UNIMINUTO
Corporación Universitaria Minuto de Dios
Educación de calidad al alcance de todos
Vigilada MinEducación

# Taller. Adquisición y adecuación de base de dato

Docente: Claudia Marcela Ospina Mosquera
Estudiante: Rafael Romario Roncancio Vinchery

Machine Learning NRC-878

UNIMINUTO
Corporación Universitaria Minuto de Dios
Educación de calidad al alcance de todos
Vigilada MinEducación

# Dataset Adult

Es un Dataset creado por Barry Becker en el cual extrajo realizó la extracción del la base de datos del censo de 1994. Se limpio la data pero aún cuenta con déficit en ella. Algunas de las condiciones de limpieza fueron: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0))

La data se recolectó en su momento para determinar si una persona gana más de 50.000 al año.

# Tabla de Variables

| Variable Name | Role | Type | Demographic | Description | Units | Missing Values |
|---|---|---|---|---|---|---|
| age | Feature | Integer | Age | N/A | | no |
| workclass | Feature | Categorical | Income | Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. | | yes |
| fnlwgt | Feature | Integer | | | | no |
| education | Feature | Categorical | Education Level | Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. | | no |
| education-num | Feature | Integer | Education Level | | | no |
| marital-status | Feature | Categorical | Other | Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. | | no |

# Tabla de Variables

| | | | | | |
|---|---|---|---|---|---|
| occupation | Feature | Categorical | Other | Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. | yes |
| relationship | Feature | Categorical | Other | Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. | no |
| race | Feature | Categorical | Race | White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. | no |
| sex | Feature | Binary | Sex | Female, Male. | no |
| capital-gain | Feature | Integer | | | no |
| capital-loss | Feature | Integer | | | no |
| hours-per-week | Feature | Integer | | | no |

# Tabla de Variables

| | | | | | |
|---|---|---|---|---|---|
| native-country | Feature | Categorical | Other | United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands. | yes |
| income | Target | Binary | Income | >50K, <=50K. | no |

Fuente: Adaptado de [UC Irvine Machine Learning Repository], disponible en https://archive.ics.uci.edu/dataset/2/adult

# Cargar data en objeto pandas

```
import json
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

#path de mi set de datos
file_path = "./adult.data"
nRowsRead = None
#encabezado de mis datos el header de mi tabla
COLUMNS = (
    'age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status',
    'occupation', 'relationship', 'race', 'sex',
    'capital-gain', 'capital-loss', 'hours-per-week', 'native-country', 'income')


#Objeto dataset que cargara mis datos,
#primer parametro recibe el archivo a trabajar
# delimiter=',' para que verifique la separación de los datos por ese simbolo la coma
#nrwos
#names=COLUMNS determino como se llamara cada una de las columnas para poder identificar
#encoding = "ISO-8859-1" lo uso en caso de caracteres especiales
file_data = pd.read_csv(file_path, delimiter=',', nrows=nRowsRead, names=COLUMNS, encoding = "ISO-8859-1")

file_data.head(12)
```

# visualización de data

| | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | income |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| **1** | 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| **2** | 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |
| **3** | 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| **4** | 28 | Private | 338409 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Wife | Black | Female | 0 | 0 | 40 | Cuba | <=50K |
| **5** | 37 | Private | 284582 | Masters | 14 | Married-civ-spouse | Exec-managerial | Wife | White | Female | 0 | 0 | 40 | United-States | <=50K |
| **6** | 49 | Private | 160187 | 9th | 5 | Married-spouse-absent | Other-service | Not-in-family | Black | Female | 0 | 0 | 16 | Jamaica | <=50K |
| **7** | 52 | Self-emp-not-inc | 209642 | HS-grad | 9 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 45 | United-States | >50K |
| **8** | 31 | Private | 45781 | Masters | 14 | Never-married | Prof-specialty | Not-in-family | White | Female | 14084 | 0 | 50 | United-States | >50K |
| **9** | 42 | Private | 159449 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 5178 | 0 | 40 | United-States | >50K |
| **10** | 37 | Private | 280464 | Some-college | 10 | Married-civ-spouse | Exec-managerial | Husband | Black | Male | 0 | 0 | 80 | United-States | >50K |
| **11** | 30 | State-gov | 141297 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Husband | Asian-Pac-Islander | Male | 0 | 0 | 40 | India | >50K |

# Data Cleaning

# 1. Búsqueda de datos faltantes

```
# Evaluación del tipo de variables por atributo (Variables categóricas y númericas)
file_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education-num   32561 non-null  int64
 5   marital-status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital-gain    32561 non-null  int64
 11  capital-loss    32561 non-null  int64
 12  hours-per-week  32561 non-null  int64
 13  native-country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

# Comprobando cantidad de datos faltantes (missing) en las variables

```
# Se contabilizan y muestran el número de datos pérdidos (nulos) para cada variable del Dataset
file_data.isnull().sum()
```

```
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

No existen datos en faltantes en null.

