

Micro Retailer Mit Lab Evidencia 2

EQUIPO 2

El primer paso para la elaboración del código fue importar las librerías requeridas para realizar la limpieza de datos, outliers, y posterior visualización de estos.

```
▶ #Instalamos librería para análisis de variables categóricas
%pip install funpymodeling
```

```
[90] #Importamos las librerias pandas, numpy y matplotlib respectivamente
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from funpymodeling.exploratory import freq_tbl
```

Luego, cargamos el archivo de microretailer_mit_lab al código para así empezar a trabajar con él

```
[91] #Cargar archivo excel desde equipo
from google.colab import files
files.upload()
```

```
▶ #Carga desde un archivo .csv sin indice
Retailer= pd.read_excel('microretailer_mit_lift_lab.xlsx')
```

Después, empezamos con la identificación y sustitución de valores nulos de todo el dataframe utilizando los métodos de “bfill” o “ffill”

```
[95] #Identificar valores nulos por columna
valores_nulos=data1.isnull().sum()
valores_nulos
```

```
[96] #Realizamos una copia del dataframe
data2=data1.copy()
```

```
[97] #Utilizamos un metodo con el que sustituimos los valores nulos que esten por detras de valores no nulos por ese mismo valor
data2= data1.fillna(method="bfill")
data2
```

```
[98] #Corroboramos valores nulos
valores_nulos=data2.isnull().sum()
valores_nulos
```

```
[99] #Debido a que todavia nos quedan unos cuantos valores nulos en el dataframe, utilizamos el metodo "ffill" para cambiar los valores nulos que esten delante de un valore no nulo
data3= data2.fillna(method="ffill")
data3
```

```
[100] #Corroboramos valores nulos
valores_nulos=data3.isnull().sum()
valores_nulos
```

Para el reemplazo de los outliers, primero filtramos todos los valores cualitativos y los separamos de los valores cuantitativos.

```
[103] #Realizamos un filtro por filas donde se incluyan unicamente los valores cuantitativos
cuantitativas=data5.iloc[:,[13,14,16,21,22,29,31,33,42,49,51,56,57,58,59,60,78,90]] #filas no consecutivas
cuantitativas

[104] #Realizamos un filtro por filas donde se incluyan unicamente los valores cualitativas
cualitativas=data5.iloc[:,[0,1,2,3,4,5,6,7,8,9,10,11,12,15,17,18,19,20,23,24,25,26,27,28,30,32,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,50,52,53,54,55,61,62,63,64,65,66,67,68,69,
cualitativas
```

Aplicamos desviación estándar a los valores cuantitativos y encontramos los outliers del dataframe

```
[106] #Aplicamos desviación estandar para encontrar valores extremos
y=cuantitativas
Limite_Superior= y.mean() +3*y.std()
Limite_Inferior= y.mean() -3*y.std()
print("Limite superior permitido", Limite_Superior)
print("Limite inferior permitido", Limite_Inferior)
```

```
[107] #Encontramos Outliers del Dataframe
outliers= cuantitativas[(y>Limite_Superior)|(y<Limite_Inferior)]
outliers
```

Una vez que encontramos los outliers del dataframe, obtenemos los valores que están limpios, y convertimos todos los valores atípicos a valores nulos para poder reemplazar esos valores con la media de cada columna.

```
[108] #Obtenemos datos limpios
data5= cuantitativas[(y<=Limite_Superior)&(y>=Limite_Inferior)]
data5
```

```
[109] #Revisamos valores atípicos (nulos) del dataframe4
valores_nulos=data5.isnull().sum()
valores_nulos
```

```
[110] #Reemplazamos valores atípicos (nulos) del dataframe con "mean"
#Realizamos una copia del dataframe
data_clean=data5.copy()
data_clean=data_clean.fillna(round(data4.mean(),1))
data_clean
```

```
▶ #Corroboramos valores nulos del dataframe LIMPIO
valores_nulos=data_clean.isnull().sum()
valores_nulos
```

Para terminar con la limpieza de datos, unimos las columnas cuantitativas y cualitativas anteriormente separadas y descargamos el dataframe limpio, sin nulos ni outliers.

```
[112] #Unimos las columnas cuantitativas y cualitativas en un mismo dataframe
      Datos_limpios = pd.concat([cualitativas, data_clean], axis=1)
      Datos_limpios
```

Descargamos el dataframe de los valores limpios

```
[113] #Convertir Dataframe a CSV
      Datos_limpios.to_csv("Datos_limpios_Micro_retailer.csv")
```

```
[114] #descargar archivo filtrado en csv
      from google.colab import files

      #files.download("Datos_limpios_Micro_retailer.csv")
```

Para comenzar con la visualización de las variables que creemos que son útiles, separamos las columnas que deseamos y obtenemos un nuevo dataframe únicamente con esas variables para luego descargarlo.

