



1. Generar gráficas para entender y procesar los datos:
 - i. Generar graficas y reportes del total de personas vacunadas.

```
In [718]: #Generar graficas y reportes del total de personas vacunadas
print("Vacunados Primera Dosis: ",archivo['primera_dosis'].sum())
print("Vacunados Segunda Dosis: ",archivo['segunda_dosis'].sum())
print("Restan un total de ",(archivo['primera_dosis'].sum()-archivo['segunda_dosis'].sum()))

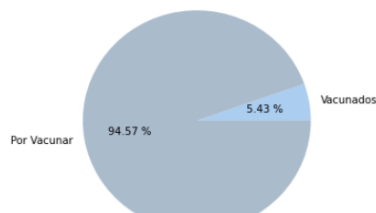
print("*****")
poblacion=17000000
print("Existe una poblacion de ",poblacion,"personas en el ecuador y tenemos ",archivo['primera_dosis'].sum())
print("Nos resta un total de ",poblacion-archivo['primera_dosis'].sum(),"por vacunar")
```

```
Vacunados Primera Dosis: 923656
Vacunados Segunda Dosis: 257524
Restan un total de 666132
*****
Existe una poblacion de 17000000 personas en el ecuador y tenemos 923656
Nos resta un total de 16076344 por vacunar
```

```
In [719]: # Generar un grafico de cual es su pie
```

```
plt.figure(figsize=(5,5))
etiquetas = ['Vacunados', 'Por Vacunar']
colores = ['#abccdf', '#aabbcc']
plt.pie([archivo['primera_dosis'].sum(), poblacion-archivo['primera_dosis'].sum()], labels=etiquetas, colors=colores, autopct='%2f %%')
plt.title('Grafico 1')
plt.show()
```

Grafico 1



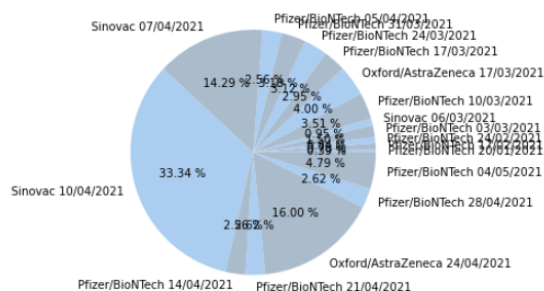
- ii. Generar gráfico de pie para el fabricante de la vacuna.

```
In [720]: nombrefecha=[]
fechatotalvacunas=[]
for i in range(len(fabricantes['total'])):
    aux=fabricantes['vaccine'][i]+" "+fabricantes['arrived_at'][i]
    nombrefecha.append(aux)
    fechatotalvacunas.append(fabricantes['total'][i])

# Generar un grafico de cual es su pie diestro
plt.figure(figsize=(5,5))

etiquetas = nombrefecha
colores = ['#abccdf', '#aabbcc']
plt.pie(fechatotalvacunas, labels=etiquetas, colors=colores, autopct='%2f %%')
plt.title('Grafico 2')
plt.show()
```