# 2. Columnas Irrelevantes

Todas las columnas pueden ser relevantes para análisis futuros, no existen constantes.

# 3. Filas Repetidas

```
print(f'tamaño de dataset antes de eliminar duplicados: {file_data.shape}')
file_data.drop_duplicates(inplace=True)
print(f'tamaño de dataset despues de eliminar duplicados: {file_data.shape}')
```

```
tamaño de dataset antes de eliminar duplicados: (32561, 15)
tamaño de dataset despues de eliminar duplicados: (32537, 15)
```

# 4. Outliers

## 4.1 Outliers tipo String

### Variable workclass

```
print(f'tamaño de dataset antes de eliminar workclass == ?: {file_data.shape}')
file_data = file_data[~file_data['workclass'].str.contains('\?')]
print(f'tamaño de dataset despues de eliminar workclass == ?: {file_data.shape}')
```

```
tamaño de dataset antes de eliminar workclass == ?: (32537, 15)
tamaño de dataset despues de eliminar workclass == ?: (30701, 15)
```

### Variable ocupation

```
print(f'tamaño de dataset antes de eliminar occupation == ?: {file_data.shape}')
file_data = file_data[~file_data['occupation'].str.contains('\?')]
print(f'tamaño de dataset despues de eliminar occupation == ?: {file_data.shape}')
```

```
tamaño de dataset antes de eliminar occupation == ?: (30701, 15)
tamaño de dataset despues de eliminar occupation == ?: (30694, 15)
```

### Variable "native-country"

```
print(f'tamaño de dataset antes de eliminar country == ?: {file_data.shape}')
#file_data = file_data[file_data['native-country']!='?']
file_data = file_data[~file_data['native-country'].str.contains('\?')]
print(f'tamaño de dataset despues de eliminar country == ?: {file_data.shape}')
```

```
tamaño de dataset antes de eliminar country == ?: (30694, 15)
tamaño de dataset despues de eliminar country == ?: (30139, 15)
```

Se encuentran valores en varias columnas con el valor ?, valor que en la tabla de variables no es una opción por ende debe tomarse como un valor faltante.

Columna: estado civil - marital-status data = [ "never-married", "married-civ-spouse", "divorced", "married-spouse-absent", "separated", "married-af-spouse" "widowed" ]

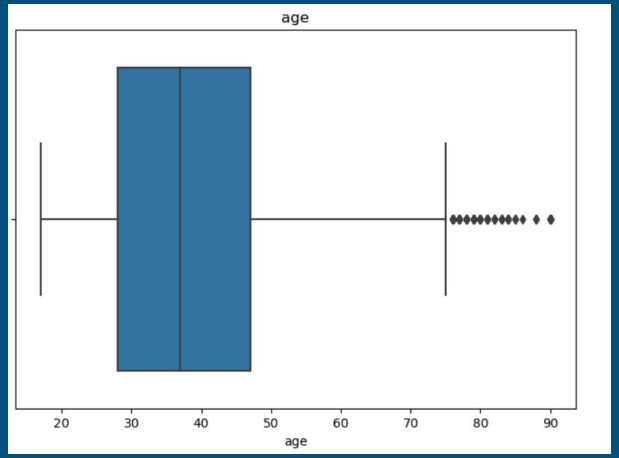
en español es datos : "nunca casado": 0, "cónyuge-civil-casado": 1, "divorciado": 2, "cónyuge-casado-ausente": 3, "separados": 4, "casado-de-cónyuge": 5, "viudo": 6 }

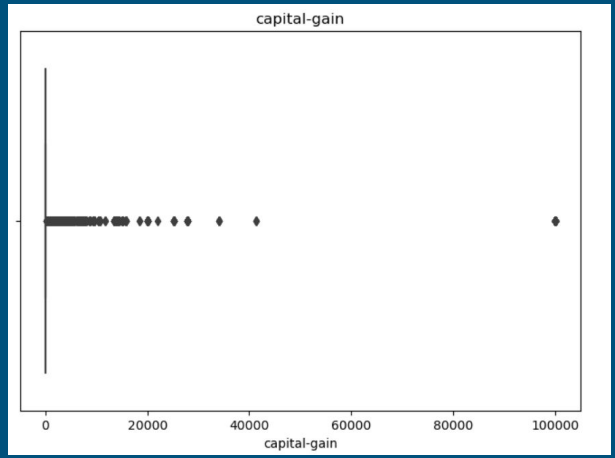
esos datos tienen que estar separados no es posible unificar debido a que el término estado civil encierra todos estos, inclusive en otros países existe la unión de hecho como por ejemplo en Colombia