Tratamiento de variables (filtrado)

```
[115] #Realizamos otro filtro por filas donde se incluyan las variables que se van a utilizar para la visualización y extracción
      variables_buenas=data4.iloc[:, [9,23,24,26,27,35,47,52,53,84,85,89,91,98,]]
      variables_buenas
```

```
▶ #Obtengo un analisis univariado de las variables categóricas
   freq_tbl(variables_buenas)
```

Descargamos el dataframe de los valores seleccionados

```
[117] #Convertir Dataframe a CSV
      variables_buenas.to_csv("variables_buenas_Micro_retailer.csv")
```

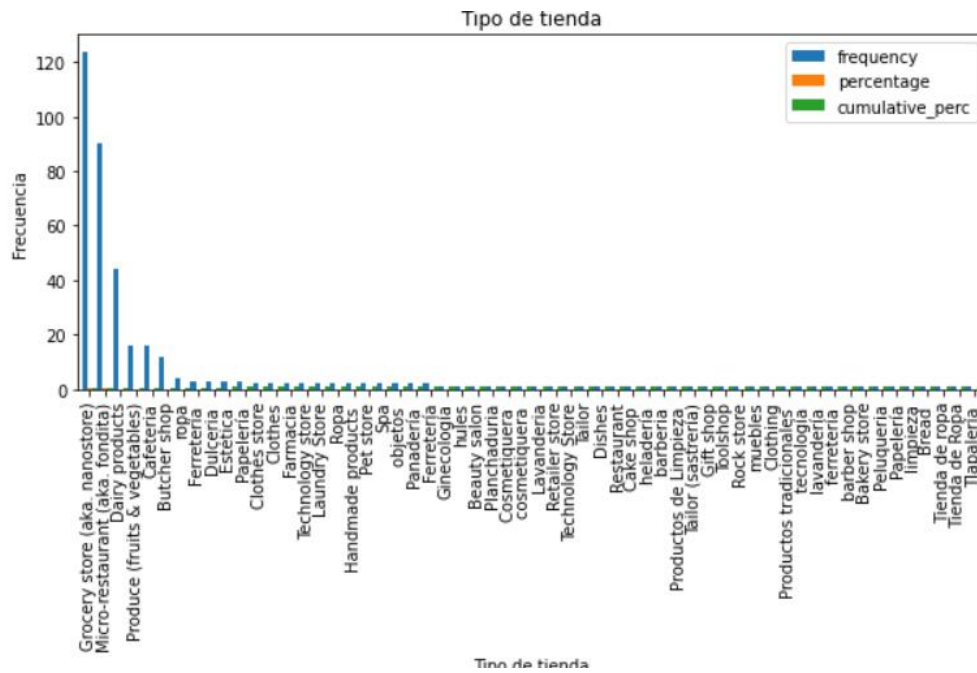
```
[118] #descargar archivo filtrado en csv
      from google.colab import files

      #files.download("variables_buenas_Micro_retailer.csv")
```

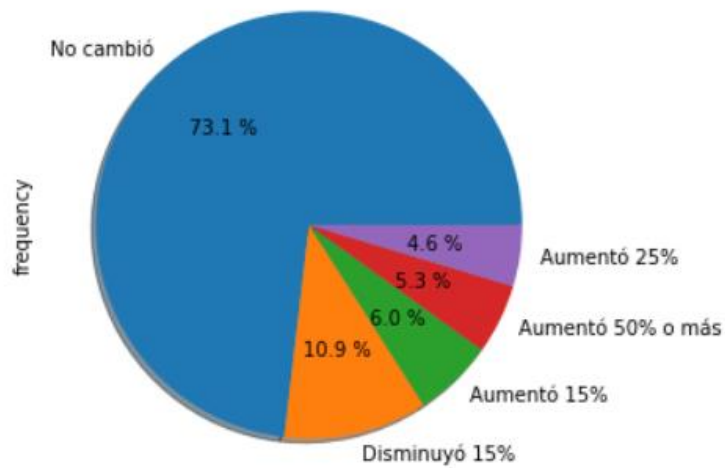
En nuestro caso, para la visualización se eligieron 14 variables diferentes que creímos útiles en nuestro proyecto.