Grafico 2



Universidad Politécnica Salesiana

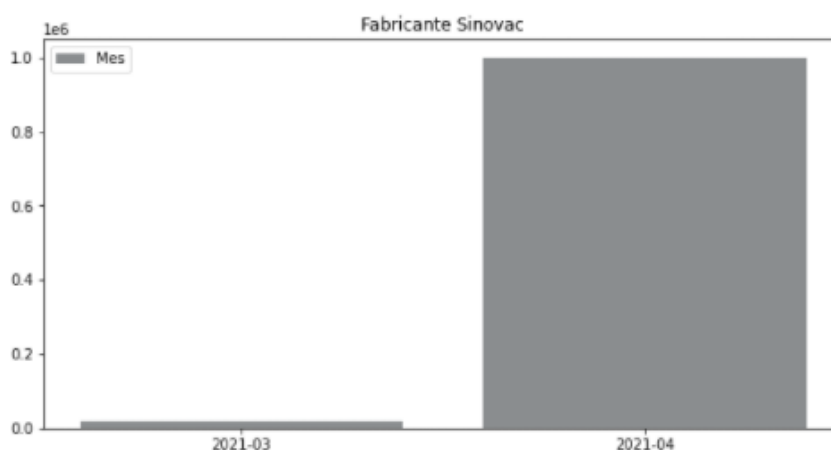
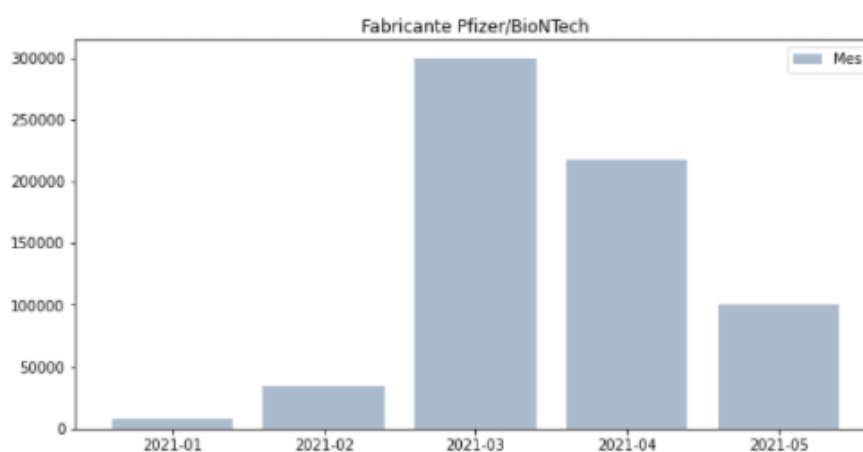
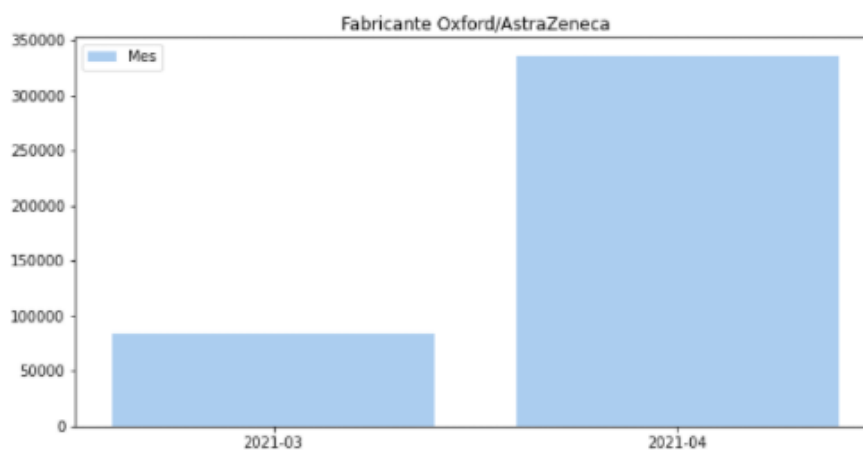


Simulación

Tema: Simulación de Eventos.



iii. Generar histogramas de vacunas por mes de llega y fabricante.



Universidad Politécnica Salesiana



Simulación

Tema: Simulación de Eventos.



