

```

from sklearn.impute import KNNImputer
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import RobustScaler

#cargar dataset
df_completo = pd.read_csv('Fortnite_players_stats.csv', sep=",")

print(df_completo.columns)

Index(['Player', 'Solo score', 'Solo top1', 'Solo kd', 'Solo
winRatio',
      'Solo matches', 'Solo kills', 'Solo minutesPlayed', 'Duos
score',
      'Duos top1', 'Duos kd', 'Duos winRatio', 'Duos matches', 'Duos
kills',
      'Duos minutesPlayed', 'Trios score', 'Trios top1', 'Trios kd',
      'Trios winRatio', 'Trios matches', 'Trios kills', 'Trios
minutesPlayed',
      'Squads score', 'Squads top1', 'Squads kd', 'Squads winRatio',
      'Squads matches', 'Squads kills', 'Squads minutesPlayed', 'LTM
score',
      'LTM top1', 'LTM top3', 'LTM kd', 'LTM winRatio', 'LTM
matches',
      'LTM kills', 'LTM minutesPlayed'],
      dtype='object')

df = df_completo[['Player', 'Solo score', 'Solo top1', 'Solo kd',
                  'Solo winRatio', 'Solo matches', 'Solo
kills', 'Solo minutesPlayed']]

```

FASE 2: Data Understanding

Verificar columnas y tipos de datos

```

columnas = df.columns
for columna in columnas:
    if df[columna].dtype == int:
        print(f"Columna {columna} ----- Tipo de variable: Cuantitativa
discreta")
    elif df[columna].dtype == float:
        print(f"Columna {columna} ----- Tipo de variable: Cuantitativa
continua")
    elif df[columna].dtype == object:
        print(f"Columna {columna} ----- Tipo de variable: Cualitativa
nominal")

```

```

elif df[columna].dtype == bool:
    print(f"Columna {columna} ----- Tipo de variable: Cualitativa ordinal")

```

```

Columna Player ----- Tipo de variable: Cualitativa nominal
Columna Solo score ----- Tipo de variable: Cuantitativa discreta
Columna Solo top1 ----- Tipo de variable: Cuantitativa discreta
Columna Solo kd ----- Tipo de variable: Cuantitativa continua
Columna Solo winRatio ----- Tipo de variable: Cuantitativa continua
Columna Solo matches ----- Tipo de variable: Cuantitativa discreta
Columna Solo kills ----- Tipo de variable: Cuantitativa discreta
Columna Solo minutesPlayed ----- Tipo de variable: Cuantitativa discreta

```

```

# Se filtran las 10 primeras Filas
df.head(10)

```

```

{"column_count":8,"columns":
[{"dtype":"object","name":"Player","stats":{"categories":
[{"count":1,"name":"Prospering"}, {"count":1,"name":"BH nixxxay"},
{"count":8,"name":"8 others"}], "nan_count":0,"unique_count":10}},
{"dtype":"int64","name":"Solo score","stats":{"histogram":
[{"bin_end":691496.4,"bin_start":15872,"count":3},
{"bin_end":1367120.8,"bin_start":691496.4,"count":2},
{"bin_end":2042745.2000000002,"bin_start":1367120.8,"count":0},
{"bin_end":2718369.6,"bin_start":2042745.2000000002,"count":2},
{"bin_end":3393994,"bin_start":2718369.6,"count":1},
{"bin_end":4069618.4000000004,"bin_start":3393994,"count":0},
{"bin_end":4745242.8,"bin_start":4069618.4000000004,"count":1},
{"bin_end":5420867.2,"bin_start":4745242.8,"count":0},
{"bin_end":6096491.6000000001,"bin_start":5420867.2,"count":0},
{"bin_end":6772116,"bin_start":6096491.6000000001,"count":1}], "max":"67
72116", "min":"15872", "nan_count":0,"unique_count":10}},
{"dtype":"int64","name":"Solo top1","stats":{"histogram":
[{"bin_end":653.4,"bin_start":6,"count":3},
{"bin_end":1300.8,"bin_start":653.4,"count":0},
{"bin_end":1948.1999999999998,"bin_start":1300.8,"count":5},
{"bin_end":2595.6,"bin_start":1948.1999999999998,"count":0},
{"bin_end":3243,"bin_start":2595.6,"count":0},
{"bin_end":3890.3999999999996,"bin_start":3243,"count":0},
{"bin_end":4537.8,"bin_start":3890.3999999999996,"count":0},
{"bin_end":5185.2,"bin_start":4537.8,"count":1},
{"bin_end":5832.599999999999,"bin_start":5185.2,"count":0},
{"bin_end":6480,"bin_start":5832.599999999999,"count":1}], "max":"6480"
,"min":"6", "nan_count":0,"unique_count":10}},
{"dtype":"float64","name":"Solo kd","stats":{"histogram":
[{"bin_end":1.383,"bin_start":0.47,"count":3},
{"bin_end":2.296,"bin_start":1.383,"count":1},
{"bin_end":3.2089999999999996,"bin_start":2.296,"count":0},
{"bin_end":4.122,"bin_start":3.2089999999999996,"count":2},

```

```

{"bin_end":5.034999999999999,"bin_start":4.122,"count":1},
{"bin_end":5.9479999999999995,"bin_start":5.034999999999999,"count":0}
,
{"bin_end":6.860999999999999,"bin_start":5.9479999999999995,"count":1}
,
{"bin_end":7.773999999999999,"bin_start":6.860999999999999,"count":0},
{"bin_end":8.687,"bin_start":7.773999999999999,"count":0},
{"bin_end":9.6,"bin_start":8.687,"count":2}], "max": "9.6", "min": "0.47",
"nan_count": 0, "unique_count": 10}}, {"dtype": "float64", "name": "Solo
winRatio", "stats": {"histogram":
[{"bin_end":3.85,"bin_start":0.5,"count":3},
{"bin_end":7.2,"bin_start":3.85,"count":0},
{"bin_end":10.55,"bin_start":7.2,"count":2},
{"bin_end":13.9,"bin_start":10.55,"count":0},
{"bin_end":17.25,"bin_start":13.9,"count":0},
{"bin_end":20.6,"bin_start":17.25,"count":1},
{"bin_end":23.95,"bin_start":20.6,"count":0},
{"bin_end":27.3,"bin_start":23.95,"count":0},
{"bin_end":30.650000000000002,"bin_start":27.3,"count":2},
{"bin_end":34,"bin_start":30.650000000000002,"count":2}], "max": "34.0",
"min": "0.5", "nan_count": 0, "unique_count": 10}},
{"dtype": "int64", "name": "Solo matches", "stats": {"histogram":
[{"bin_end":2372.5,"bin_start":429,"count":2},
{"bin_end":4316,"bin_start":2372.5,"count":1},
{"bin_end":6259.5,"bin_start":4316,"count":2},
{"bin_end":8203,"bin_start":6259.5,"count":0},
{"bin_end":10146.5,"bin_start":8203,"count":0},
{"bin_end":12090,"bin_start":10146.5,"count":1},
{"bin_end":14033.5,"bin_start":12090,"count":2},
{"bin_end":15977,"bin_start":14033.5,"count":0},
{"bin_end":17920.5,"bin_start":15977,"count":0},
{"bin_end":19864,"bin_start":17920.5,"count":2}], "max": "19864", "min": "
429", "nan_count": 0, "unique_count": 10}}, {"dtype": "int64", "name": "Solo
kills", "stats": {"histogram":
[{"bin_end":8728.1,"bin_start":200,"count":3},
{"bin_end":17256.2,"bin_start":8728.1,"count":0},
{"bin_end":25784.300000000003,"bin_start":17256.2,"count":2},
{"bin_end":34312.4,"bin_start":25784.300000000003,"count":0},
{"bin_end":42840.5,"bin_start":34312.4,"count":3},
{"bin_end":51368.600000000006,"bin_start":42840.5,"count":0},
{"bin_end":59896.700000000004,"bin_start":51368.600000000006,"count":0
}, {"bin_end":68424.8,"bin_start":59896.700000000004,"count":1},
{"bin_end":76952.900000000001,"bin_start":68424.8,"count":0},
{"bin_end":85481,"bin_start":76952.900000000001,"count":1}], "max": "8548
1", "min": "200", "nan_count": 0, "unique_count": 10}},
{"dtype": "int64", "name": "Solo minutesPlayed", "stats": {"histogram":
[{"bin_end":29048.6,"bin_start":739,"count":3},
{"bin_end":57358.2,"bin_start":29048.6,"count":2},
{"bin_end":85667.79999999999,"bin_start":57358.2,"count":2},

