

Examen Inter ciclo Simulación

Realizado por: Tatiana Cárdenas Jho

Parte 1:

El golpe económico de la crisis sanitaria del coronavirus no va a ser cosa de semanas, sino de meses. Dentro de una de las etapas importantes posteriores a las elecciones presidenciales son el análisis y tendencia que tiene el actual presidente. Para ello se planteó realizar un sistema de regresión que permita identificar cual es la tendencia en base al manejo de las redes sociales de Twitter del presidente electo el 24 de mayo, Guillermo Lasso.

Para la recopilación de datos se ha procedido a crear una cuenta de desarrollador en Twitter para poder registrar una nueva APP, de tal manera que con la ayuda de Tweepy, siendo esta una librería para Python 2.6, 2.7, y 3. x que permite acceder al API de Twitter, de tal manera que provee acceso a todos los métodos API RESTful, incluyendo la lectura y escritura de tweets, en este caso, vamos a leer los Tweets de Guillermo Lasso.

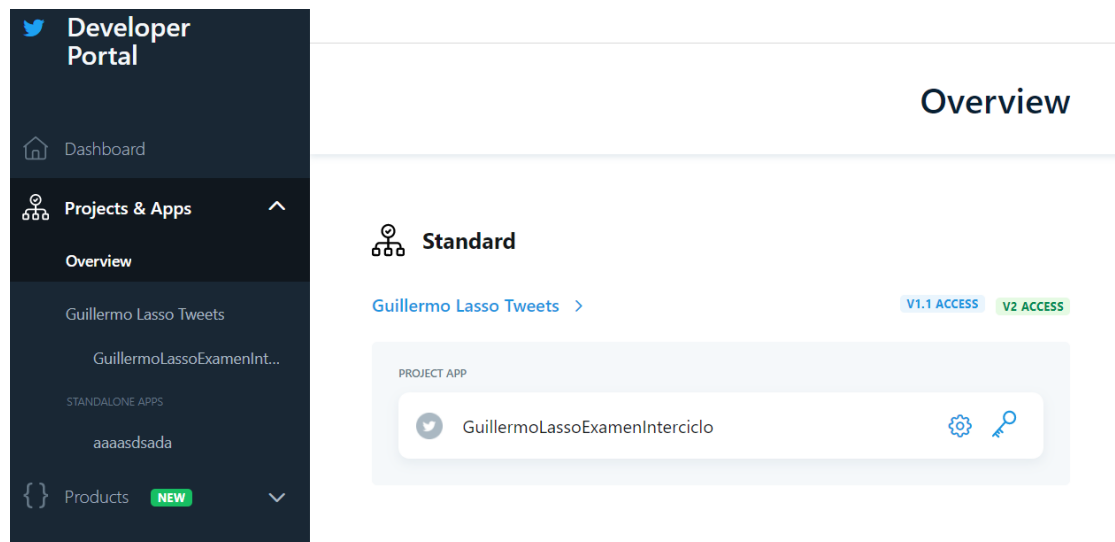


Ilustración 1 Creación de nueva app en Twitter

Tras crear el api de Twitter, procedemos a revisar los tokens que necesitaremos para poder acceder a la información del mandatario, estos tokens, se visualizan en la api que se creó como se muestra a continuación:

