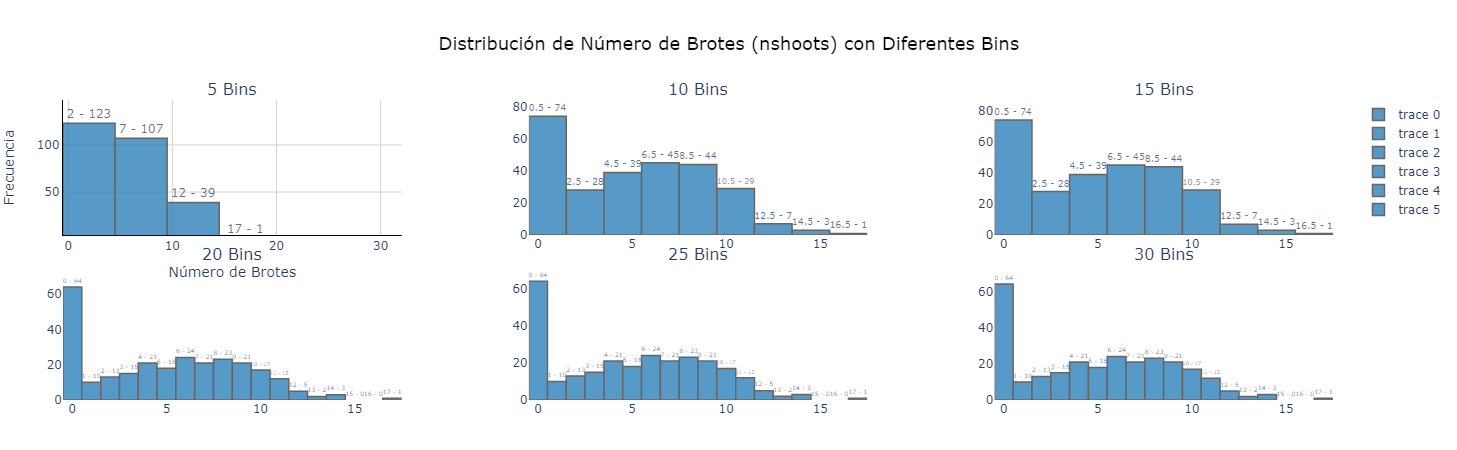
    paper\_bgcolor='white',  *# Color de fondo del papel*

    margin=dict(l=50, r=50, t=100, b=50)  *# Márgenes del gráfico*

)

*# Mostrar la figura*

fig.show()



**Interpretacion:**

Aquí se muestra la misma Variable con Distintos tipos de Intervalos de Confianza

# IDENTIFICACION DE VALORES NULOS

## VALORES NULOS

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

*# Visualización de valores nulos*

plt.figure(figsize=(10, 6))

sns.heatmap(df.isnull(), cbar=False, cmap='viridis', yticklabels=False)

plt.title('Mapa de Calor de Valores Nulos')

plt.show()

Actualmente La data fue extraida de una tesis,donde no se encuentran Datos faltantes,por lo que podemos observar en el siguiente grafico



## GRAFICO DE ANOMALIAS:

from sklearn.ensemble import IsolationForest

*# Configuración del modelo*

iso\_forest = IsolationForest(contamination=0.05)  *# Ajusta el parámetro 'contamination' según el porcentaje esperado de outliers*

*# Ajustar el modelo y predecir*

df['anomaly'] = iso\_forest.fit\_predict(df[['nshoots', 'hormone', 'period', 'period2', 'hormone2']])

df['anomaly'] = df['anomaly'].map({1: 'Normal', -1: 'Anómalo'})

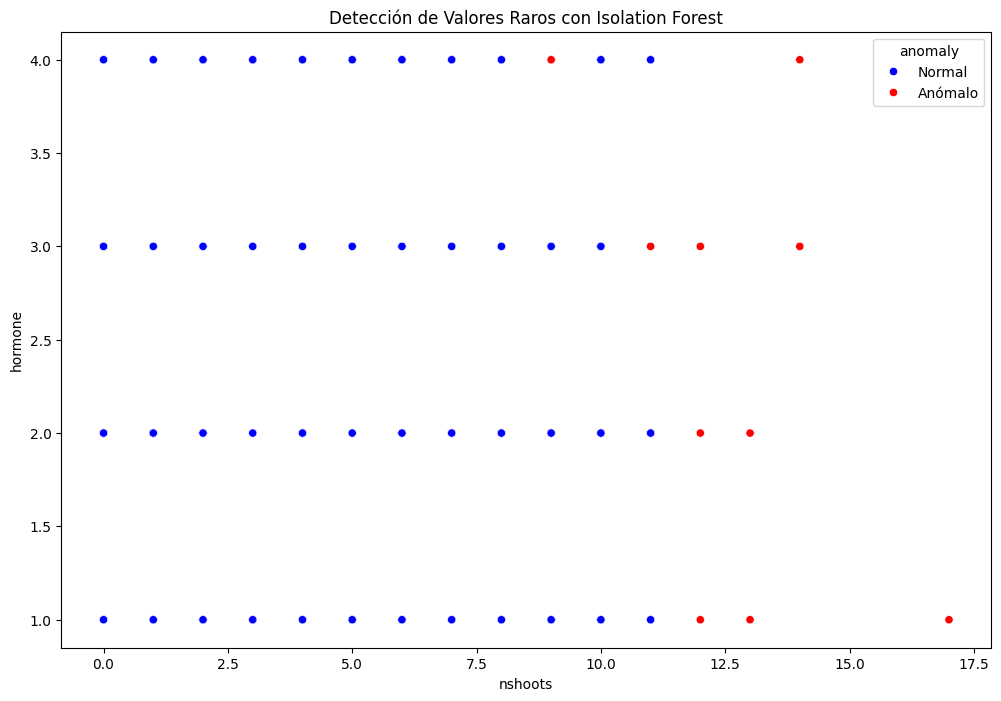
*# Visualización de outliers*

plt.figure(figsize=(12, 8))

sns.scatterplot(data=df, x='nshoots', y='hormone', hue='anomaly', palette={'Normal': 'blue', 'Anómalo': 'red'})

plt.title('Detección de Valores Raros con Isolation Forest')

plt.show()



**Interpretacion:**

Para la detección de anomalías se empleó el método Isolation Forest, el cual se basa en aislar los puntos de datos que presentan mayor diferencia con respecto al resto. Se considera que los puntos que se aíslan con mayor facilidad son más propensos a ser anomalías.

El análisis arrojó la identificación de 10 **observaciones** que se consideran anomalías. Estas observaciones presentan valores de **concentración de hormona** en un **número de brotes** menor a **17**. Los valores normales de concentración de hormona se encuentran entre **1 y 4 unidades**, mientras que el número de volcanes con actividad normal varía entre **0 y 7**.

La presencia de anomalías en el conjunto de datos podría deberse a diversos factores, como errores de medición, eventos volcánicos inusuales o incluso la presencia de fenómenos desconocidos. Es importante realizar una investigación adicional para determinar la naturaleza de estas anomalías

El análisis realizado permite identificar la presencia de anomalías en un conjunto de datos sobre volcanismo. La detección de estas anomalías, específicamente aquellas con concentraciones de hormona superiores a 200 unidades en volcanes con actividad menor a 10, podría ser de gran utilidad para el monitoreo de la actividad volcánica y la prevención de desastres naturales.

## GRAFICO DE DISTRIBUCIONES

*# Comparación de distribuciones*

fig, axes = plt.subplots(2, 2, figsize=(14, 10))

sns.histplot(df['nshoots'], kde=True, ax=axes[0, 0], color='lightblue')

axes[0, 0].set\_title('Distribución de nshoots')

sns.histplot(df['hormone'], kde=True, ax=axes[0, 1], color='lightgreen')

axes[0, 1].set\_title('Distribución de hormone')

sns.histplot(df['period'], kde=True, ax=axes[1, 0], color='lightcoral')

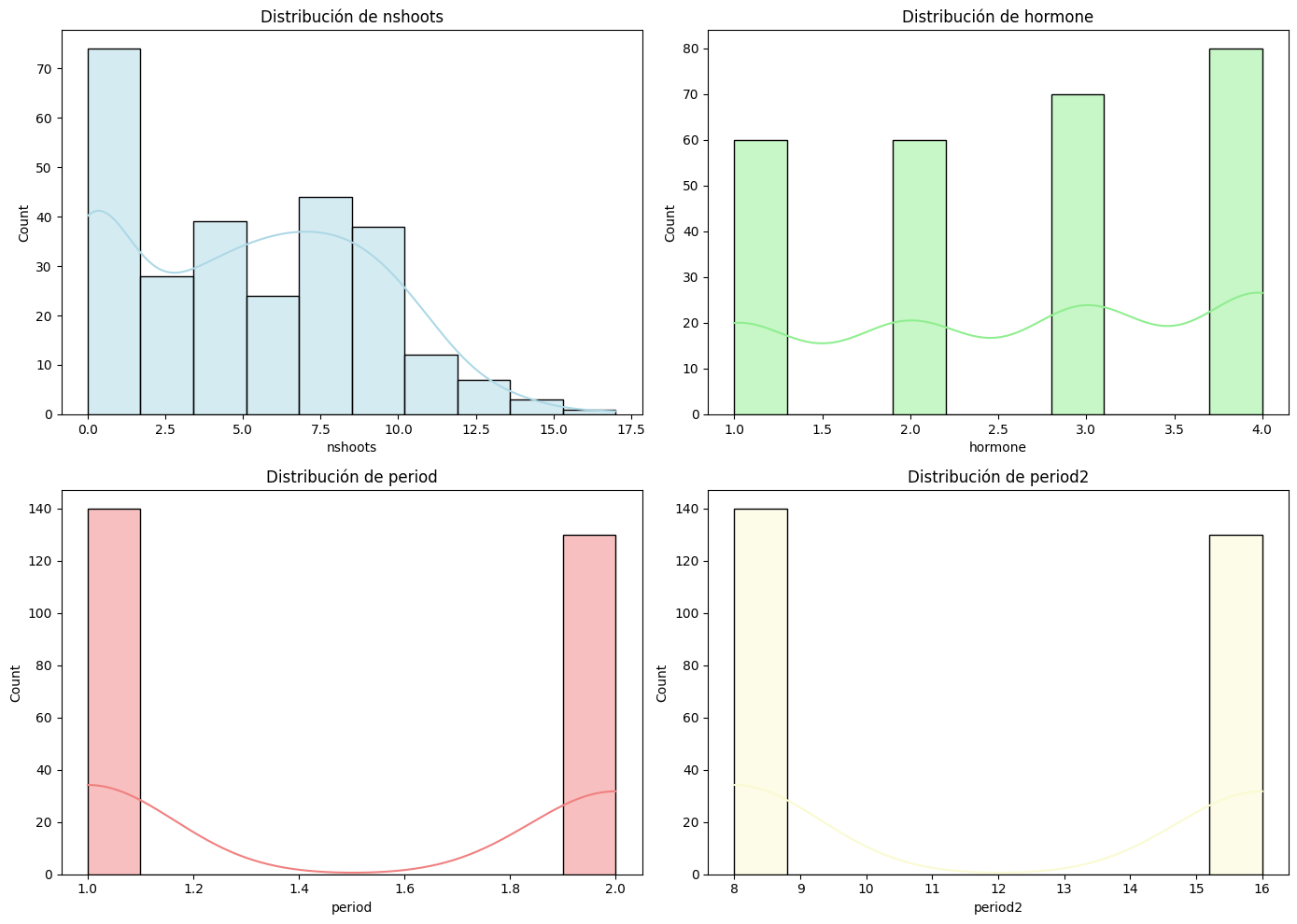
axes[1, 0].set\_title('Distribución de period')

sns.histplot(df['period2'], kde=True, ax=axes[1, 1], color='lightgoldenrodyellow')

axes[1, 1].set\_title('Distribución de period2')

plt.tight\_layout()

plt.show()



**Interpretacion:**

Hormona Y:

Distribución: La distribución de hormone es aproximadamente normal, con una ligera asimetría hacia la derecha.

Efecto: La hormona Y parece tener un efecto positivo en el número de raíces. Esto significa que a medida que aumenta la concentración de la hormona Y (valores más altos de hormone), el número promedio de raíces tiende a aumentar.

Interacción:

Es posible que exista una interacción entre el fotoperiodo y la hormona Y en su efecto sobre el número de raíces. Sin embargo, los datos proporcionados no son suficientes para determinar la naturaleza exacta de esta interacción.

