```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "K6aUpDf7KC_p"
   },
   "source": [
    "# Comprensión de los Datos"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "id": "Xhds4njPKC_u"
   },
   "outputs": [],
   "source": [
    "#importa librerías\n",
    "import pandas as pd"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "JdYf-8XyKC_x"
   },
   "source": [
    "# Descripción de Variables"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "Y7EVetS8KC_y"
   },
   "source": [
    "\n",
    "Pclass Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd): Categórica Nominal \n",
    "survival Survival (0 = No; 1 = Yes)  \n",
    "name Name  \n",
    "sex Sex  \n",
    "age Age  \n",
    "sibsp Number of Siblings/Spouses Aboard  \n",
    "parch Number of Parents/Children Aboard  \n",
    "ticket Ticket Number  \n",
    "fare Passenger Fare (British pound)  \n",
    "cabin Cabin  \n",
    "embarked Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)  \n",
    "boat Lifeboat  \n",
    "body Body Identification Number  \n",
    "home.dest Home/Destination"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "B68geAa3KC_z"
   },
   "source": [
    "**Ejemplo:** Crear un objeto DataFrame con base en un archivo .csv (poner \"df = \")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {
    "id": "dLr3UCdTKC_1"
   },
   "outputs": [],
   "source": [
    "#lee archivo csv\n",
    "df = pd.read_csv(\"diabetes.csv\")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "(768, 9)"
      ]
     },
     "execution_count": 5,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Usa función shape para revisar el total de renglones y columnas\n",
    "df.shape"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "metadata": {
    "id": "NLa_otfWKC_1",
    "outputId": "8ce7dd9a-5f3d-4fe7-d34f-ee8a02afda94",
    "scrolled": true
   },
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>Pregnancies</th>\n",
       "      <th>Glucose</th>\n",
        "      <th>BloodPressure</th>\n",
        "      <th>SkinThickness</th>\n",
        "      <th>Insulin</th>\n",
        "      <th>BMI</th>\n",
        "      <th>DiabetesPedigreeFunction</th>\n",
        "      <th>Age</th>\n",
        "      <th>Outcome</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>0</th>\n",
        "      <td>6</td>\n",
        "      <td>148</td>\n",
        "      <td>72</td>\n",
        "      <td>35</td>\n",
        "      <td>0</td>\n",
        "      <td>33.6</td>\n",
        "      <td>0.627</td>\n",
        "      <td>50</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>1</th>\n",
        "      <td>1</td>\n",
        "      <td>85</td>\n",
        "      <td>66</td>\n",
        "      <td>29</td>\n",
        "      <td>0</td>\n",
        "      <td>26.6</td>\n",
        "      <td>0.351</td>\n",
        "      <td>31</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>2</th>\n",
        "      <td>8</td>\n",
        "      <td>183</td>\n",
        "      <td>64</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "      <td>23.3</td>\n",
        "      <td>0.672</td>\n",
        "      <td>32</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>3</th>\n",
        "      <td>1</td>\n",
        "      <td>89</td>\n",
        "      <td>66</td>\n",
        "      <td>23</td>\n",
        "      <td>94</td>\n",
        "      <td>28.1</td>\n",
        "      <td>0.167</td>\n",
        "      <td>21</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>4</th>\n",
        "      <td>0</td>\n",
        "      <td>137</td>\n",
        "      <td>40</td>\n",
        "      <td>35</td>\n",
        "      <td>168</td>\n",
       "      <td>43.1</td>\n",
       "      <td>2.288</td>\n",
       "      <td>33</td>\n",
       "      <td>1</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \\\n",
       "0            6      148             72             35        0  33.6   \n",
       "1            1       85             66             29        0  26.6   \n",
       "2            8      183             64              0        0  23.3   \n",
       "3            1       89             66             23       94  28.1   \n",
       "4            0      137             40             35      168  43.1   \n",
       "\n",
       "   DiabetesPedigreeFunction  Age  Outcome  \n",
       "0                     0.627   50        1  \n",
       "1                     0.351   31        0  \n",
       "2                     0.672   32        1  \n",
       "3                     0.167   21        0  \n",
       "4                     2.288   33        1  "
      ]
     },
     "execution_count": 6,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Revisa los primeros 5 renglones del dataset usando la función head()\n",
    "df.head()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>Pregnancies</th>\n",
       "      <th>Glucose</th>\n",
       "      <th>BloodPressure</th>\n",
       "      <th>SkinThickness</th>\n",
    "      <th>Insulin</th>\n",
    "      <th>BMI</th>\n",
    "      <th>DiabetesPedigreeFunction</th>\n",
    "      <th>Age</th>\n",
    "      <th>Outcome</th>\n",
    "    </tr>\n",
    "  </thead>\n",
    "  <tbody>\n",
    "    <tr>\n",
    "      <th>763</th>\n",
    "      <td>10</td>\n",
    "      <td>101</td>\n",
    "      <td>76</td>\n",
    "      <td>48</td>\n",
    "      <td>180</td>\n",
    "      <td>32.9</td>\n",
    "      <td>0.171</td>\n",
    "      <td>63</td>\n",
    "      <td>0</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>764</th>\n",
    "      <td>2</td>\n",
    "      <td>122</td>\n",
    "      <td>70</td>\n",
    "      <td>27</td>\n",
    "      <td>0</td>\n",
    "      <td>36.8</td>\n",
    "      <td>0.340</td>\n",
    "      <td>27</td>\n",
