



Actividad 2.2

Regresión Lineal Simple y Múltiple

Carlos Fonseca Olivares- A01734538

Emily Bueno Romero - A01736939

Juan José Zamorano Balderas - A01642396

Juan Sebastián Mejía De Gortari - A01722536



Limpieza de Nulos y Outliers

```
Data columns (total 7 columns):
```

#	Column	Non-Null Count	Dtype
0	Administrador	5865 non-null	object
1	Usuario	5865 non-null	object
2	mini juego	5709 non-null	object
3	color presionado	5103 non-null	object
4	dificultad	5865 non-null	object
5	fecha	5865 non-null	object
6	Juego	5865 non-null	object

dtypes: object(7)

```
Data columns (total 6 columns):
```

#	Column	Non-Null Count	Dtype
0	botón correcto	5103 non-null	float64
1	tiempo de interacción	5103 non-null	float64
2	número de interacción	5103 non-null	float64
3	auto push	5103 non-null	float64
4	tiempo de lección	5688 non-null	float64
5	tiempo de sesión	5259 non-null	float64

dtypes: float64(6)



Datos Nulos

```
1 cualis_bfill=data_cuali.fillna(method="bfill")
2 cualis_ffill=data_cuali.fillna(method="ffill")
3 cuantias_mean=data_cuanti.fillna(round(data_cuanti.mean(),1))
4
5 data_sin_nulos = pd.concat([cuantias_mean,cualis_ffill],axis=1)
6
7 data_sin_nulos
```

	botón correcto	tiempo de interacción	número de interacción	auto push	tiempo de lección	tiempo de sesión	Administrador	Usuario	mini juego	color presionado	dificultad	fecha	Juego
0	1.0	5.399169	1.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	blue	Episodio 1	25/01/2024 09:26:42 a. m.	Astro
1	0.0	1.283400	2.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	violet	Episodio 1	25/01/2024 09:26:46 a. m.	Astro
2	1.0	2.700226	3.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	green	Episodio 1	25/01/2024 09:26:48 a. m.	Astro
3	0.0	3.050262	4.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	green	Episodio 1	25/01/2024 09:26:57 a. m.	Astro
4	0.0	4.750256	5.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	green	Episodio 1	25/01/2024 09:26:58 a. m.	Astro
...
5860	0.5	10.000000	13.3	0.0	6.300000	332.240000	ALEIDA	ESMERALDA	Restaurante	green	Episodio 1	28/05/2024 04:15:49 p. m.	Astro
5861	0.0	2.135419	1.0	0.0	0.000000	0.000000	ALEIDA	JOSE JAVIER	Asteroides	green	Episodio 3	04/06/2024 11:09:54 a. m.	Astro
5862	0.5	10.000000	13.3	0.0	2.271806	12.400000	ALEIDA	JOSE JAVIER	Asteroides	green	Episodio 3	04/06/2024 11:09:58 a. m.	Astro
5863	0.5	10.000000	13.3	0.0	6.300000	6.478299	ALEIDA	JOSE JAVIER	Asteroides	green	Episodio 3	04/06/2024 11:09:58 a. m.	Astro
5864	0.5	10.000000	13.3	0.0	6.300000	12.400000	ALEIDA	JOSE JAVIER	MiniGame_0	green	Episodio 3	04/06/2024 11:09:59 a. m.	Astro

```
1 data=data_sin_nulos.isnull().sum().sum()
2 print(data)
```