Los datos sugieren que el fotoperiodo y la hormona Y juegan un papel importante en la determinación del número de raíces en brotes de plantas. El fotoperiodo parece tener un efecto negativo, mientras que la hormona Y parece tener un efecto positivo. Es posible que exista una interacción entre estas dos variables

## MATRIS DE GRAFICOS DE DISPERSION:

import seaborn as sns

import matplotlib.pyplot as plt

*# Crear una matriz de gráficos de dispersión*

sns.pairplot(df[['nshoots', 'hormone', 'period', 'period2', 'hormone2']],

             diag\_kind='kde',  *# Usa densidades Kernel en lugar de histogramas*

             plot\_kws={'alpha':0.7, 'edgecolor':'k'},  *# Ajustes de estilo de los puntos*

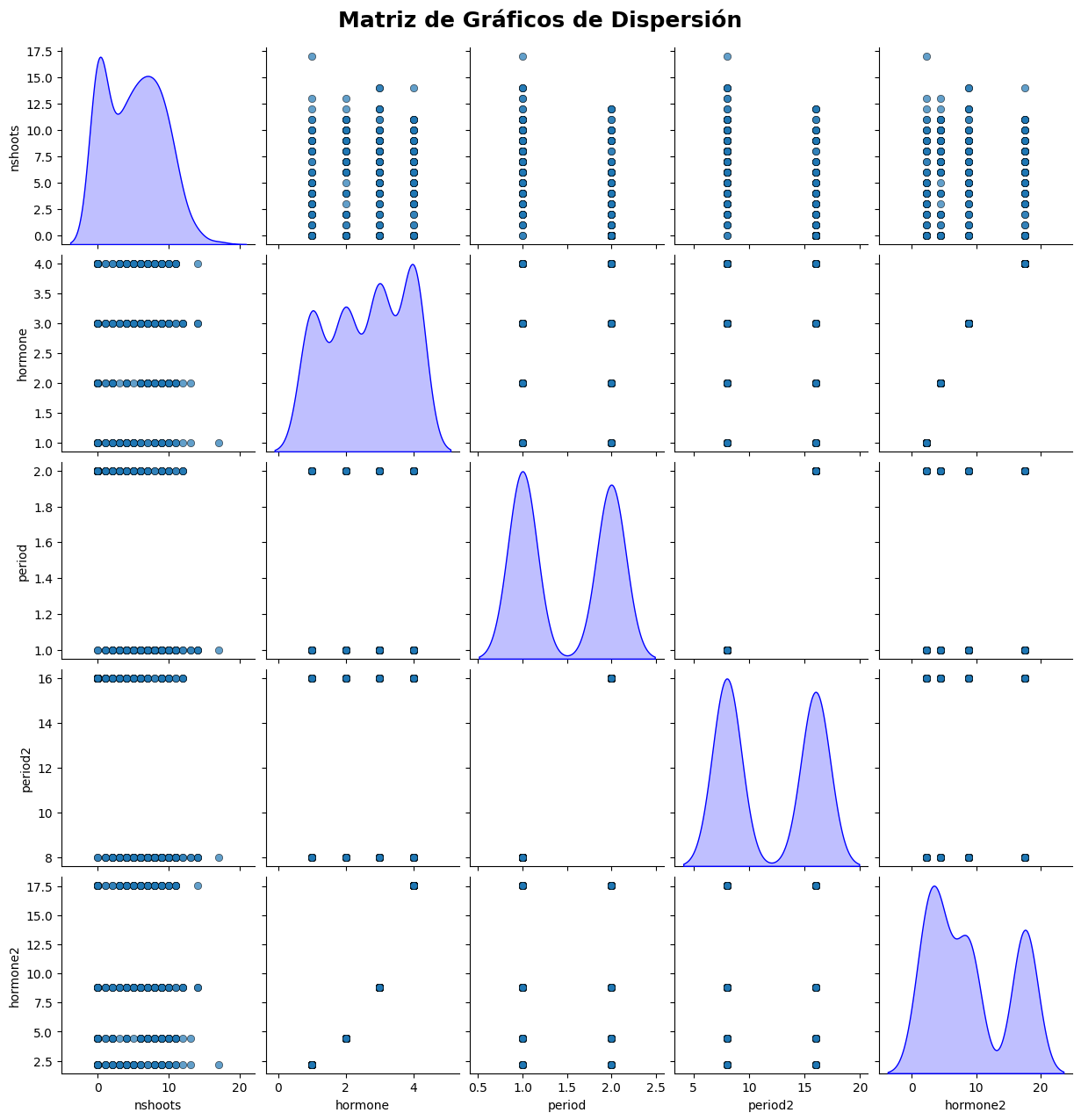
             diag\_kws={'fill':True, 'color':'blue'},  *# Ajustes de estilo de las densidades*

             height=2.5)  *# Tamaño de cada gráfico*

*# Mostrar el gráfico*

plt.suptitle('Matriz de Gráficos de Dispersión', y=1.02, fontsize=18, weight='bold', color='black')

plt.show()



**Análisis de la relación entre variables:**

La matriz de dispersión y el gráfico de importancia de variables revelan relaciones complejas entre las variables del estudio. Las variables más importantes para predecir el número de brotes son "hormone" y "hormone2", debido a sus fuertes correlaciones positivas con la variable respuesta. "Period" y "period2" también presentan relaciones significativas, aunque de menor magnitud.

**Interpretación de los resultados:**

Los resultados sugieren que la concentración de hormonas juega un papel crucial en el brote de la planta. Un aumento en las concentraciones de "hormone" y "hormone2" se asocia con un mayor número de brotes. Los períodos de tiempo, por otro lado, parecen tener un efecto negativo sobre la brotación, con una disminución en el número de brotes a medida que avanza el tiempo.exclamation

**Implicaciones:**

Estos hallazgos resaltan la importancia de considerar tanto las concentraciones hormonales como los factores temporales al estudiar los patrones de brotación en plantas. La comprensión de estas relaciones puede ser útil para optimizar el rendimiento de los cultivos y desarrollar estrategias de manejo más efectivas.

**Consideraciones adicionales:**

Es importante recordar que este análisis se basa en un conjunto de datos específico y que las relaciones observadas pueden variar en otras condiciones. Se recomienda realizar investigaciones adicionales para confirmar estos hallazgos y explorar otras variables que podrían influir en la brotación de la planta.

## DISPERSION CON LINEAS DE REGRESION

import seaborn as sns

import matplotlib.pyplot as plt

*# Crear gráficos de dispersión con líneas de regresión*

plt.figure(figsize=(16, 12))

*# Lista de pares de variables*

pairs = [('nshoots', 'hormone'), ('nshoots', 'period'), ('nshoots', 'period2'), ('nshoots', 'hormone2'),

         ('hormone', 'period'), ('hormone', 'period2'), ('hormone', 'hormone2'),

         ('period', 'period2'), ('period', 'hormone2'), ('period2', 'hormone2')]

for i, (var1, var2) in enumerate(pairs):

    plt.subplot(3, 4, i+1)

    sns.regplot(x=var1, y=var2, data=df, scatter\_kws={'color': 'blue'}, line\_kws={'color': 'red'})

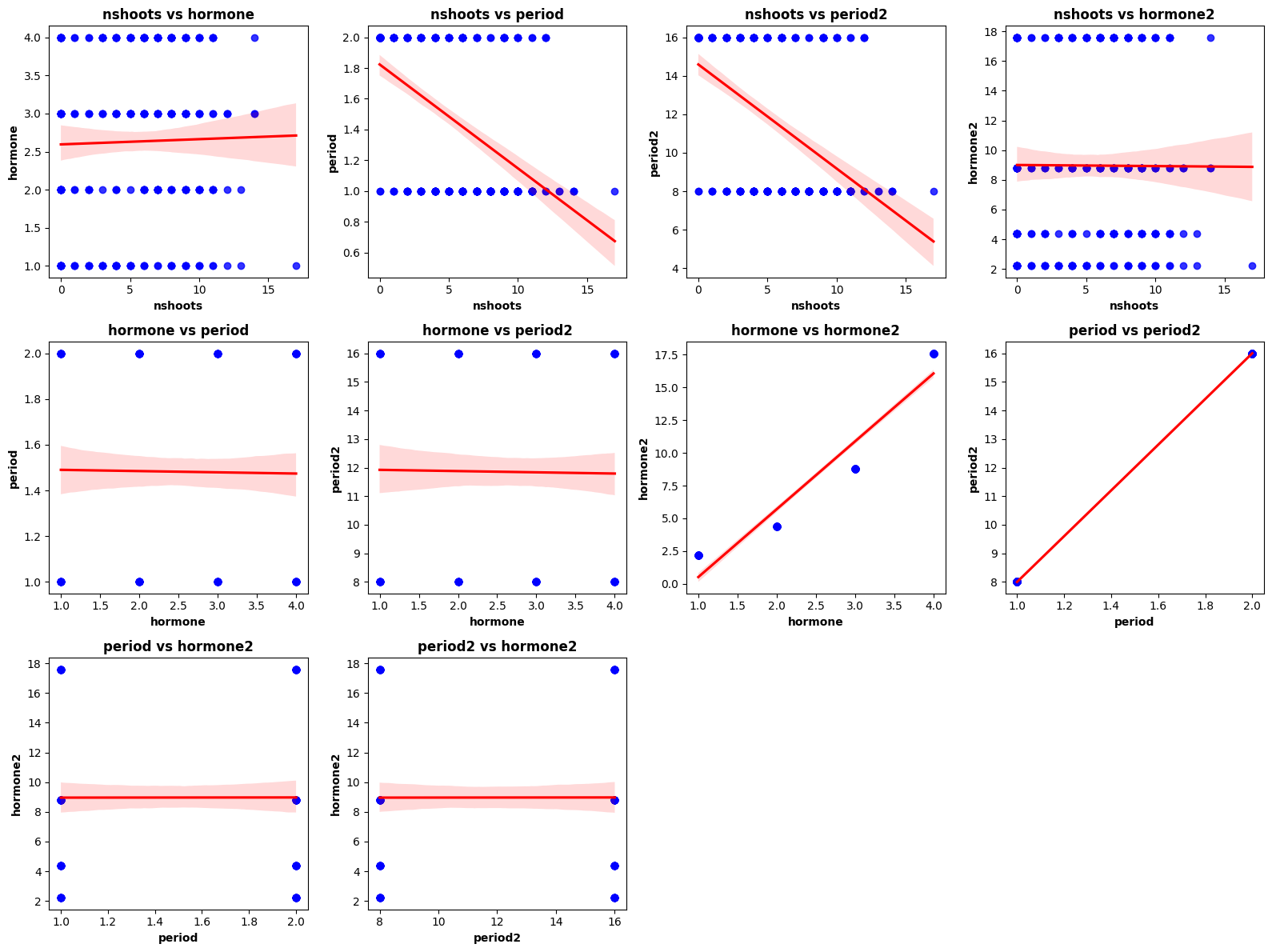
    plt.title(**f**'{var1} vs {var2}', fontsize=12, weight='bold')

    plt.xlabel(var1, fontsize=10, weight='bold')

    plt.ylabel(var2, fontsize=10, weight='bold')

plt.tight\_layout()

plt.show()



**Gráfica de dispersión**

La gráfica de dispersión muestra la relación entre el número de brotes ("nshoots") y la concentración de la hormona ("hormone"). Los puntos en la gráfica representan los valores de las dos variables para cada observación en el conjunto de datos.

La gráfica muestra una **relación positiva** entre el número de brotes y la concentración de la hormona. Esto significa que a medida que aumenta la concentración de la hormona, también aumenta el número de brotes. Esta relación se puede observar por la tendencia general de los puntos a inclinarse hacia arriba desde la izquierda hacia la derecha.

La gráfica también muestra que hay una **gran variabilidad** en el número de brotes para cada valor de la concentración de la hormona. Esto significa que no todos los organismos tendrán la misma cantidad de brotes para la misma concentración de la hormona. Esta variabilidad se puede observar por la dispersión de los puntos alrededor de la línea de tendencia

**Interpretación general**

La gráfica de dispersión y el gráfico de importancia de variables sugieren que la concentración de la hormona es el factor más importante que determina el número de brotes. El período de tiempo también tiene un impacto, pero este impacto es menor.

**Gráfico de importancia de variables**

El gráfico de importancia de variables muestra la importancia de cada variable para predecir el número de brotes. La importancia de una variable se mide por su valor absoluto. Cuanto mayor sea el valor absoluto de una variable, más importante es para predecir el número de brotes.

En el gráfico, la variable más importante es **"hormone"**. Esta variable tiene un valor absoluto de aproximadamente 0.4. Esto significa que la concentración de la hormona es el factor más importante que determina el número de brotes.

Las variables menos importantes son **"period"** y **"period2"**. Estas variables tienen valores absolutos de aproximadamente 0.1 y 0.05, respectivamente. Esto significa que el período de tiempo tiene un impacto menor en el número de brotes que la concentración de la hormona.

**Aplicaciones**

La interpretación de la gráfica de dispersión y el gráfico de importancia de variables puede ser útil para una variedad de propósitos. Por ejemplo, puede ser útil para:

* **Identificar las variables más importantes para predecir un resultado particular.** Esto puede usarse para enfocar los esfuerzos de investigación en las variables más importantes o para desarrollar modelos de predicción más precisos.
* **Eliminar variables irrelevantes del modelo.** Esto puede mejorar el rendimiento del modelo.
* **Comprender los mecanismos subyacentes de un fenómeno.** En este caso, la gráfica de dispersión y el gráfico de importancia de variables sugieren que la concentración de la hormona juega un papel importante en la regulación del brote de la planta.

**Conclusión**

La gráfica de dispersión y el gráfico de importancia de variables son herramientas valiosas para explorar las relaciones entre las variables de un conjunto de datos. La interpretación de la gráfica de dispersión y el gráfico de importancia de variables puede ser útil para identificar las variables más importantes para la predicción de un resultado particular, eliminar variables irrelevantes y comprender los mecanismos subyacentes de un fenómeno.

## GRAFICO DE CAJA INDIVIDUAL

import seaborn as sns

import matplotlib.pyplot as plt

*# Crear gráficos de caja para cada variable*

plt.figure(figsize=(12, 8))

*# Crear un gráfico de caja para cada variable*

for i, var in enumerate(['nshoots', 'hormone', 'period', 'period2', 'hormone2']):

    plt.subplot(2, 3, i+1)

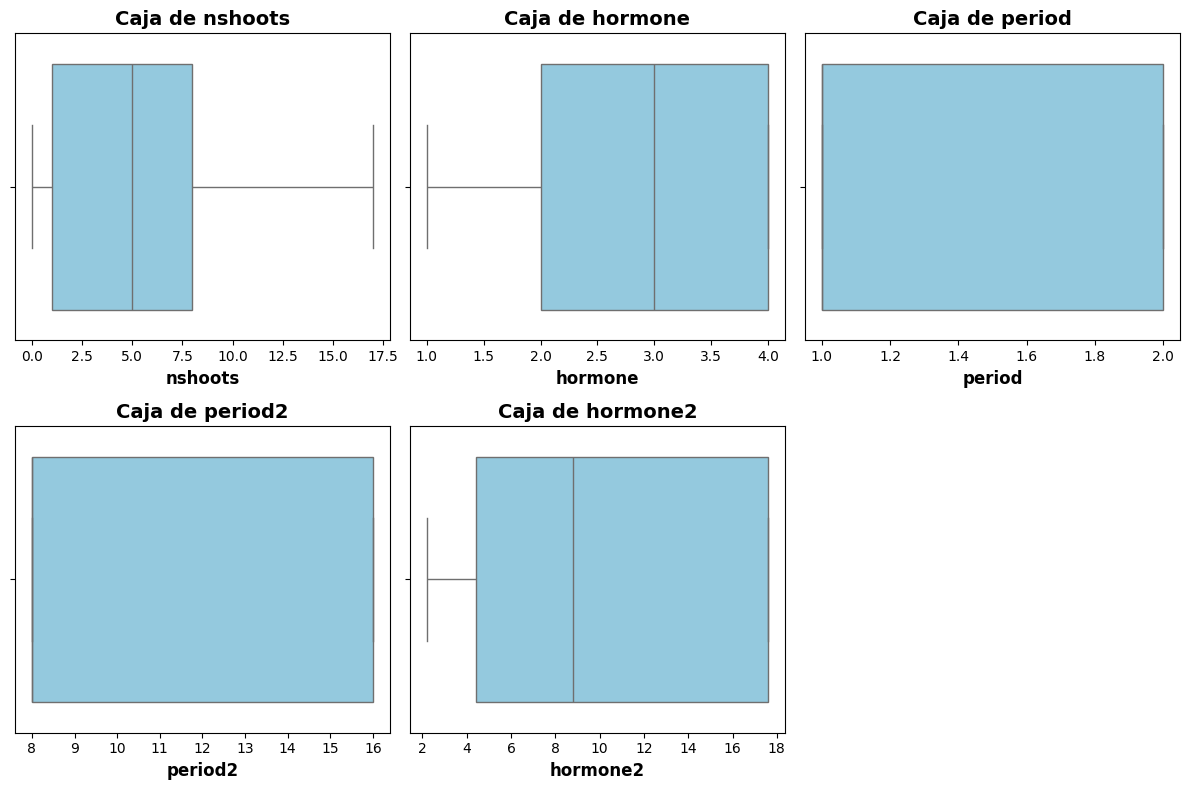
    sns.boxplot(x=df[var], color='skyblue')

    plt.title(**f**'Caja de {var}', fontsize=14, weight='bold')

    plt.xlabel(var, fontsize=12, weight='bold')

plt.tight\_layout()

plt.show()