    "      <td>0</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>765</th>\n",
    "      <td>5</td>\n",
    "      <td>121</td>\n",
    "      <td>72</td>\n",
    "      <td>23</td>\n",
    "      <td>112</td>\n",
    "      <td>26.2</td>\n",
    "      <td>0.245</td>\n",
    "      <td>30</td>\n",
    "      <td>0</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>766</th>\n",
    "      <td>1</td>\n",
    "      <td>126</td>\n",
    "      <td>60</td>\n",
    "      <td>0</td>\n",
    "      <td>0</td>\n",
    "      <td>30.1</td>\n",
    "      <td>0.349</td>\n",
    "      <td>47</td>\n",
    "      <td>1</td>\n",
    "    </tr>\n",
    "    <tr>\n",
    "      <th>767</th>\n",
    "      <td>1</td>\n",
    "      <td>93</td>\n",
    "      <td>70</td>\n",
    "      <td>31</td>\n",
    "      <td>0</td>\n",
    "      <td>30.4</td>\n",
    "      <td>0.315</td>\n",
       "      <td>23</td>\n",
       "      <td>0</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \\\n",
       "763           10      101             76             48      180  32.9   \n",
       "764            2      122             70             27        0  36.8   \n",
       "765            5      121             72             23      112  26.2   \n",
       "766            1      126             60              0        0  30.1   \n",
       "767            1       93             70             31        0  30.4   \n",
       "\n",
       "     DiabetesPedigreeFunction  Age  Outcome  \n",
       "763                     0.171   63        0  \n",
       "764                     0.340   27        0  \n",
       "765                     0.245   30        0  \n",
       "766                     0.349   47        1  \n",
       "767                     0.315   23        0  "
      ]
     },
     "execution_count": 7,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Revisa los últimos 5 renglones del dataset usando la función tail()\n",
    "df.tail()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 8,
   "metadata": {
    "id": "YcQfUpcLKC_3",
    "outputId": "4206bddf-7216-4cdc-f9d9-35500b1591bb"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 768 entries, 0 to 767\n",
      "Data columns (total 9 columns):\n",
      " #   Column                    Non-Null Count  Dtype  \n",
      "---  ------                    --------------  -----  \n",
      " 0   Pregnancies               768 non-null    int64  \n",
      " 1   Glucose                   768 non-null    int64  \n",
      " 2   BloodPressure             768 non-null    int64  \n",
      " 3   SkinThickness             768 non-null    int64  \n",
      " 4   Insulin                   768 non-null    int64  \n",
      " 5   BMI                       768 non-null    float64\n",
      " 6   DiabetesPedigreeFunction  768 non-null    float64\n",
      " 7   Age                       768 non-null    int64  \n",
      " 8   Outcome                   768 non-null    int64  \n",
      "dtypes: float64(2), int64(7)\n",
      "memory usage: 54.1 KB\n"
     ]
    }
   ],
   "source": [
    "#Revisa la información mas completa del conjunto de datos usando la función info()\n",
    "#Muestra el total de datos, las columnas y su tipo correspondiente, dice si contiene nulos o
no\n",
    "df.info()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 9,
   "metadata": {
    "id": "d70yk1VbKC_4",
    "outputId": "e7d340fd-be4a-4b6a-a205-18e7833aef88"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Pregnancies                  17\n",
       "Glucose                     136\n",
       "BloodPressure                47\n",
       "SkinThickness                51\n",
       "Insulin                     186\n",
       "BMI                         248\n",
       "DiabetesPedigreeFunction    517\n",
       "Age                          52\n",
       "Outcome                       2\n",
       "dtype: int64"
      ]
     },
     "execution_count": 9,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#revisa cuántos valores únicos tiene cada atributo del archivo usando la función nunique()\n",
    "df.nunique()"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "8nTyC2dzKC_5"
   },
   "source": [
    "## Exploración de Datos"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 10,
   "metadata": {
    "id": "0CsuORYXKC_5",
    "outputId": "8425ef66-1f29-492c-81cd-f72230bad9d2"
   },
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "      <th></th>\n",
        "      <th>Pregnancies</th>\n",
        "      <th>Glucose</th>\n",
        "      <th>BloodPressure</th>\n",
        "      <th>SkinThickness</th>\n",
        "      <th>Insulin</th>\n",
        "      <th>BMI</th>\n",
        "      <th>DiabetesPedigreeFunction</th>\n",
        "      <th>Age</th>\n",
        "      <th>Outcome</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>count</th>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "      <td>768.000000</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>mean</th>\n",
        "      <td>3.845052</td>\n",
        "      <td>120.894531</td>\n",
        "      <td>69.105469</td>\n",
        "      <td>20.536458</td>\n",
        "      <td>79.799479</td>\n",
        "      <td>31.992578</td>\n",
        "      <td>0.471876</td>\n",
        "      <td>33.240885</td>\n",
        "      <td>0.348958</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>std</th>\n",
        "      <td>3.369578</td>\n",
        "      <td>31.972618</td>\n",
        "      <td>19.355807</td>\n",
        "      <td>15.952218</td>\n",
        "      <td>115.244002</td>\n",
        "      <td>7.884160</td>\n",
        "      <td>0.331329</td>\n",
        "      <td>11.760232</td>\n",
        "      <td>0.476951</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>min</th>\n",