✓ 0.0s

0

Outliers

```
1 cuantitativas=data_sucio.select_dtypes(include=["float64","int64","float","int"])
2 cualitativas=data_sucio.select_dtypes(include=["object","datetime","category"])
3 y=cuantitativas
4
5 percentile25=y.quantile(0.25)
6 percentile75=y.quantile(0.75)
7 iqr= percentile75-percentile25
8
9 Limite_Superior_iqr= percentile75+1.5*iqr
10 Limite_Inferior_iqr= percentile25-1.5*iqr
11 iqr=cuantitativas[(y<=Limite_Superior_iqr)&y>=(Limite_Inferior_iqr)]
12 iqr2=iqr.fillna(round(iqr.mean(),1))
13 rango_intercuartilico=pd.concat([cualitativas,iqr2],axis=1)
14 rango_intercuartilico
```

```
1 cualis_bfill=data_cuali.fillna(method="bfill")
2 cualis_ffill=data_cuali.fillna(method="ffill")
3 cuantias_mean=data_cuanti.fillna(round(data_cuanti.mean(),1))
4
5 data = pd.concat([cuantias_mean,cualis_ffill],axis=1)
6
7 data
```

	botón correcto	tiempo de interacción	número de interacción	auto push	tiempo de lección	tiempo de sesión	Administrador	Usuario	mini juego	color presionado	dificultad	fecha	Juego
0	1.0	5.399169	1.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	blue	Episodio 1	25/01/2024 09:26:42 a. m.	Astro
1	0.0	1.283400	2.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	violet	Episodio 1	25/01/2024 09:26:46 a. m.	Astro
2	1.0	2.700226	3.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	green	Episodio 1	25/01/2024 09:26:48 a. m.	Astro
3	0.0	3.050262	4.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	green	Episodio 1	25/01/2024 09:26:57 a. m.	Astro
4	0.0	4.750256	5.0	0.0	0.000000	0.000000	nicolas	nicolas	Despegue	green	Episodio 1	25/01/2024 09:26:58 a. m.	Astro
...
5860	0.5	10.000000	13.3	0.0	6.300000	332.240000	ALEIDA	ESMERALDA	Restaurante	green	Episodio 1	28/05/2024 04:15:49 p. m.	Astro
5861	0.0	2.135419	1.0	0.0	0.000000	0.000000	ALEIDA	JOSE JAVIER	Asteroides	green	Episodio 3	04/06/2024 11:09:54 a. m.	Astro
5862	0.5	10.000000	13.3	0.0	2.271806	12.400000	ALEIDA	JOSE JAVIER	Asteroides	green	Episodio 3	04/06/2024 11:09:58 a. m.	Astro
5863	0.5	10.000000	13.3	0.0	6.300000	6.478299	ALEIDA	JOSE JAVIER	Asteroides	green	Episodio 3	04/06/2024 11:09:58 a. m.	Astro
5864	0.5	10.000000	13.3	0.0	6.300000	12.400000	ALEIDA	JOSE JAVIER	MiniGame_0	green	Episodio 3	04/06/2024 11:09:59 a. m.	Astro

```

1 outliers_limpios=data.isnull().sum().sum()
2 print(outliers_limpios)

```

✓ 0.0s

0



Convertir las variables categóricas ordinales

- ✓ Administrador
- ✓ Usuario
- ✓ mini juego
- ✓ color presionado
- ✓ dificultad
- ✓ Juego

```
#Administrador
frecuencia_ad =
data["Administrador"].value_counts().reset_index()
frecuencia_ad.columns = ["Administrador", "Frecuencia"]
frecuencia_ad
```

	Administrador	Frecuencia
0	ALEIDA	3260
1	nicolas	440
2	LEONARDO	371
3	DENISSE	302
4	SERGIO ANGEL	243
5	CARLOS ENRIQUE	228
6	Yael DAVID	224
7	AUSTIN	199
8	VALENTIN	163
9	erick	158
10	IKER BENJAMIN	128
11	KYTZIA	98
12	BENJAMIN	51

```
#Usuario
#(ADRIAN, ALEIDA, ARLETT, ASHLEY, AUSTIN)
frecuencia_us = data["Usuario"].value_counts().reset_index()
frecuencia_us.columns = ["Usuario", "Frecuencia"]
frecuencia_us
```