- Generar un reporte parametrizado que pueda ingresar los datos de las fechas inicio y fin para obtener la información de las graficas vistas en el primer punto.

```
In [65]: #parameters
# Parameters
fecha_inicio=''
fecha_fin=''

In [72]: import pandas as pd
import numpy
from datetime import datetime, timedelta
import matplotlib.pyplot as plt

#Cargamos nuestros Dataset
total=pd.read_csv('total_vacunas.csv')
vacunas=pd.read_csv('fabricantes.csv')
vacunas['arrived_at']=pd.to_datetime(vacunas['arrived_at'],format='%d/%m/%Y')
total['fecha']=pd.to_datetime(total['fecha'],format='%d/%m/%Y')

In [73]: filtro=vacunas[(vacunas['arrived_at'] >= fecha_inicio) & (vacunas['arrived_at'] <= fecha_fin)]
grupo=filtro.groupby(['vaccine'], as_index=False)['total'].sum()
print("Total de vacunas por fabricante")
print(grupo)

Total de vacunas por fabricante
vaccine total
0 Oxford/Astrazeneca 84000
1 Pfizer/BioNTech 274950
2 Sinovac 20000

Generar graficas y reportes del total de personas vacunadas.

In [74]: #Generar graficas y reportes del total de personas vacunadas
print("Vacunados Primera Dosis: ",total['primera_dosis'].sum())
print("Vacunados Segunda Dosis: ",total['segunda_dosis'].sum())
print("Restan un total de ",(total['primera_dosis'].sum()-total['segunda_dosis'].sum()))

print("*****")
poblacion=17000000
print("Existe una poblacion de ",poblacion,"personas en el Ecuador y tenemos ",total['primera_dosis'].sum())
print("Nos resta un total de ",poblacion-total['primera_dosis'].sum(),"por vacunar")

plt.figure(figsize=(5,5))

etiquetas = ['Vacunados', 'Por Vacunar']
colores = ['#abcdef', '#aabbcc']
plt.pie([total['primera_dosis'].sum(), poblacion-total['primera_dosis'].sum()], labels=etiquetas, colors=colores, autopct='%2f %%%')
plt.title('Grafico 1')
plt.show()

Vacunados Primera Dosis: 14679606
Vacunados Segunda Dosis: 5131438
Restan un total de 9548168
*****
Existe una poblacion de 17000000 personas en el Ecuador y tenemos 14679606
```

Uso de papermill

Papermill es una herramienta que nos permite parametrizar y ejecutar cuadernos. Transforma su cuaderno Jupyter en una herramienta de flujo de trabajo de datos, ejecutando cada celda secuencialmente, sin tener que abrir JupyterLab (o Notebook). Intenta llenar el vacío de la automatización y el registro al ofrecernos una forma de ejecutar cuadernos como archivos y también al generar un informe para cada ejecución.

Generación de grafico por fecha

```
# Parameters
fecha_inicio=''
fecha_fin=''
```



- Generar un modelo matemático de predicción basado en regresión, del procesos de vacunación en base al numero actual de vacunados (1 y 2 dosis) y a la llegada de nuevas vacunas.

```
In [723]: import datetime as dt
archivo['fecha'] = pd.to_datetime(archivo['fecha'])
archivo['dia'] = (archivo['fecha'] - archivo['fecha'].min()) / np.timedelta64(1, 'D')
archivo['dia'] = archivo.iloc[:, 3:4].values
archivo['dia'] = archivo['dia'].astype(int)

features = archivo.iloc[:, 3:4].values
target = archivo.iloc[:, 1:2].values
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2)
model_lineal = LinearRegression()
model_lineal = model_lineal.fit(X_train, y_train)
print("Linear Score : ", model_lineal.score(X_test, y_test))

model_poly = Pipeline([('poly', PolynomialFeatures(degree=4)), ('linear', LinearRegression(fit_intercept=True))])
model_poly = model_poly.fit(X_train, y_train)
print("Poly Score : ", model_poly.score(X_test, y_test))
y_pred_pr_Ecuador = model_lineal.predict(X_test_chile)

#Variables Respaldo para grafico
X_test_ecuador=X_test
y_test_ecuador=y_test
y_pred=model_lineal.predict(X_test).reshape(-1)

Linear Score : 0.6272827459454688
Poly Score : 0.6318777521542099

In [724]: print("¿A partir de 2021-05-06 cuantos dias en adelante desea predecir?Ingrese solo el numero")
print("Ejemplo: 2 que en fecha seria 2021-05-08 a partir del 2021-05-06 ")
valor_prediccion = input()
datos = [i for i in range(105, 105+int(valor_prediccion))]
predicciones = ['{}'].format(i):[model_lineal.predict(np.array(i).reshape(1, -1))[0], model_poly.predict(np.array(i).reshape(1, -1))[0]]

¿A partir de 2021-05-06 cuantos dias en adelante desea predecir?Ingrese solo el numero
Ejemplo: 2 que en fecha seria 2021-05-08 a partir del 2021-05-06
600

In [725]: fecha_inicio = datetime(2021, 5, 6, 0, 0, 00, 00000)
for i in range(len(predicciones)):
    new_date = fecha_inicio + timedelta(days=i)
    print("Fecha")
    print(new_date)
    print("Valor De Prediccion:[prediccion Lineal,Prediccion Polinomial]")
    print("Lineal: ", predicciones.get(str(i+105))[0])
    print("Polinomial: ", abs(predicciones.get(str(i+105))[1]))
    print("*****")
```