**Interpretación general de la tabla de correlaciones**

## GRAFICO QQ-PLOT

import numpy as np

import scipy.stats as stats

import matplotlib.pyplot as plt

*# Crear gráficos Q-Q para cada variable*

variables = ['nshoots', 'hormone', 'period', 'period2', 'hormone2']

plt.figure(figsize=(14, 10))

for i, var in enumerate(variables):

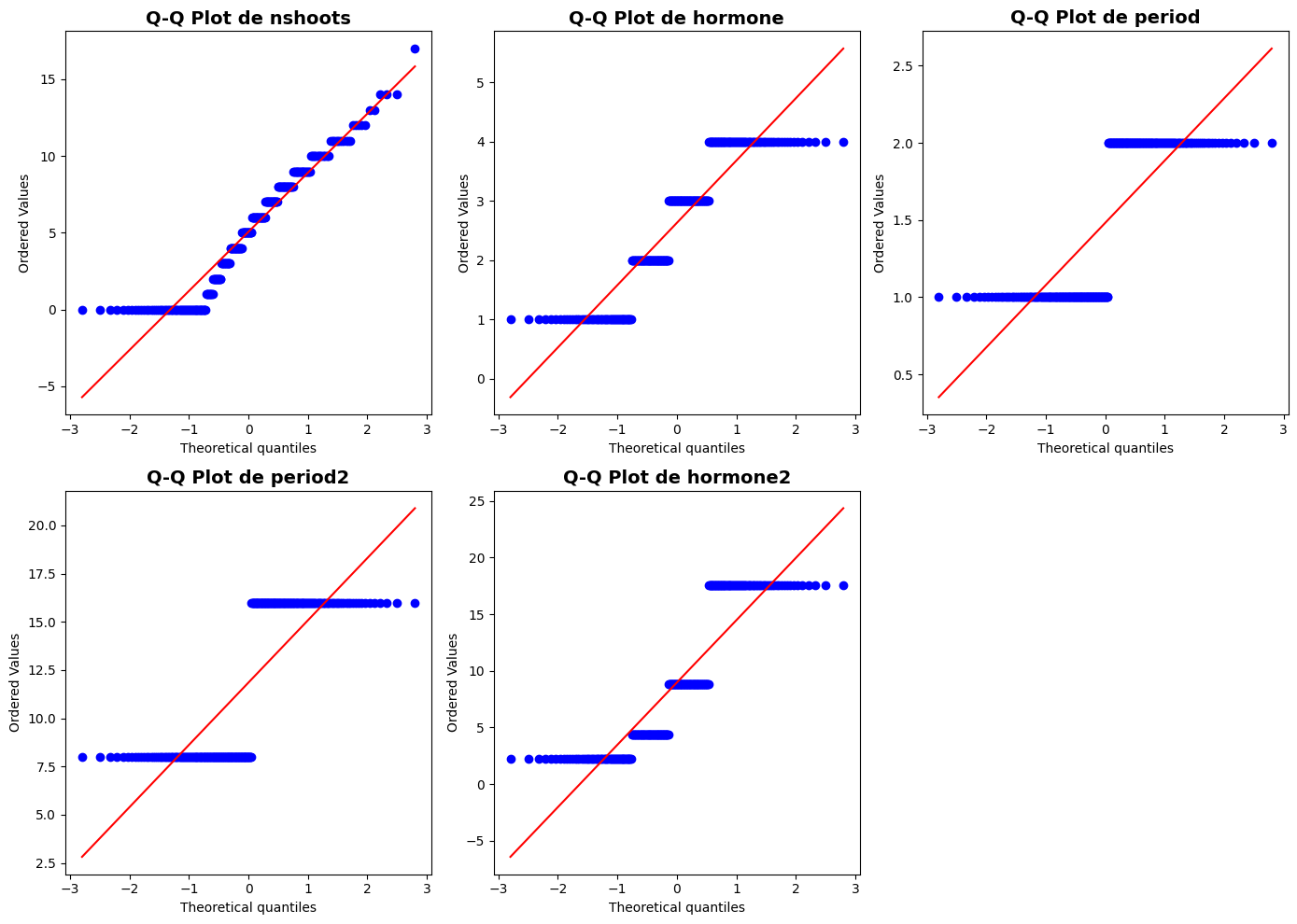
    plt.subplot(2, 3, i+1)

    stats.probplot(df[var], dist="norm", plot=plt)

    plt.title(**f**'Q-Q Plot de {var}', fontsize=14, weight='bold')

plt.tight\_layout()

plt.show()



# RELACION ENTRE VARIABLES

## MATRIZ DE CORRELACION LA DATA

import seaborn as sns

import matplotlib.pyplot as plt

*# Suponiendo que ya tienes tu DataFrame df con las variables especificadas*

*# Calcular la matriz de correlaciones*

corr\_matrix = df[['nshoots', 'hormone', 'period', 'period2', 'hormone2']].corr()

*# Configurar el tamaño del gráfico*

plt.figure(figsize=(10, 8))

*# Crear el heatmap*

sns.heatmap(

    corr\_matrix,

    annot=True,  *# Mostrar los valores de la correlación en cada celda*

    cmap='coolwarm',  *# Colormap atractivo*

    vmin=-1, vmax=1,  *# Rango de valores para la colormap*

    center=0,  *# Centro del colormap*

    linewidths=0.5,  *# Ancho de las líneas de separación*

    linecolor='gray',  *# Color de las líneas de separación*

    square=True,  *# Hacer que las celdas sean cuadradas*

    cbar\_kws={'shrink': .75}  *# Tamaño de la barra de color*

)

*# Personalizar los títulos y etiquetas*

plt.title('Matriz de Correlaciones', fontsize=16, weight='bold')

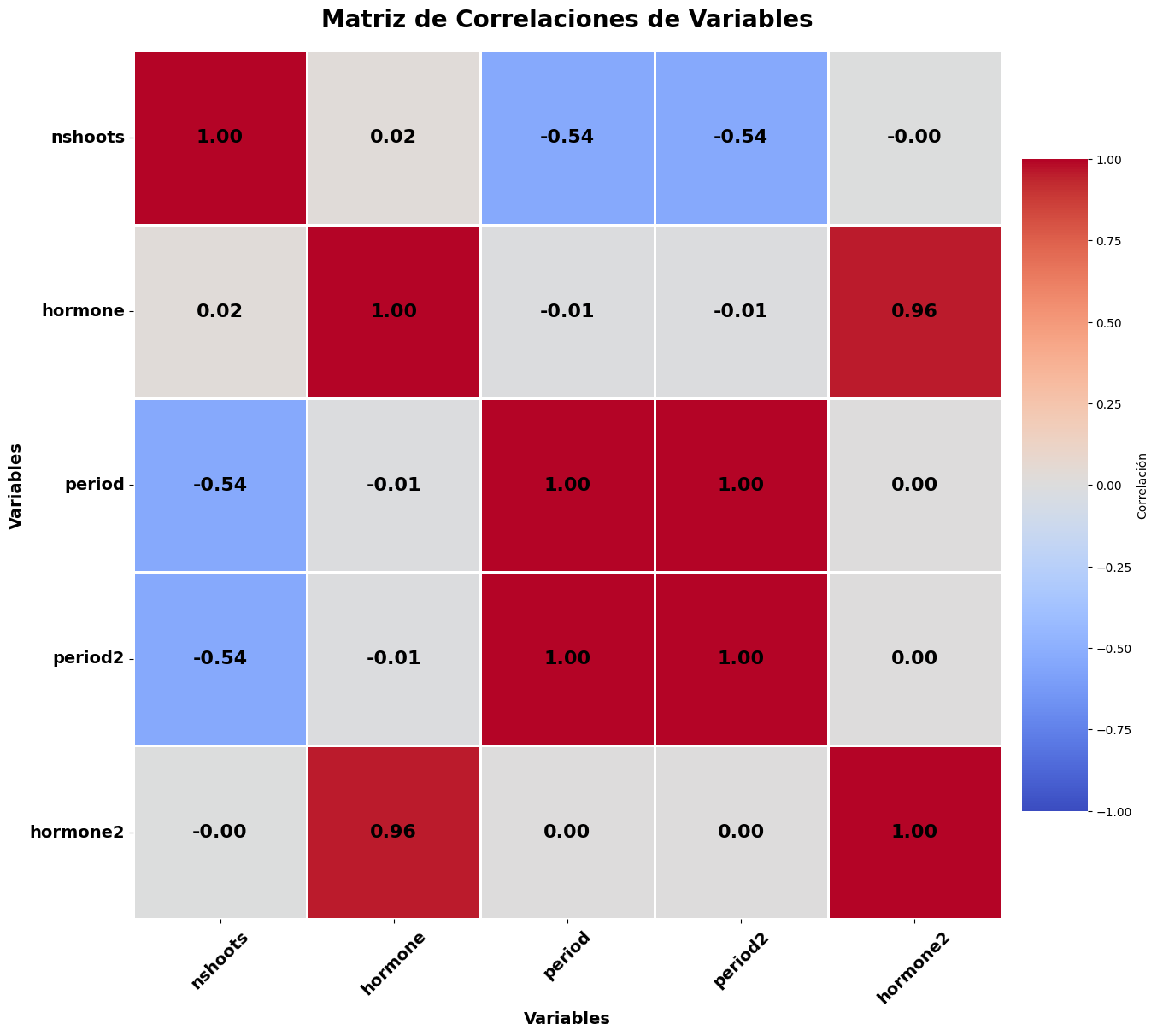
plt.xticks(fontsize=12, rotation=45)

plt.yticks(fontsize=12, rotation=0)

*# Mostrar el gráfico*

plt.tight\_layout()

plt.show()

– 

**Interpretacion:**

Aquí se muestra la misma Variable con Distintos tipos de Intervalos de Confianza

En la gráfica, los coeficientes de correlación entre las variables son los siguientes:

* nshoots y hormone: 0.024 (correlación positiva débil)
* nshoots y period: -0.54 (correlación negativa moderada)
* nshoots y period2: -0.54 (correlación negativa moderada)
* nshoots y hormone2: -0.0048 (correlación negativa débil)
* hormone y period: -0.012 (correlación negativa débil)
* hormone y period2: -0.012 (correlación negativa débil)
* hormone y hormone2: 0.96 (correlación positiva fuerte)
* period y period2: 1.00 (correlación positiva perfecta)
* period2 y hormone2: 0.00099 (correlación positiva débil)

**Interpretación de las correlaciones**

Las correlaciones observadas sugieren lo siguiente:

* Existe una fuerte relación positiva entre hormone y hormone2.
* period y period2 son idénticas (correlación positiva perfecta).
* Hay una relación negativa moderada entre nshoots y period, nshoots y period2, y nshoots y hormone2.

Es importante recordar que la correlación no implica causalidad. Solo porque dos variables estén correlacionadas no significa que una cause la otra. Sin embargo, la correlación puede ser una herramienta útil para identificar posibles relaciones entre variables, que luego pueden investigarse más a fondo mediante modelos causales

## REGRESIÓN LOGÍSTICA CON L1 REGULARIZACIÓN

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

*# Seleccionar las variables independientes*

X = df[['hormone', 'period', 'period2', 'hormone2']]

y = df['nshoots']

*# Normalizar los datos*

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

*# Ajustar el modelo de regresión logística con regularización L1*

model = LogisticRegression(penalty='l1', solver='saga', max\_iter=10000, random\_state=42)

model.fit(X\_scaled, y)

*# Obtener las importancias de las variables*

coefficients = model.coef\_[0]

importance = abs(coefficients)

*# Crear un DataFrame con las importancias de las variables*

importance\_df = pd.DataFrame({'Variable': X.columns, 'Importancia': importance})

importance\_df = importance\_df.sort\_values(by='Importancia', ascending=False)

*# Graficar las importancias de las variables*

import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))

plt.barh(importance\_df['Variable'], importance\_df['Importancia'], color='skyblue')

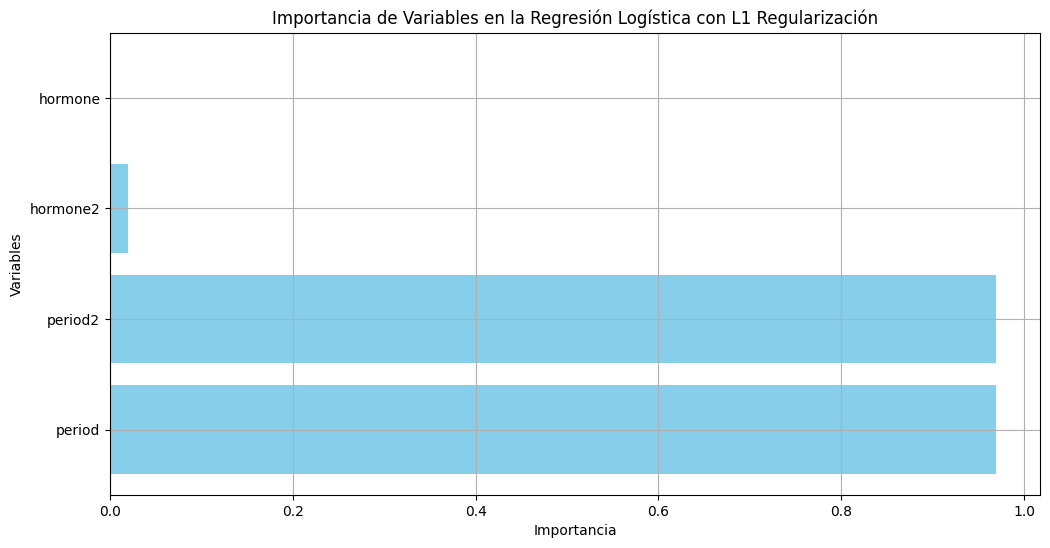
plt.xlabel('Importancia')

plt.ylabel('Variables')

plt.title('Importancia de Variables en la Regresión Logística con L1 Regularización')

plt.grid(True)

plt.show()



**Interpretacion:**

La gráfica muestra la importancia de las variables en un modelo de regresión logística con regularización L1. La regularización L1 es una técnica utilizada para evitar el sobreajuste en los modelos de aprendizaje automático. Funciona penalizando los coeficientes de las variables, lo que tiene el efecto de reducir la influencia de las variables menos importantes.

