

# Equipo 79 - TEAM DSR

---

David Martínez Celis  
Sofía Sampogna  
Ricardo González

# Problema

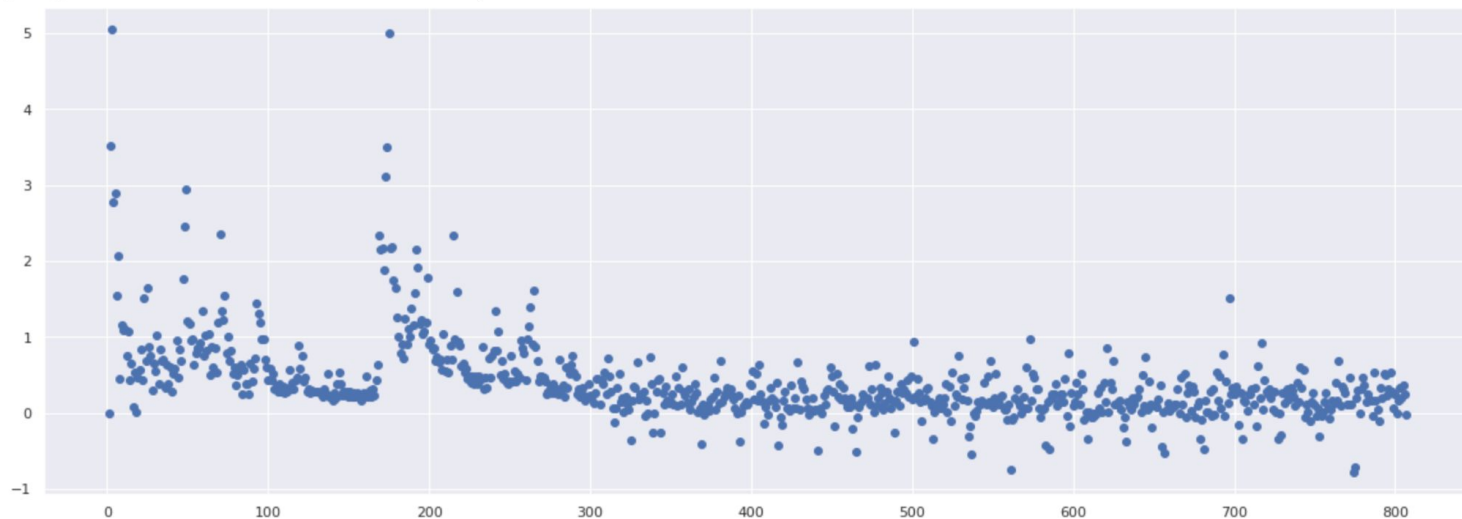
Para poder pronosticar la variación anual de la inflación general y subyacente, se tiene que generar un modelo de nowcasting que sea entrenado utilizando los datos desde enero del 2000 a diciembre de 2020. Los datos pronosticados serán esos desde enero del 2021 a agosto 2021. La eficiencia y precisión de los datos pronosticados serán definidos por su comparación a los datos reales conseguidos en el año 2021.

# Fase 1 - Recolectar dataset

- Se recolectaron los datos de la variación quincenal del INPC desde la 1Q de Enero de 1988, hasta la 1Q de Agosto de 2021, de la página del INEGI.
- Para ordenar los datos, se decidió convertir las fechas a números enteros, empezando del 1 hasta 804, para poder utilizarlo con más facilidad dentro del programa
- También se decidió incluir en la base de datos diferentes cifras del INPP, los cuales se tomaron de igual manera del INEGI

## Fase 2 - Diagrama de dispersión

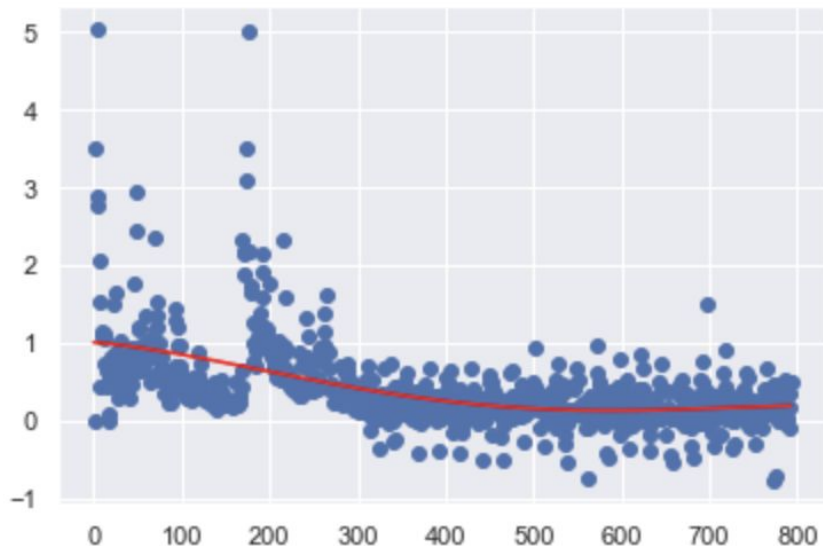
Elaboración diagrama de dispersión para analizar la tendencia general de los datos y así comenzar a planear cuál sería el mejor approach