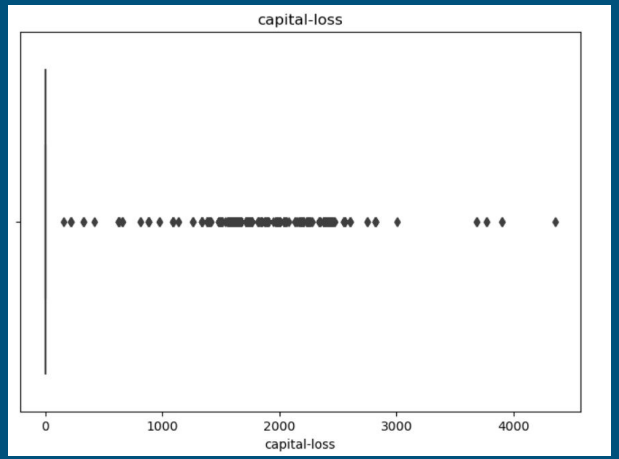
# 4.2 Outliers numéricos

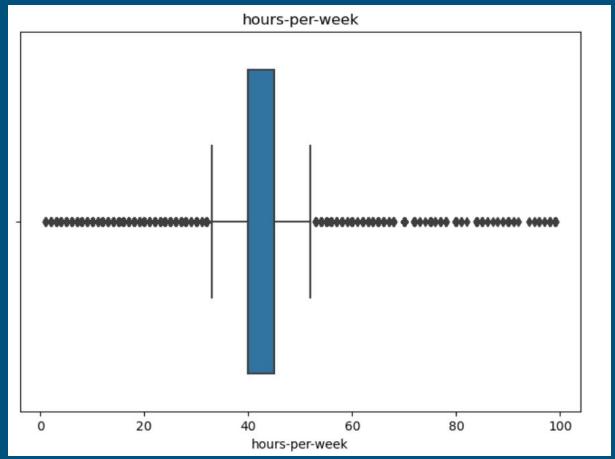
```
# Generar gráficas individuales pues las variables numéricas
# están en rangos diferentes

cols_num = ('age', 'capital-gain', 'capital-loss', 'hours-per-week')

fig, ax = plt.subplots(nrows=4, ncols=1, figsize=(8,30))
fig.subplots_adjust(hspace=0.5)

for i, col in enumerate(cols_num):
    sns.boxplot(x=col, data=file_data, ax=ax[i])
    ax[i].set_title(col)
```