Prediccion 1 y 2 dosis Ecuador

```
In [726]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures

features = archivo.iloc[:, 3:4].values
target = archivo.iloc[:, 1:2].values
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2)
model_lineal = LinearRegression()
model_lineal = model_lineal.fit(X_train, y_train)
print("Linear Score : ", model_lineal.score(X_test, y_test))

model_poly = Pipeline([('poly', PolynomialFeatures(degree=4)), ('linear', LinearRegression(fit_intercept=True))])
model_poly = model_poly.fit(X_train, y_train)
print("Poly Score : ", model_poly.score(X_test, y_test))

Linear Score : 0.4198670684477549
Poly Score : 0.5111439213182265

In [727]: print("¿A partir de 2021-05-06 cuantos dias en adelante desea predecir?Ingrese solo el numero")
print("Ejemplo: 2 que en fecha seria 2021-05-08 a partir del 2021-05-06 ")
valor_prediccion = input()
datos = [i for i in range(105, 105+int(valor_prediccion))]
predicciones = ['{}'].format(i):[model_lineal.predict(np.array(i).reshape(1, -1))[0], model_poly.predict(np.array(i).reshape(1, -1))[0]]

¿A partir de 2021-05-06 cuantos dias en adelante desea predecir?Ingrese solo el numero
Ejemplo: 2 que en fecha seria 2021-05-08 a partir del 2021-05-06
600

In [728]: fecha_inicio = datetime(2021, 5, 6, 0, 0, 00, 00000)
for i in range(len(predicciones)):
    new_date = fecha_inicio + timedelta(days=i)
    print("Fecha")
    print(new_date)
    print("Valor De Prediccion:[prediccion Lineal,Prediccion Polinomial]")
    print("Lineal: ", predicciones.get(str(i+105))[0])
    print("Polinomial: ", abs(predicciones.get(str(i+105))[1]))
    print("*****")
```



Desarrollar y generar un proceso de comparación con al menos dos países (1. Latinoamérica, 1. Europa).

Generar las gráficas de regresión y comparar.

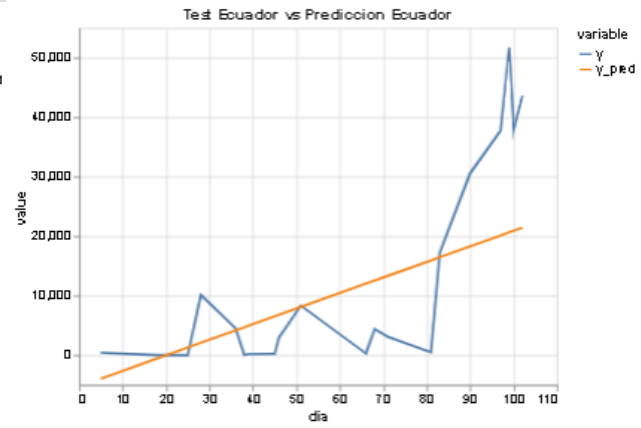
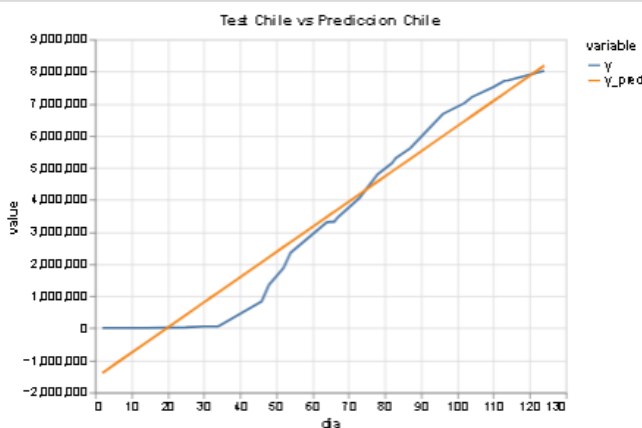
Comparacion Chile Vs Ecuador

```
In [736]: data_lineal=pd.DataFrame({'dia':X_test_chile.reshape(-1),'y':y_test_chile.reshape(-1),'y_pred':model_lineal.predict(X_test_chile).res
alt.Chart(data_lineal.melt('dia')).mark_line().encode(x='dia',y='value',color='variable').properties(title='Test Chile vs Prediccion

Out[736]:

In [735]: data_lineal=pd.DataFrame({'dia':X_test_ecuador.reshape(-1),'y':y_test_ecuador.reshape(-1),'y_pred':y_pred})
alt.Chart(data_lineal.melt('dia')).mark_line().encode(x='dia',y='value',color='variable').properties(title='Test Ecuador vs Prediccion

Out[735]:
```



