{

"cells": [

{

"cell\_type": "markdown",

"metadata": {

"id": "dByMsuzT8Tnw"

},

"source": [

"# \*\*Hotel Booking Analysis\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ADZvTFTAFLnY"

},

"source": [

"##### \*\*Project Type\*\* - Data Analysis\n",

"##### \*\*Contribution\*\* - Individual Project\n",

"##### \*\*name\*\* - krishna kumar R "

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "eS4TAWwsFLna"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zrib1RqKFLnc"

},

"source": [

"### <u>\*\*Project Summary\*\*<u> "

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "VeMo4Hv4eND1"

},

"source": [

"##### For this hotel booking analysis, the goal was to explore the customer data of a hotel and identify any potential trends or correlations. The purpose of this exploratory data analysis (EDA) was to explore the hotel booking data set and identify potential relationships between key variables.\n",

"\n",

"##### The data set included customer booking information. As part of the analysis, descriptive statistics were calculated for each variable, and visualizations were created to explore the relationships between various variables. To get insight from the dataset, we built a variety of charts, including a count plot, bar plot, kdeplot, heatmap, pairplot, violin plot, and boxplot. \n",

"\n",

"##### The data set was composed of over 119390 hotel bookings, each containing several variables such as 'hotel', 'is\_canceled', 'lead\_time', For this hotel booking analysis, the goal was to explore the customer data of a hotel and identify any potential trends or correlations. The purpose of this exploratory data analysis (EDA) was to explore the hotel booking data set and identify potential relationships between key variables.\n",

"\n",

"##### The data set included customer booking information. As part of the analysis, descriptive statistics were calculated for each variable, and visualizations were created to explore the relationships between various variables. To get insight from the'arrival\_date\_year', 'arrival\_date\_month', 'arrival\_date\_week\_number', 'arrival\_date\_day\_of\_month', 'stays\_in\_weekend\_nights', 'stays\_in\_week\_nights', 'adults', 'children', 'babies', 'meal', 'country', 'market\_segment', 'distribution\_channel', 'is\_repeated\_guest', 'previous\_cancellations', 'previous\_bookings\_not\_canceled', 'reserved\_room\_type', 'assigned\_room\_type', 'booking\_changes', 'deposit\_type', 'agent', 'company', 'days\_in\_waiting\_list', 'customer\_type', 'adr', 'required\_car\_parking\_spaces', 'total\_of\_special\_requests', 'reservation\_status', and 'reservation\_status\_date'.\n",

"\n",

"##### Dataset variables are in int64, float64, and object datatypes. There are 32 variables: 12 variables are objects, 16 are int64, and 4 are float64. 31994 duplicate values were removed. The variables country had 452, children had 4, agent had 12193, and company had 82137 null values. We replaced the null value with the mode of each variable (country, children, agent) for these variables, but the variable \"company\" had more than 50% null value, so we removed it. Further, we removed outliers from lead\_time and adr. The final dataset had 87396 observations.\n",

"\n",

"##### We also changed the data types of variables children, agent, and reservation\_status\_date to int64, int64, and datetime64, respectively. We performed some feature engineering for more convenience and created new variables: total\_stays, total\_people, total\_childrens, reserved\_room\_assigned, guest\_category, and lead\_time\_category. Now total\_people and total\_childrens are in the floated 64 datatype, so we converted them to int64. Now We were removed from the observation because having total\_people at 0 made no sense.\n",

"\n",

"##### After data cleaning, exploratory data analysis revealed several interesting findings as following :\n",

"\n",

"- The top country with the most number of bookings is PRT, and the number one agent with the most number of bookings is 9.\n",

"- Customers favored city hotels more than resort hotels by a margin of 61.07 percent.\n",

"- One of the four reservations is canceled.\n",

"- The most popular food is BB.\n",

"- The Online (internet) platform is used to make the majority of bookings.\n",

"- The majority of the bookings are made using TA/TO, the leading distribution channel.\n",