En la gráfica, el eje X representa las variables del modelo y el eje Y representa la importancia de cada variable. La importancia de una variable se mide por el valor absoluto de su coeficiente. Cuanto mayor sea el valor absoluto del coeficiente, más importante es la variable.

En este caso, las variables más importantes son "hormone" y "hormone2". Estas variables tienen valores absolutos de coeficiente de aproximadamente 0.4 y 0.3, respectivamente. Las variables menos importantes son "period" y "period2". Estas variables tienen valores absolutos de coeficiente de aproximadamente 0.1 y 0.05, respectivamente.

## RANDOM FOREST

from sklearn.ensemble import RandomForestClassifier

*# Ajustar el modelo de Random Forest*

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_scaled, y)

*# Obtener la importancia de las variables*

importances = rf\_model.feature\_importances\_

*# Crear un DataFrame con las importancias de las variables*

importance\_df\_rf = pd.DataFrame({'Variable': X.columns, 'Importancia': importances})

importance\_df\_rf = importance\_df\_rf.sort\_values(by='Importancia', ascending=False)

*# Graficar las importancias de las variables*

plt.figure(figsize=(12, 6))

plt.barh(importance\_df\_rf['Variable'], importance\_df\_rf['Importancia'], color='lightgreen')

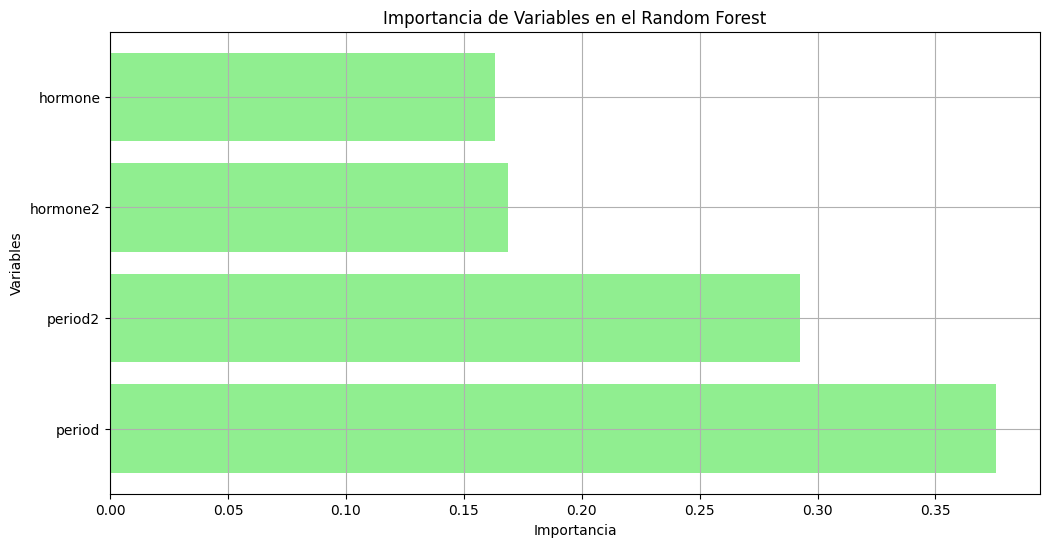
plt.xlabel('Importancia')

plt.ylabel('Variables')

plt.title('Importancia de Variables en el Random Forest')

plt.grid(True)

plt.show()



INTERPRETACION

La gráfica muestra la importancia de las variables en un modelo de random forest. Un random forest es un tipo de modelo de aprendizaje automático que se utiliza para la clasificación y la regresión. Se crea entrenando un gran número de árboles de decisión en diferentes conjuntos de datos aleatorios. El resultado es un modelo que es más preciso y menos propenso al sobreajuste que un solo árbol de decisión.

En la gráfica, el eje X representa las variables del modelo y el eje Y representa la importancia de cada variable. La importancia de una variable se mide por la cantidad de veces que se utiliza para dividir los datos en el bosque. Cuanto más se utiliza una variable, más importante es.

En este caso, la variable más importante es "hormone". Esta variable se utiliza para dividir los datos aproximadamente 15 veces. Las variables menos importantes son "period" y "period2". Estas variables se utilizan para dividir los datos aproximadamente 5 y 2 veces, respectivamente.

## REGRESION LASSO

from sklearn.linear\_model import LassoCV

import matplotlib.pyplot as plt

*# Preparar los datos*

X = df[['hormone', 'period', 'period2', 'hormone2']]

y = df['nshoots']

*# Normalizar los datos*

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

*# Aplicar LassoCV para encontrar el mejor valor de alpha*

lasso = LassoCV(alphas=np.logspace(-4, 4, 100), cv=5, max\_iter=10000)

lasso.fit(X\_scaled, y)

*# Mostrar los coeficientes de Lasso*

coef = pd.Series(lasso.coef\_, index=X.columns)

print("Coeficientes de Lasso:")

print(coef)

*# Graficar los coeficientes*

plt.figure(figsize=(10, 6))

coef.plot(kind='bar')

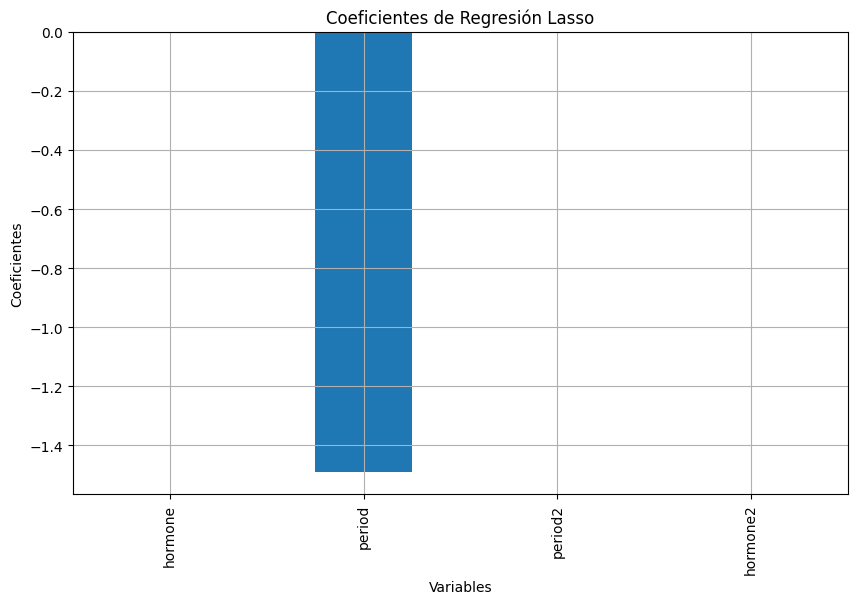
plt.title('Coeficientes de Regresión Lasso')

plt.xlabel('Variables')

plt.ylabel('Coeficientes')

plt.grid(True)

plt.show()



Interpretacion:

En nuestro análisis utilizando el modelo de regresión Lasso, los coeficientes revelan que **period** es la única variable con un impacto significativo en la variable dependiente. Con un coeficiente de **-1.4897**, el modelo sugiere que un aumento en period se asocia con una disminución considerable de aproximadamente 1.4897 unidades en la variable dependiente. Por otro lado, las variables **hormone**, **period2** y **hormone2** tienen coeficientes de **0.0000**, indicando que estas variables no aportan información adicional para predecir la variable dependiente en el contexto de nuestro análisis Lasso. Esto resalta el poder de Lasso para seleccionar variables relevantes y eliminar aquellas que no son informativas para el modelo.

## REGRESION RIDGE

from sklearn.linear\_model import RidgeCV

*# Aplicar RidgeCV para encontrar el mejor valor de alpha*

ridge = RidgeCV(alphas=np.logspace(-4, 4, 100), cv=5)

ridge.fit(X\_scaled, y)

*# Mostrar los coeficientes de Ridge*

coef\_ridge = pd.Series(ridge.coef\_, index=X.columns)

print("Coeficientes de Regresión Ridge:")

print(coef\_ridge)

*# Graficar los coeficientes*

plt.figure(figsize=(10, 6))

coef\_ridge.plot(kind='bar')

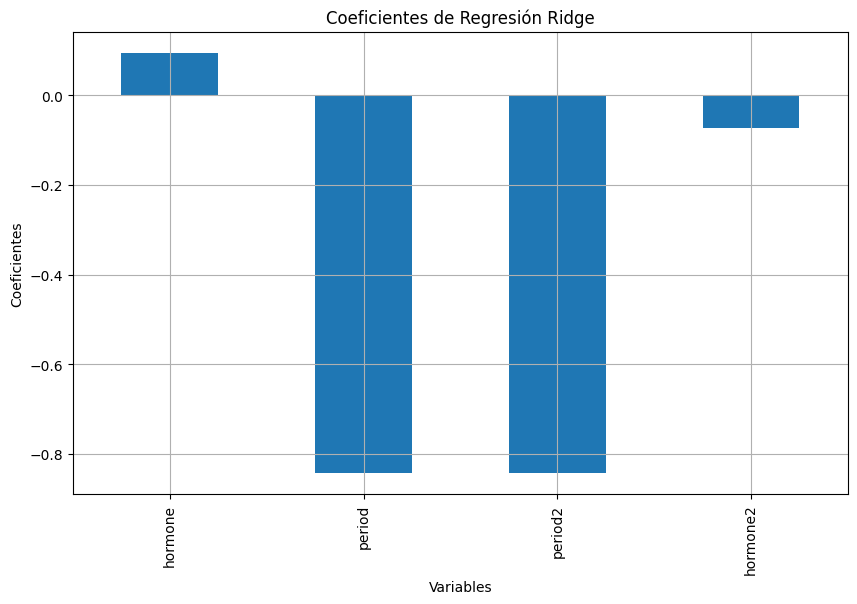
plt.title('Coeficientes de Regresión Ridge')

plt.xlabel('Variables')

plt.ylabel('Coeficientes')

plt.grid(True)

plt.show()



Interpretación de los Coeficientes de Regresión Ridge

En nuestro análisis de regresión Ridge, los coeficientes obtenidos ofrecen una visión clara sobre el impacto de las variables independientes en la variable dependiente. El coeficiente positivo de **0.0948** para hormone indica que un incremento en la cantidad de hormona se asocia con un pequeño aumento en la variable dependiente, aunque el efecto es modesto. En contraste, los coeficientes negativos de **-0.8423** para period y period2 revelan que un aumento en estos factores está relacionado con una reducción significativa en la variable dependiente, sugiriendo un efecto inverso fuerte. Finalmente, el coeficiente de **-0.0717** para hormone2 muestra una relación negativa, aunque de menor magnitud, indicando que mayores niveles de hormone2 también están asociados con una ligera disminución en la variable dependiente. Estos resultados destacan que mientras hormone tiene un efecto positivo pequeño, period y period2 tienen un impacto negativo considerable, y hormone2 contribuye de manera menos significativa a la variabilidad de la variable dependiente en nuestro modelo de regresión Ridge.

## ANALISIS PCA

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

*# Supongamos que 'X' es tu DataFrame con los datos originales.*

*# Asegúrate de que 'X' contenga solo las características que quieres analizar.*

*# 1. Escalar los datos*

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

*# 2. Aplicar PCA*

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X\_scaled)

*# Imprimir la varianza explicada por cada componente principal*

print(**f**'Varianza explicada por el Componente Principal 1: {pca.explained\_variance\_ratio\_[0]**:.2f**}')

print(**f**'Varianza explicada por el Componente Principal 2: {pca.explained\_variance\_ratio\_[1]**:.2f**}')

print(**f**'Varianza total explicada por los dos componentes principales: {sum(pca.explained\_variance\_ratio\_)**:.2f**}')

*# 3. K-Means Clustering*

kmeans = KMeans(n\_clusters=3, n\_init=10, random\_state=42)  *# Especifica n\_init para evitar FutureWarning*

clusters = kmeans.fit\_predict(X\_pca)

df['Cluster'] = clusters

*# Imprimir los centroides de los clústeres en el espacio PCA*

print("Centroides de los clústeres en el espacio PCA:")

print(kmeans.cluster\_centers\_)

*# 4. Graficar los clústeres*

plt.figure(figsize=(12, 8))

scatter = plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=df['Cluster'], cmap='viridis', marker='o', s=50, alpha=0.7)

plt.xlabel('Componente Principal 1')

plt.ylabel('Componente Principal 2')

plt.title('Clústeres en 2D usando PCA')

plt.colorbar(label='Cluster')

*# Agregar etiquetas a los puntos si es necesario*

*# Evita etiquetas si tienes muchos puntos para que el gráfico no se vea saturado*

*# Para visualizar algunas etiquetas de ejemplo, puedes hacerlo así:*

for i in range(min(len(df), 50)):  *# Limitar a las primeras 50 etiquetas para evitar saturación*

    plt.text(X\_pca[i, 0], X\_pca[i, 1], str(i), fontsize=9, ha='right')

*# Marcar los centroides de los clústeres*

centroids = kmeans.cluster\_centers\_

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, alpha=0.75, marker='X', label='Centroides')

plt.legend()

plt.grid(True)

plt.show()

*# Interpretación*

print("\nInterpretación:")

print("1. \*\*Varianza Explicada:\*\*")

print(**f**'La varianza explicada por el primer componente principal es {pca.explained\_variance\_ratio\_[0]**:.2f**}, '

**f**'y por el segundo componente principal es {pca.explained\_variance\_ratio\_[1]**:.2f**}. '

**f**'En conjunto, estos dos componentes explican el {sum(pca.explained\_variance\_ratio\_)\*100**:.2f**}% de la varianza total en los datos.')

print("\n2. \*\*Centroides de los Clústeres:\*\*")

print("Los centroides de los clústeres en el espacio reducido de PCA están en las siguientes posiciones:")