        "      <td>0.000000</td>\n",
        "      <td>0.000000</td>\n",
        "      <td>0.000000</td>\n",
         "      <td>0.000000</td>\n",
         "      <td>0.000000</td>\n",
         "      <td>0.000000</td>\n",
         "      <td>0.078000</td>\n",
         "      <td>21.000000</td>\n",
         "      <td>0.000000</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>25%</th>\n",
         "      <td>1.000000</td>\n",
         "      <td>99.000000</td>\n",
         "      <td>62.000000</td>\n",
         "      <td>0.000000</td>\n",
         "      <td>0.000000</td>\n",
         "      <td>27.300000</td>\n",
         "      <td>0.243750</td>\n",
         "      <td>24.000000</td>\n",
         "      <td>0.000000</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>50%</th>\n",
         "      <td>3.000000</td>\n",
         "      <td>117.000000</td>\n",
         "      <td>72.000000</td>\n",
         "      <td>23.000000</td>\n",
         "      <td>30.500000</td>\n",
         "      <td>32.000000</td>\n",
         "      <td>0.372500</td>\n",
         "      <td>29.000000</td>\n",
         "      <td>0.000000</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>75%</th>\n",
         "      <td>6.000000</td>\n",
         "      <td>140.250000</td>\n",
         "      <td>80.000000</td>\n",
         "      <td>32.000000</td>\n",
         "      <td>127.250000</td>\n",
         "      <td>36.600000</td>\n",
         "      <td>0.626250</td>\n",
         "      <td>41.000000</td>\n",
         "      <td>1.000000</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>max</th>\n",
         "      <td>17.000000</td>\n",
         "      <td>199.000000</td>\n",
         "      <td>122.000000</td>\n",
         "      <td>99.000000</td>\n",
         "      <td>846.000000</td>\n",
         "      <td>67.100000</td>\n",
         "      <td>2.420000</td>\n",
         "      <td>81.000000</td>\n",
         "      <td>1.000000</td>\n",
         "    </tr>\n",
         "  </tbody>\n",
         "</table>\n",
         "</div>"
        ],
        "text/plain": [
         "       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \\\n",
         "count   768.000000  768.000000     768.000000     768.000000  768.000000   \n",
         "mean      3.845052  120.894531      69.105469      20.536458   79.799479   \n",
         "std       3.369578   31.972618      19.355807      15.952218  115.244002   \n",
         "min       0.000000    0.000000       0.000000       0.000000    0.000000   \n",
         "25%       1.000000   99.000000      62.000000       0.000000    0.000000  \n",
         "50%       3.000000  117.000000      72.000000      23.000000   30.500000  \n",
         "75%       6.000000  140.250000      80.000000      32.000000  127.250000  \n",
         "max      17.000000  199.000000     122.000000      99.000000  846.000000  \n",
         "\n",
         "              BMI  DiabetesPedigreeFunction         Age     Outcome  \n",
         "count  768.000000                768.000000  768.000000  768.000000  \n",
         "mean    31.992578                  0.471876   33.240885    0.348958  \n",
         "std      7.884160                  0.331329   11.760232    0.476951  \n",
         "min      0.000000                  0.078000   21.000000    0.000000  \n",
         "25%     27.300000                  0.243750   24.000000    0.000000  \n",
         "50%     32.000000                  0.372500   29.000000    0.000000  \n",
         "75%     36.600000                  0.626250   41.000000    1.000000  \n",
         "max     67.100000                  2.420000   81.000000    1.000000  "
        ]
       },
       "execution_count": 10,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "#utiliza la función describe() para obtener estadística básica. se puede incluir -0\n",
      "df.describe()"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 14,
     "metadata": {
      "id": "6FZFAnZjKC_6",
      "outputId": "753497de-4abb-4bcb-d1b7-34275424396f"
     },
     "outputs": [
      {
       "data": {
        "text/plain": [
         "Pregnancies                 0\n",
         "Glucose                     0\n",
         "BloodPressure               0\n",
         "SkinThickness               0\n",
         "Insulin                     0\n",
         "BMI                         0\n",
         "DiabetesPedigreeFunction    0\n",
         "Age                         0\n",
         "Outcome                     0\n",
         "dtype: int64"
        ]
       },
       "execution_count": 14,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "#Revisa Valores nulos con funcion isnull().sum()\n",
      "df.isnull().sum()"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 16,
     "metadata": {},
     "outputs": [
      {
```

```
     "data": {
      "text/plain": [
       "array([ 6,  1,  8,  0,  5,  3, 10,  2,  4,  7,  9, 11, 13, 15, 17, 12, 14])"
      ]
     },
     "execution_count": 16,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Revisar valores únicos por columna usando función unique(): nombre-columna.unique()\n",
    "df.Pregnancies.unique()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 17,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([148,  85, 183,  89, 137, 116,  78, 115, 197, 125, 110, 168, 139,\n",
       "       189, 166, 100, 118, 107, 103, 126,  99, 196, 119, 143, 147,  97,\n",
       "       145, 117, 109, 158,  88,  92, 122, 138, 102,  90, 111, 180, 133,\n",
       "       106, 171, 159, 146,  71, 105, 101, 176, 150,  73, 187,  84,  44,\n",
       "       141, 114,  95, 129,  79,   0,  62, 131, 112, 113,  74,  83, 136,\n",
       "        80, 123,  81, 134, 142, 144,  93, 163, 151,  96, 155,  76, 160,\n",