```

```
{
  "bin_end": 113977.4, "bin_start": 85667.79999999999, "count": 1},
  {"bin_end": 142287, "bin_start": 113977.4, "count": 1},
  {"bin_end": 170596.59999999998, "bin_start": 142287, "count": 0},
  {"bin_end": 198906.19999999998, "bin_start": 170596.59999999998, "count": 0},
  {"bin_end": 227215.8, "bin_start": 198906.19999999998, "count": 0},
  {"bin_end": 255525.4, "bin_start": 227215.8, "count": 0},
  {"bin_end": 283835, "bin_start": 255525.4, "count": 1}], "max": "283835", "min": "739", "nan_count": 0, "unique_count": 10}},
{"dtype": "int64", "name": "_deepnote_index_column"}], "row_count": 10, "rows": [
  {"Player": "Prospering", "Solo kd": 4.37, "Solo kills": 36328, "Solo matches": 10150, "Solo minutesPlayed": 81389, "Solo score": 2476763, "Solo top1": 1828, "Solo winRatio": 18, "_deepnote_index_column": 0},
  {"Player": "BH nixxay", "Solo kd": 8.71, "Solo kills": 35895, "Solo matches": 5817, "Solo minutesPlayed": 12732, "Solo score": 439562, "Solo top1": 1694, "Solo winRatio": 29.1, "_deepnote_index_column": 1},
  {"Player": "Ranger", "Solo kd": 9.6, "Solo kills": 85481, "Solo matches": 13488, "Solo minutesPlayed": 122171, "Solo score": 4519465, "Solo top1": 4582, "Solo winRatio": 34, "_deepnote_index_column": 2},
  {"Player": "Idk_Pi", "Solo kd": 0.84, "Solo kills": 3005, "Solo matches": 3687, "Solo minutesPlayed": 32453, "Solo score": 752869, "Solo top1": 121, "Solo winRatio": 3.3, "_deepnote_index_column": 3},
  {"Player": "Twitch.GryphonRB", "Solo kd": 6.32, "Solo kills": 19591, "Solo matches": 4429, "Solo minutesPlayed": 36245, "Solo score": 1136282, "Solo top1": 1327, "Solo winRatio": 30, "_deepnote_index_column": 4},
  {"Player": "Twitch Kayotica", "Solo kd": 3.23, "Solo kills": 39131, "Solo matches": 13438, "Solo minutesPlayed": 96777, "Solo score": 2919037, "Solo top1": 1310, "Solo winRatio": 9.7, "_deepnote_index_column": 5},
  {"Player": "FaZe Replays", "Solo kd": 3.84, "Solo kills": 66161, "Solo matches": 18670, "Solo minutesPlayed": 76258, "Solo score": 2389537, "Solo top1": 1454, "Solo winRatio": 7.8, "_deepnote_index_column": 6},
  {"Player": "NiteGamerYT 190k", "Solo kd": 0.47, "Solo kills": 200, "Solo matches": 429, "Solo minutesPlayed": 739, "Solo score": 15872, "Solo top1": 6, "Solo winRatio": 1.4, "_deepnote_index_column": 7},
  {"Player": "Boss >", "Solo kd": 1.39, "Solo kills": 18610, "Solo matches": 19864, "Solo minutesPlayed": 283835, "Solo score": 6772116, "Solo top1": 6480, "Solo winRatio": 32.6, "_deepnote_index_column": 8},
  {"Player": "CIUPEA 144.HZ", "Solo kd": 0.61, "Solo kills": 1174, "Solo matches": 1938, "Solo minutesPlayed": 2441, "Solo score": 54479, "Solo top1": 9, "Solo winRatio": 0.5, "_deepnote_index_column": 9}]]
```

```
#Se Filtra el tipo de dato de la columnas
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1435 entries, 0 to 1434
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Player                1435 non-null   object
1   Solo score            1435 non-null   int64
```

2	Solo top1	1435	non-null	int64
3	Solo kd	1435	non-null	float64
4	Solo winRatio	1435	non-null	float64
5	Solo matches	1435	non-null	int64
6	Solo kills	1435	non-null	int64
7	Solo minutesPlayed	1435	non-null	int64

dtypes: float64(2), int64(5), object(1)
memory usage: 89.8+ KB

Verificación de datos nulos y posibles errores en DataFrame.

```
# Verificación de datos nulos en el dataset
df.isnull().sum()

Player          0
Solo score      0
Solo top1       0
Solo kd         0
Solo winRatio   0
Solo matches    0
Solo kills      0
Solo minutesPlayed 0
dtype: int64

# Verificación de valores nulos
nulos = df.isnull().sum()

# Mostrar solo las columnas con valores nulos
nulos = nulos[nulos > 0]
print(nulos)

Series([], dtype: int64)

# Porcentaje de valores nulos por columna
porcentaje_nulos = (nulos / len(df)) * 100
print(porcentaje_nulos)

Series([], dtype: float64)
```

Verificamos extensamente en la búsqueda de datos nulos pero de momento los datos a utilizar no arrojan datos nulos, pero existe la posibilidad que hayan datos negativos que los interprete como datos no nulos por lo que se verificara con el siguiente código.

```
# Verificar si hay valores negativos en las columnas relevantes
for col in ['Solo score', 'Solo kills', 'Solo kd', 'Solo winRatio']:
    negativos = df[df[col] < 0]
    if not negativos.empty:
        print(f"Valores negativos en {col}:")
        print(negativos)
```

Posibles columnas con datos negativos: Seleccionamos las columnas mas relevantes para saber si existen valores negativos. La cual se encontraron múltiples jugadores que tienen valores negativos en la columna "Solo kills". Esto es problemático porque en el contexto de los videojuegos, es ilógico tener un número negativo de "kills" (muertes o eliminaciones de jugadores). Un "kill" siempre debe ser un número no negativo.

En la columna "Solo minutesPlayed", algunos jugadores tienen valores negativos. Al igual que con "Solo kills", no debería haber tiempos de juego negativos, ya que el tiempo es siempre positivo.

Algunos valores negativos aparecen en "Solo kd" (K/D ratio), lo que también es inusual. El K/D ratio debe estar en un rango no negativo, a menos que haya una interpretación específica en los datos.

```
df.loc[:, 'Solo kills'] = df['Solo kills'].apply(lambda x: x if x >= 0 else np.nan)
df.loc[:, 'Solo minutesPlayed'] = df['Solo minutesPlayed'].apply(lambda x: x if x >= 0 else np.nan)
```

Reemplazo de los valores: Ahora reemplazamos los valores negativos con NaN (valores nulos), para luego decidir si se deben rellenar, ignorar o eliminar. También reemplazamos con cero si tiene sentido en el contexto (especialmente en kills o tiempo jugado).

```
# Verificar si hay datos negativos
(df[['Solo score', 'Solo top1', 'Solo kd', 'Solo winRatio', 'Solo matches', 'Solo kills', 'Solo minutesPlayed']] < 0).sum()
```

Solo score	0
Solo top1	0
Solo kd	0
Solo winRatio	0
Solo matches	0
Solo kills	0
Solo minutesPlayed	0
dtype:	int64

Ahora al no existir datos negativos y nulos, estamos listos para comenzar las medidas de posición.

Medidas de posición

```
#Resumen de Medidas de posición, para datos numéricos
print(df.describe())
```

	Solo score	Solo top1	Solo kd	Solo winRatio	Solo matches
count	1.435000e+03	1435.000000	1435.000000	1435.000000	1435.000000
mean	1.088202e+06	328.909408	1.816300	4.853937	6911.060627

std	8.872639e+05	529.192568	1.156026	5.330249
5408.219523				
min	4.453000e+03	0.000000	0.080000	0.000000
30.000000				
25%	4.355170e+05	65.500000	1.050000	1.600000
2875.000000				
50%	9.070010e+05	162.000000	1.480000	2.900000
5728.000000				
75%	1.506555e+06	366.000000	2.200000	6.100000
9585.000000				
max	6.772116e+06	6480.000000	9.600000	44.100000
36487.000000				

	Solo kills	Solo minutesPlayed
count	1435.000000	1435.000000
mean	11781.843206	43517.135889
std	11417.340071	36314.877771
min	28.000000	210.000000
25%	3861.000000	16922.000000
50%	8452.000000	35762.000000
75%	15993.000000	59471.000000
max	89023.000000	283835.000000

Excluir columnas no numéricas

```
df_numeric = df.select_dtypes(include=[float, int])
```

Calcular la matriz de correlación solo con las columnas numéricas

```
correlation_matrix = df_numeric.corr()
```

Mostrar la matriz de correlación

```
print(correlation_matrix)
```

	Solo score	Solo top1	Solo kd	Solo winRatio	\
Solo score	1.000000	0.598385	0.203172	0.245719	
Solo top1	0.598385	1.000000	0.659677	0.768525	
Solo kd	0.203172	0.659677	1.000000	0.854501	
Solo winRatio	0.245719	0.768525	0.854501	1.000000	
Solo matches	0.860244	0.361533	0.036813	-0.022918	
Solo kills	0.715041	0.748277	0.613477	0.448650	
Solo minutesPlayed	0.979049	0.484908	0.067514	0.133830	

	Solo matches	Solo kills	Solo minutesPlayed
Solo score	0.860244	0.715041	0.979049
Solo top1	0.361533	0.748277	0.484908
Solo kd	0.036813	0.613477	0.067514
Solo winRatio	-0.022918	0.448650	0.133830
Solo matches	1.000000	0.705988	0.859965
Solo kills	0.705988	1.000000	0.601728
Solo minutesPlayed	0.859965	0.601728	1.000000