A continuación, se muestran cada columna la variable seleccionada con su respectiva gráfica.

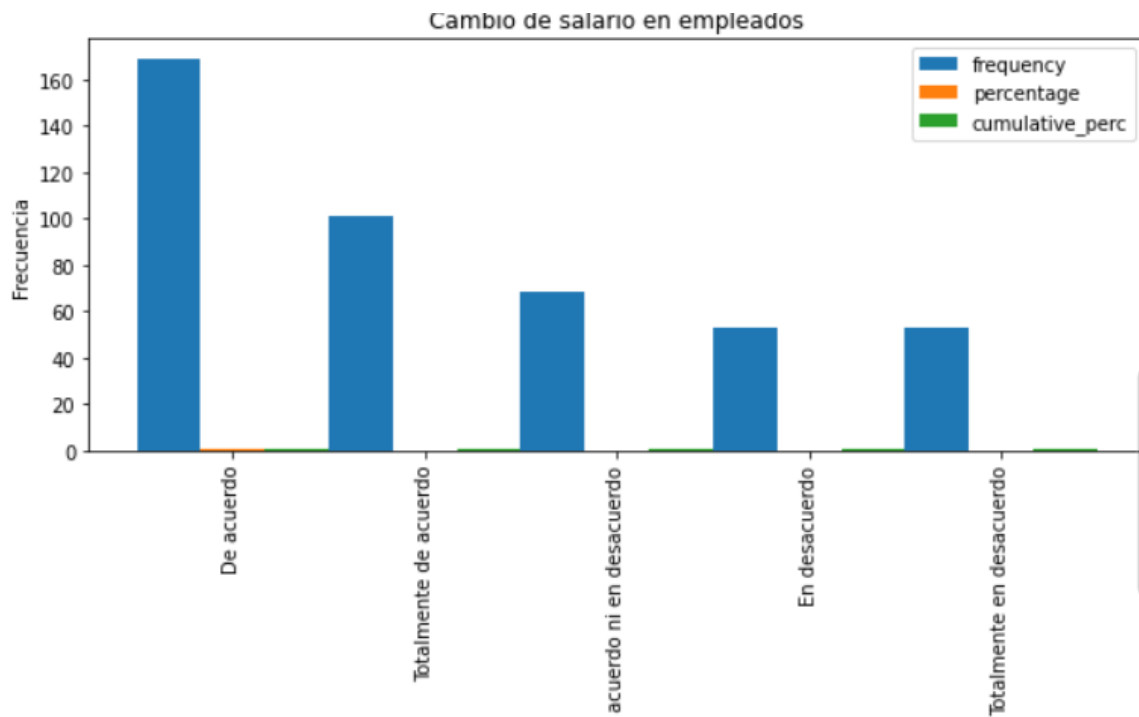
232_type_of_store