       "       124, 162, 132, 120, 173, 170, 128, 108, 154,  57, 156, 153, 188,\n",
       "       152, 104,  87,  75, 179, 130, 194, 181, 135, 184, 140, 177, 164,\n",
       "        91, 165,  86, 193, 191, 161, 167,  77, 182, 157, 178,  61,  98,\n",
       "       127,  82,  72, 172,  94, 175, 195,  68, 186, 198, 121,  67, 174,\n",
       "       199,  56, 169, 149,  65, 190])"
      ]
     },
     "execution_count": 17,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Revisar valores únicos por columna usando función unique(): nombre-columna.unique()\n",
    "df.Glucose.unique()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 18,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([1, 0])"
      ]
     },
     "execution_count": 18,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Revisar valores únicos por columna usando función unique(): nombre-columna.unique()\n",
    "df.Outcome.unique()"
```

```
     ]
    },
    {
     "cell_type": "markdown",
     "metadata": {
      "id": "ae13SbMmKC_7"
     },
     "source": [
      "## Variables Cuantitativas\n",
      "\n",
      "### Medidas de tendencia central\n"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 20,
     "metadata": {
      "id": "vDNofiEZKC_7",
      "outputId": "63524fd6-ca06-484e-fa9e-0dbbf5be2d6d"
     },
     "outputs": [
      {
       "name": "stdout",
       "output_type": "stream",
       "text": [
        "Mean_pregnancies: 3.8450520833333335\n",
        "Median_pregnancies: 3.0\n",
        "Mode_pregnancies: 0    1\n",
        "Name: Pregnancies, dtype: int64\n"
       ]
      }
     ],
     "source": [
      "#Pregnancies\n",
      "#Se puede obtener la media, mediana y moda para\n",
      "mean_pregnancies = df['Pregnancies'].mean()\n",
      "median_pregnancies = df['Pregnancies'].median()\n",
      "mode_pregnancies = df['Pregnancies'].mode()\n",
      "print(\"Mean_pregnancies:\",mean_pregnancies)\n",
      "print(\"Median_pregnancies:\",median_pregnancies)\n",
      "print(\"Mode_pregnancies:\",mode_pregnancies)"
     ]
    },
    {
     "cell_type": "markdown",
     "metadata": {
      "id": "BxOaUFllKC_8"
     },
     "source": [
      "Conclusiones:\n",
      "El promedio de embarazos fue de 3  \n",
      "La cantidad de embarazos al centro es 3  \n",
      "La cantidad de embarazos más repetida fue de 0 y 1"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 22,
     "metadata": {
      "id": "vDNofiEZKC_7",
      "outputId": "63524fd6-ca06-484e-fa9e-0dbbf5be2d6d"
     },
     "outputs": [
      {
       "name": "stdout",
     "output_type": "stream",
     "text": [
      "Mean_glucose: 120.89453125\n",
      "Median_glucose: 117.0\n",
      "Mode_glucose: 0     99\n",
      "1    100\n",
      "Name: Glucose, dtype: int64\n"
     ]
    }
   ],
   "source": [
    "#Glucose\n",
    "#Se puede obtener la media, mediana y moda para\n",
    "mean_glucose = df['Glucose'].mean()\n",
    "median_glucose = df['Glucose'].median()\n",
    "mode_glucose = df['Glucose'].mode()\n",
    "print(\"Mean_glucose:\",mean_glucose)\n",
    "print(\"Median_glucose:\",median_glucose)\n",
    "print(\"Mode_glucose:\",mode_glucose)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "BxOaUFllKC_8"
   },
   "source": [
    "Conclusiones:  \n",
    "El promedio de glucosa fue de 120  \n",
    "La glucosa al centro es 117  \n",
    "La glucosa más repetida fue de 0, 1, 99 y 100"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 24,
   "metadata": {
    "id": "vDNofiEZKC_7",
    "outputId": "63524fd6-ca06-484e-fa9e-0dbbf5be2d6d"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Mean_outcome: 0.3489583333333333\n",
      "Median_outcome: 0.0\n",
      "Mode_outcome: 0    0\n",
      "Name: Outcome, dtype: int64\n"
     ]
    }
   ],
   "source": [
    "#Outcome\n",
    "#Se puede obtener la media, mediana y moda para\n",
    "mean_outcome = df['Outcome'].mean()\n",
    "median_outcome = df['Outcome'].median()\n",
    "mode_outcome = df['Outcome'].mode()\n",
    "print(\"Mean_outcome:\",mean_outcome)\n",
    "print(\"Median_outcome:\",median_outcome)\n",
    "print(\"Mode_outcome:\",mode_outcome)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "BxOaUFllKC_8"
   },
   "source": [
    "Conclusiones:\n",
    "El promedio de resultado fue de 0.3  \n",
    "El resultado al centro es 0  \n",
    "El resultado más repetido fue de 0"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Cabe mencionar que, en las discusiones en Kaggle, se menciona que el dato \"0\" significa que el
paciente es sano y el dato \"1\" significa que el paciente tiene diabetes. Por lo tanto, la mayoría
de los datos indican un paciente sano."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "ru1HJsXHKC_9"
   },
   "source": [
    "# Variables Categóricas"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 21,
   "metadata": {
    "id": "hu2J0Q7NKC_9",
    "outputId": "b0509fc7-7e9d-4b5b-8fbf-0a9563c10d55"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Pregnancies\n",
       "1     135\n",
       "0     111\n",
       "2     103\n",
       "3      75\n",
       "4      68\n",
       "5      57\n",
       "6      50\n",
       "7      45\n",
       "8      38\n",
       "9      28\n",
       "10     24\n",
       "11     11\n",
       "13     10\n",
       "12      9\n",
       "14      2\n",
       "15      1\n",
       "17      1\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 21,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts():
\n",
    "# nombreDataframe.columna.value_counts()\n",
    "# nombreDataframe['columna'].value_counts()\n",
    "df.Pregnancies.value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 25,
   "metadata": {
    "id": "hu2J0Q7NKC_9",