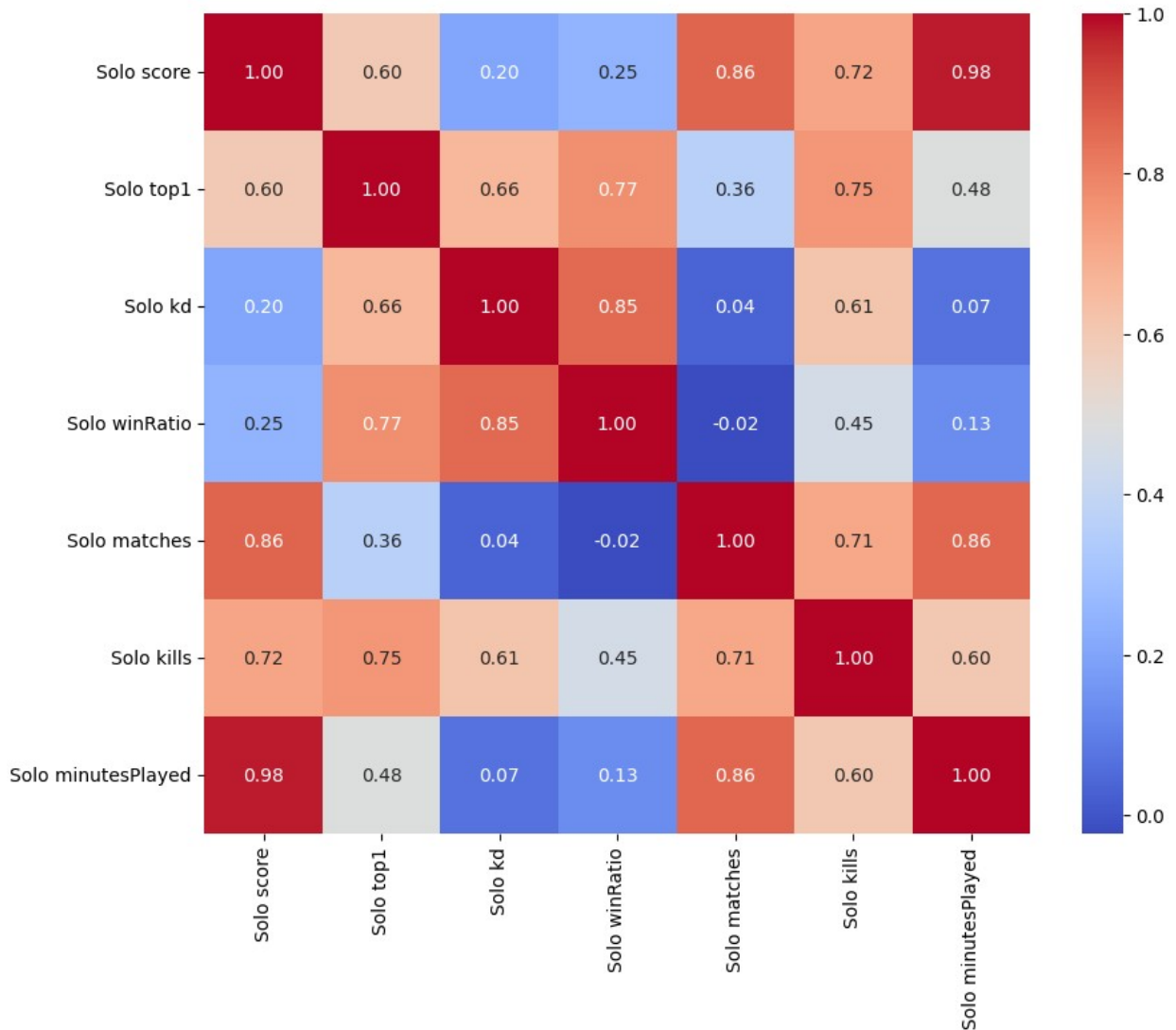
Matriz de correlación

Las variables Solo score, Solo matches y Solo minutesPlayed están fuertemente correlacionadas, podemos decir que el tiempo jugado y el número de partidas tienen un gran impacto en la puntuación total.

Solo winRatio está fuertemente correlacionado con la capacidad del jugador de estar en el Solo top1 y con su habilidad de combate (Solo kd).

No hay correlación significativa entre algunas variables, como Solo matches y Solo kd, lo que puede sugerir que jugar más partidas no mejora la relación de asesinatos/muertes de un jugador.

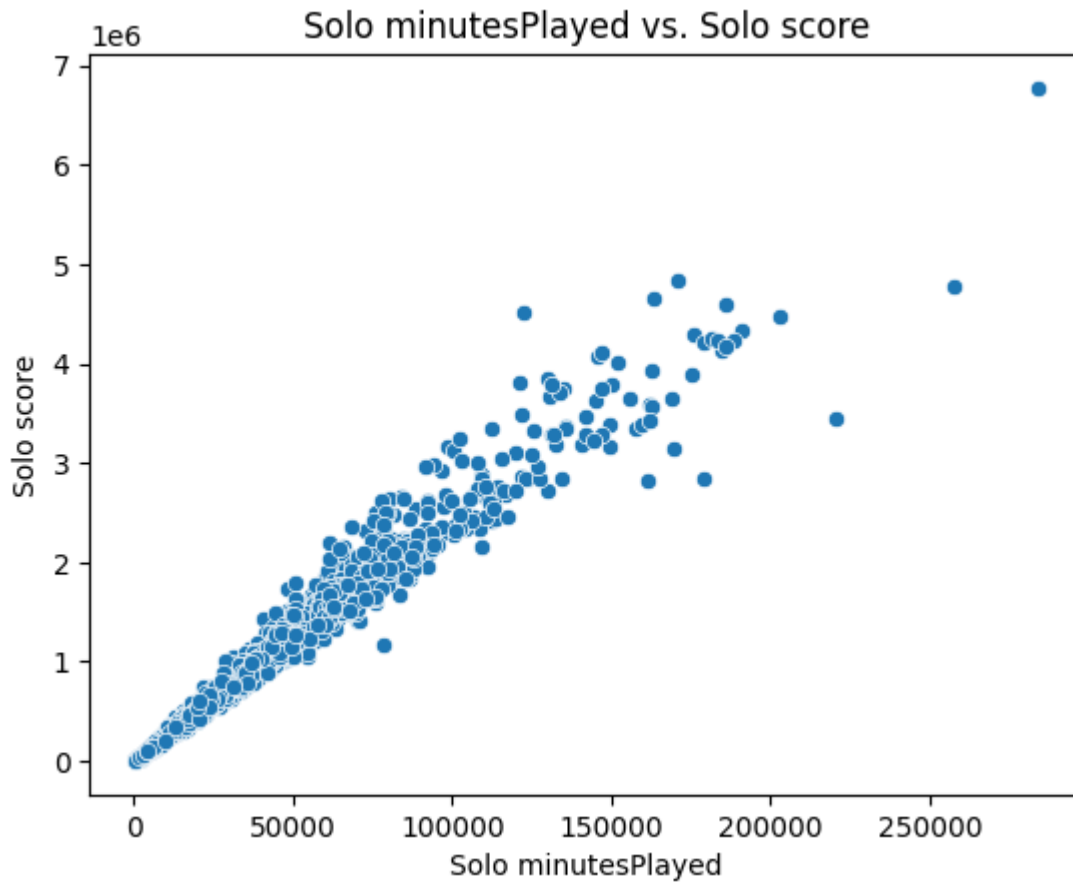
```
plt.figure(figsize=(10, 8))
sb.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.show()
```

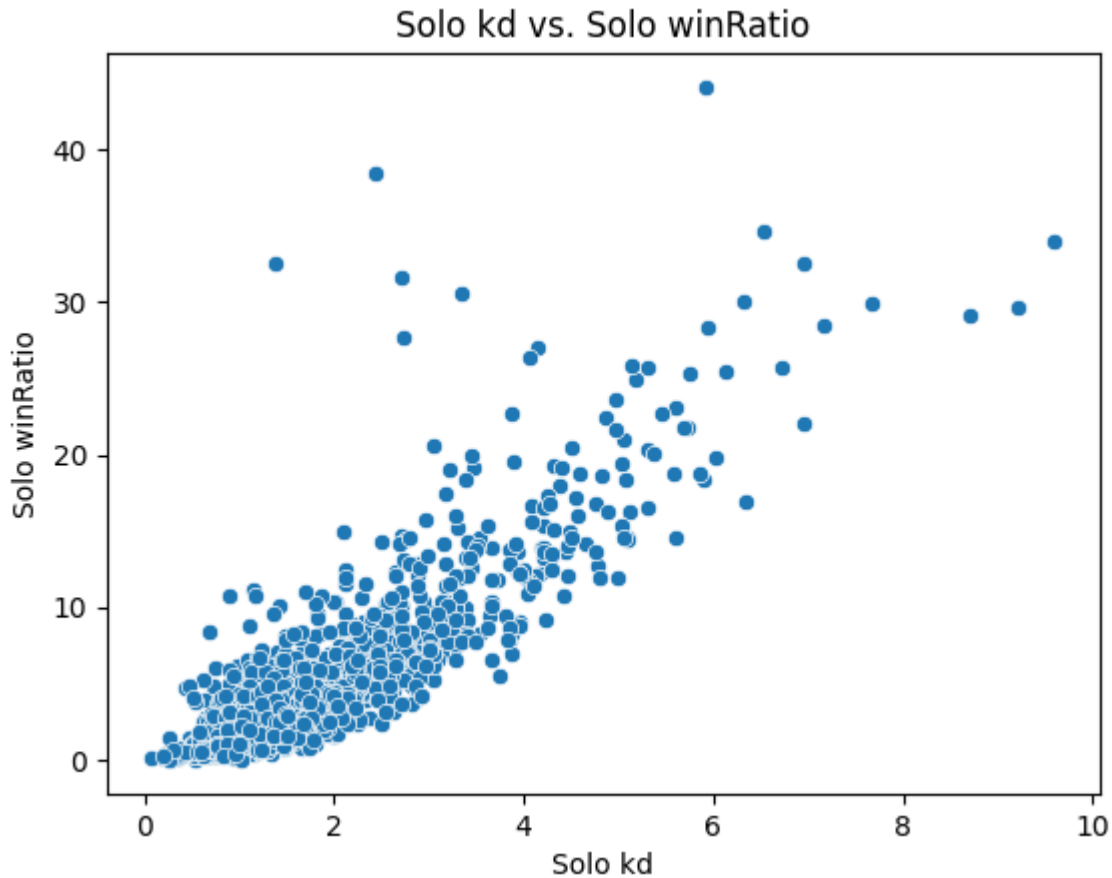


Scatter Plots (Diagramas de Dispersión):

```
sb.scatterplot(x='Solo minutesPlayed', y='Solo score', data=df)
plt.title('Solo minutesPlayed vs. Solo score')
plt.show()
```

```
sb.scatterplot(x='Solo kd', y='Solo winRatio', data=df)
plt.title('Solo kd vs. Solo winRatio')
plt.show()
```





Solo minutesPlayed vs. Solo score: Dado que existe una fuerte correlación entre el tiempo jugado y el puntaje, un scatter plot puede mostrar cómo el puntaje de los jugadores aumenta con el tiempo jugado. Esto es una clara tendencia positiva, lo que sugiere que a medida que los jugadores juegan más minutos, su puntuación total aumenta. Esto es lógico, ya que mientras más juegas generalmente proporcionas más oportunidades de obtener más puntaje.