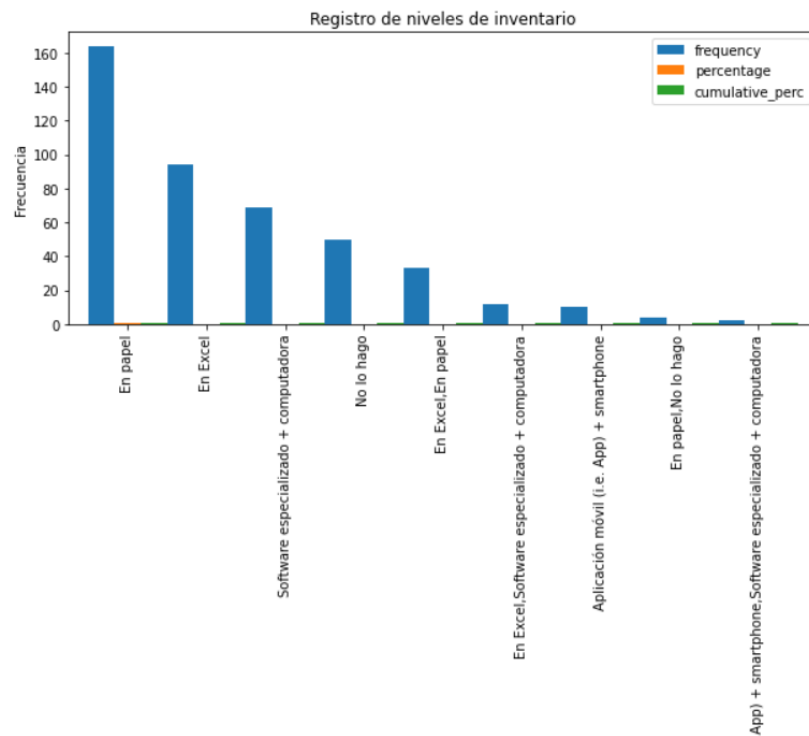
5_change_store_space_last_year



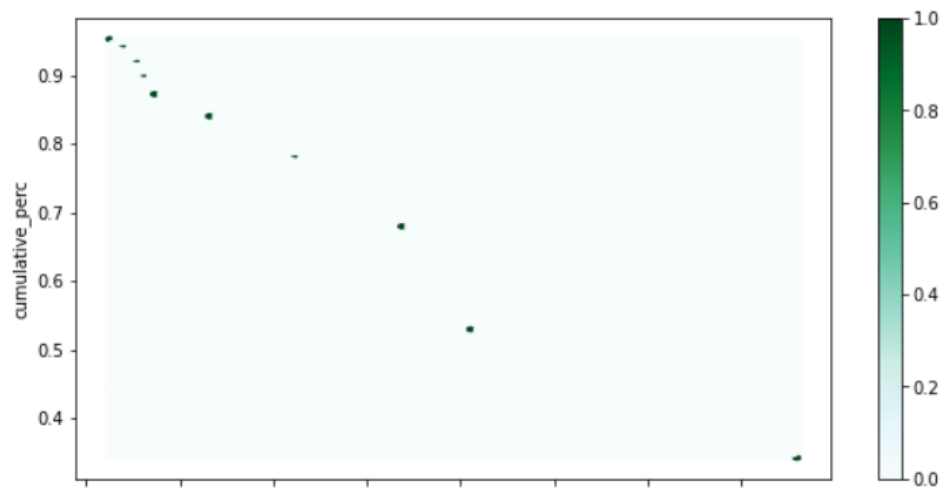
6_change_employees_average_salary_last_year