No se encontraron datos outliers con valores numéricos



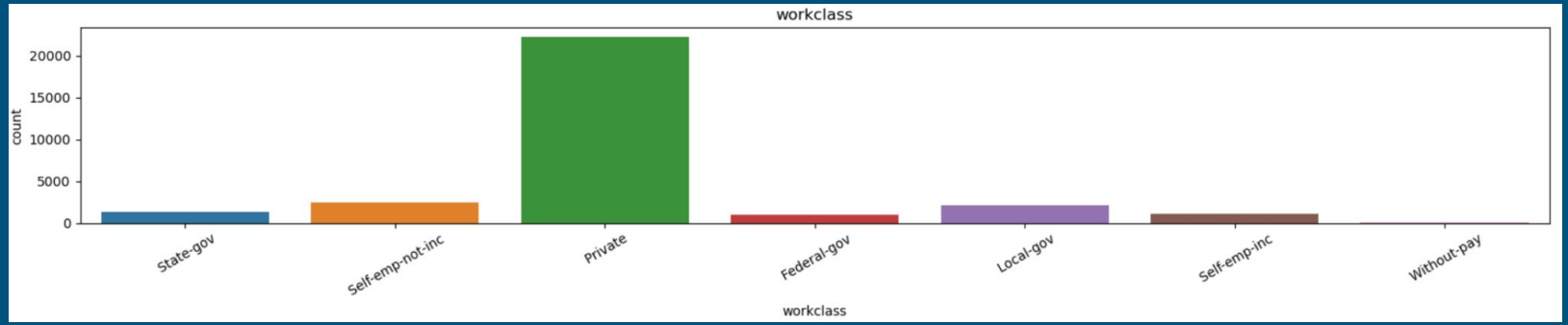



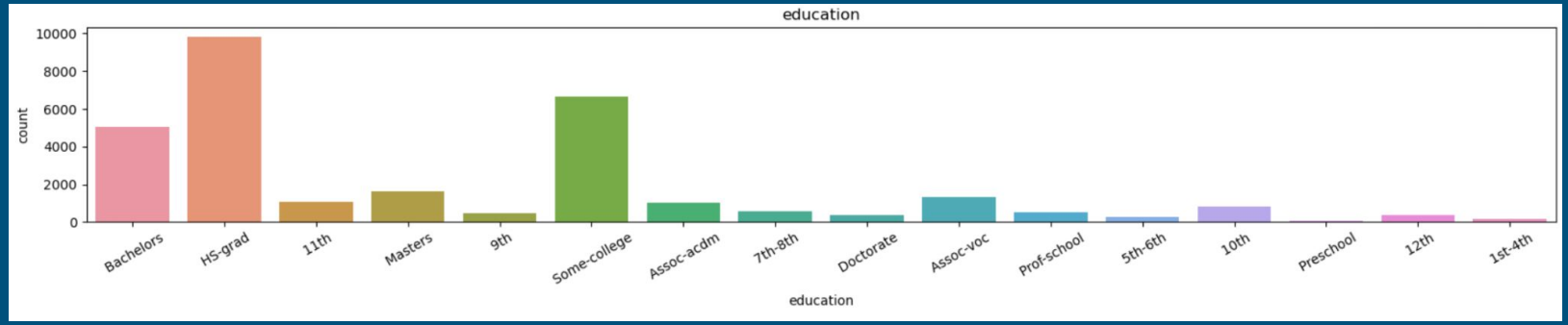



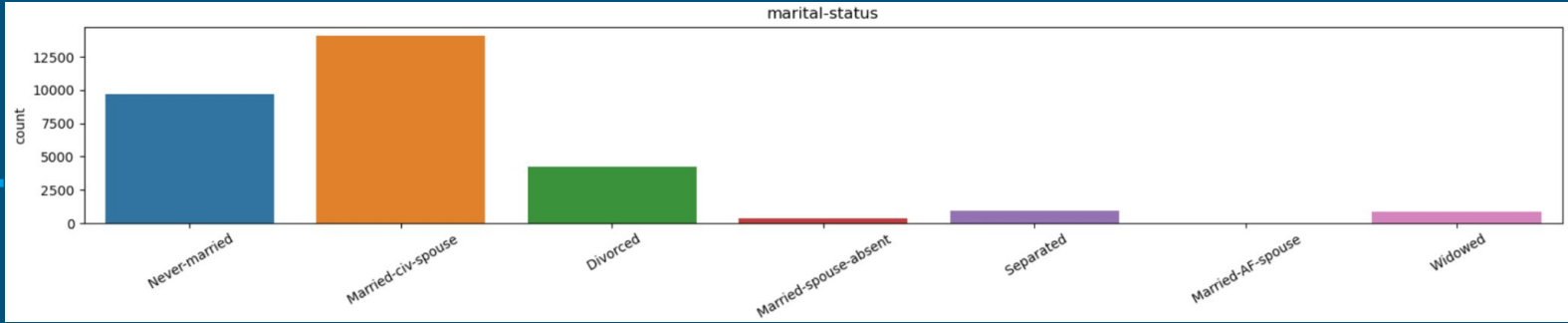



# 5 Errores tipográficos en variables categóricas

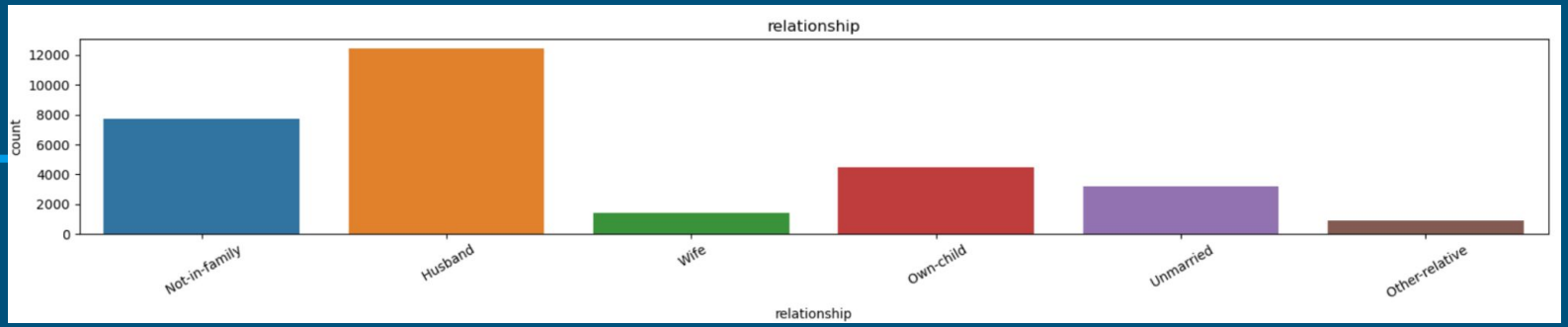
```
COLUMNS = (
    'workclass', 'education', 'marital-status',
    'occupation', 'relationship', 'race',
    'native-country', 'income')

fig, ax = plt.subplots(nrows=8, ncols=1, figsize=(20,40))
fig.subplots_adjust(hspace=1)

for i, col in enumerate(COLUMNS):
    sns.countplot(x=col, data=file_data, ax=ax[i])
    ax[i].set_title(col)
    ax[i].set_xticklabels(ax[i].get_xticklabels(),rotation=30)
```