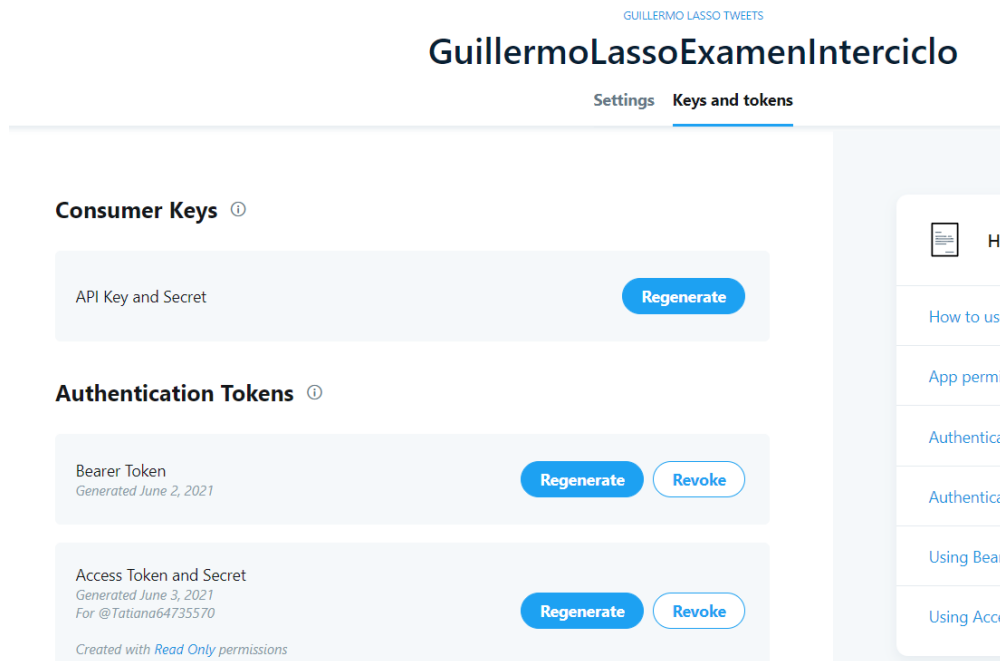


Ilustración 2 Tokens de la API generados en Twitter

Para poder acceder a los mismos es necesario seleccionar el botón de regenerar y aparecerá sin ningún problema, sin embargo, cada vez que se regenere un token, es necesario cambiar en el código donde se esté trabajando.

```
#Api para acceder a los tweets y claves de la api para poder acceder.
access_token="1400175729267163138-IDTuUVPeNNxhBNkohIruIg6XaKdm2g"
access_token_secret="79ftXKrMLTDGBIquTPhVHryO9N9qWdrf5d0Kc8H2fS4Ed"
consumer_key="1Y60IOoBpUJotVhezHqQyJKk"
consumer_secret="AQAr8GVvXaLsdMqMyMokF7IZteCaBSRUdtietwQysVnKRjSftF"
auth= tweepy.OAuthHandler(consumer_key,consumer_secret)
auth.set_access_token(access_token,access_token_secret)

#Se trabajará con 1000 Tweets
tweets_num=1000
```

Ilustración 3 Importación del Access token y el consumer token

A continuación, se presenta la función que permitirá almacenar los tweets: los tweets que contengan la palabra "LassoGuillermo", se filtrarán y de manera inicial se almacenará dentro de un archivo de texto.

```

tweetsObtenidos=[]
class Tweets(tweepy.StreamListener):

    def __init__(self,api=None):
        super(Tweets,self).__init__()
        self.num_tweets=0
        self.file=open("tweetsGuillermoLasso.txt","w")

    def on_status(self,status):
        tweet=status._json
        self.file.write(json.dumps(tweet)+ '\n')
        tweetsObtenidos.append(status)
        self.num_tweets+=1
        if self.num_tweets<tweets_num:
            return True
        else:
            return False
        self.file.close()

#crear objeto de transmisión y autenticar
tweet = Tweets()
stream =tweepy.Stream(auth,tweet)

# Filtra flujos para capturar datos por palabras clave
stream.filter(track=['LassoGuillermo'])

```

Ilustración 4 Funcion que recibe los tweets por filtro de palabra

Posterior se haya creado el archivo .txt, procedemos a crear un archivo .csv donde se integrarán los datos obtenidos y lo leeremos para el análisis de datos como se presenta a continuación:

```

#Carga de datos

#Variables
#dataset = pd.read_csv('dataset.csv')
#print(dataset.head())

df=pd.read_csv('tweetsGuillermoLasso.csv')
df = df[df.Text.notna()]
df.sample(5)

```

Ilustración 5 Carga de datos del archivo .csv con todos los tweets obtenidos

A continuación, se realizará un proceso de análisis de datos a través de las funciones proporcionadas por pandas, ya que la limpieza de datos es el proceso de detectar y corregir o eliminar registros corruptos o inexactos de un conjunto de registros, tabla o base de datos y se refiere a la identificación de partes incompletas, incorrectas, inexactas o irrelevantes de los datos, para su posterior sustitución, modificación o eliminación de los datos sucios o poco precisos, para lo cual utilice las funciones:

- head
- info
- keys
- describe
- sample
- dtypes

Posterior a este ejercicio se realizó la regresión lineal y polinomial en cuanto a la predicción de un tweet, a continuación, se presenta las graficas representativas de los resultados de las regresiones y el acurracy de cada una:

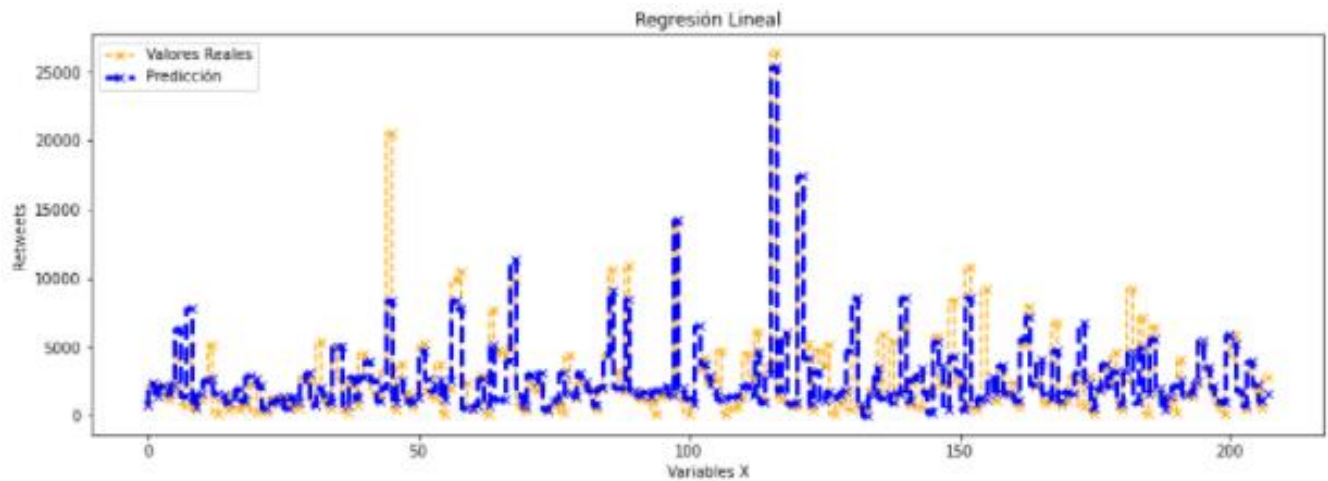


Ilustración 6 Grafica regresión lineal

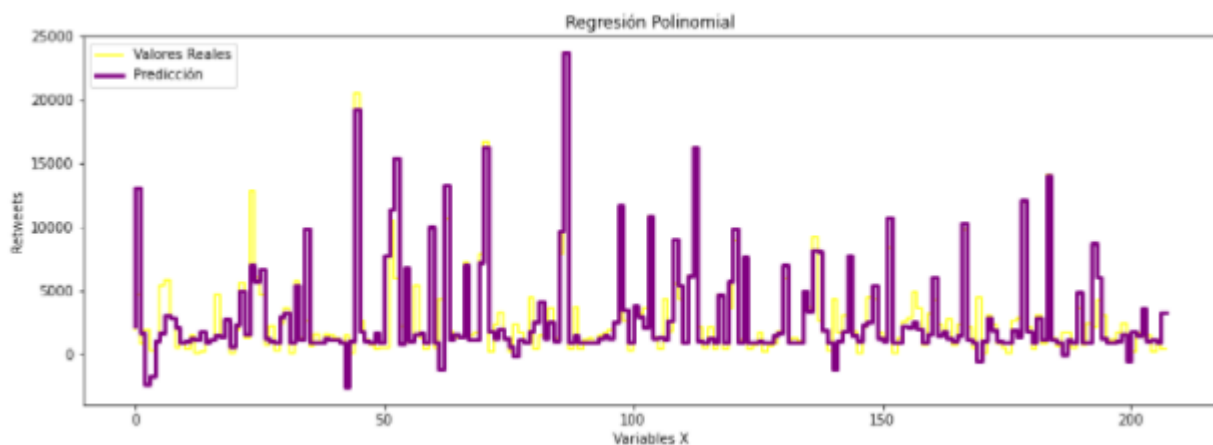


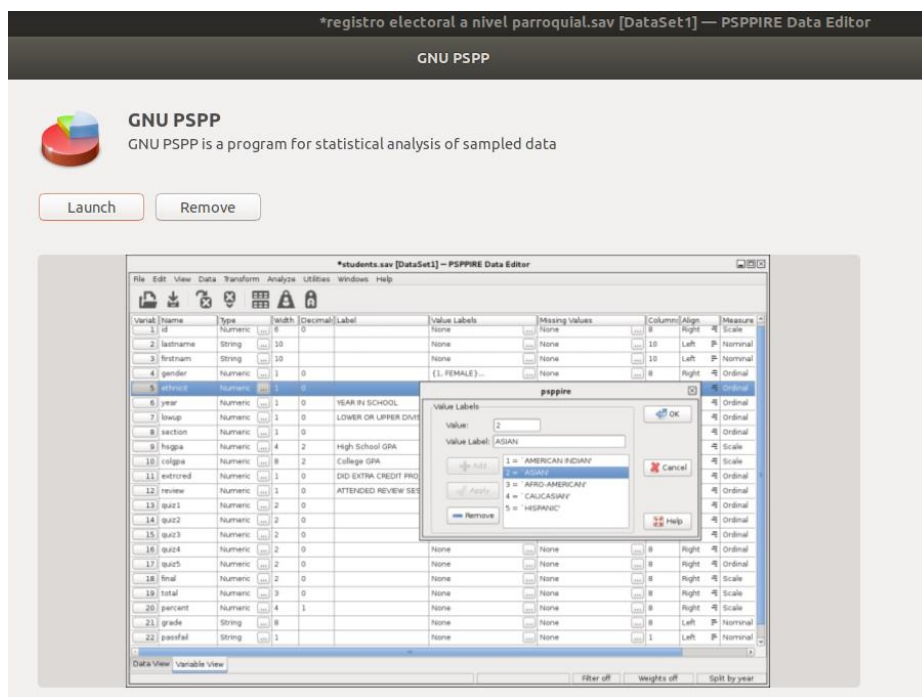
Ilustración 7 Gráfica regresión polinomial

Predicción regresión lineal: `[[6158.90830662]]`
 Predicción regresión polinomial: `[[10292.23326259]]`

Ilustración 8 Predicción lineal vs Predicción polinomial

Parte 2:

Tras la pandemia mundial denominada COVID-19, se incrementó 611 recintos electorales para el 2021, de tal manera que, según la información proporcionada por el (Consejo Nacional Electoral, 2021), dentro de las estadísticas del registro electoral de las elecciones realizadas en el 2021, se presenta un documento de manera transparente en el cual se evidencia la información detallada de los votantes de acuerdo a su parroquia a la que fue asignada, no obstante, este documento se encuentra en formato .sav, para lo cual se tuvo que instalar el programa GNU PSPP en Linux para que pueda hacer la lectura del mismo como se presenta a continuación:



Posterior a la instalación se procedió a abrir el documento el cual nos entregaría una data completa de:

- Provincia
- Nombre de provincia
- Código de cantón
- Nombre de cantón
- Parroquia
- Nombre de Parroquia
- Sexo de la junta
- Electores