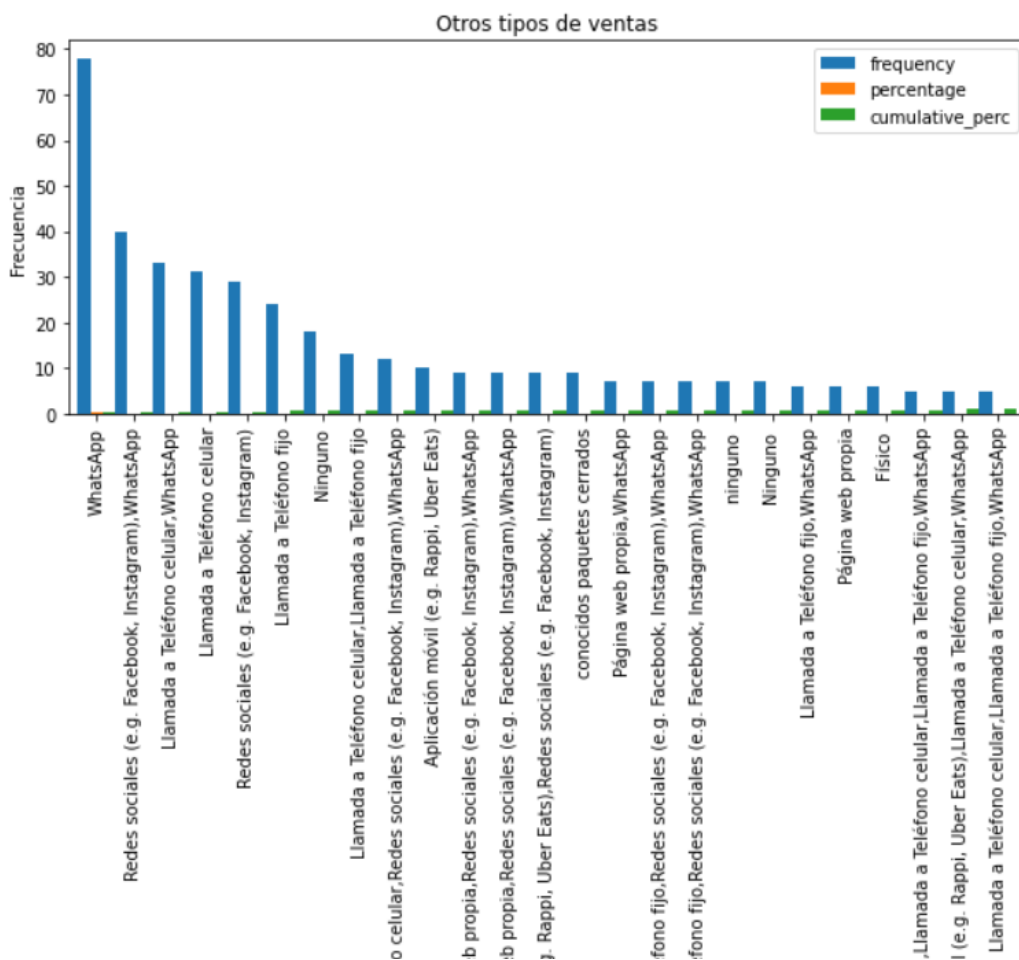
49_inventory_records



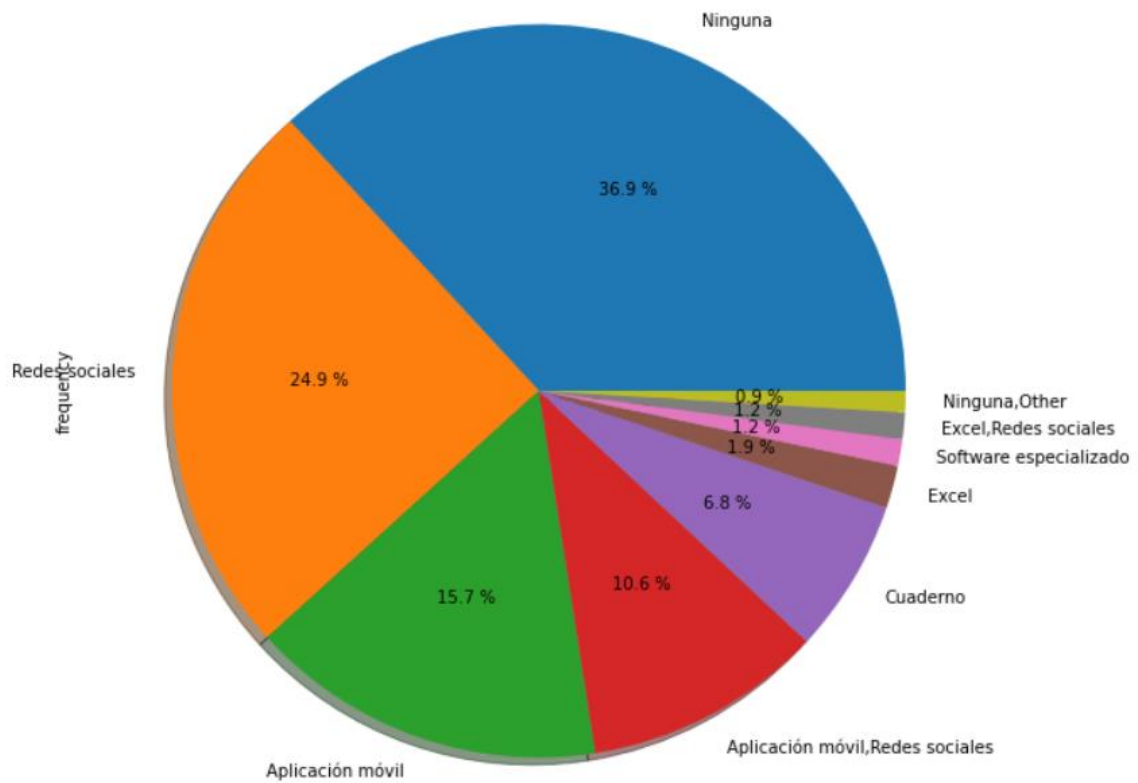
18_sales_records



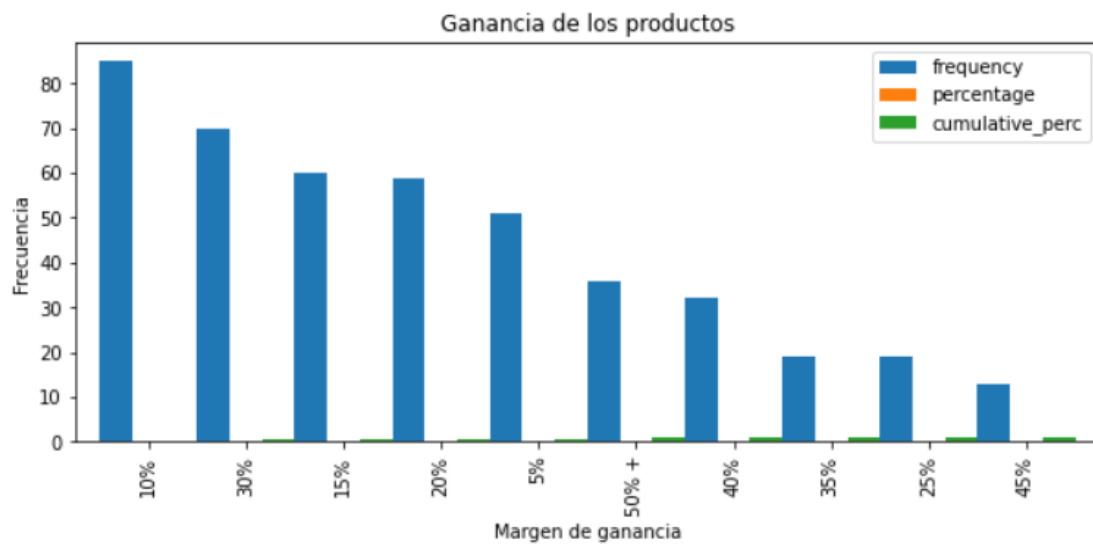
210_sales_channels



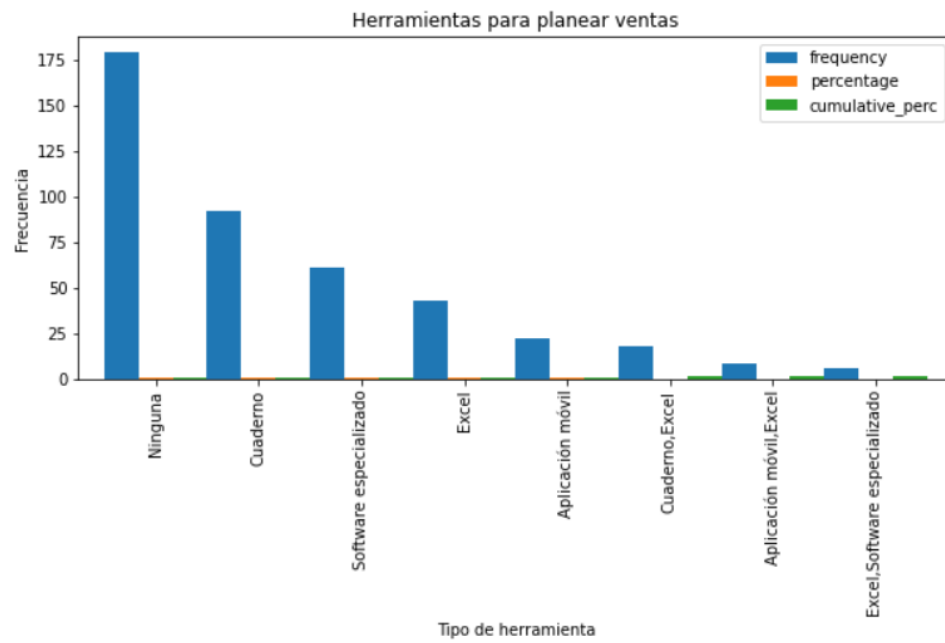