    "outputId": "b0509fc7-7e9d-4b5b-8fbf-0a9563c10d55"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Glucose\n",
       "99     17\n",
       "100    17\n",
       "111    14\n",
       "129    14\n",
       "125    14\n",
       "       ..\n",
       "191     1\n",
       "177     1\n",
       "44      1\n",
       "62      1\n",
       "190     1\n",
       "Name: count, Length: 136, dtype: int64"
      ]
     },
     "execution_count": 25,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts():
\n",
    "# nombreDataframe.columna.value_counts()\n",
    "# nombreDataframe['columna'].value_counts()\n",
    "df.Glucose.value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 26,
   "metadata": {
    "id": "hu2J0Q7NKC_9",
    "outputId": "b0509fc7-7e9d-4b5b-8fbf-0a9563c10d55"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Outcome\n",
       "0    500\n",
       "1    268\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 26,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts():
\n",
    "# nombreDataframe.columna.value_counts()\n",
    "# nombreDataframe['columna'].value_counts()\n",
    "df.Outcome.value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 27,
   "metadata": {
    "id": "hu2J0Q7NKC_9",
    "outputId": "b0509fc7-7e9d-4b5b-8fbf-0a9563c10d55"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Pregnancies\n",
       "1     135\n",
       "0     111\n",
       "2     103\n",
       "3      75\n",
       "4      68\n",
       "5      57\n",
       "6      50\n",
       "7      45\n",
       "8      38\n",
       "9      28\n",
       "10     24\n",
       "11     11\n",
       "13     10\n",
       "12      9\n",
       "14      2\n",
       "15      1\n",
       "17      1\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 27,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts():
\n",
    "# nombreDataframe.columna.value_counts()\n",
    "# nombreDataframe['columna'].value_counts()\n",
    "df[\"Pregnancies\"].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 28,
   "metadata": {
    "id": "hu2J0Q7NKC_9",
    "outputId": "b0509fc7-7e9d-4b5b-8fbf-0a9563c10d55"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Glucose\n",
       "99     17\n",
       "100    17\n",
       "111    14\n",
       "129    14\n",
       "125    14\n",
       "       ..\n",
       "191     1\n",
       "177     1\n",
       "44      1\n",
       "62      1\n",
       "190     1\n",
       "Name: count, Length: 136, dtype: int64"
      ]
     },
     "execution_count": 28,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts(): \n",
    "# nombreDataframe.columna.value_counts()\n",
    "# nombreDataframe['columna'].value_counts()\n",
    "df[\"Glucose\"].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 29,
   "metadata": {
    "id": "hu2J0Q7NKC_9",
    "outputId": "b0509fc7-7e9d-4b5b-8fbf-0a9563c10d55"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Outcome\n",
       "0    500\n",
       "1    268\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 29,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts(): \n",
    "# nombreDataframe.columna.value_counts()\n",
    "# nombreDataframe['columna'].value_counts()\n",
    "df[\"Outcome\"].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 10,
   "metadata": {
    "id": "r3T8XkonKC_-",
    "outputId": "cd7251fd-e296-48ac-eb48-d883e5af0f35"
   },
   "outputs": [],
   "source": [
    "#Revisa conteo de varias columnas"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 42,
   "metadata": {
    "id": "Q7VSqNEtKC__",
    "outputId": "567aec70-2bd2-4f2f-8128-5a3ce5459baa"
   },
   "outputs": [],
   "source": [
    "# Crear variable totalPregDiabetic que incluya la suma de las columnas Pregnancies y Outcome con
valor \"1\"\n",
    "# Mostrar el total por cada tamaño de familia\n",
    "conteo_preg = df[\"Pregnancies\"].count()\n",
    "conteo_outcome = (df[\"Outcome\"] == 1).sum()\n",
    "df[\"totalPregDiabetic\"] = df[\"Pregnancies\"] + (df[\"Outcome\"] == 1).astype(int)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 43,
   "metadata": {
    "id": "WLb1AfB0KDAA"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<bound method NDFrame.head of      Pregnancies  Glucose  BloodPressure  SkinThickness
Insulin   BMI  \\\n",
       "0              6      148             72             35        0  33.6   \n",
       "1              1       85             66             29        0  26.6   \n",
       "2              8      183             64              0        0  23.3   \n",
       "3              1       89             66             23       94  28.1   \n",
       "4              0      137             40             35      168  43.1   \n",
       "..           ...      ...            ...            ...      ...   ...   \n",
       "763           10      101             76             48      180  32.9   \n",
       "764            2      122             70             27        0  36.8   \n",
       "765            5      121             72             23      112  26.2   \n",
       "766            1      126             60              0        0  30.1   \n",
       "767            1       93             70             31        0  30.4   \n",
       "\n",
       "     DiabetesPedigreeFunction  Age  Outcome  totalPregDiabetic  \n",
       "0                       0.627   50        1                  7  \n",
       "1                       0.351   31        0                  1  \n",
       "2                       0.672   32        1                  9  \n",
       "3                       0.167   21        0                  1  \n",
       "4                       2.288   33        1                  1  \n",
       "..                        ...  ...      ...                ...  \n",
       "763                     0.171   63        0                 10  \n",