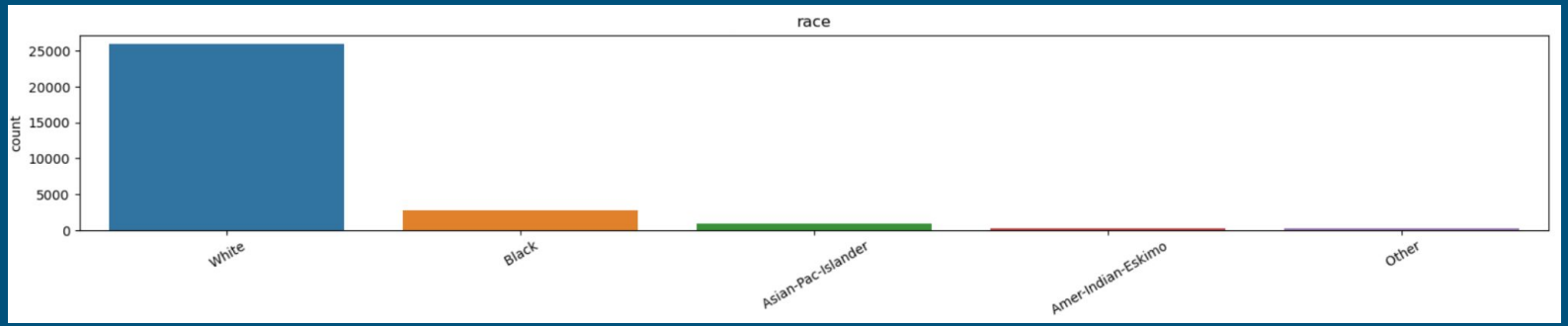

workclass
count
20000
15000
10000
5000
0
State-gov
Self-emp-not-inc
Private
Federal-gov
Local-gov
Self-emp-inc
Without-pay
workclass

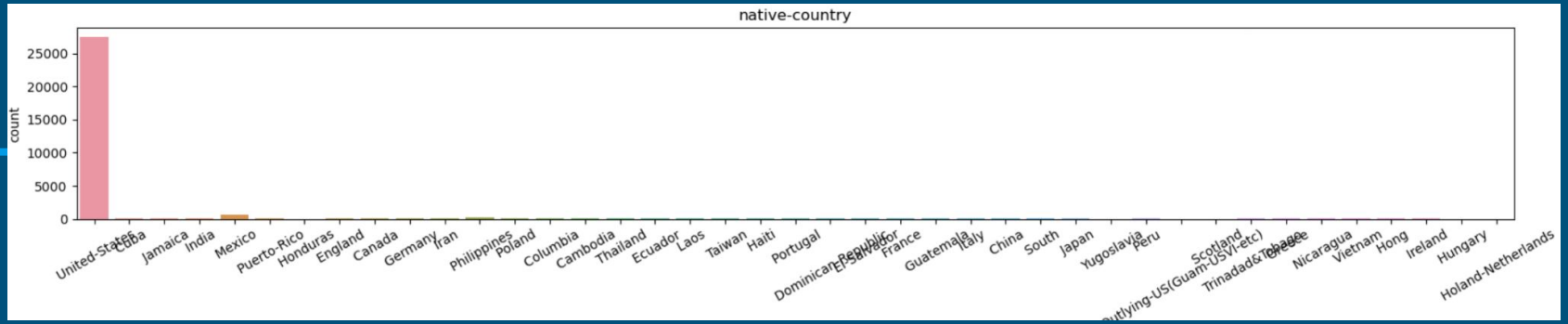

education
count
10000
8000
6000
4000
2000
0
Bachelors
HS-grad
11th
Masters
9th
Some-college
Assoc-acdm
7th-8th
Doctorate
Assoc-voc
Prof-school
5th-6th
10th
Preschool
12th
1st-4th
education

