

IXL  
Senior Developer  
Challenge

## **Documentación Javascript Echarts**

Carlos Alejandro Montiel Lorenzana

## Objetivos

- Crear una página web el cual permita seleccionar una columna de la data
- Analizar y procesar la data
- Utilizar Echarts Apache

## Planteamiento del Problema

Para este problema se tiene que estudiar el uso de la librería javascript Echarts, posteriormente analizar y procesar la data para posteriormente utilizar echarts para que se vea en un sitio web.

## Marco Teórico

Para analizar la data utilizaremos Python, con las librerías numpy y pandas, tambien utilizaremos Jupyter como interfaz grafica o IDE de python, en la siguiente imagen removemos todas las filas que tengan 'NaN' en todas las columnas

```
•[10]: #Remove NA where all data is missing
data.dropna(axis='rows',how='all')
```

Posterior a eso, llenamos todos los NaN restantes con el string '(en blanco)' para que tambien sean clasificados.

```
[59]: data.fillna('(en blanco)', inplace=True)
```

Luego, obtenemos todas las clases y contamos cuántas ocurrencias tiene cada una

```
#Get all the possible classes
seniority = data['Seniority Level'].value_counts(dropna=False)
position = data['Position'].value_counts(dropna=False)
related = data['Is Job Related to Innovation'].value_counts(dropna=False)
department = data['Department'].value_counts(dropna=False)
industry = data['Industry'].value_counts(dropna=False)
employee = data['Employee'].value_counts(dropna=False)
country = data['Country'].value_counts(dropna=False)
ternure = data['Tenure'].value_counts(dropna=False)
```

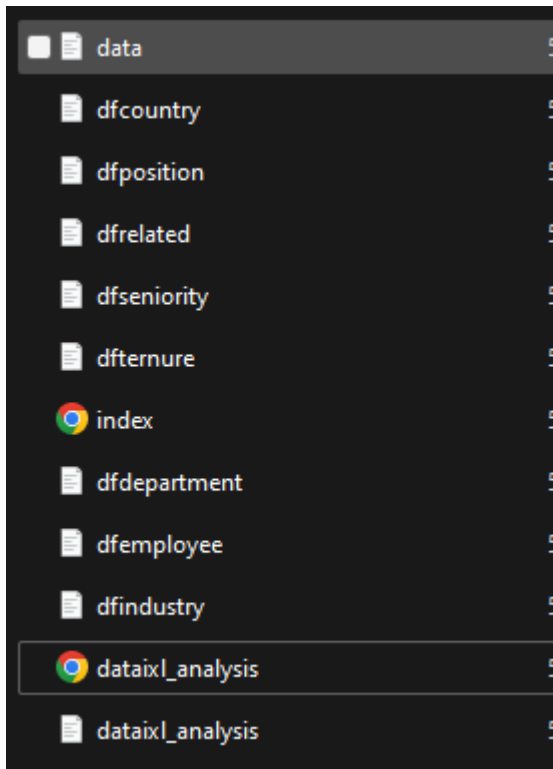
seniority

```
Seniority Level
(en blanco)      1223
Middle Management    205
Senior Management   181
Junior Management   146
Name: count, dtype: int64
```

Esta documentacion la puede encontrar en el repositorio, con el nombre de 'dataixl\_analysis.html'

Obteniendo ese valor, luego cada set de datos lo convertimos en un CSV diferente, cabe mencionar que esto podría ser un proceso automatizado si se tuviera un recurso Python en la nube.

En la estructura de datos, existen varios .csv que representan a la data clasificada por columna:



Para finalizar, con echarts se utilizó un selector de bootstrap el cual llama una función que asigna el nombre del dataset, esta llama a otra función que hace fetch de la información y la muestra.

Selección de columna:

```
<ul class="dropdown-menu" aria-labelledby="navbarDropdown">
  <li><a class="dropdown-item" href="#" onclick="fetchSeniority()">Seniority Level</a></li>
  <li><a class="dropdown-item" href="#" onclick="fetchPosition()">Position</a></li>
```

Función de activación:

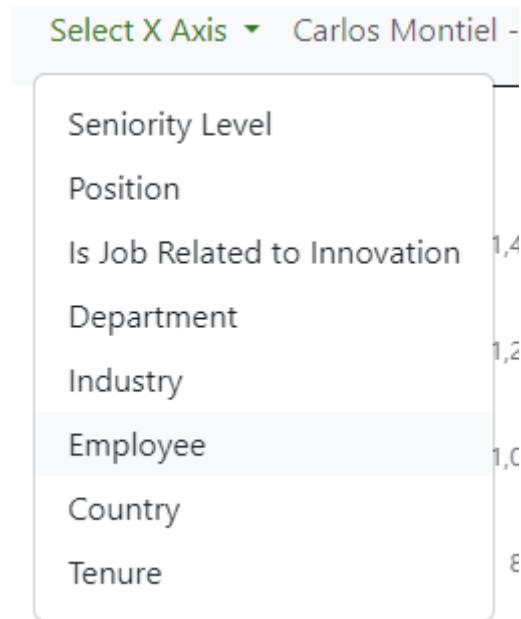
```
<script>
function fetchSeniority(){
  |   showData('dfseniority.csv')
}
function fetchPosition(){
  |   showData('dfposition.csv')
}
```

Funcion que permite mostrar la data en las echarts:

```
function showData(params){
  var dataframe = params;
  fetch(dataframe)
  .then(response => response.text())
  .then(data => {
    // Process CSV data
    var dataArray = data.split('\n').map(function (row) {
      return row.split(',');
    });
  });
}
```

Ejemplo:

- Click en <https://ixl.montiieldiamonds.com/>
- Seleccionar Columna



- Ver informacion