	Usuario	Frecuencia
0	LEONARDO	546
1	ALEIDA	487
2	nicolas	440
3	JOSE JAVIER	394
4	LEONARDO	371
5	JESUS ALEJANDRO	356
6	ramiro isai	332
7	ADRIAN	280
8	SERGIO ANGEL	243
9	DENISSE	228
10	CARLOS ENRIQUE	228
11	Yael DAVID	224
12	VALENTIN	163
13	erick	158
14	IKER BENJAMIN	152
15	NICOLAS	128
16	ERICK OSVALDO	126
17	CONCEPCION	103
18	KYTZIA	98
19	AUSTIN	96
20	JOSE IGNACIO TADEO	94
21	JOSE IAN	86
22	ASHLEY	75
23	JOSHUA	74
24	YEREMI YAZMIN	71
25	MA DEL ROSARIO	64
26	BENJAMIN	51
27	INGRID	49
28	RENE	44
29	CARLOS ABEL	39
30	ARLETT	29
31	ESMERALDA	24
32	IRVING	10
33	jesus eduardo	2

```
#Color presionado
frecuencia_co = data["color presionado"].value_counts().reset_index()
frecuencia_co.columns = ["color presionado", "Frecuencia"]
frecuencia_co
```

	color presionado	Frecuencia
0	violet	1609
1	green	1455
2	yellow	1425
3	blue	1342
4	red	34

```
#Mini juego
frecuencia_mj = data["mini juego"].value_counts().reset_index()
frecuencia_mj.columns = ["mini juego", "Frecuencia"]
frecuencia_mj
```

	mini juego	Frecuencia
0	Asteroides	1088
1	Restaurante	888
2	Estrellas	837
3	Gusanos	768
4	sonidos y animales	765
5	animales y colores	563
6	figuras y colores	409
7	partes del cuerpo	329
8	Despegue	197
9	MiniGame_0	8
10	MiniGame_3	5
11	MiniGame_2	5
12	MiniGame_1	3

```
#Dificultad
frecuencia_di = data["dificultad"].value_counts().reset_index()
frecuencia_di.columns = ["dificultad", "Frecuencia"]
frecuencia_di
```

	dificultad	Frecuencia
0	Episodio 1	2357
1	Episodio 2	1493
2	Episodio 3	1292
3	Episodio 4	723

```
#Juego
frecuencia_ju = data["Juego"].value_counts().reset_index()
frecuencia_ju.columns = ["Juego", "Frecuencia"]
frecuencia_ju
```

	Juego	Frecuencia
0	Astro	3799
1	Cadetes	2066

Analizar las correlaciones lineales simples ("Heatmap")