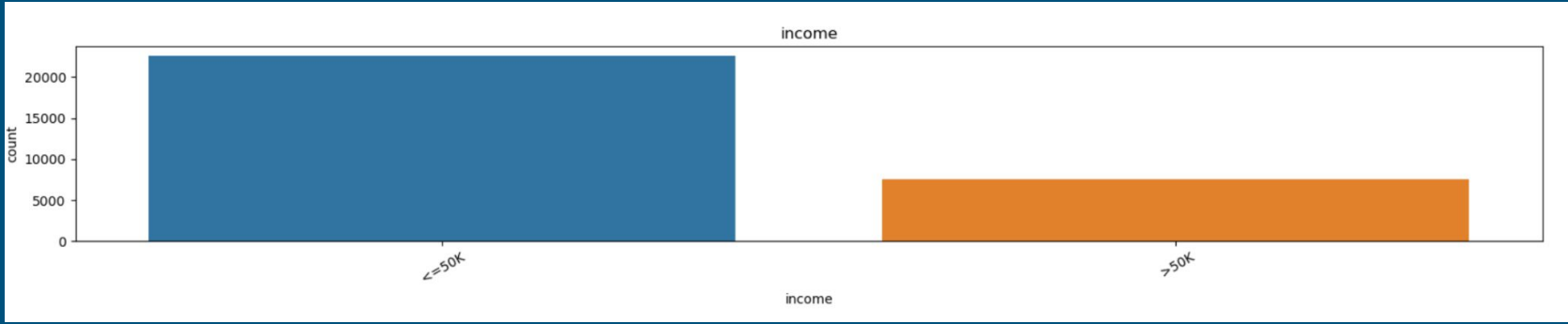

marital-status
count
12500
10000
7500
5000
2500
0
Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Married-AF-spouse
Widowed


occupation
count
4000
3000
2000
1000
0
Adm-clerical
Exec-managerial
Handlers-cleaners
Prof-specialty
Other-service
Sales
Transport-moving
Farming-fishing
Machine-op-inspct
Tech-support
Craft-repair
Protective-serv
Armed-Forces
Priv-house-serv
occupation


relationship
count
12000
10000
8000
6000
4000
2000
0
Not-in-family
Husband
Wife
Own-child
Unmarried
Other-relative
relationship


race
count
25000
20000
15000
10000
5000
0
White
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
Other


native-country
count
25000
20000
15000
10000
5000
0
United-States
Cuba
Jamaica
India
Mexico
Puerto-Rico
Honduras
England
Canada
Germany
Iran
Philippines
Poland
Columbia
Cambodia
Thailand
Ecuador
Laos
Taiwan
Haiti
Portugal
France
Guatemala
Italy
China
South
Japan
Yugoslavia
Peru
Scotland
Nicaragua
Vietnam
Hong
Ireland
Hungary
Holand-Netherlands


income
count
20000
15000
10000
5000
0
<=50K
>50K
income

Error u observación:
Se encontró una observación muchas valores tenían espacios en el inicio y final de los contenidos, los espacios en inicio o final son realmente irrelevantes por lo que se deben borrar para cuando se haga el proceso de variables de tendencia central para valores categóricas es necesario

```
COLUMNS = (
    'workclass', 'education', 'marital-status',
    'occupation', 'relationship', 'race',
    'native-country', 'income')
print(f'Tamaño de dataset antes de modificar: {file_data.shape}')
for column in COLUMNS:
    # Imprime el tamaño del DataFrame antes de realizar cambios
    #print(f'Tamaño de dataset antes de modificar native-country: {file_data.shape}')
    file_data[column] = file_data[column].str.strip().str.lower()
    #print(f'Tamaño de dataset después de modificar native-country: {file_data.shape}')
print(f'Tamaño de dataset despues de modificar: {file_data.shape}')
```

```
Tamaño de dataset antes de modificar: (30139, 15)
Tamaño de dataset despues de modificar: (30139, 15)
```

# Análisis de medidas de tendencia central

# Medidas de tendencia central de variables numéricas

```python
print(f'\n\nMedidad de tendencia central de valores numericos')

COLUMNS = ['age', 'capital-gain',
           'capital-loss', 'hours-per-week']

def get_measures_of_central_tendency(file_data, column:str ):
    mean = file_data[column].mean()
    median = file_data[column].median()
    mode = file_data[column].mode()[0]
    print(f"MEDIDA DE TENDENCIA CENTRAL DE {column}")
    print("La media es ", mean)
    print("La mediana es ", median)
    print("La moda es ", mode)
    print("-"*50)

for column in COLUMNS:
    print(get_measures_of_central_tendency(
            file_data=file_data,
            column=column)
    )
```

```
Medidad de tendencia central de valores numericos
MEDIDA DE TENDENCIA CENTRAL DE age
La media es  38.44172003052523
La mediana es  37.0
La moda es  36
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE capital-gain
La media es  1092.8412024287468
La mediana es  0.0
La moda es  0
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE capital-loss
La media es  88.43992833206144
La mediana es  0.0
La moda es  0
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE hours-per-week
La media es  40.934702544875414
La mediana es  40.0
La moda es  40
--------------------------------------------------
None
```