Solo kd vs. Solo winRatio: Con una correlación alta entre el ratio KD y el winRatio, un scatter plot puede ayudar a visualizar cómo se relacionan estas dos métricas. Lo que muestra que a medida que aumenta el KD de un jugador, también tiende a aumentar su ratio de victorias.

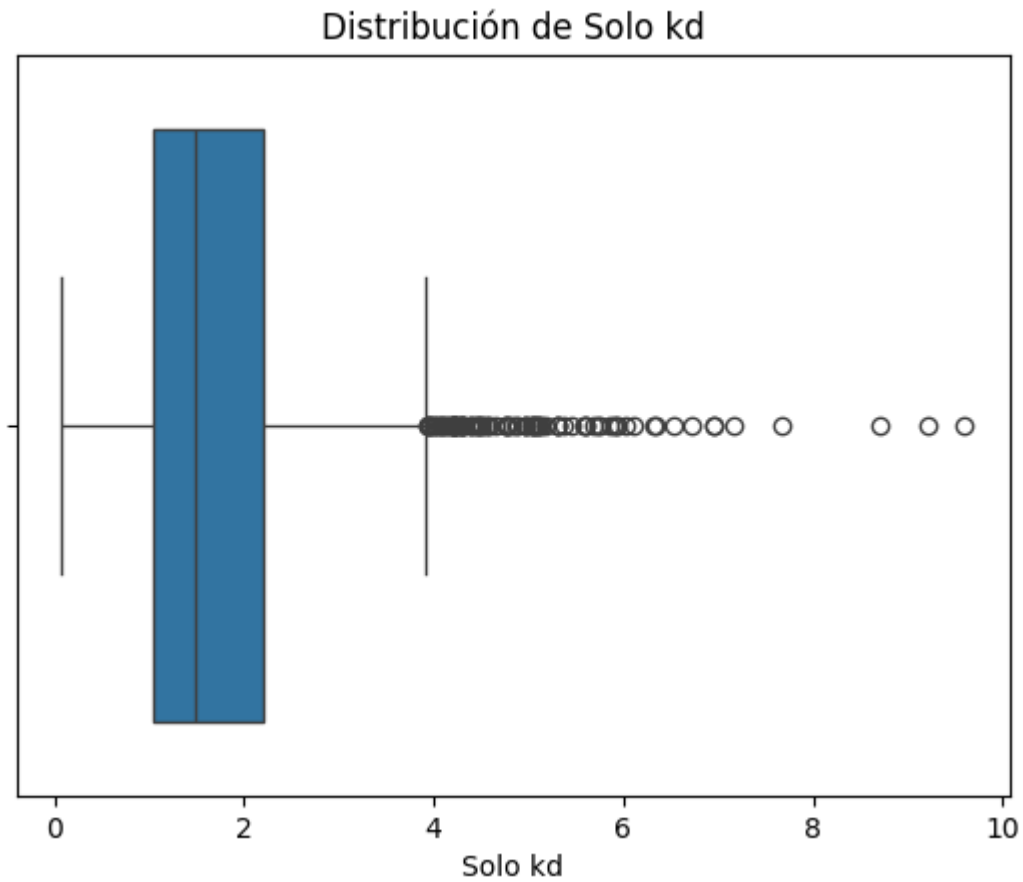
Los puntos en la parte superior derecha del gráfico (alto KD y alto winRatio) representan jugadores con alto potencial. Estos jugadores tienen un KD superior a 4 y un winRatio por encima del 20%. Epic Games podría enfocarse en estos jugadores para programas de jugadores destacados, torneos o como modelos a seguir para la comunidad.

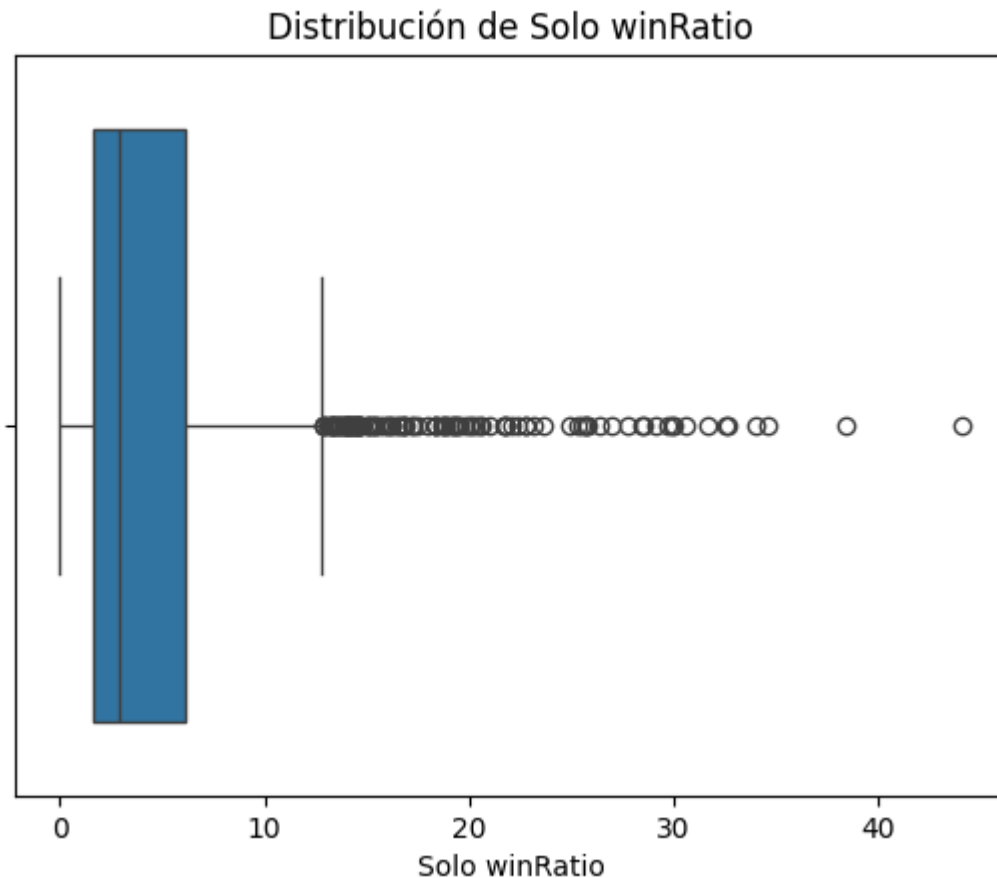
Si bien jugar más aumenta la puntuación, la cantidad de victorias y KD podrían depender más de la habilidad que solo del tiempo jugado. Es decir, un jugador que ha jugado muchas horas no necesariamente tendrá un buen KD o un gran número de victorias si su rendimiento en el juego no es eficiente.

Box Plot

```
sb.boxplot(x='Solo kd', data=df)
plt.title('Distribución de Solo kd')
plt.show()

sb.boxplot(x='Solo winRatio', data=df)
plt.title('Distribución de Solo winRatio')
plt.show()
```





Solo KD: La línea dentro de la caja representa la mediana del Solo kd, que parece estar alrededor de 1. Esto significa que el 50% de los jugadores tienen un ratio de eliminación-muerte menor o igual a 1.

Se observa un grupo considerable de outliers a la derecha del gráfico, es decir, jugadores que tienen un Solo kd significativamente mayor que el promedio. Algunos jugadores tienen valores de Solo kd que llegan hasta 10, pero son casos poco comunes.