214_customer_relationship_tools



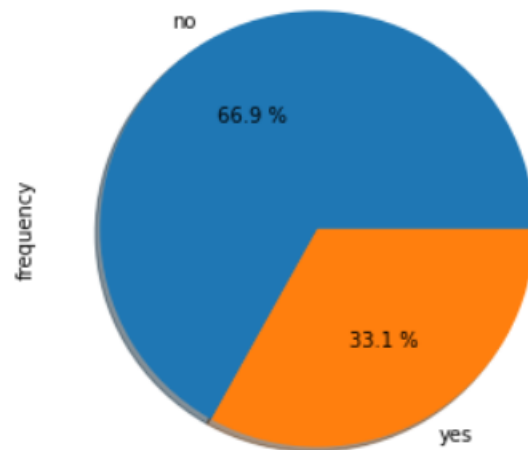
38_average_margin_profits



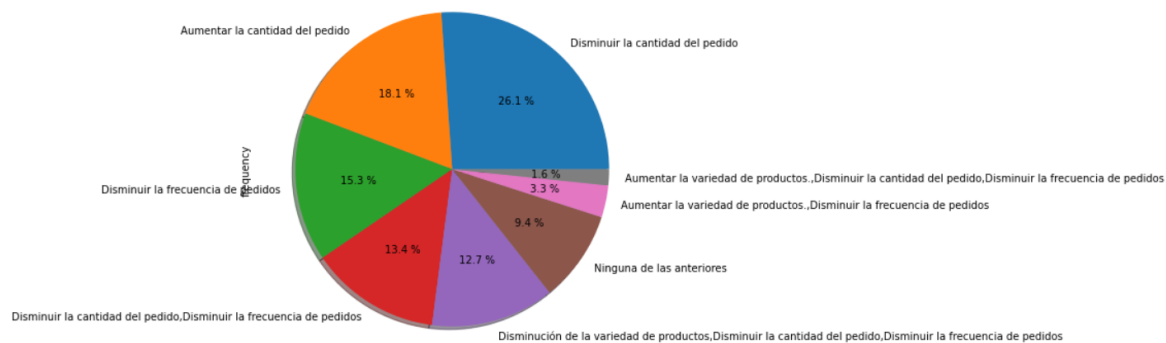
193_sales_planning_tools



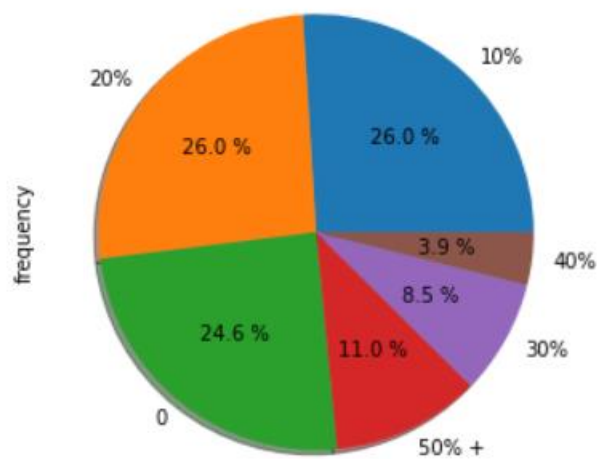