"- The vast majority of hotel bookings are made by new guests. Almost no consumers (3.86%) returned.\n",

"- The customer wants Room A to be reserved the most.\n",

"- Customers do not wish to make a bookings with a pre-deposit.\n",

"- Customers (80%) favored making a hotel reservation for a short visit.\n",

"- Only 10% of people require space to park their cars.\n",

"- Most visitors are couples.\n",

"- The inability to assign a reserved room to a customer is not grounds for cancellation.\n",

"- Booking cancellations are not caused by a longer Lead time.\n",

"- A city hotel is busier than a resort.\n",

"- The busiest months for hotels are October and September. There isn't a lengthy wait for reservations in July.\n",

"- Not assigning a reserved room does not affect ADR.\n",

"\n",

"##### We had some difficulties with the data when we were cleaning and analyzing it. There were a lot of duplicate values in the dataset. Null values were present in the dataset. Choosing the most effective visualization method is difficult. Performing feature engineering was more challenging."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "5JF9nGliFLne"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zg0xrP7dFLni"

},

"source": [

"### <u>\*\*GitHub Link\*\*<u>"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "5UKV7Fo\_FLnk"

},

"source": [

"[Github Link]( https://github.com/au813521105013/DataScienceFundamentals.git)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "6IVWWPi3FLnl"

},

"source": [

"https://github.com/ajitmane36/EDA-Hotel-Booking-Analysis.git"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "dM-YP1\_8FLnn"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "w2CjYYvPFLno"

},

"source": [

"### <u>\*\*Problem Statement\*\*<u>"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Y06xIdG26kRF"

},

"source": [

"#### <b> Have you ever wondered when the best time of year to book a hotel room is? Or the optimal length of stay in order to get the best daily rate? What if you wanted to predict whether or not a hotel was likely to receive a disproportionately high number of special requests? This hotel booking dataset can help you explore those questions!\n",

"#### <b>This data set contains booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things. All personally identifying information has been removed from the data. </b>\n",

"#### <b> Explore and analyze the data to discover important factors that govern the bookings. </b>"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "qjhXc0FLFLnr"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ALyOsvTQFLnr"

},

"source": [

"### \*\*<u>Business Objective<u>\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "qLBQB5CGFLns"

},

"source": [

"Our primary goal is to conduct EDA on the provided dataset and derive valuable conclusions about broad hotel booking trends and how various factors interact to affect hotel bookings."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "tnBpjeNbFLnt"

},

"source": [

"."

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"id": "znZnJFE-CSRp"

},

"outputs": [],

"source": [

"#Importing necessary libraries\n",

"import numpy as np\n",

"import pandas as pd\n",

"import matplotlib.pyplot as plt\n",

"import seaborn as sns\n",

"%matplotlib inline\n",

"\n",

"#To ignore warnings\n",

"import warnings\n",

"warnings.filterwarnings('ignore')"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "bVYjGDNnFnX2",

"outputId": "a266e9ec-eeea-4394-8605-80cdac622891"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Mounted at /content/drive\n"

]

}

],

"source": [

"# Mounting drive\n",

"from google.colab import drive\n",

"drive.mount('/content/drive') "

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {

"id": "JL5RxBZ5Cm8D"

},

"outputs": [],

"source": [

"# Loabding Dataset\n",

"path='/content/drive/MyDrive/Almabetter Capstone Projects/1. Exploratory Data Analysis/Hotel Booking Analysis - Capstone Project/Hotel Bookings.csv'\n",

"df=pd.read\_csv(path)"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 386

},

"id": "03M21iRdDd8H",

"outputId": "9ddbd717-6cef-447c-97b7-c3d5d77504e5"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" hotel is\_canceled lead\_time arrival\_date\_year arrival\_date\_month \\\n",