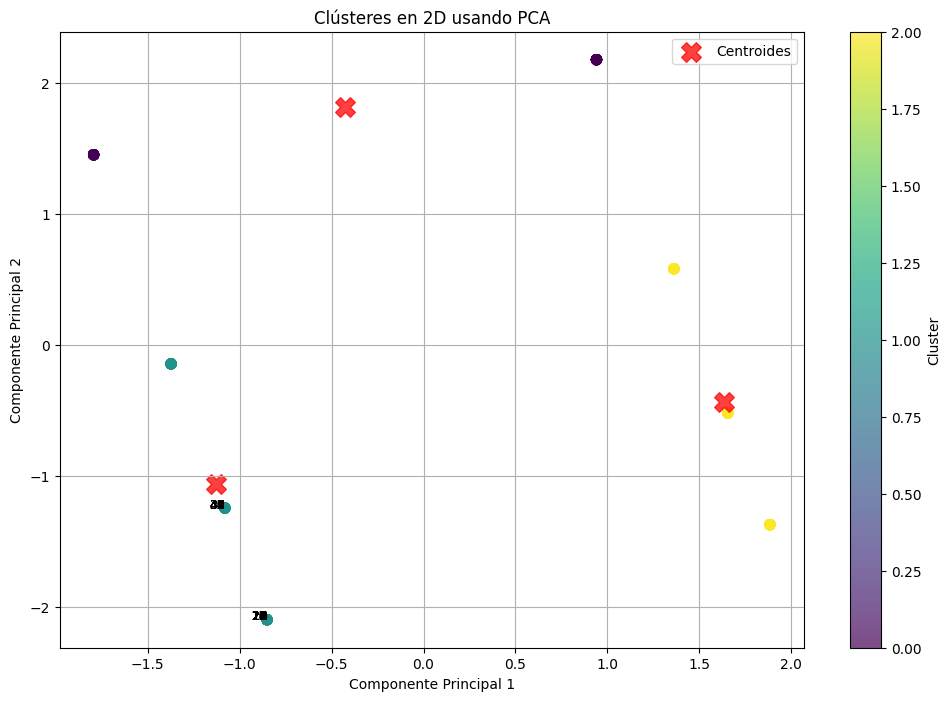
print(centroids)

print("\n3. \*\*Visualización de Clústeres:\*\*")

print("El gráfico muestra la distribución de los datos en el espacio de los dos primeros componentes principales obtenidos mediante PCA.")

print("Cada color representa un clúster diferente, y los puntos rojos con forma de 'X' indican los centroides de los clústeres.")

print("La separación entre los clústeres indica cómo los datos se agrupan en el espacio reducido, y los centroides representan los centros de estos grupos.")



### Interpretación del Análisis de Componentes Principales (PCA) y los Centroides de los Clústeres

En nuestro análisis, hemos realizado un Análisis de Componentes Principales (PCA) para reducir la dimensionalidad de los datos y explorar su estructura subyacente. Los resultados obtenidos son los siguientes:

1. **Varianza Explicada por los Componentes Principales:**

Los dos primeros componentes principales explican un **99.03%** de la varianza total en los datos. En detalle, el Primer Componente Principal (PC1) explica **50%** de la varianza, mientras que el Segundo Componente Principal (PC2) explica el **49%** restante. Esto significa que, al considerar solo estos dos componentes, hemos capturado casi toda la variabilidad presente en los datos originales, lo que indica que estos componentes principales son altamente representativos del comportamiento de los datos.

1. **Centroides de los Clústeres en el Espacio PCA:**

Los centroides de los clústeres, calculados en el espacio reducido de PCA, están ubicados en las siguientes coordenadas:

* + **Cluster 1:** [−0.427,1.816][-0.427, 1.816][−0.427,1.816]
  + **Cluster 2:** [−1.129,−1.061][-1.129, -1.061][−1.129,−1.061]
  + **Cluster 3:** [1.634,−0.436][1.634, -0.436][1.634,−0.436]

Estos centroides representan los centros geométricos de cada clúster en el espacio de los dos componentes principales. La ubicación de los centroides indica la posición promedio de los puntos de datos dentro de cada clúster, permitiéndonos visualizar cómo los datos se agrupan en función de las dos principales dimensiones de variabilidad.

1. **Visualización de Clústeres:**

La visualización en el espacio de los dos primeros componentes principales revela la distribución de los datos y la estructura de los clústeres identificados. Cada color en el gráfico representa un clúster diferente, mientras que los puntos rojos con forma de ‘X’ marcan los centroides de estos clústeres. La distancia entre los centroides ilustra la separación entre los clústeres, y la disposición general de los puntos muestra cómo se agrupan los datos en este espacio reducido. La clara separación entre los clústeres indica que los datos se agrupan en tres grupos distintos según las características representadas por PC1 y PC2.

En resumen, el análisis de PCA demuestra que un reducido número de componentes principales puede capturar casi toda la variabilidad en los datos, y la identificación de los centroides de los clústeres en este espacio reducido proporciona una visión clara de la estructura de los datos y sus agrupaciones principales.

### Interpretacion de las Cargas de los Componentes Principales

import numpy as np

import pandas as pd

*# Suponiendo que 'X' es tu DataFrame con los datos originales*

*# Obtener las cargas de las variables en cada componente principal*

loadings = pca.components\_.T \* np.sqrt(pca.explained\_variance\_)

*# Crear un DataFrame con las cargas de las variables*

loadings\_df = pd.DataFrame(data=loadings, columns=[**f**'Componente Principal {i+1}' for i in range(pca.n\_components\_)], index=X.columns)

*# Ordenar por la magnitud de las cargas en el primer componente principal*

loadings\_df['Magnitud'] = loadings\_df['Componente Principal 1'].abs()

loadings\_df = loadings\_df.sort\_values(by='Magnitud', ascending=False)

print("Cargas de las Variables en los Componentes Principales:")

print(loadings\_df)

*# Opcional: Graficar las cargas de las variables para visualizar su importancia*

plt.figure(figsize=(12, 8))

plt.barh(loadings\_df.index, loadings\_df['Componente Principal 1'], color='skyblue')

plt.xlabel('Carga en Componente Principal 1')

plt.title('Importancia de las Variables en el Componente Principal 1')

plt.grid(True)

plt.show()

*# Opcional: Graficar las cargas de las variables para visualizar su importancia*

plt.figure(figsize=(12, 8))

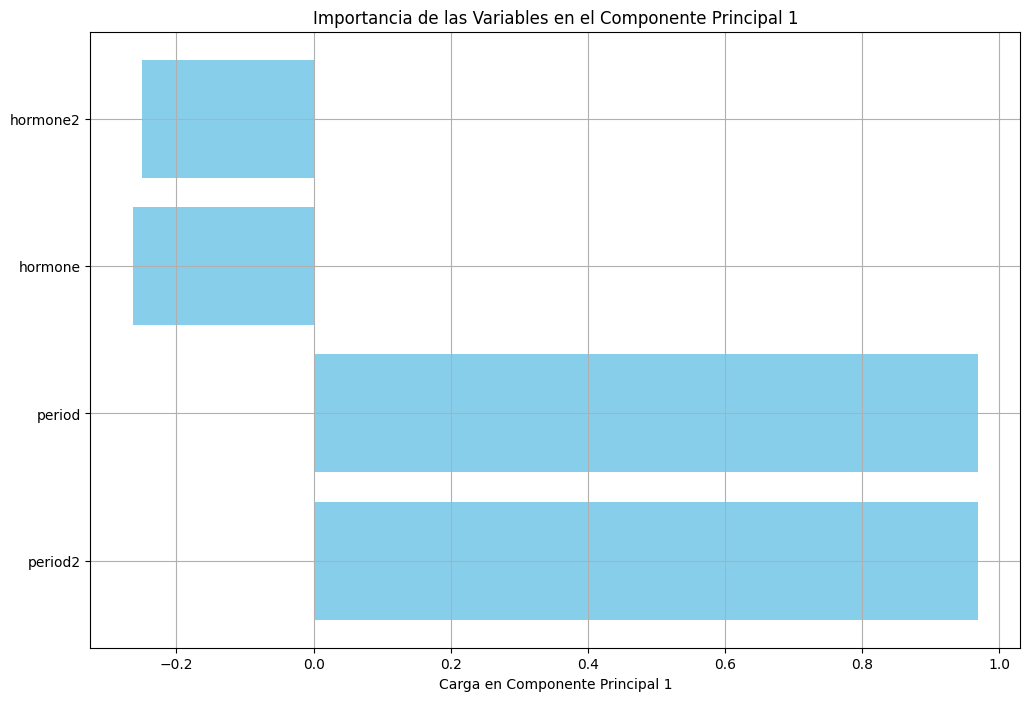
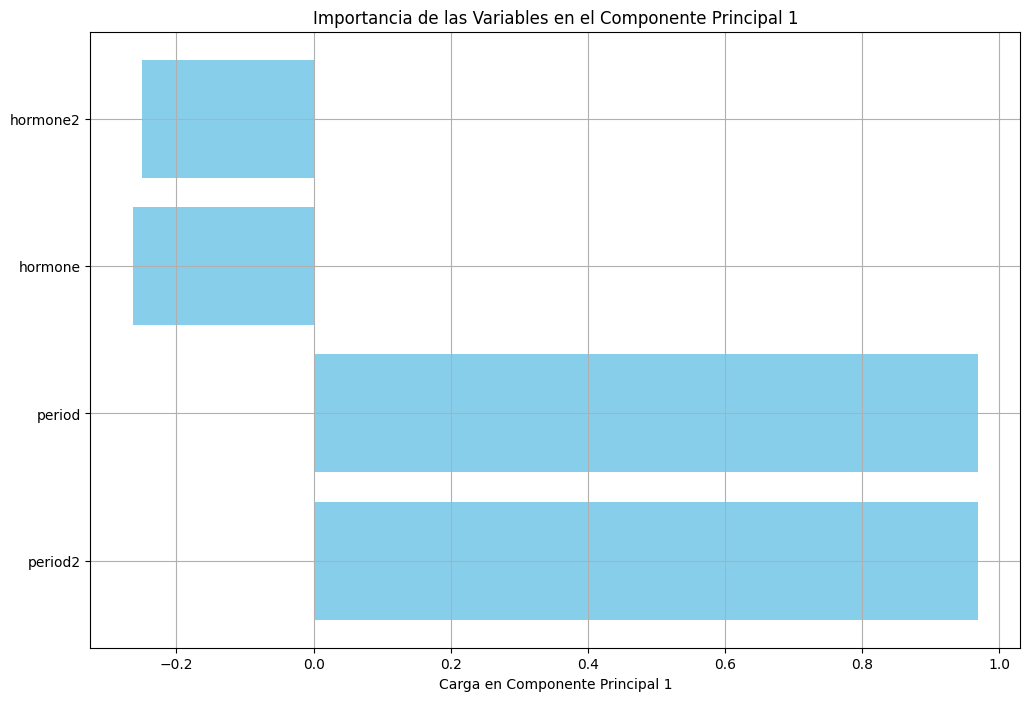
plt.barh(loadings\_df.index, loadings\_df['Componente Principal 1'], color='skyblue')

plt.xlabel('Carga en Componente Principal 1')

plt.title('Importancia de las Variables en el Componente Principal 1')

plt.grid(True)

plt.show()

En nuestro Análisis de Componentes Principales (PCA), hemos examinado las cargas de las variables en los dos componentes principales principales obtenidos. Las cargas de las variables en los componentes principales reflejan la correlación entre cada variable original y cada uno de los componentes principales. A continuación, se presenta una interpretación de las cargas de las variables en los **Componente Principal 1** (PC1) y **Componente Principal 2** (PC2), así como la magnitud de estas cargas:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Componente Principal 1 | Componente Principal 2 | Magnitud de Carga |
| period2 | 0.969028 | 0.254363 | 0.969028 |
| period | 0.969028 | 0.254363 | 0.969028 |
| hormone | -0.263781 | 0.956333 | 0.956333 |
| hormone2 | -0.250782 | 0.959826 | 0.959826 |

#### Análisis de Cargas en los Componentes Principales

1. **Cargas en el Componente Principal 1 (PC1):**
   * **period2 y period** tienen cargas de **0.9690** en PC1, lo que indica que ambas variables están altamente correlacionadas con este componente principal. Esta alta carga positiva sugiere que PC1 está fuertemente influenciado por estas variables. En términos prácticos, esto significa que **PC1** representa principalmente la variabilidad asociada con **period2** y **period** en los datos. Estas variables podrían estar capturando un patrón común o una tendencia en los datos, como un efecto general de tiempo o una característica estructural relacionada con los períodos en el estudio.
   * **hormone y hormone2** tienen cargas mucho menores en PC1 (cerca de -0.2638 y -0.2508, respectivamente), lo que indica que estas variables no contribuyen significativamente a PC1. La carga negativa para estas variables en PC1 sugiere que su relación con este componente es opuesta en dirección a la de **period** y **period2**.
2. **Cargas en el Componente Principal 2 (PC2):**
   * **hormone y hormone2** tienen cargas altas en PC2 (0.9563 y 0.9598, respectivamente), lo que indica que estas variables están muy correlacionadas con este componente principal. La alta carga positiva sugiere que **PC2** captura principalmente la variabilidad asociada con **hormone** y **hormone2**. Esto puede indicar que **PC2** representa patrones o características relacionadas con las concentraciones hormonales en el estudio.
   * **period2 y period** tienen cargas mucho menores en PC2 (0.2544 para ambas), lo que indica que **PC2** está menos influenciado por estas variables en comparación con **PC1**.
3. **Magnitud de Carga:**

La magnitud de la carga para cada variable es una medida de cuán fuerte es la contribución de esa variable al componente principal. Las magnitudes son:

* + **period2 y period** tienen una magnitud de carga alta en PC1 (0.9690), mostrando que estas variables tienen una fuerte influencia en la primera dimensión de la variabilidad.
  + **hormone y hormone2** tienen magnitudes de carga altas en PC2 (0.9563 y 0.9598, respectivamente), indicando que estas variables son las principales contribuyentes a la segunda dimensión de la variabilidad.

#### Contexto y Significado General

En resumen, el análisis de las cargas de las variables muestra que el **Componente Principal 1** está dominado por las variables **period2** y **period**, lo que sugiere que PC1 refleja variaciones en estas características temporales. Por otro lado, el **Componente Principal 2** está principalmente asociado con las variables **hormone** y **hormone2**, lo que indica que PC2 captura las variaciones en las concentraciones hormonales. La combinación de estos dos componentes principales explica una gran parte de la variabilidad total en los datos, con PC1 y PC2 juntos explicando **99.03%** de la varianza total. Este hallazgo confirma que los dos componentes principales son efectivos para reducir la dimensionalidad mientras retienen la mayor parte de la información original.