Podemos notar que en un lapso de 100 dias chile(Azul-Negro) tiene un aumento exponencial por lo cual la grafica de la regresion lineal(Negro) nos presenta un gran aumento frente a ecuador.

Lo que demuestra que no estamos al nivel de vacunacion de chile.

Comparacion Paises Bajos vs Ecuador

```
In [734]: import datetime as dt
archivo_paisesbajos = pd.read_csv('vacunacionPaisesbajos.csv')
archivo_paisesbajos['fecha'] = pd.to_datetime(archivo_paisesbajos['fecha'])
archivo_paisesbajos['dia'] = (archivo_paisesbajos['fecha'] - archivo_paisesbajos['fecha'].min()) / np.timedelta64(1,'D')
archivo_paisesbajos['dia'] = archivo_paisesbajos['dia'].values
archivo_paisesbajos['dia'] = archivo_paisesbajos['dia'].astype(int)
features = archivo_paisesbajos.iloc[:,3:4].values
target = archivo_paisesbajos.iloc[:,1:2].values
X_train_paisesbajos, X_test_paisesbajos, y_train_paisesbajos, y_test_paisesbajos = train_test_split(features, target, test_size=0.2)
model_lineal = LinearRegression()
model_lineal = model_lineal.fit(X_train_paisesbajos,y_train_paisesbajos)
print("Linear Score : ",model_lineal.score(X_test_paisesbajos,y_test_paisesbajos))
data_lineal=pd.DataFrame({'dia':X_test_paisesbajos.reshape(-1),'y':y_test_paisesbajos.reshape(-1),'y_pred':model_lineal.predict(X_test
alt.Chart(data_lineal.melt('dia')).mark_line().encode(x='dia',y='value',color='variable').properties(title='Test Paises Bajos vs Prediccion

Linear Score : 0.8602719695363671
```

Out[734]:

```
In [733]: data_lineal=pd.DataFrame({'dia':X_test_ecuador.reshape(-1),'y':y_test_ecuador.reshape(-1),'y_pred':y_pred})
alt.Chart(data_lineal.melt('dia')).mark_line().encode(x='dia',y='value',color='variable').properties(title='Test Ecuador vs Prediccion

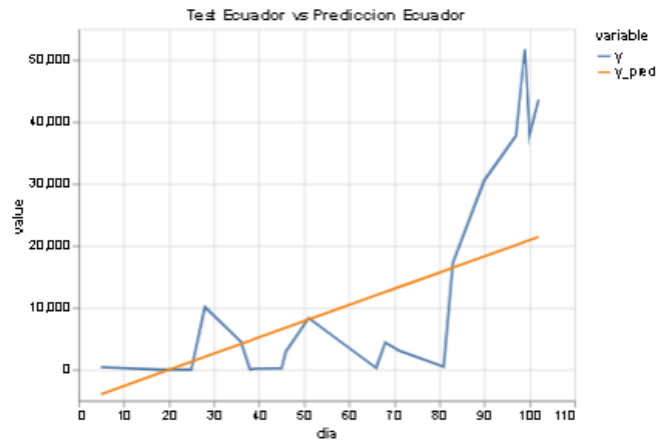
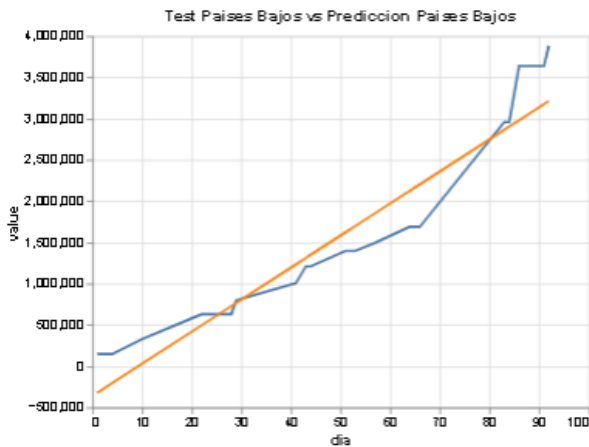
Out[733]:
```

