

```
In [ ]: import pandas as pd
```

```
In [ ]: eventos = pd.read_csv("events.csv")
```

```
In [ ]: #A

#Saco las sesiones duplicadas ya que no se repiten entre usuarios y una sesion puede tener mas de un evento
sesiones_no_duplicadas = eventos[["uid", "sessionid"]].drop_duplicates(subset=["sessionid"])

#Cuento la cantidad de sesiones que tiene cada usuario
sesiones_por_usuario = sesiones_no_duplicadas.groupby("uid").agg({"sessionid": "count"})
sesiones_por_usuario.columns = ["sesiones por usuario"]

#Calculo el promedio de sesiones por usuario
promedio_sesiones = sesiones_por_usuario["sesiones por usuario"].mean()

#Imprimo los usuarios que tienen una mayor cantidad de sesiones que el promedio, junto con las sesiones que tienen
sesiones_por_usuario[sesiones_por_usuario["sesiones por usuario"] > promedio_sesiones]
```

```
In [ ]: #B

#Al igual que antes, filtro las
sesiones_no_duplicadas = eventos.drop_duplicates(subset=["s
essionid"])
sesiones_no_duplicadas["sesiones_totales"] = sesiones_no_dup
licadas.groupby("uid")["sessionid"].transform("count")

#Filtro con el requerimiento pedido
usuarios_mas_de_25 = sesiones_no_duplicadas[sesiones_no_dup
licadas["sesiones_totales"]>25][["uid"]]

#Me quedo con todas las acciones de los usuarios con mas de
25 sesiones
usuarios_mas_de_25 = eventos[["uid", "event", "value"]].mer
ge(usuarios_mas_de_25["uid"], on="uid", how="inner")

#Filtro quedandome solo los users que realizaron los evento
s pedidos
usuarios_mas_de_25_eventos_filtrados = usuarios_mas_de_25[u
suarios_mas_de_25["event"].isin(["ecommerce.view-product",
"ecommerce.checkout", "ecommerce.conversion"])]

#Hago un pivote para generar la tabla pedida, no pongo aggf
unc porque es
#por default el promedio
tabla_final = pd.pivot_table(usuarios_mas_de_25_eventos_fil
trados, index="uid", columns = "event", values="value")

#Borro el titulo de las columnas
tabla_final= tabla_final.rename_axis(columns=None)

#Agrego mean a las columnas
tabla_final.columns = tabla_final.columns + "_mean"

#Dejo uid como columna, deja de ser indice
tabla_final.reset_index(inplace=True)

#Dejo el dataframe con el orden de columnas que pide le enu
nciado
tabla_final= tabla_final[["uid", "ecommerce.view-product_me
an", "ecommerce.checkout_mean", "ecommerce.conversion_mea
n"]]
tabla_final
```

```
In [ ]:
```