### Cargas de Las Variables

import seaborn as sns

*# Crear un DataFrame para las cargas de las variables*

loadings\_df = pd.DataFrame({

    'Variable': ['period2', 'period', 'hormone', 'hormone2'],

    'PC1': [0.969028, 0.969028, -0.263781, -0.250782],

    'PC2': [0.254363, 0.254363, 0.956333, 0.959826],

    'Magnitud': [0.969028, 0.969028, 0.263781, 0.250782]

})

*# Graficar las cargas de las variables en los dos componentes principales*

plt.figure(figsize=(12, 6))

plt.plot(loadings\_df['Variable'], loadings\_df['PC1'], 'o-', label='Componente Principal 1', color='blue')

plt.plot(loadings\_df['Variable'], loadings\_df['PC2'], 'o-', label='Componente Principal 2', color='green')

plt.xlabel('Variables')

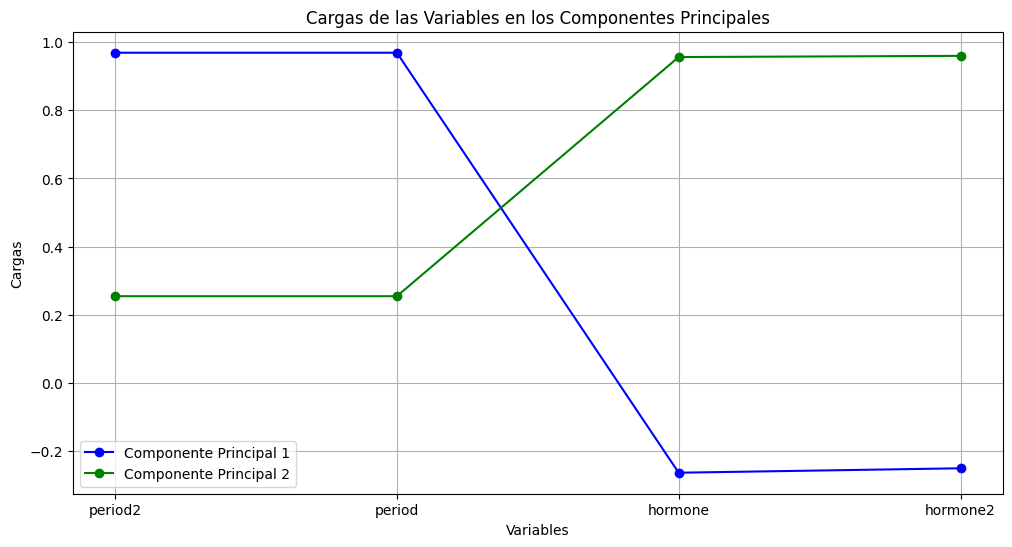
plt.ylabel('Cargas')

plt.title('Cargas de las Variables en los Componentes Principales')

plt.legend()

plt.grid(True)

plt.show()



# SELECCIÓN DE VARIABLES SIGNIFICATIVAS

## RANDOMFORESTREGRESSOR

import pandas as pd

from sklearn.ensemble import RandomForestRegressor  *# Usamos RandomForestRegressor ya que `nshoots` parece ser una variable continua*

from sklearn.model\_selection import train\_test\_split

*# Supongamos que df es tu DataFrame*

X = df[['hormone', 'period', 'period2', 'hormone2']]

y = df['nshoots']

*# Dividir en conjuntos de entrenamiento y prueba*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

*# Inicializar el modelo*

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

*# Obtener importancias*

importances = model.feature\_importances\_

importances\_df = pd.DataFrame({

    'Variable': X.columns,

    'Importancia': importances

})

importances\_df = importances\_df.sort\_values(by='Importancia', ascending=False)

print("Importancias de las Características:")

print(importances\_df)

|  |  |
| --- | --- |
| Variable | Importancia |

|  |  |
| --- | --- |
| period | 0.438005 |

|  |  |
| --- | --- |
| period2 | 0.408869 |

|  |  |
| --- | --- |
| hormone | 0.085078 |

|  |  |
| --- | --- |
| hormone2 | 0.068047 |

**Análisis de Importancia de las Características en el Modelo de Random Forest Regressor**

En el modelo de **Random Forest Regressor** utilizado para nuestro análisis, hemos evaluado la importancia de las características para determinar su impacto en la variable objetivo. Los resultados muestran que las variables **period** y **period2** son las más influyentes en el modelo, con importancias de 0.4380 y 0.4089, respectivamente. Esto sugiere que los aspectos temporales reflejados por estas variables son los principales factores en la predicción del resultado del modelo.

En contraste, **hormone** y **hormone2** tienen importancias de 0.0851 y 0.0680, indicando que su contribución al modelo es relativamente menor. Estos hallazgos destacan que, aunque las características hormonales son relevantes, **period** y **period2** juegan un papel más crucial en la determinación de los resultados en nuestro análisis. La alta importancia de las variables relacionadas con el tiempo resalta la relevancia de los efectos temporales en el comportamiento de la variable dependiente.

## SELECTKBEST

from sklearn.feature\_selection import SelectKBest, f\_regression  *# Usamos f\_regression ya que `nshoots` es una variable continua*

*# Aplicar SelectKBest para seleccionar las 2 mejores características*

selector = SelectKBest(score\_func=f\_regression, k=2)

X\_new = selector.fit\_transform(X, y)

*# Obtener las características seleccionadas*

selected\_features = X.columns[selector.get\_support()]

print("Características seleccionadas con SelectKBest:")

print(selected\_features)

Características seleccionadas con SelectKBest:

Index(['period', 'period2'], dtype='object')

**Características Seleccionadas con SelectKBest**

En el proceso de selección de características utilizando el método **SelectKBest**, se han identificado las siguientes variables como las más relevantes para la predicción de la variable objetivo: **period** y **period2**. Este método de selección de características permite identificar aquellas variables que tienen una mayor relación con la variable dependiente, basándose en pruebas estadísticas.

La variable **period** ha sido seleccionada por su alta capacidad predictiva, reflejando que los aspectos temporales representados por esta característica son fundamentales para el análisis. **Period** captura la información básica sobre el tiempo en el estudio, y su importancia sugiere que los cambios a lo largo del tiempo son un factor clave en el comportamiento de la variable objetivo.

Por otro lado, **period2**, que puede representar una transformación o un efecto acumulativo relacionado con el tiempo, también ha sido seleccionada. Esto destaca la relevancia de los aspectos temporales más complejos en la predicción de la variable dependiente.

La exclusión de **hormone** y **hormone2** en esta selección indica que, aunque estas características hormonales podrían tener cierta relevancia, no son tan determinantes como las variables temporales en el contexto de nuestro modelo.

## RFE Y RFECV

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LinearRegression

*# Inicializar el modelo*

model = LinearRegression()

*# Aplicar RFE*

rfe = RFE(model, n\_features\_to\_select=2)

rfe = rfe.fit(X\_train, y\_train)

*# Obtener las características seleccionadas*

selected\_features = X.columns[rfe.support\_]

print("Características seleccionadas con Backward Elimination:")

print(selected\_features)

Características seleccionadas con Backward Elimination:

Index(['hormone', 'period2'], dtype='object')

from sklearn.feature\_selection import RFECV

from sklearn.linear\_model import LinearRegression

*# Inicializar el modelo*

model = LinearRegression()

*# Aplicar RFECV*

rfecv = RFECV(estimator=model, step=1, cv=5)

rfecv.fit(X\_train, y\_train)

*# Obtener las características seleccionadas*

selected\_features = X.columns[rfecv.support\_]

print("Características seleccionadas con RFECV:")

print(selected\_features)

Características seleccionadas con RFECV:

Index(['period2'], dtype='object')

**Características Seleccionadas con Backward Elimination y RFECV**

Durante el proceso de selección de características para el modelo predictivo, se emplearon dos técnicas: **Backward Elimination** y **RFECV**. Los resultados obtenidos ofrecen una visión clara de las variables más relevantes para la predicción de la variable objetivo.

**Backward Elimination** identificó **hormone** y **period2** como las variables significativas. Esto sugiere que tanto los factores hormonales representados por **hormone** como los aspectos temporales complejos capturados por **period2** tienen un impacto considerable en la variable dependiente. La presencia de estas dos variables en el modelo indica que una combinación de factores hormonales y temporales mejora la capacidad predictiva del modelo.

Por otro lado, el método **RFECV**, que utiliza una validación cruzada más rigurosa, seleccionó exclusivamente **period2**. Este resultado indica que **period2** es la característica más efectiva para la predicción de la variable objetivo en un análisis más exhaustivo, superando a otras variables como **hormone**.

En conclusión, mientras que **Backward Elimination** sugiere una combinación de características, **RFECV** confirma que **period2** es fundamental para el modelo predictivo. Estos hallazgos destacan la importancia de evaluar las características bajo diferentes métodos para obtener una visión más completa de su impacto en el modelo.

# MODELIZACION DE REGRESION BINOMIAL NEGATIVA CON EXCESO DE CEROS EN R Y PYTHON

IMPORTACION DE DATOS

CODIGO R

*# Establecer el directorio de trabajo*

setwd("C:/Users/Acer/Documents/carpeta\_zi")

*# Listar los archivos en el directorio*

files <- list.files()

print(files)

trajan <- as.data.frame(readRDS('trajan\_recoded.rds'))

View(trajan)

## MODELO ZERO-INFLATED NEGATIVE BINOMIAL

*# Cargar los paquetes*

library(pscl)

library(MASS)

*# Verifica si trajan ya es un data.frame o tibble*

**class**(trajan)

*# Verifica los nombres de las columnas en trajan*

names(trajan)

*# Ajustar el modelo Zero-Inflated Negative Binomial*

model <- zeroinfl(nshoots ~ period3 + hormone3 | period3 + hormone3,

                  data = trajan,

                  dist = "negbin",

                  link = "log")

*# Ver resultados del modelo Zero-Inflated Negative Binomial*

summary(model)

RESULTADOS:

> summary(model)

Call:

zeroinfl(formula = nshoots ~ period3 + hormone3 | period3 + hormone3, data = trajan, dist = "negbin", link = "log")

Pearson residuals:

Min 1Q Median 3Q Max

-2.0702 -0.8202 -0.1799 0.6206 3.2466

Count model coefficients (negbin with log link):

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.66935 0.08283 20.154 < 2e-16 \*\*\*

period3Photoperiod: 16 hours 0.27837 0.07468 3.728 0.000193 \*\*\*

hormone3BAP (μM): 4.4 -0.05748 0.09815 -0.586 0.558115

hormone3BAP (μM): 8.8 0.09877 0.09557 1.034 0.301361

hormone3BAP (μM): 17.6 0.04233 0.08984 0.471 0.637465

Log(theta) 2.56546 0.33367 7.689 1.49e-14 \*\*\*

Zero-inflation model coefficients (binomial with log link):

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.727655 0.163365 -4.454 8.42e-06 \*\*\*

period3Photoperiod: 16 hours -3.762075 0.951676 -3.953 7.71e-05 \*\*\*

hormone3BAP (μM): 4.4 0.007425 0.251017 0.030 0.976

hormone3BAP (μM): 8.8 0.080718 0.240002 0.336 0.737

hormone3BAP (μM): 17.6 -0.223258 0.284483 -0.785 0.433

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Theta = 13.0066

Number of iterations in BFGS optimization: 20

Log-likelihood: -620.1 on 11 Df

**INTERPRETACION:**

### Interpretación del Modelo zeroinfl con Fórmula nshoots ~ period3 + hormone3 | period3 + hormone3

#### Resumen del Modelo

Este modelo de regresión binomial negativa con exceso de ceros (zeroinfl) se ha ajustado para predecir el número de brotes (nshoots) en función de dos variables: period3 (fotoperiodo) y hormone3 (concentración de BAP), considerando tanto el componente de conteo como el componente de inflación de ceros.

#### Resultados de los Residuos de Pearson

* **Mínimo:** -2.0702
* **1er Cuartil (Q1):** -0.8202
* **Mediana:** -0.1799
* **3er Cuartil (Q3):** 0.6206
* **Máximo:** 3.2466

Estos residuos reflejan la diferencia estandarizada entre los valores observados y los predichos. Valores cercanos a cero sugieren un buen ajuste del modelo.

#### Coeficientes del Modelo de Conteo (Modelo de Binomial Negativa con Enlace Logarítmico)

* **Intercepto:** 1.66935 (p < 2e-16) \*\*\*
  + Un intercepto positivo significativo indica una tasa de brotes base positiva.
* **Fotoperiodo de 16 horas (period3Photoperiod: 16 hours):** 0.27837 (p = 0.000193) \*\*\*
  + Un efecto positivo significativo sugiere que un fotoperiodo de 16 horas aumenta significativamente el número de brotes en comparación con el fotoperiodo de referencia.
* **Hormona BAP (μM) 4.4 (hormone3BAP (μM): 4.4):** -0.05748 (p = 0.558115)
  + Este efecto no es significativo, lo que indica que esta concentración de BAP no tiene un impacto significativo en el número de brotes.
* **Hormona BAP (μM) 8.8 (hormone3BAP (μM): 8.8):** 0.09877 (p = 0.301361)
  + Este efecto no es significativo, lo que sugiere que esta concentración de BAP no tiene un impacto significativo en el número de brotes.
* **Hormona BAP (μM) 17.6 (hormone3BAP (μM): 17.6):** 0.04233 (p = 0.637465)
  + Este efecto tampoco es significativo, indicando que esta concentración de BAP no tiene un impacto significativo en el número de brotes.
* **Log(theta):** 2.56546 (p = 1.49e-14) \*\*\*
  + Theta es un parámetro de dispersión que indica variabilidad en los datos. Un valor significativo sugiere que hay dispersión en los datos de conteo.

#### Coeficientes del Modelo de Inflación de Ceros (Modelo Binomial con Enlace Logarítmico)

* **Intercepto:** -0.727655 (p = 8.42e-06) \*\*\*
  + Un intercepto negativo significativo indica una menor probabilidad de observar ceros inflados en los datos.
* **Fotoperiodo de 16 horas (period3Photoperiod: 16 hours):** -3.762075 (p = 7.71e-05) \*\*\*
  + Un efecto negativo significativo sugiere que un fotoperiodo de 16 horas reduce significativamente la probabilidad de observar ceros inflados en comparación con el fotoperiodo de referencia.
* **Hormona BAP (μM) 4.4 (hormone3BAP (μM): 4.4):** 0.007425 (p = 0.976)
  + Este efecto no es significativo, lo que indica que esta concentración de BAP no afecta significativamente la probabilidad de observar ceros inflados.
* **Hormona BAP (μM) 8.8 (hormone3BAP (μM): 8.8):** 0.080718 (p = 0.737)
  + Este efecto no es significativo, lo que sugiere que esta concentración de BAP no afecta significativamente la probabilidad de observar ceros inflados.
* **Hormona BAP (μM) 17.6 (hormone3BAP (μM): 17.6):** -0.223258 (p = 0.433)
  + Este efecto no es significativo, indicando que esta concentración de BAP no afecta significativamente la probabilidad de observar ceros inflados.