Como resultado del análisis de datos, se puede observar que la columna de "Retweets" contiene el número de retweets y la palabra mil, por lo que nosotros solo necesitamos el número de este. A continuación, se procede a hacer una limpieza de datos, en la cual eliminaremos la palabra mil de la columna y otros valores que no son de nuestro interés.

```
#Este metodo cambia Los nombres de Las columnas y elimina la palabra mil de Los valroes que obtiene de Los tweets
#Se vuelve a leer para hacer el cambio de valores
df=pd.read_csv('tweetsGuillermoLasso.csv')
df = df[df.Text.notna()]

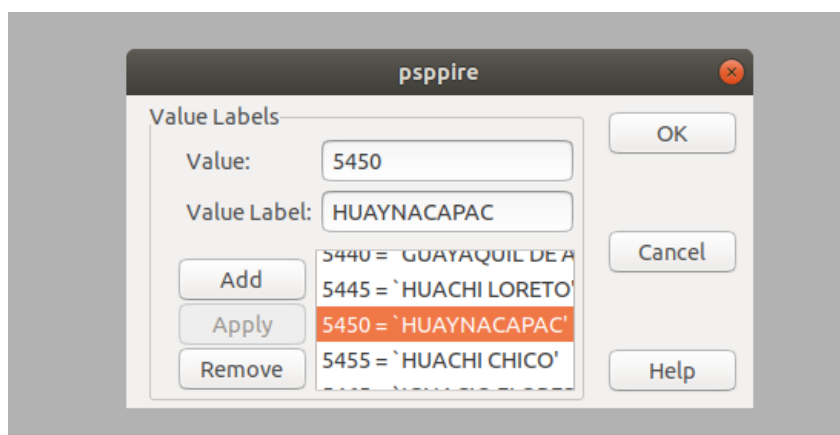
df.Comments = df.Comments.replace([r'[ mil]+$',','], ['00',''], regex=True)
df.Likes = df.Likes.replace([r'[ mil]+$',','], ['00',''], regex=True)
df.Retweets = df.Retweets.replace([r'[ mil]+$',','], ['00',''], regex=True)
df.Text = df.Text.replace([r'Citar Tweet'], [' Citar Tweet'], regex=True)

df['Hashtags'] = df.Text.str.findall(r'#.*?(?=\s|$)').str.len()
df['PalabrasTotales']= df.Text.apply(lambda x: len(x.split()))
df['links'] = df.Text.apply(lambda x: 1 if str(x).find('Citar Tweet') != -1 else 0)

df = df.astype({"Comments": float, "Likes": float, "Retweets": float, "Hashtags": float, "PalabrasTotales": float, "links": float})
```