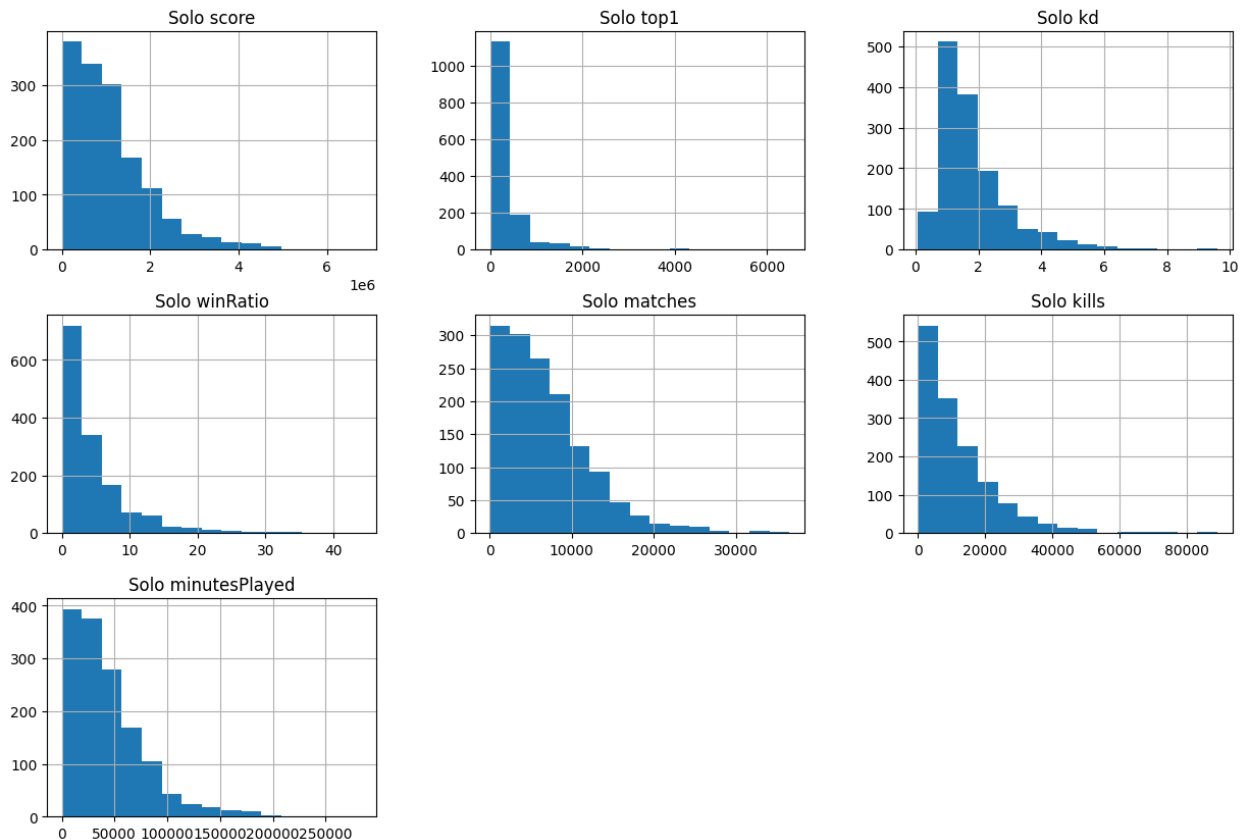
Solo winRatio: La línea en la caja sugiere que la mediana del win ratio está entre 5% y 10%, lo que indica que el 50% de los jugadores ganan menos del 10% de las partidas que juegan.

Nuevamente, se observan muchos outliers hacia la derecha, con algunos jugadores logrando ratios de victoria que superan el 40%. Estos jugadores son excepcionales en comparación con la mayoría.

Histogramas

```
df_numeric.hist(bins=15, figsize=(15, 10), layout=(3, 3))
plt.suptitle('Distribuciones de Variables Numéricas')
plt.show()
```

Distribuciones de Variables Numéricas



Solo score: -Hay una concentración de valores bajos y una larga cola hacia la derecha, lo que nos indicaría que la mayoría de los jugadores tienen una puntuación baja, mientras que muy pocos tienen puntuaciones extremadamente altas. Solo top1: -La mayoría de los jugadores tienen pocas o ninguna victoria, mientras que solo unos pocos tienen un gran número de victorias. Solo kd: -La mayoría de los jugadores tienen un kd bajo, con pocos jugadores teniendo un kd muy alto. -Tiene una distribución sesgada a la derecha, indicando que la mayoría de los jugadores no tienen una relación de asesinatos/muertes alta. Solo winRatio: -La mayoría de los jugadores tienen una proporción de victorias baja (menor a 10%), con solo unos pocos jugadores alcanzando porcentajes mayores. Solo matches: -Se tiene una distribución sesgada a la derecha, nos indica que es común tener pocos jugadores que juegan muchas partidas y que la mayoría han jugado un número relativamente bajo de partidas, mientras que unos pocos han jugado muchas más. Solo kills: -La mayoría de los jugadores tienen un número bajo de asesinatos, y pocos jugadores tienen un número muy alto. -Esta variable también muestra una distribución sesgada a la derecha. Solo minutesPlayed: -La mayoría de los jugadores tienen menos tiempo de juego, y hay algunos jugadores con una cantidad de tiempo de juego muy alta.

Identificar los jugadores con el mejor rendimiento en Modo Solitario

```
# Ordenar por rendimiento en el modo Solo
mejores_jugadores_solo = df.sort_values(by=['Solo score', 'Solo
```

```
winRatio', 'Solo kd'], ascending=False)

# Mostrar los jugadores con el mejor rendimiento en cada modo
print("Mejores jugadores en Solo:\n",
mejores_jugadores_solo[['Player', 'Solo score', 'Solo winRatio', 'Solo
kd']].head(10))
```

Mejores jugadores en Solo:

	Player	Solo score	Solo winRatio	Solo kd
8	B o s s ン	6772116	32.6	1.39
11	DarkAssaSSin._	4840719	30.6	3.34
110	TEBELKING	4784847	5.5	0.95
14	MachiTv	4659548	19.5	3.90
53	Ninjadad5	4595164	0.8	1.01
2	Ranger	4519465	34.0	9.60
144	RailCart	4479084	1.8	0.73
37	Ci2i	4323464	1.6	0.84
17	pr0Mancos	4300077	1.8	1.27
186	facu giorgetta29	4253032	7.0	1.36

Cálculo de los jugadores de menor rendimiento, riesgo de abandono en el Modo Solitario y disminución significativa en el número de partidas jugadas

```
# Filtrar jugadores con Solo kd y Solo winRatio bajos (por debajo de
la mediana)
jugadorBajoRendimiento = df[(df['Solo kd'] < df['Solo kd'].median()) &
(df['Solo winRatio'] < df['Solo
winRatio'].median())]

# Guardar el dataset filtrado como archivo CSV
jugadorBajoRendimiento.to_csv('jugadoresBajoRendimiento.csv',
index=False)

# Filtrar jugadores que han jugado pocas partidas en comparación con
el promedio
jugadoresPocasPartidas = df[df['Solo matches'] < df['Solo
matches'].quantile(0.25)]
# Guardar el dataset
jugadoresPocasPartidas.to_csv('jugadoresPocasPartidas.csv',
index=False)

# Filtrar jugadores con bajo rendimiento y baja actividad
jugadorBajoRendimientoYBajaAct = df[(df['Solo kd'] < df['Solo
kd'].median()) &
(df['Solo winRatio'] < df['Solo
winRatio'].median()) &
(df['Solo matches'] < df['Solo
```

```

matches'].quantile(0.25)]]

# Guardar el dataset
jugadorBajoRendimientoyBajaAct.to_csv('jugadorBajoRendimientoyBajaAct.
csv', index=False)

jugadorBajoRendimiento.head(10)

{"column_count":8,"columns":
[{"dtype":"object","name":"Player","stats":{"categories":
[{"count":1,"name":"NiteGamerYT 190k"}, {"count":1,"name":"CIUPEA
144.HZ"}, {"count":8,"name":"8
others"}]}, {"nan_count":0,"unique_count":10}},
{"dtype":"int64","name":"Solo score","stats":{"histogram":
[{"bin_end":444292.5,"bin_start":15872,"count":5},
{"bin_end":872713,"bin_start":444292.5,"count":1},
{"bin_end":1301133.5,"bin_start":872713,"count":1},
{"bin_end":1729554,"bin_start":1301133.5,"count":1},
{"bin_end":2157974.5,"bin_start":1729554,"count":0},
{"bin_end":2586395,"bin_start":2157974.5,"count":1},
{"bin_end":3014815.5,"bin_start":2586395,"count":0},
{"bin_end":3443236,"bin_start":3014815.5,"count":0},
{"bin_end":3871656.5,"bin_start":3443236,"count":0},
{"bin_end":4300077,"bin_start":3871656.5,"count":1}], "max":"4300077", "
min":"15872", "nan_count":0,"unique_count":10}},
{"dtype":"int64","name":"Solo top1","stats":{"histogram":
[{"bin_end":63,"bin_start":6,"count":6},
{"bin_end":120,"bin_start":63,"count":2},
{"bin_end":177,"bin_start":120,"count":0},
{"bin_end":234,"bin_start":177,"count":0},
{"bin_end":291,"bin_start":234,"count":0},
{"bin_end":348,"bin_start":291,"count":1},
{"bin_end":405,"bin_start":348,"count":0},
{"bin_end":462,"bin_start":405,"count":0},
{"bin_end":519,"bin_start":462,"count":0},
{"bin_end":576,"bin_start":519,"count":1}], "max":"576", "min":"6", "nan_
count":0,"unique_count":9}}, {"dtype":"float64","name":"Solo
kd","stats":{"histogram":
[{"bin_end":0.5499999999999999,"bin_start":0.47,"count":1},
{"bin_end":0.63,"bin_start":0.5499999999999999,"count":1},
{"bin_end":0.71,"bin_start":0.63,"count":2},
{"bin_end":0.79,"bin_start":0.71,"count":0},
{"bin_end":0.87,"bin_start":0.79,"count":1},
{"bin_end":0.95,"bin_start":0.87,"count":0},
{"bin_end":1.03,"bin_start":0.95,"count":1},
{"bin_end":1.1099999999999999,"bin_start":1.03,"count":3},
{"bin_end":1.19,"bin_start":1.1099999999999999,"count":0},
{"bin_end":1.27,"bin_start":1.19,"count":1}], "max":"1.27", "min":"0.47"
,"nan_count":0,"unique_count":9}}, {"dtype":"float64","name":"Solo
winRatio","stats":{"histogram":