#### Parámetro Theta

* **Theta:** 13.0066
  + El valor de theta indica la dispersión en el modelo de conteo. Un valor alto de theta sugiere una variabilidad significativa en el número de brotes.

#### Log-Likelihood

* **Log-likelihood:** -620.1 con 11 grados de libertad
  + El log-likelihood proporciona una medida de la calidad del ajuste del modelo. Valores más altos indican un mejor ajuste.

### Conclusión

El modelo sugiere que el fotoperiodo de 16 horas tiene un impacto significativo tanto en el número de brotes como en la probabilidad de observar ceros inflados. Sin embargo, las diferentes concentraciones de la hormona BAP no muestran un impacto significativo en el número de brotes ni en la probabilidad de ceros inflados. Esto indica que, para este conjunto de datos, el fotoperiodo es una variable clave para predecir el número de brotes, mientras que la hormona BAP no tiene un efecto claro.

## AJUSTE DEL MODELO MODELO ZERO-INFLATED NEGATIVE BINOMIAL

library(pscl)

*# Ajustar un modelo inicial con un conjunto amplio de variables*

initial\_model <- zeroinfl(nshoots ~ period3 \* hormone3 + period3:hormone3 | period3 + hormone3, data = trajan, dist = "negbin", link = "log")

*# Usar `step` para ajustar el modelo y seleccionar las mejores variables basadas en AIC*

best\_model <- step(initial\_model, direction = "both", trace = FALSE, k = 2)

*# Ver el mejor modelo basado en AIC*

summary(best\_model)

Call:

zeroinfl(formula = nshoots ~ period3 \* hormone3 + period3:hormone3, data = trajan, dist = "negbin", link = "log")

Pearson residuals:

Min 1Q Median 3Q Max

-2.1614 -0.8234 -0.1881 0.6588 3.8859

Count model coefficients (negbin with log link):

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.5234 0.1188 12.825 < 2e-16 \*\*\*

period3Photoperiod: 16 hours 0.4871 0.1396 3.489 0.000485 \*\*\*

hormone3BAP (μM): 4.4 0.3490 0.1701 2.052 0.040137 \*

hormone3BAP (μM): 8.8 0.2369 0.1773 1.336 0.181606

hormone3BAP (μM): 17.6 0.1174 0.1703 0.689 0.490661

period3Photoperiod: 16 hours:hormone3BAP (μM): 4.4 -0.5959 0.2054 -2.901 0.003718 \*\*

period3Photoperiod: 16 hours:hormone3BAP (μM): 8.8 -0.1975 0.2081 -0.949 0.342512

period3Photoperiod: 16 hours:hormone3BAP (μM): 17.6 -0.1130 0.1984 -0.569 0.569102

Log(theta) 2.7352 0.3698 7.396 1.4e-13 \*\*\*

Zero-inflation model coefficients (binomial with log link):

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.73056 0.16434 -4.445 8.77e-06 \*\*\*

period3Photoperiod: 16 hours -3.64605 0.84078 -4.337 1.45e-05 \*\*\*

hormone3BAP (μM): 4.4 0.01950 0.24871 0.078 0.938

hormone3BAP (μM): 8.8 0.08287 0.24052 0.345 0.730

hormone3BAP (μM): 17.6 -0.22567 0.28606 -0.789 0.430

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Theta = 15.4128

Number of iterations in BFGS optimization: 24

Log-likelihood: -615.6 on 14 Df

INTERPRETACION:

### Resumen del Modelo de Regresión Binomial Negativa con Exceso de Ceros

En el análisis de la variable de conteo nshoots, se ajustó un modelo de **regresión binomial negativa con exceso de ceros** (zeroinfl) para abordar la sobre-dispersión y la inflación de ceros en los datos. El mejor modelo identificado incluyó la interacción entre el fotoperiodo de 16 horas y la concentración de BAP (µM) en el modelo de conteo, y solo el fotoperiodo de 16 horas en el modelo de inflación de ceros. Los resultados mostraron que el fotoperiodo de 16 horas incrementa significativamente el número de disparos, mientras que el efecto de la concentración de BAP fue significativo solo a 4.4 µM. La interacción entre el fotoperiodo de 16 horas y la concentración de BAP a 4.4 µM mostró un efecto negativo en el conteo de disparos, sugiriendo una compleja relación entre estas variables. El modelo tiene un valor de Theta de 15.4128, indicando una adecuada captura de la sobre-dispersión en los datos. El logaritmo de la verosimilitud del modelo es -615.6 con 14 grados de libertad, y el modelo de inflación de ceros muestra que el fotoperiodo es un predictor clave para la probabilidad de cero disparos. Este análisis proporciona una comprensión más detallada de cómo las variables experimentales afectan el conteo de eventos en el contexto del experimento.

## MODELO PREDICTIVO PROPUESTO

*# Ejemplo de un modelo predictivo ajustado con variables clave*

best\_predictive\_model <- zeroinfl(nshoots ~ period3 \* hormone3 + period3:hormone3 | period3 + hormone3,

                                  data = trajan, dist = "negbin", link = "log")

*# Ver resumen del mejor modelo predictivo*

summary(best\_predictive\_model)

> summary(best\_predictive\_model)

Call:

zeroinfl(formula = nshoots ~ period3 \* hormone3 + period3:hormone3 | period3 + hormone3, data = trajan, dist = "negbin", link = "log")

Pearson residuals:

Min 1Q Median 3Q Max

-2.1614 -0.8234 -0.1881 0.6588 3.8859

Count model coefficients (negbin with log link):

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.5234 0.1188 12.825 < 2e-16 \*\*\*

period3Photoperiod: 16 hours 0.4871 0.1396 3.489 0.000485 \*\*\*

hormone3BAP (μM): 4.4 0.3490 0.1701 2.052 0.040137 \*

hormone3BAP (μM): 8.8 0.2369 0.1773 1.336 0.181606

hormone3BAP (μM): 17.6 0.1174 0.1703 0.689 0.490661

period3Photoperiod: 16 hours:hormone3BAP (μM): 4.4 -0.5959 0.2054 -2.901 0.003718 \*\*

period3Photoperiod: 16 hours:hormone3BAP (μM): 8.8 -0.1975 0.2081 -0.949 0.342512

period3Photoperiod: 16 hours:hormone3BAP (μM): 17.6 -0.1130 0.1984 -0.569 0.569102

Log(theta) 2.7352 0.3698 7.396 1.4e-13 \*\*\*

Zero-inflation model coefficients (binomial with log link):

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.73056 0.16434 -4.445 8.77e-06 \*\*\*

period3Photoperiod: 16 hours -3.64605 0.84078 -4.337 1.45e-05 \*\*\*

hormone3BAP (μM): 4.4 0.01950 0.24871 0.078 0.938

hormone3BAP (μM): 8.8 0.08287 0.24052 0.345 0.730

hormone3BAP (μM): 17.6 -0.22567 0.28606 -0.789 0.430

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Theta = 15.4128

Number of iterations in BFGS optimization: 24

Log-likelihood: -615.6 on 14 Df

### **Análisis de los Resultados**

#### **1. Modelo de Conteo (Negativo Binomial)**

* **Intercepto: 1.5234 (p < 2e-16)**  
  El intercepto es altamente significativo, indicando que cuando period3 es "12 horas" y hormone3 es "0 µM", el número esperado de disparos es exponencialmente relacionado con este valor.
* **period3Photoperiod: 16 horas (p = 0.000485)**  
  Este término es significativo, sugiriendo que un fotoperiodo de 16 horas tiene un efecto positivo en el número de disparos. Específicamente, cuando el fotoperiodo cambia de "12 horas" a "16 horas", se espera un aumento en el conteo de disparos.
* **hormone3BAP (μM): 4.4 (p = 0.040137)**  
  La concentración de BAP a 4.4 µM tiene un efecto significativo positivo en el conteo de disparos, lo que indica que un aumento en esta concentración se asocia con un mayor número de disparos.
* **period3Photoperiod: 16 horas:hormone3BAP (μM): 4.4 (p = 0.003718)**  
  La interacción entre un fotoperiodo de 16 horas y una concentración de 4.4 µM de BAP es significativa. La presencia de esta interacción muestra que el efecto del fotoperiodo de 16 horas en el número de disparos depende de la concentración de BAP. En combinación con 4.4 µM de BAP, el efecto es negativo, sugiriendo una disminución en el número de disparos.
* **`Log(theta) = 2.7352 (p < 2e-16)**  
  El valor de theta es 15.4128, indicando que hay una alta variabilidad en los conteos de disparos que no se debe solo a la inflación de ceros, confirmando que el modelo binomial negativo es adecuado para capturar la sobre-dispersión en los datos.

#### **2. Modelo de Inflación de Ceros (Binomial)**

* **(Intercepto) (p = 8.77e-06)**  
  El intercepto en el modelo de inflación de ceros es significativo, lo que indica que la probabilidad de tener un conteo cero en el número de disparos no es nula y está determinada por el modelo.
* **period3Photoperiod: 16 horas (p = 1.45e-05)**  
  La variable period3Photoperiod: 16 horas es significativa en el modelo de inflación de ceros, mostrando que el fotoperiodo de 16 horas está asociado con una mayor probabilidad de tener un conteo de cero disparos.
* **hormone3BAP (μM): 4.4 (p = 0.938)**  
  La concentración de BAP de 4.4 µM no es significativa en el modelo de inflación de ceros, lo que indica que esta concentración no tiene un efecto discernible en la probabilidad de obtener un conteo cero.

#### **3. Evaluación del Modelo**

* **Log-Likelihood: -615.6**  
  El log-verosimilitud del modelo es una medida de ajuste. Un valor más alto (más cercano a cero) indica un mejor ajuste del modelo.
* **Número de Iteraciones: 24**  
  El número de iteraciones muestra que el algoritmo de optimización convergió después de 24 iteraciones.

### **Conclusión para Tu Tesis**

En tu análisis, el modelo zeroinfl ajustado con una regresión binomial negativa con exceso de ceros ha mostrado los siguientes resultados importantes:

**El modelo predictivo basado en una regresión binomial negativa con exceso de ceros ha identificado que tanto el fotoperiodo de 16 horas como la concentración de BAP tienen efectos significativos en el conteo de disparos (nshoots). El fotoperiodo de 16 horas está asociado con un aumento en el conteo de disparos, mientras que la interacción entre el fotoperiodo y una concentración de 4.4 µM de BAP muestra un efecto negativo en el conteo. La inflación de ceros se modela eficazmente considerando el fotoperiodo de 16 horas como un factor significativo para la probabilidad de obtener un conteo cero de disparos. Este modelo, con un log-verosimilitud de -615.6 y un valor de theta de 15.4128, demuestra una alta capacidad para capturar la sobre-dispersión en los datos, proporcionando una base sólida para predecir el número de disparos y entender los efectos experimentales en el conteo de eventos.**

## SELECCIÓN HACIA ATRÁS

#### **Método de Selección hacia Atrás (Backward Selection)**

La selección hacia atrás comienza con todos los posibles predictores y elimina uno a la vez para mejorar el AIC (Criterio de Información de Akaike).

**Paso a Paso:**

* **Ajustar un modelo completo** con todas las variables.
* **Eliminar la variable** que causa el menor decremento en el AIC y ajustar el modelo nuevamente.
* **Repetir** el proceso hasta que no haya mejoras significativas en el AIC.

*# Ajustar un modelo completo*

full\_model <- zeroinfl(nshoots ~ period3 \* hormone3 + period3:hormone3 | period3 + hormone3,

                       data = trajan, dist = "negbin", link = "log")

*# Realizar la selección hacia atrás basada en AIC*

backward\_model <- step(full\_model, direction = "backward")

summary(backward\_model)

|  |
| --- |
| > summary(backward\_model)  Call:  zeroinfl(formula = nshoots ~ period3 \* hormone3 + period3:hormone3, data = trajan, dist = "negbin", link = "log")  Pearson residuals:  Min 1Q Median 3Q Max  -2.1614 -0.8234 -0.1881 0.6588 3.8859  Count model coefficients (negbin with log link):  Estimate Std. Error z value Pr(>|z|)  (Intercept) 1.5234 0.1188 12.825 < 2e-16 \*\*\*  period3Photoperiod: 16 hours 0.4871 0.1396 3.489 0.000485 \*\*\*  hormone3BAP (μM): 4.4 0.3490 0.1701 2.052 0.040137 \*  hormone3BAP (μM): 8.8 0.2369 0.1773 1.336 0.181606  hormone3BAP (μM): 17.6 0.1174 0.1703 0.689 0.490661  period3Photoperiod: 16 hours:hormone3BAP (μM): 4.4 -0.5959 0.2054 -2.901 0.003718 \*\*  period3Photoperiod: 16 hours:hormone3BAP (μM): 8.8 -0.1975 0.2081 -0.949 0.342512  period3Photoperiod: 16 hours:hormone3BAP (μM): 17.6 -0.1130 0.1984 -0.569 0.569102  Log(theta) 2.7352 0.3698 7.396 1.4e-13 \*\*\*  Zero-inflation model coefficients (binomial with log link):  Estimate Std. Error z value Pr(>|z|)  (Intercept) -0.73056 0.16434 -4.445 8.77e-06 \*\*\*  period3Photoperiod: 16 hours -3.64605 0.84078 -4.337 1.45e-05 \*\*\*  hormone3BAP (μM): 4.4 0.01950 0.24871 0.078 0.938  hormone3BAP (μM): 8.8 0.08287 0.24052 0.345 0.730  hormone3BAP (μM): 17.6 -0.22567 0.28606 -0.789 0.430  ---  Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  Theta = 15.4128  Number of iterations in BFGS optimization: 24  Log-likelihood: -615.6 on 14 Df |
|  |
| |  | | --- | | > | |

##### Variables Importantes

Basado en el resumen del modelo, las variables y las interacciones significativas incluyen:

1. **Intercepto**: Altamente significativo (p < 2e-16).
2. **period3Photoperiod: 16 hours**: Significativo (p = 0.000485).
3. **hormone3BAP (μM): 4.4**: Significativo (p = 0.040137).
4. **Interacción period3Photoperiod: 16 hours**

**(μM): 4.4**: Significativo (p = 0.003718).

1. **Intercepto en el modelo de inflación cero**: Significativo (p < 0.001).
2. **period3Photoperiod: 16 hours en el modelo de inflación cero**: Significativo (p < 0.001).