Entre "Usuario" y las demás

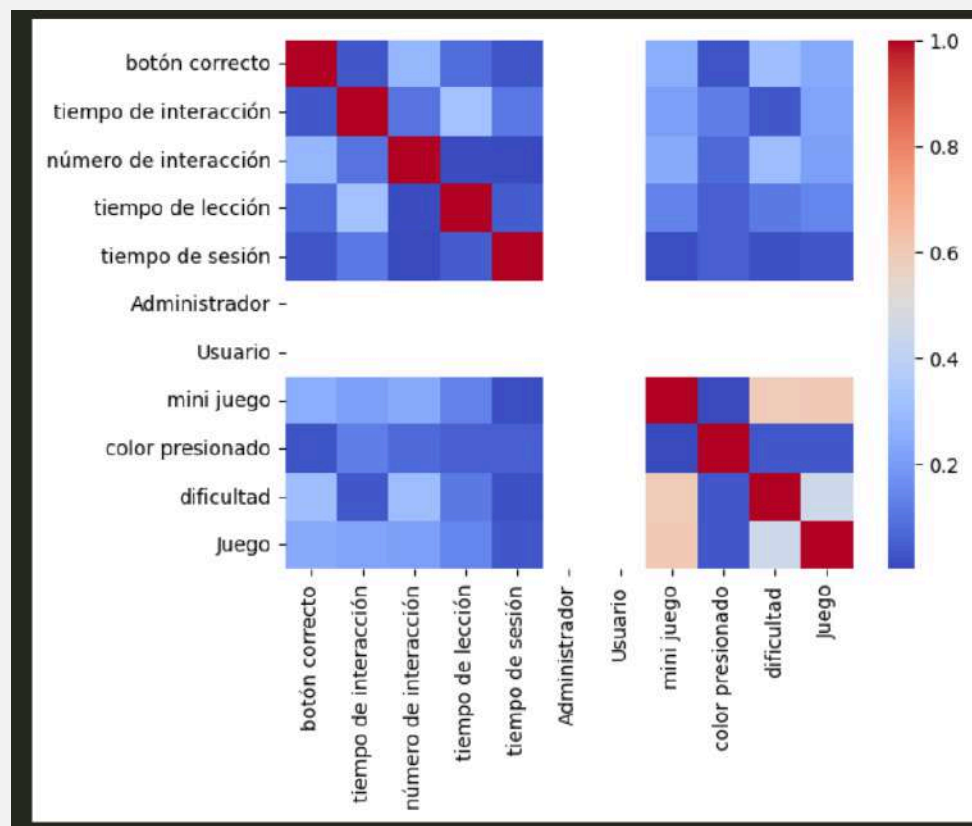
```
1 data=data.drop(["fecha"],axis=1)
2 data=data.drop(["auto push"],axis=1)
✓ 0.0s

1 aleida = 1
2 adrian = 2
3 austin = 3
4 ashley = 4
5 arlett = 5
6 ALEIDA = data[data["Usuario"] == aleida]
7 ADRIAN = data[data["Usuario"] == adrian]
8 AUSTIN = data[data["Usuario"] == austin]
9 ASHLEY = data[data["Usuario"] == ashley]
10 ARLETT = data[data["Usuario"] == arlett]
11
✓ 0.0s
```





```
Corr_Factors=ALEIDA.corr()  
Corr_Factors1=abs(Corr_Factors)  
Heat_Map_ALEIDA=sns.heatmap(Corr_Factors1,cmap="coolwarm")  
Heat_Map_ALEIDA
```



Las mayores correlaciones fueron:
mini juego - Juego
mini juego - dificultad

Las mayores correlaciones fueron:
mini juego - Juego.