Ilustración 9 Limpieza y tratamiento de datos del csv

De manera inicial, el documento no proporciona directamente el nombre de la parroquia a la cual hacer referencia la data, sino al código de esta, para lo cual, se tuvo que buscar a que parroquia hace referencia el lugar donde voy a sufragar normalmente y la relación con el código que se presenta, es decir la parroquia Huayna Capac como se presenta a continuación:



175	1	1,00	260	260	5450	5450	F	1	319
176	1	1,00	260	260	5450	5450	F	2	7254
177	1	1,00	260	260	5450	5450	F	3	1348
178	1	1,00	260	260	5450	5450	M	1	317
179	1	1,00	260	260	5450	5450	M	2	6382
180	1	1,00	260	260	5450	5450	M	3	1042

Realizando un análisis de la data, podemos diferir que existen 3 juntas de voto, y la junta receptora de la Universidad de Cuenca aproximadamente tuvo 5.554 electores.

Posterior a la obtención de la data, se procedió a realizar un cuaderno en Jupyter, en el cual se estableció las librerías, variables, funciones, resultados y graficas de los resultados con los requerimientos solicitados por el docente. Normalmente en los recintos que son bastante pequeños según los estudiantes de medicina de la Universidad del Azuay y la Universidad Católica, existen entre 12 - 15 mesas de tal manera que, el recinto de la Universidad de Cuenca es bastante grande, por lo que se establece 23 mesas de vacunación.

🚦 Librerías:

```
#Importación de librerías
import matplotlib.pyplot as plt
import simpy
import random
import datetime
from statistics import mean

#Nueva libreria para graficas
import seaborn as sns
import pygal

print('Importadas')
```

Las librerías más importantes para la resolución del examen fueron simpy, statics y random, y las librerías para realizar las gráficas fueron matplotlib y seaborn.

Seaborn: se basa en la biblioteca matplotlib. tiene muchas funciones integradas con las que puede crear gráficos llamativos con solo líneas simples de códigos. Proporciona una variedad de gráficos de visualización avanzada con una sintaxis simple como gráficos de caja, gráficos de violín, gráficos de dist, gráficos conjuntos, gráficos de pares, mapas de calor, entre otros.

🚦 Variables:

Las variables que se presenta a continuación hace referencia a aquellos valores que serán utilizados en las funciones, para que a través de los mismos el valor sea utilizado mas de una ves en el proceso de la simulación.

```

# Personas que asistieron a votar
TOTAL_PERSONAS = 5554 # REFERENCIA 1

# Unicamente el 80% de electores serán vacunados
PERSONAS_A_VACUNAR = TOTAL_PERSONAS*0.8

# Personal de vacunación "MESAS"
# Normalmente en los recintos que son bastante pequeños segun los estudiantes de la UDA existen entre 12 - 15 mesas de
#vacunacion por lo que se establece que el el recinto de la Universidad de Cuenca es bastante grande, por lo que se
#establece 23 mesas de vacunacion
PERSONAL_VACUNACION = 23

# Intervalo de tiempo de control para que la persona se registre "minutos"
REGISTRO_DATOS_MESA = 5

# Intervalo de tiempo para vacunación "minutos"
#*Realiza la vacunación en un tiempo aleatorio entre 5 a 10 minutos.
VACUNACION_PERSONA = 6

# Intervalo de tiempo para verificación de salud despues de vacunarse "minutos"
# Debe esperar 20 minutos dentro del establecimiento para verificar que no tenga problemas de salud
VERIFICACION_SALUD = 20

#*La persona recibe su certificado de vacunación y la fecha de la próxima vacuna entre 2 - 3 minutos.
CERTIFICADO_SEGUNDA = 2

# Tiempo de simulación 2H
TIEMPO_SIMULACION = 120

# Vacunados
VACUNADOS = 0

# Cantidad de personas que llegaron
TOTAL_PERSONAS = 0

# Tiempo Vacunación
TIEMPO_TOTAL_VACUNACION = 0

# Tiempo Espera
TIEMPO_ESPERA = 0

# Tiempo de vacunación de cada persona
TIEMPO_VACUNACION_PER_PERS = {}

# Tiempo de espera de cada persona
TIEMPO_ESPERA_PER_PERSONA = {}

```

A continuación, se presentan todas las funciones utilizadas para el presente examen de tal manera que la comunicación y la utilización de todas es optimo para poder compilar la simulación de la vacunación.

Existen cuatro pasos primordiales para la ejecución de la simulación en cuanto a la vacunación en el recinto de la Universidad de Cuenca:

- Paso 1: Hacer fila en caso de que las mesas estén ocupadas
- Paso 2: Vacunar a la persona
- Paso 3: Verificar la salud del elector después de haber vacunado al elector
- Paso 4: Dar certificado de vacuna al elector y la fecha de la segunda dosis

La función para ejecutar la simulación consta de personal de vacunación, siendo estas las personas que se van a encargar de vacunar, el tiempo de espera del turno, el tiempo de vacunación y el tiempo de verificar la salud.

Resultados:

En la siguiente grafica podemos visualizar los resultados obtenidos del total de vacunados que se pudo realizar en 120 minutos.

Total de personas vacunadas:

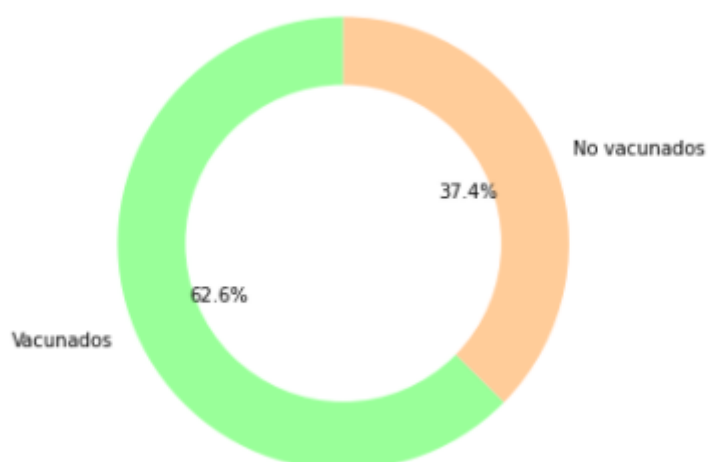
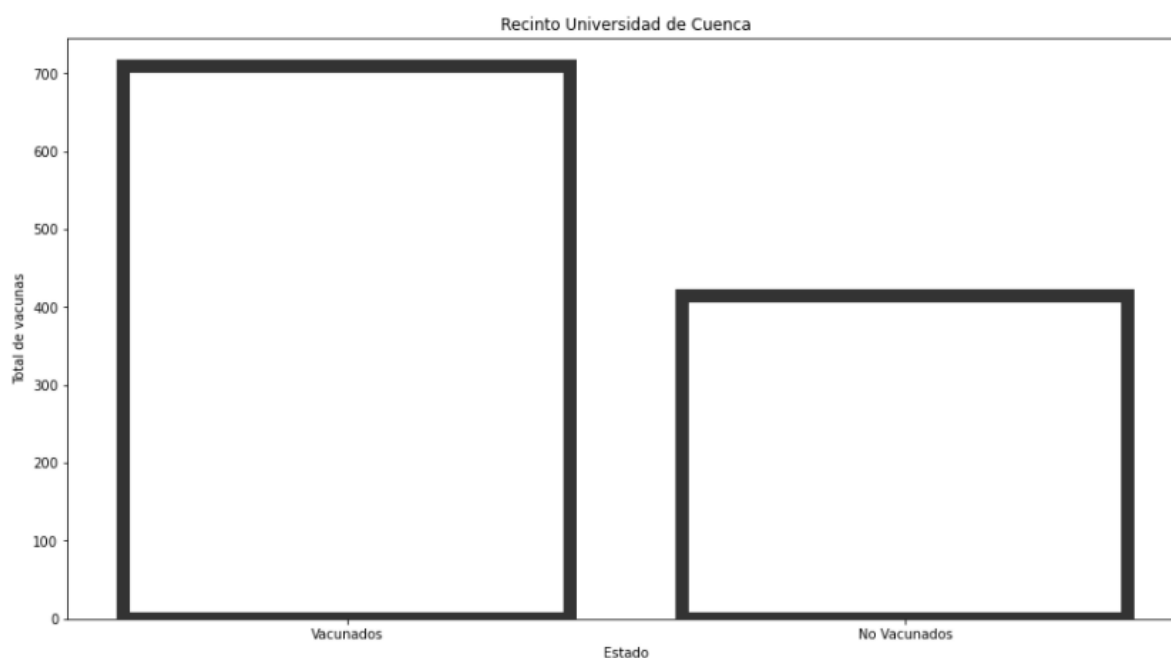
```
print('Total de electores vacunadas:', VACUNADOS)
```