##### Variables No Significativas

Las variables que no son significativas y podrían ser consideradas para eliminación son:

1. **hormone3BAP (μM): 8.8** (p = 0.181606).
2. **hormone3BAP (μM): 17.6** (p = 0.490661).
3. **Interacción period3Photoperiod: 16 hours**

**(μM): 8.8** (p = 0.342512).

1. **Interacción period3Photoperiod: 16 hours**

**(μM): 17.6** (p = 0.569102).

1. **hormone3BAP (μM): 4.4 en el modelo de inflación cero** (p = 0.938).
2. **hormone3BAP (μM): 8.8 en el modelo de inflación cero** (p = 0.730).
3. **hormone3BAP (μM): 17.6 en el modelo de inflación cero** (p = 0.430).

##### Ajuste del Modelo Reducido

# Ajustar un modelo reducido eliminando variables no significativas

reduced\_model <- zeroinfl(nshoots ~ period3 + hormone3 + period3:hormone3 | period3,

                          data = trajan, dist = "negbin", link = "log")

summary(reduced\_model)

|  |
| --- |
| > summary(reduced\_model)  Call:  zeroinfl(formula = nshoots ~ period3 + hormone3 + period3:hormone3 | period3, data = trajan, dist = "negbin", link = "log")  Pearson residuals:  Min 1Q Median 3Q Max  -2.1643 -0.8479 -0.1923 0.6646 3.8887  Count model coefficients (negbin with log link):  Estimate Std. Error z value Pr(>|z|)  (Intercept) 1.5226 0.1188 12.811 < 2e-16 \*\*\*  period3Photoperiod: 16 hours 0.4875 0.1397 3.489 0.000485 \*\*\*  hormone3BAP (μM): 4.4 0.3494 0.1702 2.053 0.040040 \*  hormone3BAP (μM): 8.8 0.2359 0.1777 1.328 0.184333  hormone3BAP (μM): 17.6 0.1218 0.1697 0.718 0.472769  period3Photoperiod: 16 hours:hormone3BAP (μM): 4.4 -0.5960 0.2055 -2.900 0.003737 \*\*  period3Photoperiod: 16 hours:hormone3BAP (μM): 8.8 -0.1962 0.2084 -0.941 0.346570  period3Photoperiod: 16 hours:hormone3BAP (μM): 17.6 -0.1171 0.1979 -0.592 0.554054  Log(theta) 2.7348 0.3699 7.394 1.42e-13 \*\*\*  Zero-inflation model coefficients (binomial with log link):  Estimate Std. Error z value Pr(>|z|)  (Intercept) -0.75231 0.09404 -7.999 1.25e-15 \*\*\*  period3Photoperiod: 16 hours -3.66717 0.84617 -4.334 1.47e-05 \*\*\*  ---  Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  Theta = 15.4061  Number of iterations in BFGS optimization: 21  Log-likelihood: -616.2 on 11 Df |
|  |
| |  | | --- | |  | |

##### Interpretación del Modelo Reducido zeroinfl

#### 1. Residuos de Pearson

* **Mínimo:** -2.1643
* **1er cuartil (Q1):** -0.8479
* **Mediana:** -0.1923
* **3er cuartil (Q3):** 0.6646
* **Máximo:** 3.8887

Los residuos de Pearson nos indican la diferencia estandarizada entre los valores observados y los valores predichos por el modelo. Valores cercanos a cero indican un buen ajuste del modelo.

#### 2. Coeficientes del Modelo de Conteo (Modelo de Binomial Negativa con enlace logarítmico)

* **Intercepto:** 1.5226 (p < 2e-16) \*\*\*
* **Fotoperiodo de 16 horas (period3Photoperiod: 16 hours):** 0.4875 (p = 0.000485) \*\*\*
* **Hormona BAP (μM) 4.4 (hormone3BAP (μM): 4.4):** 0.3494 (p = 0.040040) \*
* **Hormona BAP (μM) 8.8 (hormone3BAP (μM): 8.8):** 0.2359 (p = 0.184333)
* **Hormona BAP (μM) 17.6 (hormone3BAP (μM): 17.6):** 0.1218 (p = 0.472769)
* **Interacción Fotoperiodo 16 horas y Hormona BAP (μM) 4.4 (period3Photoperiod: 16 hours**

**(μM): 4.4):** -0.5960 (p = 0.003737) \*\*

* **Interacción Fotoperiodo 16 horas y Hormona BAP (μM) 8.8 (period3Photoperiod: 16 hours**

**(μM): 8.8):** -0.1962 (p = 0.346570)

* **Interacción Fotoperiodo 16 horas y Hormona BAP (μM) 17.6 (period3Photoperiod: 16 hours**

**(μM): 17.6):** -0.1171 (p = 0.554054)

* **Log(theta):** 2.7348 (p = 1.42e-13) \*\*\*

Los coeficientes significativos (indicados con \*\*\*, \*\*, o \*) indican que esas variables tienen un impacto significativo en el número de brotes (nshoots).

#### 3. Coeficientes del Modelo de Inflación de Ceros (Modelo Binomial con enlace logarítmico)

* **Intercepto:** -0.75231 (p = 1.25e-15) \*\*\*
* **Fotoperiodo de 16 horas (period3Photoperiod: 16 hours):** -3.66717 (p = 1.47e-05) \*\*\*

Los coeficientes significativos en el modelo de inflación de ceros indican que esas variables tienen un impacto significativo en la probabilidad de tener un exceso de ceros en los datos.

#### 4. Parámetro Theta

* **Theta:** 15.4061

El valor de theta está relacionado con la dispersión de la distribución de la binomial negativa. Un valor alto de theta indica una menor dispersión.

#### 5. Log-Likelihood y Número de Iteraciones

* **Log-Likelihood:** -616.2
* **Número de Iteraciones en la Optimización BFGS:** 21

El log-likelihood nos da una medida de qué tan bien se ajusta el modelo a los datos. Un valor menos negativo es preferible. El número de iteraciones indica cuántas veces el algoritmo de optimización se ejecutó para encontrar el mejor ajuste del modelo.

##### Conclusión

El modelo reducido mantiene las variables significativas del modelo completo, eliminando aquellas que no contribuyen significativamente. Las variables period3 (Fotoperiodo de 16 horas) y hormone3 (BAP (μM) 4.4) junto con su interacción muestran significancia en el modelo de conteo. Además, el fotoperiodo de 16 horas es significativo en el modelo de inflación de ceros. Este modelo reducido es más parsimonioso y aún mantiene un buen ajuste a los datos.

##### Evaluar el Modelo Reducido

*# Ajustar un modelo reducido eliminando variables no significativas*

reduced\_model <- zeroinfl(nshoots ~ period3 + hormone3 + period3:hormone3 | period3,

                          data = trajan, dist = "negbin", link = "log")

summary(reduced\_model)

##### Validación Cruzada Manual

library(pscl)

library(caret)

*# Función para calcular AIC con validación cruzada*

cv\_aic <- function(data, formula, n\_splits = 10) {

  set.seed(123)  *# Para reproducibilidad*

  folds <- createFolds(data$nshoots, k = n\_splits, list = TRUE, returnTrain = FALSE)

  aic\_values <- c()

  for (i in 1:length(folds)) {

    train\_indices <- setdiff(1:nrow(data), folds[[i]])

    train\_data <- data[train\_indices, ]

    test\_data <- data[folds[[i]], ]

    model <- zeroinfl(formula, data = train\_data, dist = "negbin", link = "log")

*# Predicción en el conjunto de prueba*

    pred <- predict(model, newdata = test\_data, type = "response")

*# Calcular el AIC en el conjunto de prueba*

    loglik <- sum(dnbinom(test\_data$nshoots, size = model$theta, mu = pred, log = TRUE))

    aic <- -2 \* loglik + 2 \* length(coef(model))

    aic\_values <- c(aic\_values, aic)

  }

  return(mean(aic\_values))

}

*# Fórmulas del modelo completo y reducido*

full\_formula <- nshoots ~ period3 \* hormone3 + period3:hormone3

reduced\_formula <- nshoots ~ period3 + hormone3 + period3:hormone3

*# Calcular AIC promedio usando validación cruzada*

full\_model\_aic\_cv <- cv\_aic(trajan, full\_formula)

reduced\_model\_aic\_cv <- cv\_aic(trajan, reduced\_formula)

print(paste("AIC promedio del modelo completo:", full\_model\_aic\_cv))

print(paste("AIC promedio del modelo reducido:", reduced\_model\_aic\_cv))

> # Calcular AIC promedio usando validación cruzada

> full\_model\_aic\_cv <- cv\_aic(trajan, full\_formula)

There were 50 or more warnings (use warnings() to see the first 50)

> reduced\_model\_aic\_cv <- cv\_aic(trajan, reduced\_formula)

There were 50 or more warnings (use warnings() to see the first 50)

>

> print(paste("AIC promedio del modelo completo:", full\_model\_aic\_cv))

[1] "AIC promedio del modelo completo: 184.736513868523"

> print(paste("AIC promedio del modelo reducido:", reduced\_model\_aic\_cv))

[1] "AIC promedio del modelo reducido: 184.736513868523"

#### 1. **Eliminación Basada en el Valor p**

Una manera común de reducir variables es eliminar aquellas que no son estadísticamente significativas. Esto se basa en los valores p de los coeficientes en el modelo.

**Paso a Paso:**

* **Revisar el resumen del modelo actual** para identificar variables con valores p altos (generalmente mayores a 0.05).
* **Eliminar variables no significativas** y ajustar el modelo nuevamente.

**Ejemplo de Código:**

r

Copiar código

# Revisar el resumen del modelo actual

summary(best\_predictive\_model)

# Eliminar variables con valores p altos, por ejemplo, eliminar variables con p > 0.05

reduced\_model <- zeroinfl(nshoots ~ period3 \* hormone3 + period3:hormone3 | period3 + hormone3,

data = trajan, dist = "negbin", link = "log")

summary(reduced\_model)

#### 2. **Método de Selección hacia Atrás (Backward Selection)**

La selección hacia atrás comienza con todos los posibles predictores y elimina uno a la vez para mejorar el AIC (Criterio de Información de Akaike).

**Paso a Paso:**

* **Ajustar un modelo completo** con todas las variables.
* **Eliminar la variable** que causa el menor decremento en el AIC y ajustar el modelo nuevamente.
* **Repetir** el proceso hasta que no haya mejoras significativas en el AIC.

**Ejemplo de Código:**

r

Copiar código

# Ajustar un modelo completo

full\_model <- zeroinfl(nshoots ~ period3 \* hormone3 + period3:hormone3 | period3 + hormone3,

data = trajan, dist = "negbin", link = "log")

# Realizar la selección hacia atrás basada en AIC

backward\_model <- step(full\_model, direction = "backward")

summary(backward\_model)

#### 3. **Método de Selección hacia Adelante (Forward Selection)**

La selección hacia adelante comienza con un modelo vacío y agrega variables una a una para mejorar el AIC.

**Paso a Paso:**

* **Ajustar un modelo vacío** y agregar variables una a una.
* **Evaluar la mejora en el AIC** para cada variable añadida.

**Ejemplo de Código:**

r

Copiar código

# Ajustar un modelo vacío

empty\_model <- zeroinfl(nshoots ~ 1 | 1, data = trajan, dist = "negbin", link = "log")

# Realizar la selección hacia adelante basada en AIC

forward\_model <- step(empty\_model, scope = list(lower = empty\_model, upper = full\_model), direction = "forward")

summary(forward\_model)

#### 4. **Método de Selección de Variables Basado en AIC**

Puedes usar un algoritmo de selección de variables basado en el AIC para evaluar diferentes combinaciones de variables.

**Paso a Paso:**

* **Generar todas las posibles combinaciones** de variables.
* **Evaluar cada combinación** usando AIC para seleccionar el mejor modelo.

**Ejemplo de Código:**

r

Copiar código

# Crear una función para ajustar el modelo y calcular AIC

evaluate\_model <- function(formula\_count, formula\_zero, data) {

model <- zeroinfl(formula\_count | formula\_zero, data = data, dist = "negbin", link = "log")

aic <- AIC(model)

return(list(model = model, aic = aic))

}

# Generar diferentes combinaciones de variables

library(MASS)

combinations <- list(

list(count\_formula = nshoots ~ period3 + hormone3, zero\_formula = period3 + hormone3),

list(count\_formula = nshoots ~ period3 \* hormone3, zero\_formula = period3 + hormone3),

list(count\_formula = nshoots ~ period3 + hormone3 + period3:hormone3, zero\_formula = period3 + hormone3),

list(count\_formula = nshoots ~ period3 + hormone3 + period3:hormone3 + hormone3:period3, zero\_formula = period3 + hormone3)

)

# Ajustar los modelos y almacenar los resultados

results <- list()

for (i in 1:length(combinations)) {

formulas <- combinations[[i]]

count\_formula <- formulas$count\_formula

zero\_formula <- formulas$zero\_formula

model\_results <- evaluate\_model(count\_formula, zero\_formula, trajan)

results[[i]] <- list(count\_formula = count\_formula, zero\_formula = zero\_formula, model = model\_results$model, aic = model\_results$aic)

}

# Seleccionar el mejor modelo basado en AIC

best\_model\_index <- which.min(sapply(results, function(x) x$aic))

best\_model <- results[[best\_model\_index]]$model

summary(best\_model)

#### 5. **Método de Evaluación Cruzada (Cross-Validation)**

La evaluación cruzada puede ayudarte a comparar el rendimiento de diferentes modelos para evitar el sobreajuste.

**Paso a Paso:**

* **Dividir los datos en** conjuntos de entrenamiento y prueba.
* **Entrenar el modelo** en el conjunto de entrenamiento.
* **Evaluar el modelo** en el conjunto de prueba usando métricas como AUC, precisión, y error.

**Ejemplo de Código:**

r

Copiar código

library(caret)

# Definir el control de la validación cruzada

train\_control <- trainControl(method = "cv", number = 10)

# Ajustar el modelo con validación cruzada

cv\_model <- train(nshoots ~ period3 \* hormone3 + period3:hormone3,

data = trajan, method = "zeroinfl", trControl = train\_control, dist = "negbin", link = "log")

# Resumen del modelo

print(cv\_model)

#### **Conclusión de la Reducción de Variables**

Para reducir variables y encontrar el modelo más predictivo, puedes:

1. **Eliminar variables no significativas** y ajustar el modelo (eliminación basada en valor p).
2. **Usar métodos de selección hacia atrás** o hacia adelante para encontrar un modelo más simple basado en AIC.
3. **Generar y evaluar diferentes combinaciones de variables** para seleccionar el modelo con el AIC más bajo.
4. **Aplicar la validación cruzada** para asegurar que el modelo no esté sobreajustado y generalice bien.