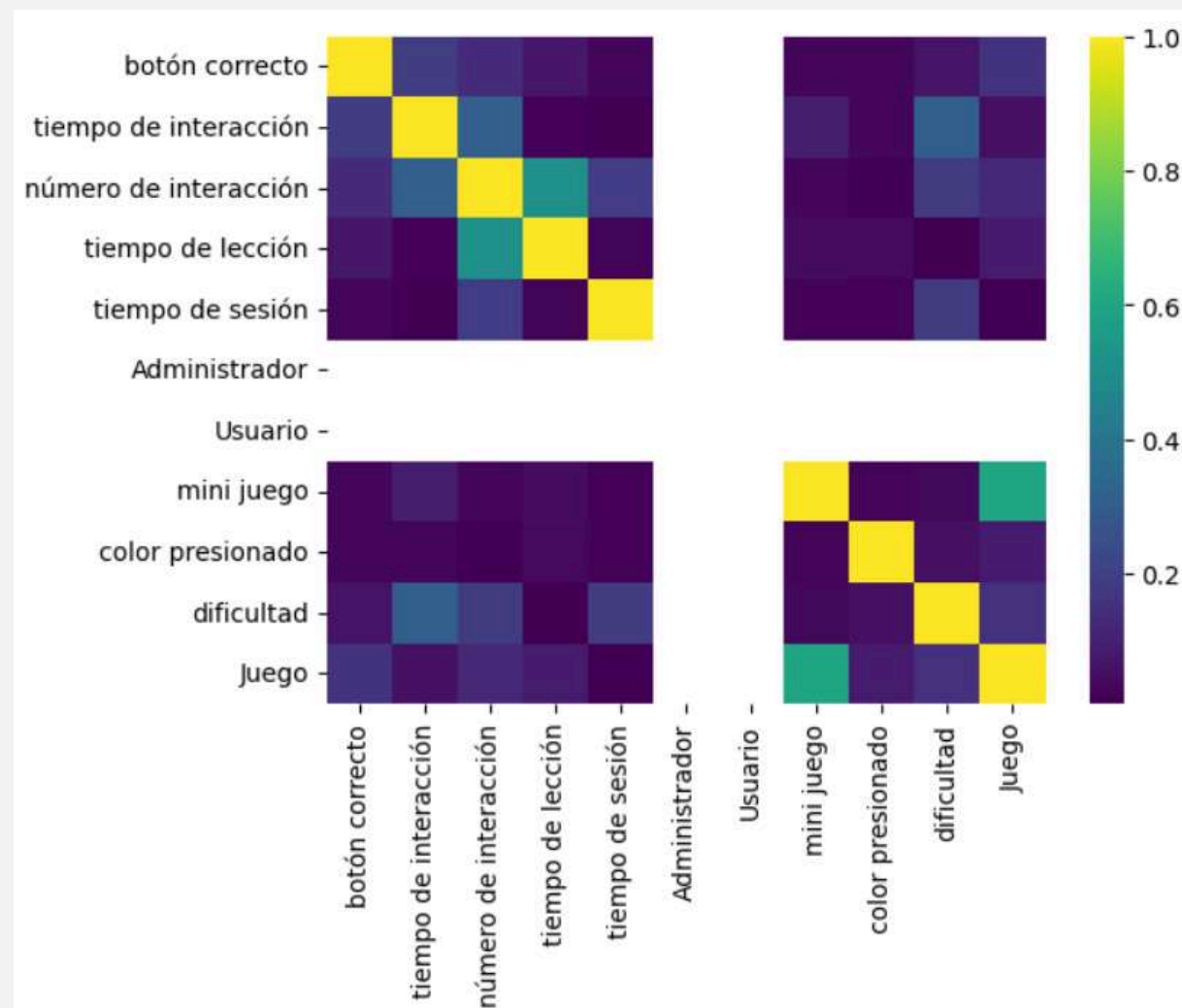
Seguido de cerca por: tiempo de interacción - dificultad | tiempo de interacción - número de interacción.