# Medidas de tendencia central de variables categóricas

## Designar variable categórica

```
COLUMNS = (
    'workclass', 'education', 'marital-status',
    'occupation', 'relationship', 'race',
    'native-country', 'income')

convention_column_dict = {}
for column in COLUMNS:
    attr_unique_list = file_data[column].unique().tolist()
    aux_convention_dict = {}
    for index, value in enumerate(attr_unique_list):
        aux_convention_dict[value] = index
    convention_column_dict[column] = aux_convention_dict

print("CONVENCIONES")
for column in COLUMNS:
    print(f'\nColumn: {column}')
    __json_data = json.dumps(convention_column_dict[column], indent=4)
    print(f'data:{__json_data}')
```

debido al código estos valores generan dependiendo del orden en el que se encuentren en la base de datos

```
Column: workclass
data:{
    "state-gov": 0,
    "self-emp-not-inc": 1,
    "private": 2,
    "federal-gov": 3,
    "local-gov": 4,
    "self-emp-inc": 5,
    "without-pay": 6
}

Column: education
data:{
    "bachelors": 0,
    "hs-grad": 1,
    "11th": 2,
    "masters": 3,
    "9th": 4,
    "some-college": 5,
    "assoc-acdm": 6,
    "7th-8th": 7,
    "doctorate": 8,
    "assoc-voc": 9,
    "prof-school": 10,
    "5th-6th": 11,
    "10th": 12,
    "preschool": 13,
    "12th": 14,
    "1st-4th": 15
}
```

```
Column: marital-status
data:{
    "never-married": 0,
    "married-civ-spouse": 1,
    "divorced": 2,
    "married-spouse-absent": 3,
    "separated": 4,
    "married-af-spouse": 5,
    "widowed": 6
}

Column: occupation
data:{
    "adm-clerical": 0,
    "exec-managerial": 1,
    "handlers-cleaners": 2,
    "prof-specialty": 3,
    "other-service": 4,
    "sales": 5,
    "transport-moving": 6,
    "farming-fishing": 7,
    "machine-op-inspct": 8,
    "tech-support": 9,
    "craft-repair": 10,
    "protective-serv": 11,
    "armed-forces": 12,
    "priv-house-serv": 13
}
```

```
Column: relationship
data:{
    "not-in-family": 0,
    "husband": 1,
    "wife": 2,
    "own-child": 3,
    "unmarried": 4,
    "other-relative": 5
}

Column: race
data:{
    "white": 0,
    "black": 1,
    "asian-pac-islander": 2,
    "amer-indian-eskimo": 3,
    "other": 4
}
```

```
Column: income
data:{
    "<=50k": 0,
    ">50k": 1
}
```

```
Column: native-country
data:{
    "united-states": 0,
    "cuba": 1,
    "jamaica": 2,
    "india": 3,
    "mexico": 4,
    "puerto-rico": 5,
    "honduras": 6,
    "england": 7,
    "canada": 8,
    "germany": 9,
    "iran": 10,
    "philippines": 11,
    "poland": 12,
    "columbia": 13,
    "cambodia": 14,
    "thailand": 15,
    "ecuador": 16,
    "laos": 17,
    "taiwan": 18,
    "haiti": 19,
    "portugal": 20,
    "dominican-republic": 21,
    "el-salvador": 22,
    "france": 23,
    "guatemala": 24,
    "italy": 25,
    "china": 26,
    "south": 27,
    "japan": 28,
    "yugoslavia": 29,
    "peru": 30,
    "outlying-us(guam-usvi-etc)": 31,
    "scotland": 32,
    "trinadad&tobago": 33,
    "greece": 34,
    "nicaragua": 35,
    "vietnam": 36,
    "hong": 37,
    "ireland": 38,
    "hungary": 39,
    "holand-netherlands": 40
}
```