```



```
[{"bin_end":0.73,"bin_start":0.5,"count":3},
{"bin_end":0.96,"bin_start":0.73,"count":0},
{"bin_end":1.19,"bin_start":0.96,"count":2},
{"bin_end":1.42,"bin_start":1.19,"count":1},
{"bin_end":1.65,"bin_start":1.42,"count":0},
{"bin_end":1.88,"bin_start":1.65,"count":1},
{"bin_end":2.11,"bin_start":1.88,"count":0},
{"bin_end":2.34,"bin_start":2.11,"count":1},
{"bin_end":2.57,"bin_start":2.34,"count":1},
{"bin_end":2.8,"bin_start":2.57,"count":1}],{"max":"2.8","min":"0.5","nan_count":0,"unique_count":10}},{"dtype":"int64","name":"Solo matches","stats":{"histogram":
[{"bin_end":3675.9,"bin_start":429,"count":6},
{"bin_end":6922.8,"bin_start":3675.9,"count":0},
{"bin_end":10169.7,"bin_start":6922.8,"count":1},
{"bin_end":13416.6,"bin_start":10169.7,"count":2},
{"bin_end":16663.5,"bin_start":13416.6,"count":0},
{"bin_end":19910.4,"bin_start":16663.5,"count":0},
{"bin_end":23157.3,"bin_start":19910.4,"count":0},
{"bin_end":26404.2,"bin_start":23157.3,"count":0},
{"bin_end":29651.100000000002,"bin_start":26404.2,"count":0},
{"bin_end":32898,"bin_start":29651.100000000002,"count":1}],{"max":"32898","min":"429","nan_count":0,"unique_count":10}},
{"dtype":"int64","name":"Solo kills","stats":{"histogram":
[{"bin_end":4274.2,"bin_start":200,"count":6},
{"bin_end":8348.4,"bin_start":4274.2,"count":1},
{"bin_end":12422.599999999999,"bin_start":8348.4,"count":2},
{"bin_end":16496.8,"bin_start":12422.599999999999,"count":0},
{"bin_end":20571,"bin_start":16496.8,"count":0},
{"bin_end":24645.199999999997,"bin_start":20571,"count":0},
{"bin_end":28719.399999999998,"bin_start":24645.199999999997,"count":0},
{"bin_end":32793.6,"bin_start":28719.399999999998,"count":0},
{"bin_end":36867.799999999996,"bin_start":32793.6,"count":0},
{"bin_end":40942,"bin_start":36867.799999999996,"count":1}],{"max":"40942","min":"200","nan_count":0,"unique_count":10}},
{"dtype":"int64","name":"Solo minutesPlayed","stats":{"histogram":
[{"bin_end":18229.4,"bin_start":739,"count":5},
{"bin_end":35719.8,"bin_start":18229.4,"count":1},
{"bin_end":53210.200000000004,"bin_start":35719.8,"count":0},
{"bin_end":70700.6,"bin_start":53210.200000000004,"count":2},
{"bin_end":88191,"bin_start":70700.6,"count":0},
{"bin_end":105681.400000000001,"bin_start":88191,"count":1},
{"bin_end":123171.800000000002,"bin_start":105681.400000000001,"count":0},
{"bin_end":140662.2,"bin_start":123171.800000000002,"count":0},
{"bin_end":158152.6,"bin_start":140662.2,"count":0},
{"bin_end":175643,"bin_start":158152.6,"count":1}],{"max":"175643","min":"739","nan_count":0,"unique_count":10}},
{"dtype":"int64","name":"_deepnote_index_column"}],{"row_count":10,"rows":[{"Player":"NiteGamerYT 190k","Solo kd":0.47,"Solo kills":200,"Solo
```

```

matches":429,"Solo minutesPlayed":739,"Solo score":15872,"Solo
top1":6,"Solo winRatio":1.4,"_deepnote_index_column":7},
{"Player":"CIUPEA 144.HZ","Solo kd":0.61,"Solo kills":1174,"Solo
matches":1938,"Solo minutesPlayed":2441,"Solo score":54479,"Solo
top1":9,"Solo winRatio":0.5,"_deepnote_index_column":9},
{"Player":"UçaN_HamSi","Solo kd":0.68,"Solo kills":1471,"Solo
matches":2182,"Solo minutesPlayed":3199,"Solo score":67758,"Solo
top1":25,"Solo winRatio":1.1,"_deepnote_index_column":28},
{"Player":"Trips","Solo kd":0.7,"Solo kills":1785,"Solo
matches":2581,"Solo minutesPlayed":21679,"Solo score":478280,"Solo
top1":25,"Solo winRatio":1,"_deepnote_index_column":13},
{"Player":"KING nax nax","Solo kd":0.8,"Solo kills":8356,"Solo
matches":10460,"Solo minutesPlayed":63105,"Solo score":1407541,"Solo
top1":70,"Solo winRatio":0.7,"_deepnote_index_column":24},
{"Player":"Deathfury420","Solo kd":0.95,"Solo kills":8101,"Solo
matches":8568,"Solo minutesPlayed":55671,"Solo score":1257973,"Solo
top1":51,"Solo winRatio":0.6,"_deepnote_index_column":23},
{"Player":"Scribell","Solo kd":1.03,"Solo kills":1505,"Solo
matches":1491,"Solo minutesPlayed":8945,"Solo score":212388,"Solo
top1":34,"Solo winRatio":2.3,"_deepnote_index_column":18},
{"Player":"Qx_9u انسٲا","Solo kd":1.03,"Solo kills":3281,"Solo
matches":3269,"Solo minutesPlayed":8903,"Solo score":228620,"Solo
top1":79,"Solo winRatio":2.4,"_deepnote_index_column":19},
{"Player":"Kid_Fighter","Solo kd":1.06,"Solo kills":12348,"Solo
matches":11967,"Solo minutesPlayed":103551,"Solo score":2428145,"Solo
top1":331,"Solo winRatio":2.8,"_deepnote_index_column":29},
{"Player":"pr0Mancos","Solo kd":1.27,"Solo kills":40942,"Solo
matches":32898,"Solo minutesPlayed":175643,"Solo score":4300077,"Solo
top1":576,"Solo winRatio":1.8,"_deepnote_index_column":17}}]

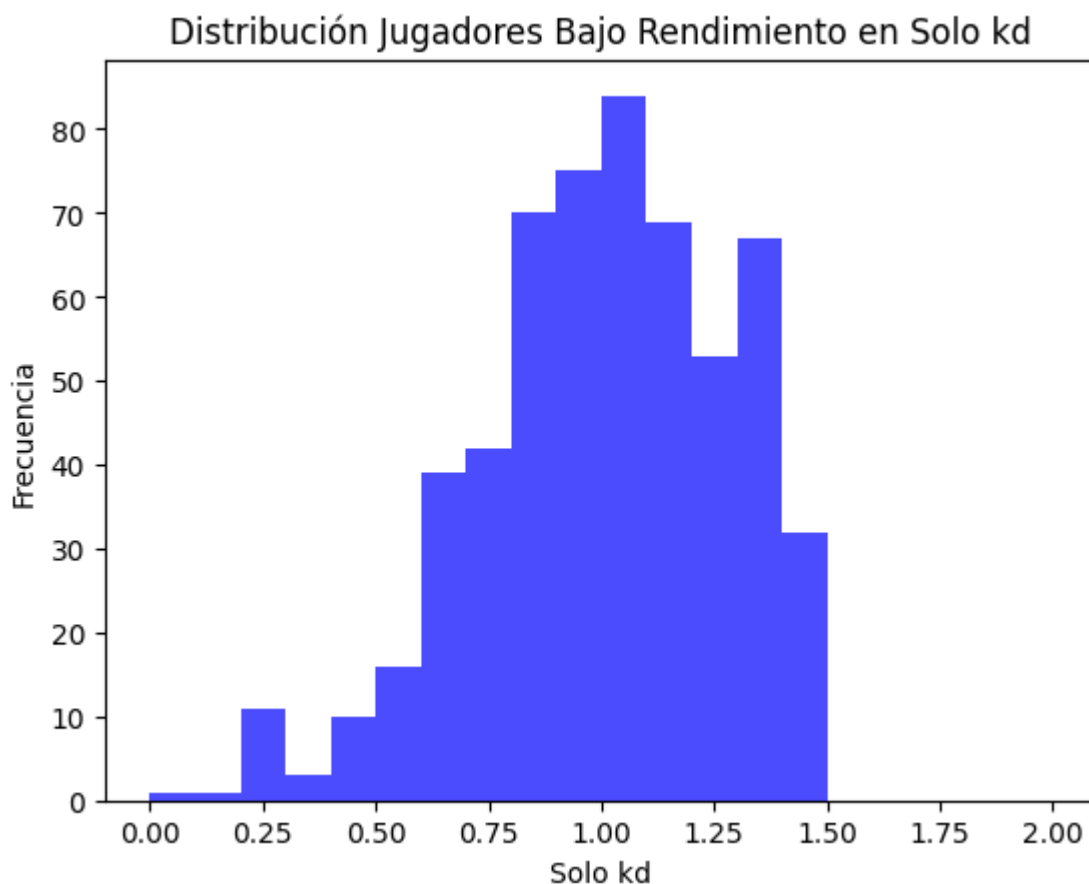
```

```
print(jugadorBajoRendimiento.describe())
```