56_procurement_changes_pandemic



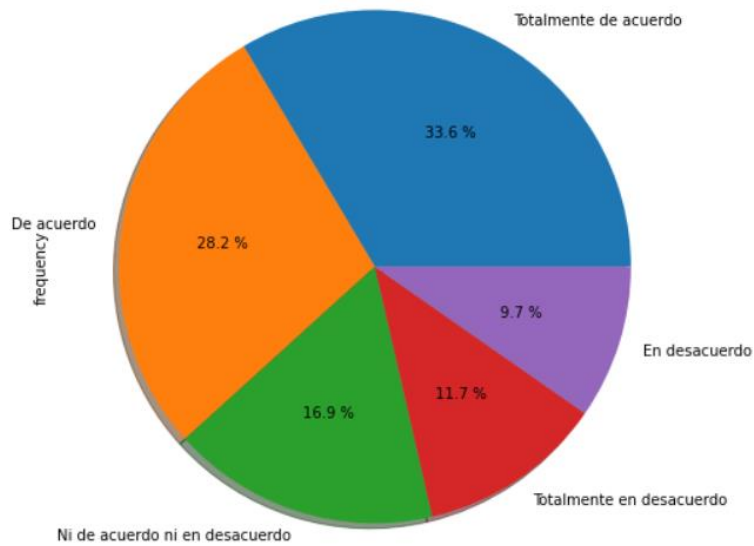
279_changes_procurement_pandemic



162_inventory_extra_space



249_technology_training



263_technology_secure_sales