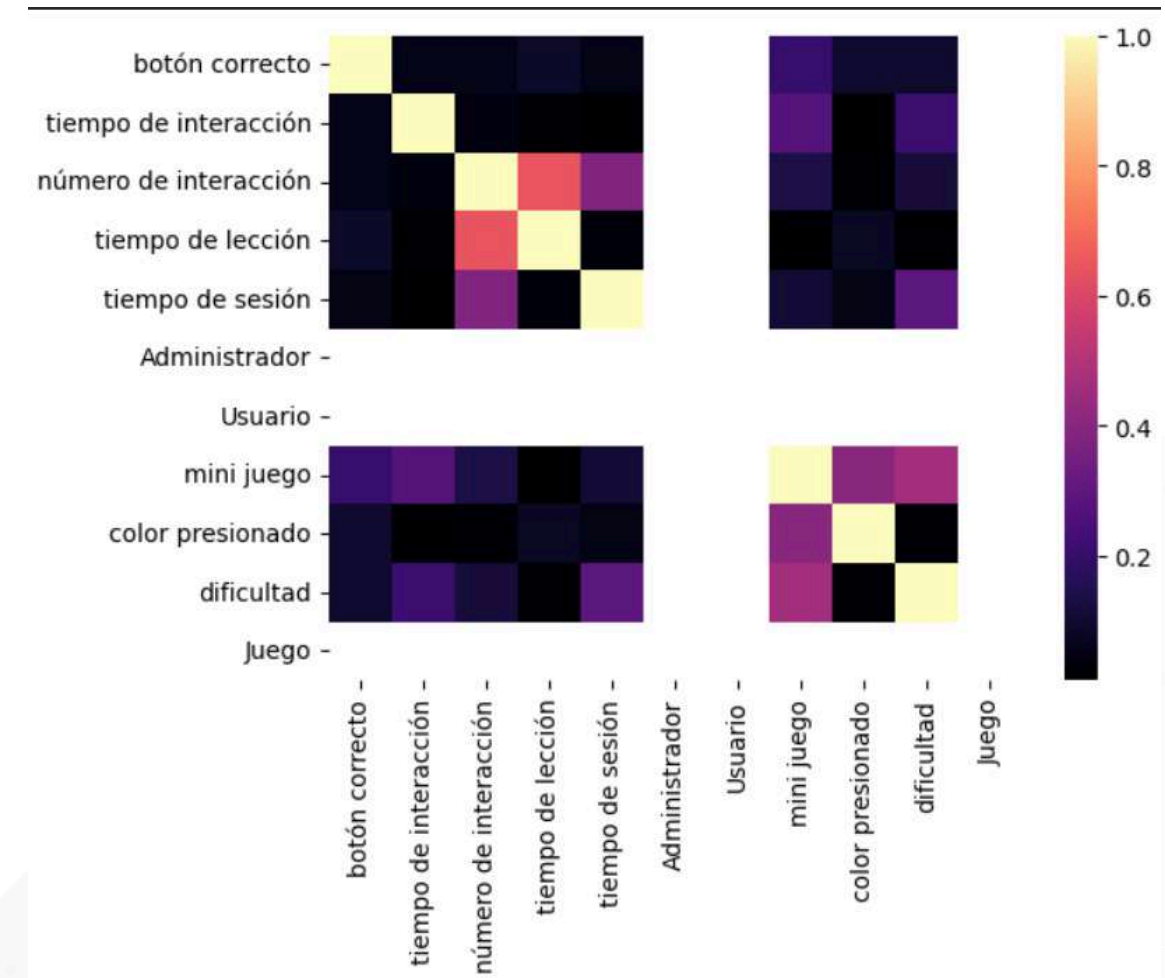
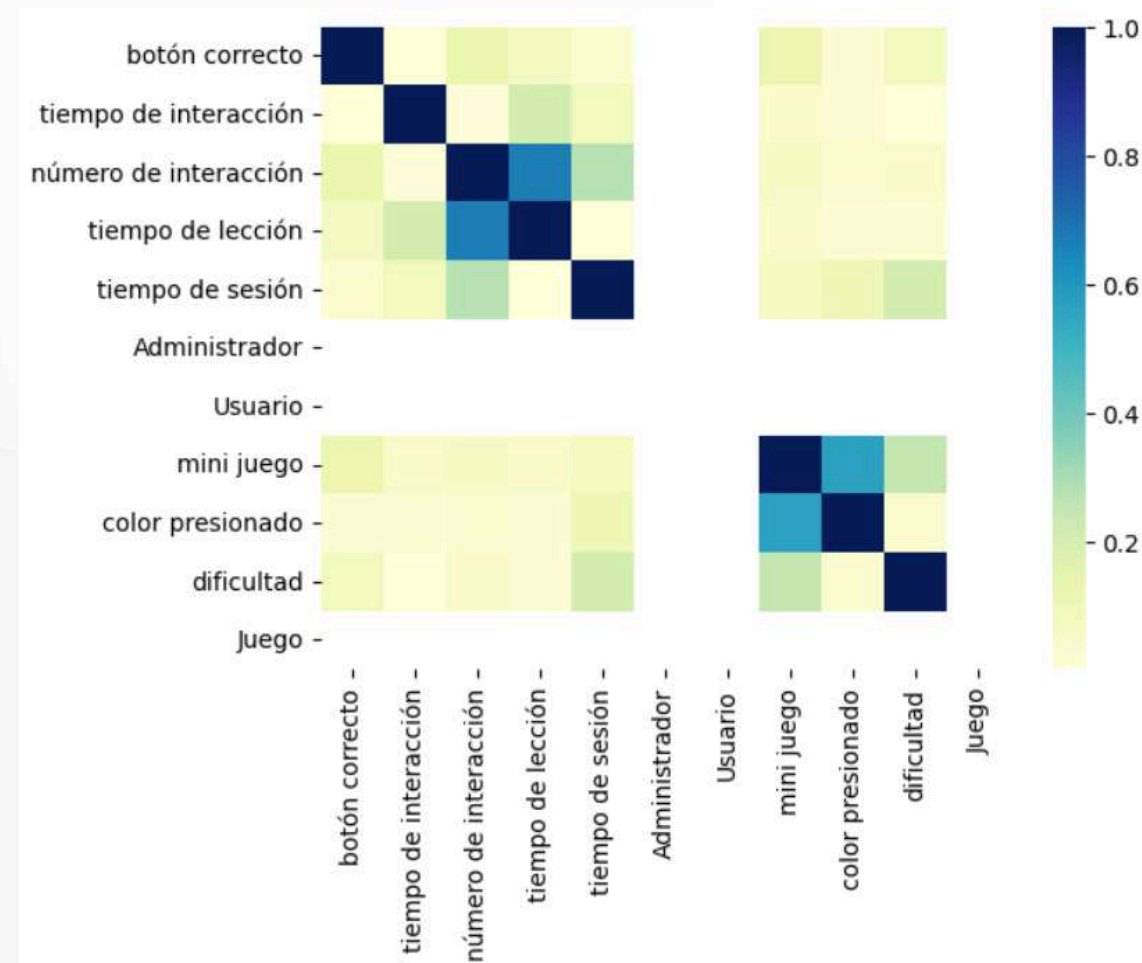
```
Corr_Factors=ADRIAN.corr()
```

```
Corr_Factors1=abs(Corr_Factors)
```

```
Heat_Map_ADRIAN=sns.heatmap(Corr_Factors1,cmap="viridis")
```

```
Heat_Map_ADRIAN
```





```
Corr_Factors=AUSTIN.corr()
Corr_Factors1=abs(Corr_Factors)
Heat_Map_AUSTIN=sns.heatmap(Corr_Fa
ctors1,cmap="YlGnBu")
Heat_Map_AUSTIN
```