       "764                     0.340   27        0                  2  \n",
       "765                     0.245   30        0                  5  \n",
       "766                     0.349   47        1                  2  \n",
       "767                     0.315   23        0                  1  \n",
       "\n",
       "[768 rows x 10 columns]>"
      ]
     },
    "execution_count": 43,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "df.head"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
   "id": "G13IyhcDKDAT"
  },
  "source": [
   "## Consulta"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 44,
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "Pregnancies                   6.000\n",
      "Glucose                     148.000\n",
      "BloodPressure                72.000\n",
      "SkinThickness                35.000\n",
      "Insulin                       0.000\n",
      "BMI                          33.600\n",
      "DiabetesPedigreeFunction      0.627\n",
      "Age                          50.000\n",
      "Outcome                       1.000\n",
      "totalPregDiabetic             7.000\n",
      "Name: 0, dtype: float64"
     ]
    },
    "execution_count": 44,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "# df.iloc[i]: Accede a la fila en la posición i.\n",
   "# Acceder a la primera fila\n",
   "df.iloc[0]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 45,
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/html": [
      "<div>\n",
      "<style scoped>\n",
      "    .dataframe tbody tr th:only-of-type {\n",
      "        vertical-align: middle;\n",
      "    }\n",
      "\n",
      "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "      <th></th>\n",
        "      <th>Pregnancies</th>\n",
        "      <th>Glucose</th>\n",
        "      <th>BloodPressure</th>\n",
        "      <th>SkinThickness</th>\n",
        "      <th>Insulin</th>\n",
        "      <th>BMI</th>\n",
        "      <th>DiabetesPedigreeFunction</th>\n",
        "      <th>Age</th>\n",
        "      <th>Outcome</th>\n",
        "      <th>totalPregDiabetic</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>0</th>\n",
        "      <td>6</td>\n",
        "      <td>148</td>\n",
        "      <td>72</td>\n",
        "      <td>35</td>\n",
        "      <td>0</td>\n",
        "      <td>33.6</td>\n",
        "      <td>0.627</td>\n",
        "      <td>50</td>\n",
        "      <td>1</td>\n",
        "      <td>7</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>1</th>\n",
        "      <td>1</td>\n",
        "      <td>85</td>\n",
        "      <td>66</td>\n",
        "      <td>29</td>\n",
        "      <td>0</td>\n",
        "      <td>26.6</td>\n",
        "      <td>0.351</td>\n",
        "      <td>31</td>\n",
        "      <td>0</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>2</th>\n",
        "      <td>8</td>\n",
        "      <td>183</td>\n",
        "      <td>64</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "      <td>23.3</td>\n",
        "      <td>0.672</td>\n",
        "      <td>32</td>\n",
        "      <td>1</td>\n",
        "      <td>9</td>\n",
        "    </tr>\n",
        "  </tbody>\n",
        "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \\\n",
       "0            6      148             72             35        0  33.6   \n",
       "1            1       85             66             29        0  26.6   \n",
       "2            8      183             64              0        0  23.3   \n",
       "\n",
       "   DiabetesPedigreeFunction  Age  Outcome  totalPregDiabetic  \n",
       "0                     0.627   50        1                  7  \n",
       "1                     0.351   31        0                  1  \n",
       "2                     0.672   32        1                  9  "
      ]
     },
     "execution_count": 45,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "# Acceder a las dos primeras filas\n",
    "df.iloc[:3]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 46,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>Pregnancies</th>\n",
       "      <th>Glucose</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>0</th>\n",
       "      <td>6</td>\n",
       "      <td>148</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>1</th>\n",
       "      <td>1</td>\n",
       "      <td>85</td>\n",
       "    </tr>\n",
         "    <tr>\n",
         "      <th>2</th>\n",
         "      <td>8</td>\n",
         "      <td>183</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>3</th>\n",
         "      <td>1</td>\n",
         "      <td>89</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>4</th>\n",
         "      <td>0</td>\n",
         "      <td>137</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>...</th>\n",
         "      <td>...</td>\n",
         "      <td>...</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>763</th>\n",
         "      <td>10</td>\n",
         "      <td>101</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>764</th>\n",
         "      <td>2</td>\n",
         "      <td>122</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>765</th>\n",
         "      <td>5</td>\n",
         "      <td>121</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>766</th>\n",
         "      <td>1</td>\n",
         "      <td>126</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>767</th>\n",
         "      <td>1</td>\n",
         "      <td>93</td>\n",
         "    </tr>\n",
         "  </tbody>\n",
         "</table>\n",
         "<p>768 rows × 2 columns</p>\n",
         "</div>"
        ],
        "text/plain": [
         "     Pregnancies  Glucose\n",
         "0              6      148\n",