## Fase 3 - Primer Modelo



Se hizo prueba con modelo lineal de regresión para observar la calificación de entrada de este



```
In [111]: r2 = r2_score(train_y, mymodel(train_x))  
           print(r2)  
0.2654534979826165
```

## Fase 4 - Limpieza de Datos

Observamos que nuestros datos contenían un alto número de “outliers” que afectaban el modelo de regresión, por lo que decidimos suavizarlos utilizando el método de “z score”

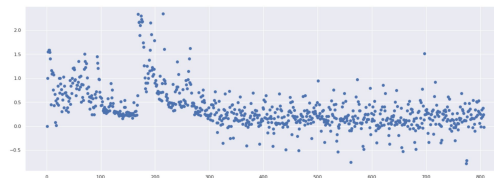
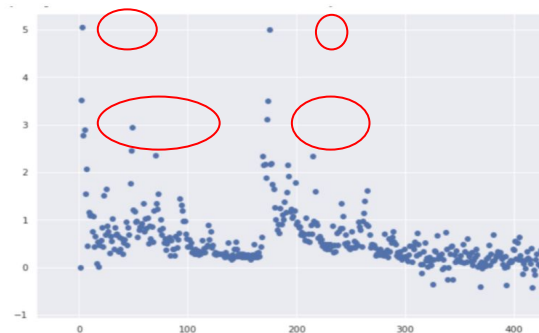
How to calculate Z score?

$z = (\text{data point} - \text{mean}) / \text{standard deviation}$

$$Z = \frac{(x - \mu)}{\sigma}$$

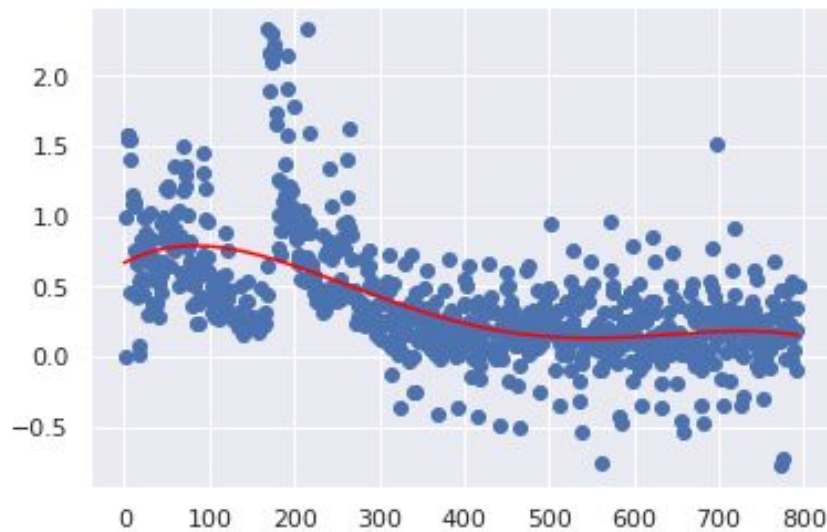
Diagram illustrating the Z-score formula components:

- Data point** (labeled  $x$ )
- Mean** (labeled  $\mu$ )
- Standard deviation** (labeled  $\sigma$ )



## Fase 5 - Nuevo ajuste del modelo

Para conseguir un mejor resultado, decidimos utilizar Training Test para conseguir mejores predicciones de nuestros datos sin tomar en cuenta las anomalías presentadas.



```
[15]
r2 = r2_score(train_y, mymodel(train_x))
print(r2)

0.3303782192830963
```