La mayor correlación fue: mini juego - color presionado.

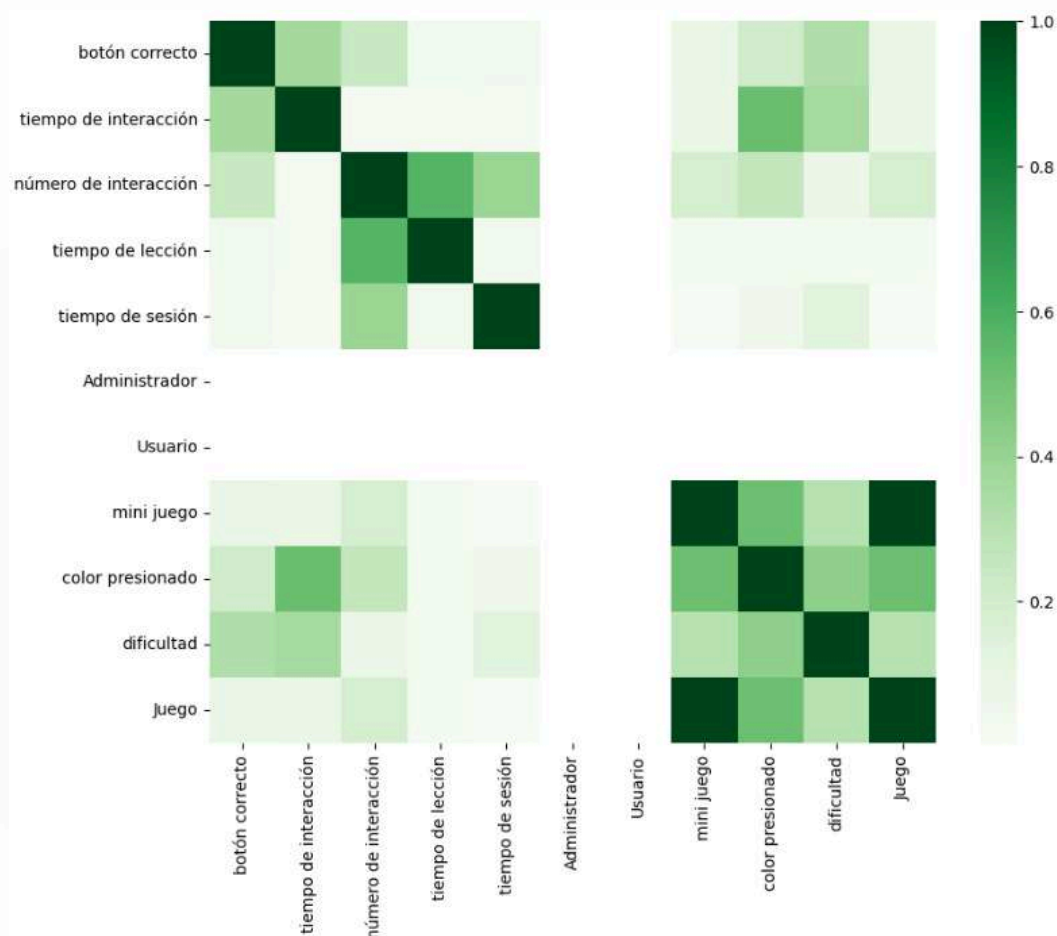


```
Corr_Factors=ASHLEY.corr()
Corr_Factors1=abs(Corr_Factors)
Heat_Map_ASHLEY=sns.heatmap(Corr_F
actors1,cmap="magma")
Heat_Map_ASHLEY
```