         "1              1       85\n",
         "2              8      183\n",
         "3              1       89\n",
         "4              0      137\n",
         "..           ...      ...\n",
         "763           10      101\n",
         "764            2      122\n",
         "765            5      121\n",
         "766            1      126\n",
         "767            1       93\n",
         "\n",
         "[768 rows x 2 columns]"
```

```
      ]
     },
     "execution_count": 46,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Seleccionar columnas, indicando entre corchetes [nombreColumna, nombreColumna]\n",
    "df[[\"Pregnancies\", \"Glucose\"]]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 50,
   "metadata": {},
   "outputs": [],
   "source": [
    "#Selección de filas [indicar dataframe[columna] operador valor]\n",
    "embarazos = df[df[\"Pregnancies\"] == 0]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 51,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>Pregnancies</th>\n",
       "      <th>Glucose</th>\n",
       "      <th>BloodPressure</th>\n",
       "      <th>SkinThickness</th>\n",
       "      <th>Insulin</th>\n",
       "      <th>BMI</th>\n",
       "      <th>DiabetesPedigreeFunction</th>\n",
       "      <th>Age</th>\n",
       "      <th>Outcome</th>\n",
       "      <th>totalPregDiabetic</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>4</th>\n",
       "      <td>0</td>\n",
       "      <td>137</td>\n",
        "      <td>40</td>\n",
        "      <td>35</td>\n",
        "      <td>168</td>\n",
        "      <td>43.1</td>\n",
        "      <td>2.288</td>\n",
        "      <td>33</td>\n",
        "      <td>1</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>528</th>\n",
        "      <td>0</td>\n",
        "      <td>117</td>\n",
        "      <td>66</td>\n",
        "      <td>31</td>\n",
        "      <td>188</td>\n",
        "      <td>30.8</td>\n",
        "      <td>0.493</td>\n",
        "      <td>22</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>511</th>\n",
        "      <td>0</td>\n",
        "      <td>139</td>\n",
        "      <td>62</td>\n",
        "      <td>17</td>\n",
        "      <td>210</td>\n",
        "      <td>22.1</td>\n",
        "      <td>0.207</td>\n",
        "      <td>21</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>506</th>\n",
        "      <td>0</td>\n",
        "      <td>180</td>\n",
        "      <td>90</td>\n",
        "      <td>26</td>\n",
        "      <td>90</td>\n",
        "      <td>36.5</td>\n",
        "      <td>0.314</td>\n",
        "      <td>35</td>\n",
        "      <td>1</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>487</th>\n",
        "      <td>0</td>\n",
        "      <td>173</td>\n",
        "      <td>78</td>\n",
        "      <td>32</td>\n",
        "      <td>265</td>\n",
        "      <td>46.5</td>\n",
        "      <td>1.159</td>\n",
        "      <td>58</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>...</th>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "      <td>...</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>247</th>\n",
        "      <td>0</td>\n",
        "      <td>165</td>\n",
        "      <td>90</td>\n",
        "      <td>33</td>\n",
        "      <td>680</td>\n",
        "      <td>52.3</td>\n",
        "      <td>0.427</td>\n",
        "      <td>23</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>239</th>\n",
        "      <td>0</td>\n",
        "      <td>104</td>\n",
        "      <td>76</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "      <td>18.4</td>\n",
        "      <td>0.582</td>\n",
        "      <td>27</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>237</th>\n",
        "      <td>0</td>\n",
        "      <td>179</td>\n",
        "      <td>90</td>\n",
        "      <td>27</td>\n",
        "      <td>0</td>\n",
        "      <td>44.1</td>\n",
        "      <td>0.686</td>\n",
        "      <td>23</td>\n",
        "      <td>1</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>297</th>\n",
        "      <td>0</td>\n",
        "      <td>126</td>\n",
        "      <td>84</td>\n",
        "      <td>29</td>\n",
        "      <td>215</td>\n",
        "      <td>30.7</td>\n",
        "      <td>0.520</td>\n",
        "      <td>24</td>\n",
        "      <td>0</td>\n",
        "      <td>0</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>757</th>\n",
        "      <td>0</td>\n",
        "      <td>123</td>\n",
         "      <td>72</td>\n",
         "      <td>0</td>\n",
         "      <td>0</td>\n",
         "      <td>36.3</td>\n",
         "      <td>0.258</td>\n",
         "      <td>52</td>\n",
         "      <td>1</td>\n",
         "      <td>1</td>\n",
         "    </tr>\n",
         "  </tbody>\n",
         "</table>\n",
         "<p>111 rows × 10 columns</p>\n",
         "</div>"
        ],
        "text/plain": [
         "     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \\\n",
         "4              0      137             40             35      168  43.1   \n",
         "528            0      117             66             31      188  30.8   \n",
         "511            0      139             62             17      210  22.1   \n",
         "506            0      180             90             26       90  36.5   \n",
         "487            0      173             78             32      265  46.5   \n",
         "..           ...      ...            ...            ...      ...   ...   \n",
         "247            0      165             90             33      680  52.3   \n",
         "239            0      104             76              0        0  18.4   \n",