# Fase 6 - Prueba con otros modelos

## Regresión Ridge

### Ridge Regression

```
In [36]: train_x = x1[:793]
         train_y = y1[:793]

         test_x = x1[793:]
         test_y = y1[793:]

         train_x = train_x.values.reshape(-1, 1)
         test_x = test_x.values.reshape(-1, 1)

         ridge = Ridge().fit(train_x, train_y)
         print("Training set score: {:.2f}".format(ridge.score(train_x, train_y)))
         print("Test set score: {:.2f}".format(ridge.score(test_x, test_y)))

Training set score: 0.28
Test set score: -2.88
```

```
In [37]: train_x = x1[:793]
         train_y = y1[:793]

         test_x = x1[793:]
         test_y = y1[793:]

         train_x = train_x.values.reshape(-1, 1)
         test_x = test_x.values.reshape(-1, 1)

         ridge10 = Ridge(alpha=10).fit(train_x, train_y)
         print("Training set score: {:.2f}".format(ridge10.score(train_x, train_y)))
         print("Test set score: {:.2f}".format(ridge10.score(test_x, test_y)))

Training set score: 0.28
Test set score: -2.88
```



# Fase 6 - Prueba con otros modelos

## Árbol de Decisión

### Decision Tree

```
In [38]: #no se puede usar para pronosticos
from sklearn.tree import DecisionTreeRegressor

train_x = x1[:793]
train_y = y1[:793]

test_x = x1[793:]
test_y = y1[793:]

train_x = train_x.values.reshape(-1, 1)
test_x = test_x.values.reshape(-1, 1)

tree = DecisionTreeRegressor(random_state=0)
tree.fit(train_x, train_y)
print("Training set score: {:.2f}".format(tree.score(train_x, train_y)))
print("Test set score: {:.2f}".format(tree.score(test_x, test_y)))
```

```
Training set score: 1.00
Test set score: -2.75
```

# Fase 6 - Prueba con otros modelos

## Random Forest

### Random Forest

```
In [56]: from sklearn.ensemble import RandomForestRegressor

train_x = x1[:793]
train_y = y1[:793]

test_x = x1[793:]
test_y = y1[793:]

train_x = train_x.values.reshape(-1, 1)
test_x = test_x.values.reshape(-1, 1)

forest1 = RandomForestRegressor(random_state=0,n_estimators=1000)
forest1.fit(train_x, train_y)
print("Training set score: {:.2f}".format(forest1.score(train_x,train_y)))
print("Test set score: {:.2f}".format(forest1.score(test_x,test_y)))
```

```
Training set score: 0.94
Test set score: -0.86
```

# Fase 7 - Pronósticos y Error

Fecha	Variacion	Pronóstico	EAM				
1	0	0.8500923	0.0004051		Promedio	0.383135068	
2	1	0.8489728	7.19699E-05		n	804	
3	1.55	0.8478532	0.000334598				
4	2.1	0.8467336	0.000597227				
5	2.2	0.8456141	0.000645414				
6	1.54	0.8444945	0.000331434				
7	2.07	0.843375	0.000584531				

791	0.34	-0.0343601	0.000178396		
792	0.19	-0.0354796	0.000107449		
793	0.51	-0.0365992	0.000260474	2021	
794	0.5	-0.0377188	0.000256242		
795	0.23	-0.0388383	0.000128111		
796	0.31	-0.0399579	0.000166768		
797	0.53	-0.0410774	0.000272139		
798	0.28	-0.042197	0.000153539		
799	0.06	-0.0433166	4.92341E-05		
800	0.25	-0.0444361	0.000140309		
801	-0.01	-0.0455557	1.69436E-05		
802	0.17	-0.0466752	0.000103254		
803	0.34	-0.0477948	0.000184798		
804	0.2	-0.0489144	0.000118617		
805	0.37	-0.0500339	0.000200161		
806	0.24	-0.0511535	0.000138745		
807	-0.02	-0.052273	1.53793E-05		

# Conclusión

Al final del día, utilizamos una variedad de métodos para conseguir nuestro resultado final. Nos dimos cuenta que para tener un mejor modelo, hay que aprender a identificar las anomalías en los datos para no tomarlas en cuenta, y que nuestro programa sea más eficiente. El promedio final del EAM es de 0.000146981.

# Fuentes

[www.inegi.org.mx](http://www.inegi.org.mx)

[https://www.w3schools.com/python/python\\_ml\\_train\\_test.asp](https://www.w3schools.com/python/python_ml_train_test.asp)

<https://medium.com/analytics-vidhya/how-to-remove-outliers-for-machine-learning-24620c4657e8>

<https://scikit-learn.org/stable/>

[https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8\\_525](https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_525)