"0 Resort Hotel 0 342 2015 July \n",

"1 Resort Hotel 0 737 2015 July \n",

"2 Resort Hotel 0 7 2015 July \n",

"3 Resort Hotel 0 13 2015 July \n",

"4 Resort Hotel 0 14 2015 July \n",

"\n",

" arrival\_date\_week\_number arrival\_date\_day\_of\_month \\\n",

"0 27 1 \n",

"1 27 1 \n",

"2 27 1 \n",

"3 27 1 \n",

"4 27 1 \n",

"\n",

" stays\_in\_weekend\_nights stays\_in\_week\_nights adults ... deposit\_type \\\n",

"0 0 0 2 ... No Deposit \n",

"1 0 0 2 ... No Deposit \n",

"2 0 1 1 ... No Deposit \n",

"3 0 1 1 ... No Deposit \n",

"4 0 2 2 ... No Deposit \n",

"\n",

" agent company days\_in\_waiting\_list customer\_type adr \\\n",

"0 NaN NaN 0 Transient 0.0 \n",

"1 NaN NaN 0 Transient 0.0 \n",

"2 NaN NaN 0 Transient 75.0 \n",

"3 304.0 NaN 0 Transient 75.0 \n",

"4 240.0 NaN 0 Transient 98.0 \n",

"\n",

" required\_car\_parking\_spaces total\_of\_special\_requests reservation\_status \\\n",

"0 0 0 Check-Out \n",

"1 0 0 Check-Out \n",

"2 0 0 Check-Out \n",

"3 0 0 Check-Out \n",

"4 0 1 Check-Out \n",

"\n",

" reservation\_status\_date \n",

"0 2015-07-01 \n",

"1 2015-07-01 \n",

"2 2015-07-02 \n",

"3 2015-07-02 \n",

"4 2015-07-03 \n",

"\n",

"[5 rows x 32 columns]"

],

"text/html": [

"\n",

" <div id=\"df-4f967374-c9d1-4b5c-8ab6-36fa7514f1b9\">\n",

" <div class=\"colab-df-container\">\n",

" <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>hotel</th>\n",

" <th>is\_canceled</th>\n",

" <th>lead\_time</th>\n",

" <th>arrival\_date\_year</th>\n",

" <th>arrival\_date\_month</th>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <th>adults</th>\n",

" <th>...</th>\n",

" <th>deposit\_type</th>\n",

" <th>agent</th>\n",

" <th>company</th>\n",

" <th>days\_in\_waiting\_list</th>\n",

" <th>customer\_type</th>\n",

" <th>adr</th>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <th>total\_of\_special\_requests</th>\n",

" <th>reservation\_status</th>\n",

" <th>reservation\_status\_date</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>342</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>0.0</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-01</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>737</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>0.0</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-01</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>7</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>75.0</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-02</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>13</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>304.0</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>75.0</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-02</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>14</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>240.0</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>98.0</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-03</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 32 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-4f967374-c9d1-4b5c-8ab6-36fa7514f1b9')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-4f967374-c9d1-4b5c-8ab6-36fa7514f1b9 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-4f967374-c9d1-4b5c-8ab6-36fa7514f1b9');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 4

}

],

"source": [

"# Dataset First Look\n",

"df.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 386

},

"id": "CS28W\_MkoQmp",

"outputId": "6589c2eb-07e6-425a-d9aa-ad282483a4a1"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" hotel is\_canceled lead\_time arrival\_date\_year \\\n",