	Solo score	Solo top1	Solo kd	Solo winRatio	Solo
matches \					
count	5.730000e+02	573.000000	573.000000	573.000000	
mean	9.795136e+05	99.668412	1.001850	1.451134	
std	9.044167e+05	92.986643	0.277303	0.693415	
min	4.453000e+03	0.000000	0.080000	0.000000	
25%	2.862850e+05	28.000000	0.830000	0.900000	
50%	7.330330e+05	74.000000	1.020000	1.400000	
75%	1.359115e+06	143.000000	1.230000	2.000000	
max	4.595164e+06	591.000000	1.470000	2.800000	

	Solo kills	Solo minutesPlayed
count	573.000000	573.000000
mean	7384.303665	42627.394415
std	6923.464940	40263.510377
min	28.000000	210.000000
25%	2147.000000	12418.000000
50%	5460.000000	31656.000000
75%	10606.000000	59160.000000
max	45588.000000	220536.000000

```
# Histograma de Solo kd para jugadores de bajo rendimiento
plt.hist(jugadorBajoRendimiento['Solo kd'], bins=20, color='blue',
alpha=0.7, range=(0,2))
plt.title('Distribución Jugadores Bajo Rendimiento en Solo kd')
plt.xlabel('Solo kd')
plt.ylabel('Frecuencia')
plt.show()
```



La mayor concentración de jugadores de menor rendimiento está alrededor de los valores de 1.0 y 1.25 en Solo kd. Esto significa que la mayoría de los jugadores considerados de bajo rendimiento tienen una proporción de 1 o un poco más de 1 muerte por cada vez que mueren.

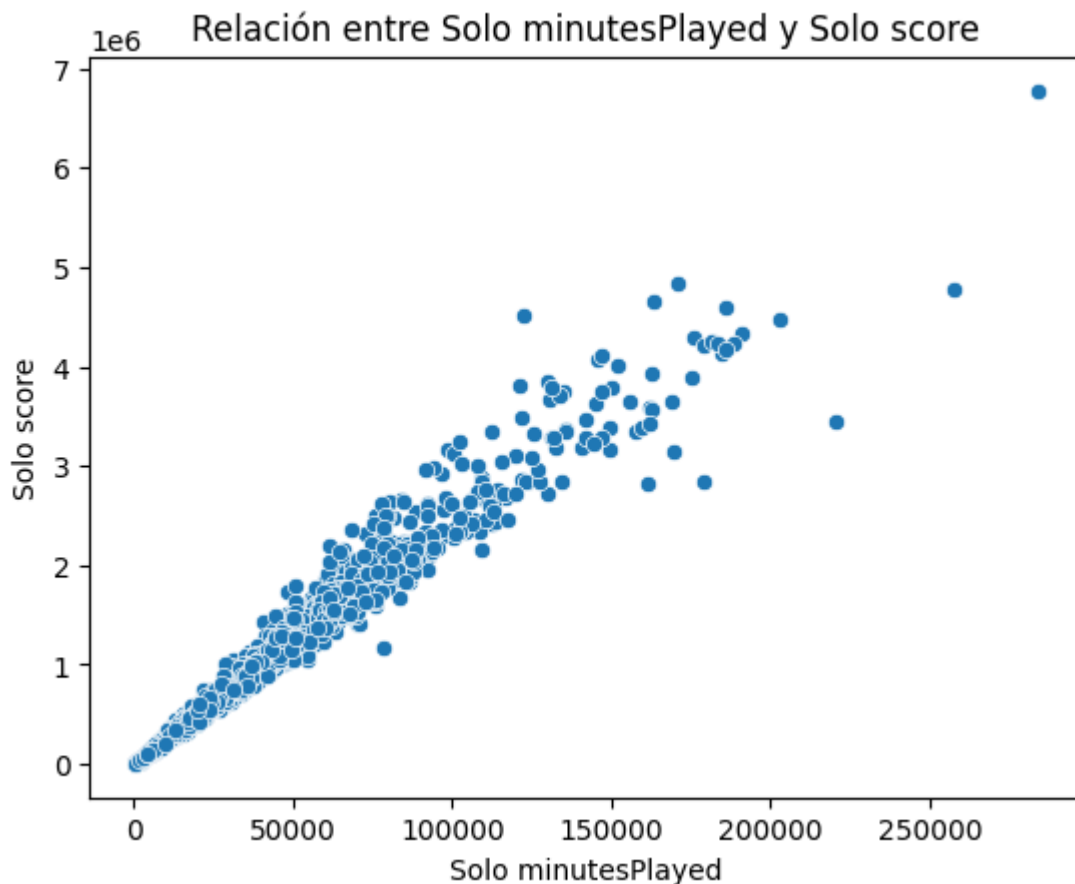
Casi no hay jugadores con un Solo kd superior a 1.5, lo que refuerza la idea de que estos jugadores están en la categoría de bajo rendimiento y con altas posibilidades de abandono de juego.

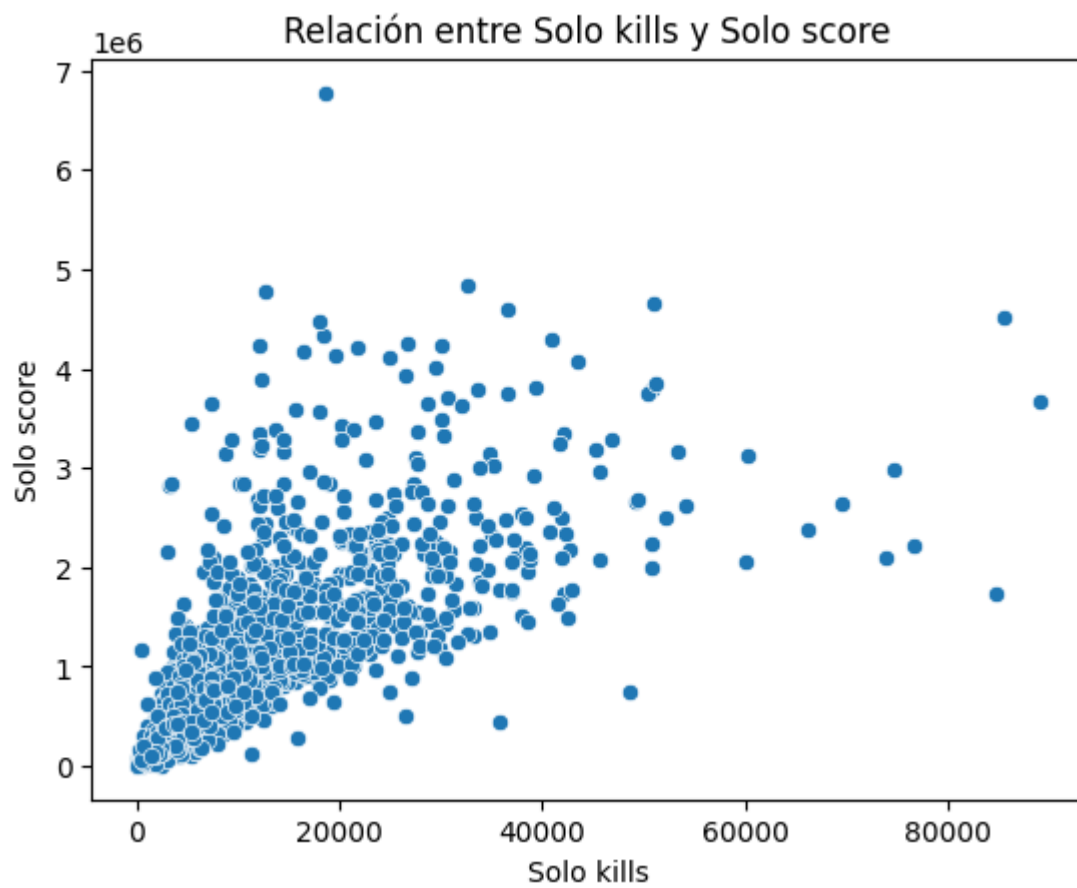
¿Qué factores influyen en la puntuación (Solo score) de un jugador?