Universidad Politécnica Salesiana



Simulación

Tema: Simulación de Eventos.



En un lapso de 100 días y con una población entre 17 millones tenemos un total de 50 mil personas vacunadas frente a un total de 4 millones en el pasar de 100 días lo que demuestra el deficiente plan de vacunación en el Ecuador.

Identificar cual es la fecha tentativa en la que todos los Ecuatorianos podrán ser vacunados con las dos dosis.

Obtenemos un accuracy del

```
Linear Score : 0.49588588319185223  
Poly Score : 0.7940945812509957
```

Podemos identificar que la regresión polinomial es la más acertada para nuestra predicción.

- **Fecha Tentativa en la que todos los Ecuatorianos podrán ser vacunados con las dos dosis:**

```
Fecha  
2022-02-06 00:00:00  
Valor De Prediccion:[prediccion Lineal,Prediccion Polinomial]  
Lineal: 94399.89782943949  
○ Polinomial: 17404749.310474027
```

En el Ecuador somos un total de 17.32 millones

Desde el día de realización de la prueba restaban un total de:

Universidad Politécnica Salesiana



Simulación

Tema: Simulación de Eventos.



```
Vacunados Primera Dosis: 923656
Vacunados Segunda Dosis: 257524
Restan un total de 666132
*****
Existe una poblacion de 17000000 personas en el ecuador y tenemos 923656
Nos resta un total de 16076344 por vacunar
```

La predicción es en:

```
Fecha
2022-02-01 00:00:00
Valor De Predicción:[predicción Lineal,Predicción Polinomial]
Lineal: 93091.93299615908
Polinomial: 16418822.923112445
.....
```

Análisis

- **Cual tiene una mejor predicción.**
 - Ofrece una mejor predicción la polinomial ya que desde el ingreso de datos aumenta la cantidad de vacunas ingresadas exponencialmente por lo cual la polinomial se desempeña de mejor manera.
- **Ventajas y desventajas de los modelos.**

Regresión Polinomial

Ventajas

- Funciona con cualquier tamaño de muestra.
- Trabaja bien sobre datos no lineales.

Desventaja

- Se requiere elegir el grado correcto del polinomio para una buena relación sesgo/varianza.

Regresión Lineal

Ventajas

- Trabaja con cualquier tamaño de muestra.
- Informa sobre la relevancia de cada variable dependiente.

Desventaja

- Asume que los datos se comportan de forma lineal

- **Opinión**
 - El uso de una regresión se debe implementar en función al tamaño de la muestra por lo cual se debe tener en claro cada punto.
- **Conclusiones**

A través de esta prueba comprendimos el uso de la regresión y la implementación de gráficos interactivos. El uso de la herramienta papermill nos ayuda a mejorar la interacción con el usuario. El uso de la regresión polinomial en esta prueba fue clave para obtener una predicción acertada sobre el tiempo que tomará la vacunación en el ecuador.
- **Recomendaciones.**
 - Al realizar una regresión debemos entender como esta formado nuestro dataset para elegir una regresión y de esta manera obtendremos un buen resultado.

Universidad Politécnica Salesiana



Simulación

Tema: Simulación de Eventos.