"119385 City Hotel 0 23 2017 \n",

"119386 City Hotel 0 102 2017 \n",

"119387 City Hotel 0 34 2017 \n",

"119388 City Hotel 0 109 2017 \n",

"119389 City Hotel 0 205 2017 \n",

"\n",

" arrival\_date\_month arrival\_date\_week\_number \\\n",

"119385 August 35 \n",

"119386 August 35 \n",

"119387 August 35 \n",

"119388 August 35 \n",

"119389 August 35 \n",

"\n",

" arrival\_date\_day\_of\_month stays\_in\_weekend\_nights \\\n",

"119385 30 2 \n",

"119386 31 2 \n",

"119387 31 2 \n",

"119388 31 2 \n",

"119389 29 2 \n",

"\n",

" stays\_in\_week\_nights adults ... deposit\_type agent company \\\n",

"119385 5 2 ... No Deposit 394.0 NaN \n",

"119386 5 3 ... No Deposit 9.0 NaN \n",

"119387 5 2 ... No Deposit 9.0 NaN \n",

"119388 5 2 ... No Deposit 89.0 NaN \n",

"119389 7 2 ... No Deposit 9.0 NaN \n",

"\n",

" days\_in\_waiting\_list customer\_type adr \\\n",

"119385 0 Transient 96.14 \n",

"119386 0 Transient 225.43 \n",

"119387 0 Transient 157.71 \n",

"119388 0 Transient 104.40 \n",

"119389 0 Transient 151.20 \n",

"\n",

" required\_car\_parking\_spaces total\_of\_special\_requests \\\n",

"119385 0 0 \n",

"119386 0 2 \n",

"119387 0 4 \n",

"119388 0 0 \n",

"119389 0 2 \n",

"\n",

" reservation\_status reservation\_status\_date \n",

"119385 Check-Out 2017-09-06 \n",

"119386 Check-Out 2017-09-07 \n",

"119387 Check-Out 2017-09-07 \n",

"119388 Check-Out 2017-09-07 \n",

"119389 Check-Out 2017-09-07 \n",

"\n",

"[5 rows x 32 columns]"

],

"text/html": [

"\n",

" <div id=\"df-8b64b84e-0588-4382-8843-81720ceb0686\">\n",

" <div class=\"colab-df-container\">\n",

" <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>hotel</th>\n",

" <th>is\_canceled</th>\n",

" <th>lead\_time</th>\n",

" <th>arrival\_date\_year</th>\n",

" <th>arrival\_date\_month</th>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <th>adults</th>\n",

" <th>...</th>\n",

" <th>deposit\_type</th>\n",

" <th>agent</th>\n",

" <th>company</th>\n",

" <th>days\_in\_waiting\_list</th>\n",

" <th>customer\_type</th>\n",

" <th>adr</th>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <th>total\_of\_special\_requests</th>\n",

" <th>reservation\_status</th>\n",

" <th>reservation\_status\_date</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>119385</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>23</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>30</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>394.0</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>96.14</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-06</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119386</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>102</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>31</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>3</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>225.43</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119387</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>34</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>31</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>157.71</td>\n",

" <td>0</td>\n",

" <td>4</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119388</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>109</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>31</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>89.0</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>104.40</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119389</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>205</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>29</td>\n",

" <td>2</td>\n",

" <td>7</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>NaN</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>151.20</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 32 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-8b64b84e-0588-4382-8843-81720ceb0686')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-8b64b84e-0588-4382-8843-81720ceb0686 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-8b64b84e-0588-4382-8843-81720ceb0686');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 5

}

],

"source": [

"#Last five observations\n",

"df.tail()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "iCvNFRA-FLn2"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "He5fd8b-8OZf"

},

"source": [

"### <u>\*\*Dataset Inispection\*\*<u>"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "d4YE03nWDfz9",

"outputId": "8f89276d-ff16-41f4-a567-a964688c85cc"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(119390, 32)"

]

},

"metadata": {},

"execution\_count": 6

}

],

"source": [

"#check the shape of dataset\n",

"df.shape"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "\_VqO0KbAoD3t"

},

"source": [

"Dataset having 119390 observations and 32 varibles"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "aXxMtdELKExU",

"outputId": "9833edec-c679-4f95-a9b0-4662fca456ad"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Index(['hotel', 'is\_canceled', 'lead\_time', 'arrival\_date\_year',\n",