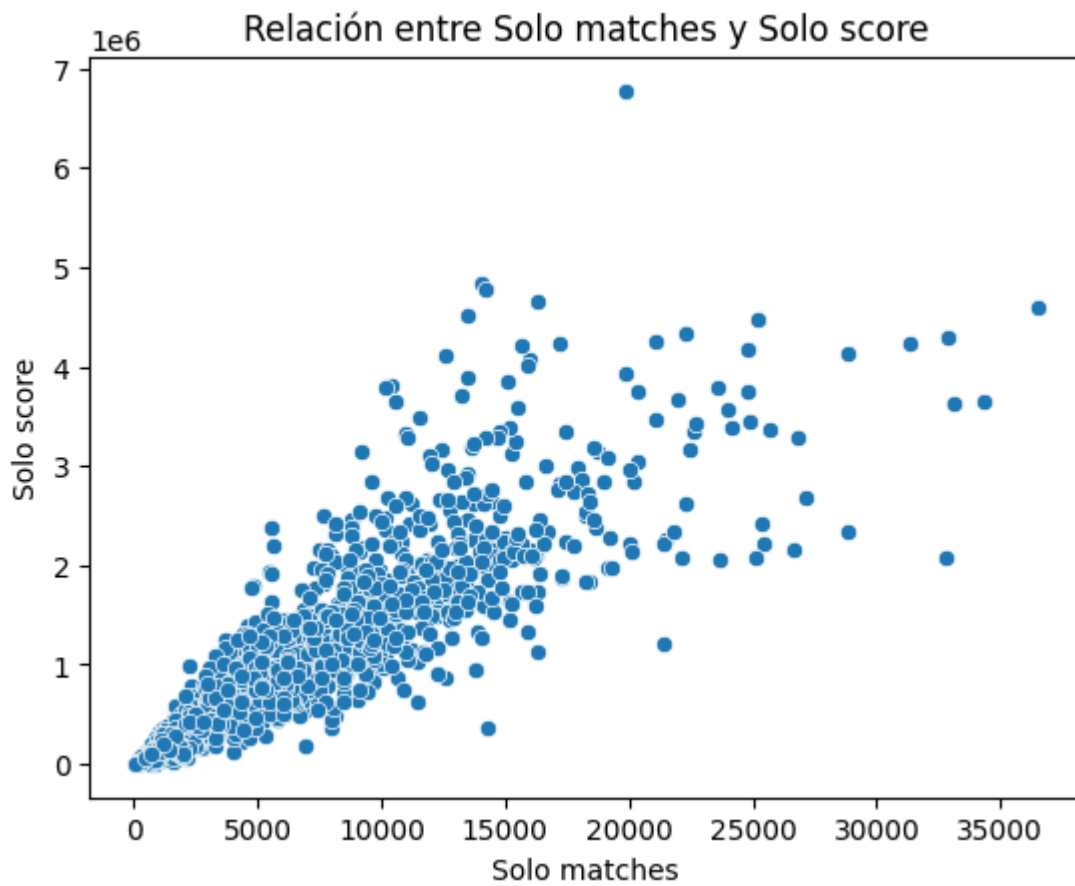
```
variables = ['Solo minutesPlayed', 'Solo kills', 'Solo matches', 'Solo winRatio']
```

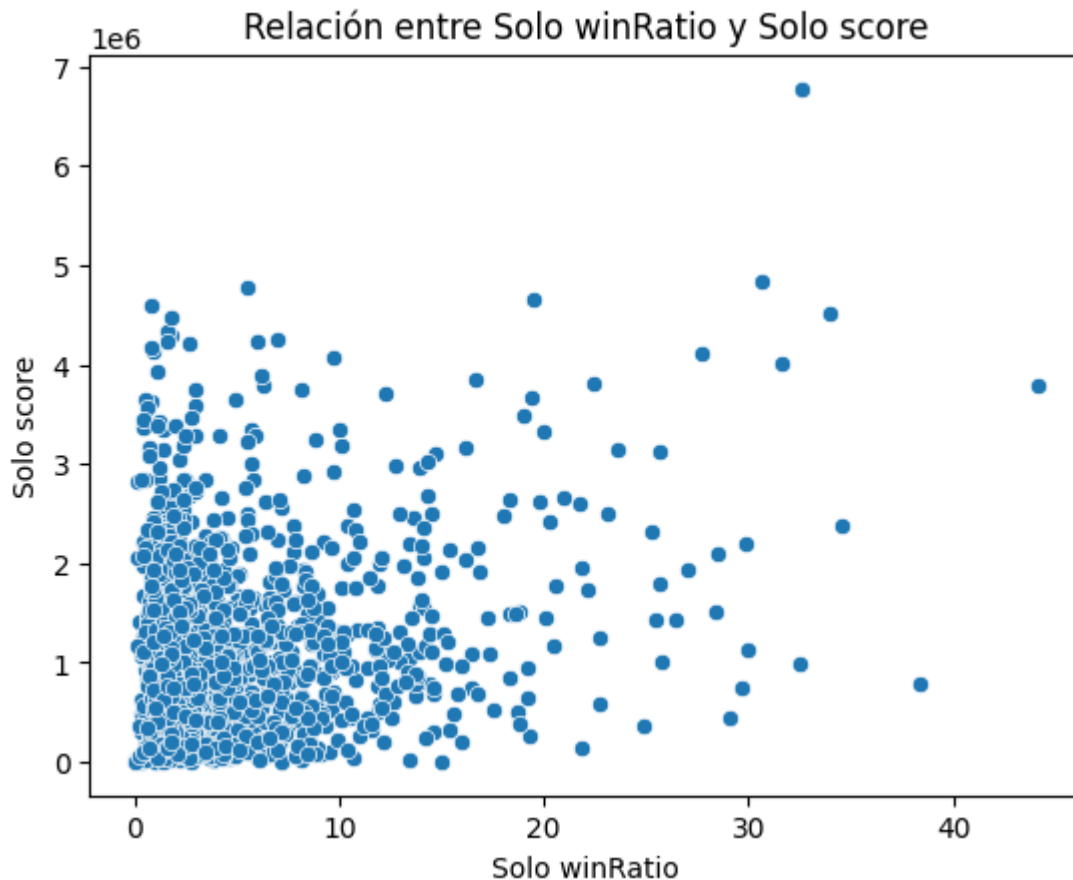
```
# Gráficos de dispersión
```

```
for var in variables:  
    plt.figure()  
    sb.scatterplot(x=df[var], y=df['Solo score'])  
    plt.title(f'Relación entre {var} y Solo score')  
    plt.xlabel(var)  
    plt.ylabel('Solo score')  
    plt.show()
```









Existe una fuerte correlación entre el tiempo jugado (Solo minutesPlayed), el número de partidas (Solo matches), y el puntaje (Solo score). Esto indica que el puntaje es influenciado significativamente por la cantidad de tiempo y partidas jugadas. Aunque Solo kills también muestra una correlación positiva con el Solo Score, la dispersión muestra que la presencia de otros factores influyentes. Solo winRatio muestra la menor correlación, lo que indica que ganar una mayor proporción de partidas no siempre asegura un puntaje más alto.

Existe una relación significativa entre el tiempo jugado (Solo minutesPlayed) y el ratio de victorias (Solo winRatio)?

```
# Crear gráficos de dispersión
plt.figure(figsize=(14, 6))

# Scatter plot entre Solo minutesPlayed y Solo kills
plt.subplot(1, 2, 1)
sb.scatterplot(x=df['Solo minutesPlayed'], y=df['Solo kills'])
plt.title('Relación entre Solo minutesPlayed y Solo kills')
plt.xlabel('Solo minutesPlayed')
plt.ylabel('Solo kills')

# Scatter plot entre Solo minutesPlayed y Solo winRatio
plt.subplot(1, 2, 2)
```

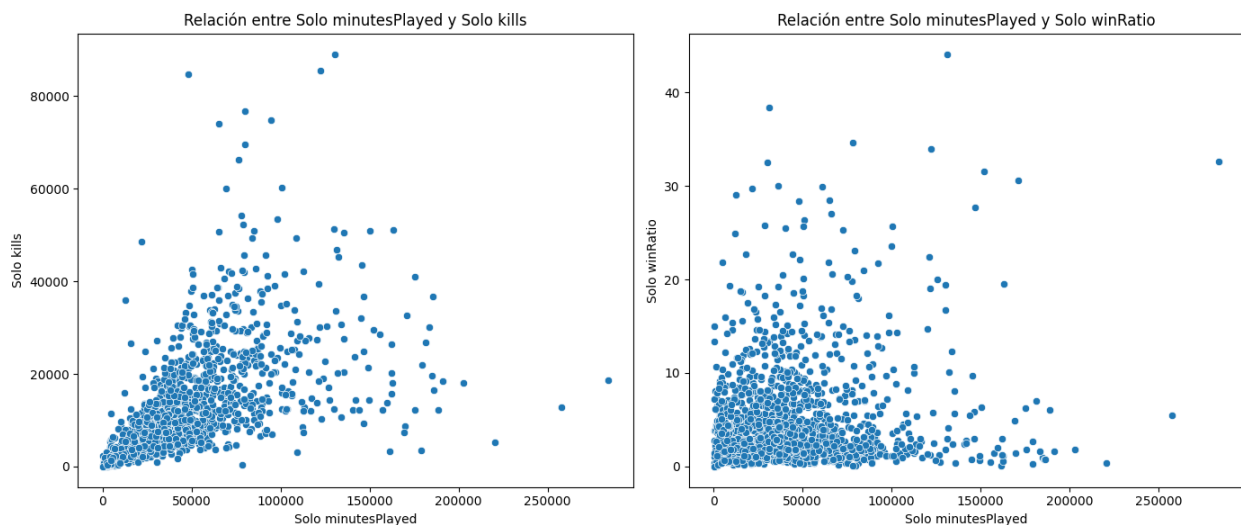


```

sb.scatterplot(x=df['Solo minutesPlayed'], y=df['Solo winRatio'])
plt.title('Relación entre Solo minutesPlayed y Solo winRatio')
plt.xlabel('Solo minutesPlayed')
plt.ylabel('Solo winRatio')

plt.tight_layout()
plt.show()

```



En el primer gráfico se observa que hay una tendencia positiva entre SoloMinutesPlayed y Solo Kills, según la matriz de correlación es de un 0.60, lo que indica que a medida que el tiempo de juego aumenta, también aumenta el número de eliminaciones.

Podemos decir que los jugadores que juegan más tiempo tienden a conseguir más eliminaciones. Pero, la separación de puntos nos demuestra que son muy dispersas, lo que nos señala que no todos los jugadores con mucho tiempo de juego tienen un alto número de eliminaciones.

En el gráfico de Relación entre SoloMinutesPlayed y SoloWinRatio no hay una relación clara entre estas dos variables, ya que los puntos están distribuidos de manera más dispersa y no existe una tendencia clara.

Fase 3 Data Preparation

```

# Histograma para visualizar la distribución
df[['Solo score', 'Solo kd', 'Solo winRatio']].hist(bins=20,
figsize=(10, 8))
plt.show()

```