         "237            0      179             90             27        0  44.1   \n",
         "297            0      126             84             29      215  30.7   \n",
         "757            0      123             72              0        0  36.3   \n",
         "\n",
         "     DiabetesPedigreeFunction  Age  Outcome  totalPregDiabetic  \n",
         "4                       2.288   33        1                  1  \n",
         "528                     0.493   22        0                  0  \n",
         "511                     0.207   21        0                  0  \n",
         "506                     0.314   35        1                  1  \n",
         "487                     1.159   58        0                  0  \n",
         "..                        ...  ...      ...                ...  \n",
         "247                     0.427   23        0                  0  \n",
         "239                     0.582   27        0                  0  \n",
         "237                     0.686   23        1                  1  \n",
         "297                     0.520   24        0                  0  \n",
         "757                     0.258   52        1                  1  \n",
         "\n",
         "[111 rows x 10 columns]"
        ]
       },
       "execution_count": 51,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "#ordenar usando funcion sort_values(by=atributo, ascending=True/false\n",
      "embarazos.sort_values(by=\"Pregnancies\", ascending=True)"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 52,
     "metadata": {},
     "outputs": [
      {
       "data": {
        "text/plain": [
         "Glucose\n",
         "57     0.0\n",
         "67     0.0\n",
       "73     0.0\n",
       "74     0.0\n",
       "78     0.0\n",
       "      ... \n",
       "180    1.0\n",
       "181    1.0\n",
       "188    1.0\n",
       "189    1.0\n",
       "198    1.0\n",
       "Name: Outcome, Length: 64, dtype: float64"
      ]
     },
     "execution_count": 52,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#Agrupar por un atributo y calcular función de agregación utilizando groupby(atributo)
['atributoAgregacion].funcion() (ej. mean() para promedio)\n",
    "embarazos.groupby(\"Glucose\")[\"Outcome\"].mean()"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "PnlaaWEmKDAU"
   },
   "source": [
    "Crea un subconjunto de **glucose** para el valor mayor a 180"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 56,
   "metadata": {
    "id": "o7-EK3csKDAU",
    "outputId": "498702ea-89b2-480c-dec1-d5a7704b6104"
   },
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>Pregnancies</th>\n",
       "      <th>Glucose</th>\n",
       "      <th>BloodPressure</th>\n",
       "      <th>SkinThickness</th>\n",
        "      <th>Insulin</th>\n",
        "      <th>BMI</th>\n",
        "      <th>DiabetesPedigreeFunction</th>\n",
        "      <th>Age</th>\n",
        "      <th>Outcome</th>\n",
        "      <th>totalPregDiabetic</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>440</th>\n",
        "      <td>0</td>\n",
        "      <td>189</td>\n",
        "      <td>104</td>\n",
        "      <td>25</td>\n",
        "      <td>0</td>\n",
        "      <td>34.3</td>\n",
        "      <td>0.435</td>\n",
        "      <td>41</td>\n",
        "      <td>1</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>561</th>\n",
        "      <td>0</td>\n",
        "      <td>198</td>\n",
        "      <td>66</td>\n",
        "      <td>32</td>\n",
        "      <td>274</td>\n",
        "      <td>41.3</td>\n",
        "      <td>0.502</td>\n",
        "      <td>28</td>\n",
        "      <td>1</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>595</th>\n",
        "      <td>0</td>\n",
        "      <td>188</td>\n",
        "      <td>82</td>\n",
        "      <td>14</td>\n",
        "      <td>185</td>\n",
        "      <td>32.0</td>\n",
        "      <td>0.682</td>\n",
        "      <td>22</td>\n",
        "      <td>1</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>753</th>\n",
        "      <td>0</td>\n",
        "      <td>181</td>\n",
        "      <td>88</td>\n",
        "      <td>44</td>\n",
        "      <td>510</td>\n",
        "      <td>43.3</td>\n",
        "      <td>0.222</td>\n",
        "      <td>26</td>\n",
        "      <td>1</td>\n",
        "      <td>1</td>\n",
        "    </tr>\n",
        "  </tbody>\n",
        "</table>\n",
        "</div>"
       ],
      "text/plain": [
       "     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \\\n",
       "440            0      189            104             25        0  34.3   \n",
       "561            0      198             66             32      274  41.3   \n",
       "595            0      188             82             14      185  32.0   \n",
       "753            0      181             88             44      510  43.3   \n",
       "\n",
       "     DiabetesPedigreeFunction  Age  Outcome  totalPregDiabetic  \n",
       "440                     0.435   41        1                  1  \n",
       "561                     0.502   28        1                  1  \n",
       "595                     0.682   22        1                  1  \n",
       "753                     0.222   26        1                  1  "
      ]
     },
     "execution_count": 56,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "# usa el criterio para extraer solo los boletos caros con fare > 50\n",
    "embarazos[embarazos[\"Glucose\"] > 180]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "colab": {
   "provenance": []
  },
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.8.10"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 4
}
```