" 'arrival\_date\_month', 'arrival\_date\_week\_number',\n",

" 'arrival\_date\_day\_of\_month', 'stays\_in\_weekend\_nights',\n",

" 'stays\_in\_week\_nights', 'adults', 'children', 'babies', 'meal',\n",

" 'country', 'market\_segment', 'distribution\_channel',\n",

" 'is\_repeated\_guest', 'previous\_cancellations',\n",

" 'previous\_bookings\_not\_canceled', 'reserved\_room\_type',\n",

" 'assigned\_room\_type', 'booking\_changes', 'deposit\_type', 'agent',\n",

" 'company', 'days\_in\_waiting\_list', 'customer\_type', 'adr',\n",

" 'required\_car\_parking\_spaces', 'total\_of\_special\_requests',\n",

" 'reservation\_status', 'reservation\_status\_date'],\n",

" dtype='object')"

]

},

"metadata": {},

"execution\_count": 7

}

],

"source": [

"# Columns names\n",

"df.columns"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "wAaxwrGXKN8w",

"outputId": "e2ba26b3-1ee1-4379-9065-c2aa31f9ef09"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 119390 entries, 0 to 119389\n",

"Data columns (total 32 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 hotel 119390 non-null object \n",

" 1 is\_canceled 119390 non-null int64 \n",

" 2 lead\_time 119390 non-null int64 \n",

" 3 arrival\_date\_year 119390 non-null int64 \n",

" 4 arrival\_date\_month 119390 non-null object \n",

" 5 arrival\_date\_week\_number 119390 non-null int64 \n",

" 6 arrival\_date\_day\_of\_month 119390 non-null int64 \n",

" 7 stays\_in\_weekend\_nights 119390 non-null int64 \n",

" 8 stays\_in\_week\_nights 119390 non-null int64 \n",

" 9 adults 119390 non-null int64 \n",

" 10 children 119386 non-null float64\n",

" 11 babies 119390 non-null int64 \n",

" 12 meal 119390 non-null object \n",

" 13 country 118902 non-null object \n",

" 14 market\_segment 119390 non-null object \n",

" 15 distribution\_channel 119390 non-null object \n",

" 16 is\_repeated\_guest 119390 non-null int64 \n",

" 17 previous\_cancellations 119390 non-null int64 \n",

" 18 previous\_bookings\_not\_canceled 119390 non-null int64 \n",

" 19 reserved\_room\_type 119390 non-null object \n",

" 20 assigned\_room\_type 119390 non-null object \n",

" 21 booking\_changes 119390 non-null int64 \n",

" 22 deposit\_type 119390 non-null object \n",

" 23 agent 103050 non-null float64\n",

" 24 company 6797 non-null float64\n",

" 25 days\_in\_waiting\_list 119390 non-null int64 \n",

" 26 customer\_type 119390 non-null object \n",

" 27 adr 119390 non-null float64\n",

" 28 required\_car\_parking\_spaces 119390 non-null int64 \n",

" 29 total\_of\_special\_requests 119390 non-null int64 \n",

" 30 reservation\_status 119390 non-null object \n",

" 31 reservation\_status\_date 119390 non-null object \n",

"dtypes: float64(4), int64(16), object(12)\n",

"memory usage: 29.1+ MB\n"

]

}

],

"source": [

"#Basic information about dataset\n",

"df.info()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "nu2FQNvzouim"

},

"source": [

"Features children, country, company and agent are having null values"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "33\_hqF6VFLn9"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "-odc5O-CreBx"

},

"source": [

"### <u>\*\*Handling Duplicate Values\*\*<u>"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "qBs9s\_4Mp\_Uh",

"outputId": "87ef7f06-5538-4dd1-8a77-ecae71c3a033"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"31994"

]

},

"metadata": {},

"execution\_count": 9

}

],

"source": [

"#checking for duplicate values\n",

"df.duplicated().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {

"id": "tvz6MOlDp\_O\_"

},

"outputs": [],

"source": [

"#Remove duplicate values\n",

"df.drop\_duplicates(inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "kFCBzzELr71-",

"outputId": "079abc6a-10f2-49e4-d82b-15e44aa79864"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0"

]

},

"metadata": {},

"execution\_count": 11

}

],

"source": [

"df.duplicated().sum()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "u\_k2xfZgsASx"