# Agregar variable categórica en tabla

```
for column in COLUMNS:
    __aux_dict = convention_column_dict[column]
    file_data[f'convention_{column}'] = file_data[column].map(__aux_dict)

print(file_data)
```

| | convention_marital-status | convention_occupation | \ |
|---|---|---|---|
| 0 | 0 | 0 | |
| 1 | 1 | 1 | |
| 2 | 2 | 2 | |
| 3 | 1 | 2 | |
| 4 | 1 | 3 | |
| ... | ... | ... | |
| 32556 | 1 | 9 | |
| 32557 | 1 | 8 | |
| 32558 | 6 | 0 | |
| 32559 | 0 | 0 | |
| 32560 | 1 | 1 | |

| convention_workclass | convention_education | \ |
|---|---|---|
| 0 | 0 | |
| 1 | 0 | |
| 2 | 1 | |
| 2 | 2 | |
| 2 | 0 | |
| ... | ... | |
| 2 | 6 | |
| 2 | 1 | |
| 2 | 1 | |
| 2 | 1 | |
| 5 | 1 | |

| | convention_relationship | convention_race | convention_native-country | \ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 1 | 1 | 0 | 0 | |
| 2 | 0 | 0 | 0 | |
| 3 | 1 | 1 | 0 | |
| 4 | 2 | 1 | 1 | |
| ... | ... | ... | ... | |
| 32556 | 2 | 0 | 0 | |
| 32557 | 1 | 0 | 0 | |
| 32558 | 4 | 0 | 0 | |
| 32559 | 3 | 0 | 0 | |
| 32560 | 2 | 0 | 0 | |

| | convention_income |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| ... | ... |
| 32556 | 0 |
| 32557 | 1 |
| 32558 | 0 |
| 32559 | 0 |
| 32560 | 1 |

# medidas de tendencia central

```python
COLUMNS = (
    'convention_workclass', 'convention_education',
    'convention_marital-status',
    'convention_occupation', 'convention_relationship',
    'convention_race',
    'convention_native-country', 'convention_income')


def get_measures_of_central_tendency(file_data, column:str ):
    mean = file_data[column].mean()
    median = file_data[column].median()
    mode = file_data[column].mode()[0]
    print(f"MEDIDA DE TENDENCIA CENTRAL DE {column}")
    #print("La media es ", mean)
    #print("La mediana es ", median)
    print(f"La moda es {mode}, favor revisar CONVENCIONES" )
    print("-"*50)

for column in COLUMNS:
    print(get_measures_of_central_tendency(
            file_data=file_data,
            column=column)
    )
```

```
MEDIDA DE TENDENCIA CENTRAL DE convention_workclass
La moda es 2, favor revisar CONVENCIONES
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE convention_education
La moda es 1, favor revisar CONVENCIONES
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE convention_marital-status
La moda es 1, favor revisar CONVENCIONES
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE convention_occupation
La moda es 3, favor revisar CONVENCIONES
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE convention_relationship
La moda es 1, favor revisar CONVENCIONES
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE convention_race
La moda es 0, favor revisar CONVENCIONES
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE convention_native-country
La moda es 0, favor revisar CONVENCIONES
--------------------------------------------------
None
MEDIDA DE TENDENCIA CENTRAL DE convention_income
La moda es 0, favor revisar CONVENCIONES
--------------------------------------------------
None
```

# Conclusiones

1. En el data cleaning los atributos o columnas de tipo categoría muchas veces tienen en la cadena de texto datos que tiene la misma intención, se diferencian levemente con espacios, mayúsculas y minúsculas, es indispensable darles forma para poder hacer un análisis óptimo por medio los scripts que ejecutan esa data.
2. Los datos outliers son aquellos que no tiene lógica pero no se encontró ninguno que fuese imposible a pesar que la persona trabajaba más horas en la semana era de aproximadamente 100 horas calculo que nos dice que trabaja 14.28 horas diarias número que es posible. se podría alegar que legalmente es injusto o físicamente desgastante; pero para los datos es factible y por ende estará en la muestra para el análisis
3. La mayoría de las personas en la base de datos no logra los $50.000 al año.
4. Para el análisis matemático como las medidas de tendencia central de variables cualitativas es necesario designar una variable categórica (convenciones key:str(category), value:entero)
5. En el Ejercicio de data cleaning es inevitable pensar en eliminar algunos registros pero lo más importante es el análisis y aprender cuando se debe hacer y en qué casos darle forma a los datos sin afectar el estudio que se quiere hacer con ellos.

# Referencias

pandas.pydata.org. (2024, Enero 20). pandas.api.extensions.ExtensionArray.tolist. pandas. https://pandas.pydata.org/docs/reference/api/pandas.api.extensions.ExtensionArray.tolist.html

Becker, B & Kohavi, R. (1996, Abril 30). Adult. UC Irvine Machine Learning Repository. https://archive.ics.uci.edu/dataset/2/adult