La mayor correlación fue: mini juego - dificultad. Seguido de cerca por: mini juego - color presionado.

La mayor correlación fue: mini juego - Juego.

```
Corr_Factors=ARLETT.corr()  
Corr_Factors1=abs(Corr_Factors)  
plt.figure(figsize=(10, 8))  
Heat_Map_ARLETT=sns.heatmap(Corr_Factors1,cmap="Greens")  
Heat_Map_ARLETT
```



Encontrar un modelo lineal múltiple que supere la mejor correlación lineal simple encontrada

```
y_pred= model.predict(X=data[["Administrador","botón  
correcto","tiempo de interacción","mini juego","número de  
interacción","color presionado","dificultad","Juego","tiempo  
de lección","tiempo de sesión"]])
```

y_pred

```
array([0.19199212, 0.14727468, 0.18424371, ...,  
0.33547098, 0.33686001,  
0.14771464])
```

Corroboramos cual es el coeficiente de Determinación y Correlación

```
1 #Corroboramos cual es el coeficiente de Determinación de nuestro modelo
2 coef_Deter=model.score(X=Vars_Indep, y=Var_Dep)
3 print("Coeficiente de Determinación",coef_Deter)
4 #Corroboramos cual es el coeficiente de Correlación de nuestro modelo
5 coef_Correl=np.sqrt(coef_Deter)
6 print("Coeficiente de Correlación",coef_Correl)
```

✓ 0.0s

Coeficiente de Determinación 0.04013733821604415
Coeficiente de Correlación 0.20034305132957356



Proponer un modelo de regresión multiple que supere las correlaciones obtenidas en los heatmap de cada usuario.

```
1 print(f"Coeficiente de Determinación (R²): {coef_deter:.4f}")
2 print(f"Coeficiente de Correlación (R): {coef_correl:.4f}")
3
4 # Evaluar el error cuadrático medio (MSE)
5 mse = mean_squared_error(y_test, y_pred)
6 print(f"Error Cuadrático Medio (MSE): {mse:.2f}")
✓ 0.0s
```

Coeficiente de Determinación (R^2): 0.0452
Coeficiente de Correlación (R): 0.2126
Error Cuadrático Medio (MSE): 0.62



Gracias