Total de electores vacunadas: 710

Promedio de tiempo de espera:

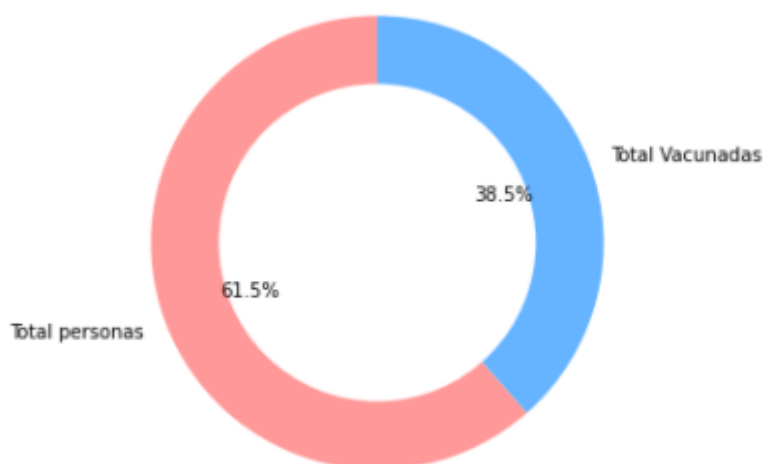
```
new_1 = list(map(int, TIEMPO_ESPERA_PER_PERSONA)) # or just map(int, L) in Python 2
promedio_espera = mean(new_1)
print('El promedio de espera desde hacer fila hasta que le entregan el certificado es:', round(promedio_espera, 2))
```

El promedio de espera desde hacer fila hasta que le entregan el certificado es: 54



Finalmente, se presenta una grafica comparativa entre el total de personas que asistieron al recinto para el proceso de vacunación versus el total de personas que fueron vacunadas tras el proceso de control de salud que se realiza previamente antes de pasar a la que el encargado le

asigne la vacuna.



Conclusiones:

Podemos diferir que el análisis de datos en un tema muy importante antes de realizar algún tipo de actividad, proceso, función y/o modelo, ya que es importante identificar el objetivo de trabajo, de tal manera que a través de un examen con información actual, nos permite simular actividades de la vida cotidiana en el Ecuador.

La simulación permite obtener un prototipo físico y ver qué sucede antes de entrar al proceso de producción, como puede ser el caso particular de las vacunas en el Ecuador, con los puntos de los recintos electorales, siendo el máximo beneficio reducir la posibilidad de que se presenten defectos o errores en el proceso de vacunación.

Referencias:

- Consejo Nacional Electoral. (2021). Bases de datos - Registro Electoral | Consejo Nacional Electoral. <http://portal.cne.gob.ec/es/estadisticas/bases-de-datos/category/1954-registro-electoral>