},

"source": [

"Dataset having 31994 duplicate values and those are successfully removed"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "lDqaRROmsaIh",

"outputId": "38641887-a816-494f-e097-000da4b5f485"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(87396, 32)"

]

},

"metadata": {},

"execution\_count": 12

}

],

"source": [

"# Checking for new shape of dataset\n",

"df.shape"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "qr2iAVysFLoC"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "wsGU7ekNtmRu"

},

"source": [

"### <u>\*\*Handling Missing/Null Values\*\*<u>"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"id": "lrDLhWhAsy14",

"outputId": "9f36b1d7-2da1-473e-cc3b-2f7e3f4517c4"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" number\_of\_nulls\_values\n",

"hotel 0\n",

"is\_canceled 0\n",

"lead\_time 0\n",

"arrival\_date\_year 0\n",

"arrival\_date\_month 0\n",

"arrival\_date\_week\_number 0\n",

"arrival\_date\_day\_of\_month 0\n",

"stays\_in\_weekend\_nights 0\n",

"stays\_in\_week\_nights 0\n",

"adults 0\n",

"children 4\n",

"babies 0\n",

"meal 0\n",

"country 452\n",

"market\_segment 0\n",

"distribution\_channel 0\n",

"is\_repeated\_guest 0\n",

"previous\_cancellations 0\n",

"previous\_bookings\_not\_canceled 0\n",

"reserved\_room\_type 0\n",

"assigned\_room\_type 0\n",

"booking\_changes 0\n",

"deposit\_type 0\n",

"agent 12193\n",

"company 82137\n",

"days\_in\_waiting\_list 0\n",

"customer\_type 0\n",

"adr 0\n",

"required\_car\_parking\_spaces 0\n",

"total\_of\_special\_requests 0\n",

"reservation\_status 0\n",

"reservation\_status\_date 0"

],

"text/html": [

"\n",

" <div id=\"df-5a516b3f-4801-4678-bb86-24225b821e5e\">\n",

" <div class=\"colab-df-container\">\n",

" <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>number\_of\_nulls\_values</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>hotel</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>is\_canceled</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>lead\_time</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_year</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_month</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>adults</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>children</th>\n",

" <td>4</td>\n",

" </tr>\n",

" <tr>\n",

" <th>babies</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>meal</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>country</th>\n",

" <td>452</td>\n",

" </tr>\n",

" <tr>\n",

" <th>market\_segment</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>distribution\_channel</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>is\_repeated\_guest</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>previous\_cancellations</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>previous\_bookings\_not\_canceled</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>reserved\_room\_type</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>assigned\_room\_type</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>booking\_changes</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>deposit\_type</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>agent</th>\n",

" <td>12193</td>\n",

" </tr>\n",

" <tr>\n",

" <th>company</th>\n",

" <td>82137</td>\n",

" </tr>\n",

" <tr>\n",

" <th>days\_in\_waiting\_list</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>customer\_type</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>adr</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>total\_of\_special\_requests</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>reservation\_status</th>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>reservation\_status\_date</th>\n",

" <td>0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-5a516b3f-4801-4678-bb86-24225b821e5e')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-5a516b3f-4801-4678-bb86-24225b821e5e button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-5a516b3f-4801-4678-bb86-24225b821e5e');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 13

}

],

"source": [

"# Check for missing/null values in Dataset\n",

"null\_df=pd.DataFrame(df.isna().sum()).rename(columns={0:'number\_of\_nulls\_values'})\n",

"null\_df"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "9EhTDRvzFLoF"

},

"source": [

"Null values for the features children, country, agent, and company are 4, 452, 12193, and 82137, respectively."

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 470

},

"id": "70XJHVYzFLoG",

"outputId": "7e4a02c1-76d5-46c3-bcba-3d314c5f2e1d"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Text(0.5, 1.0, 'Graph of Number of null values with respect to its Variable')"

]

},

"metadata": {},

"execution\_count": 14

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 504x504 with 0 Axes>"

]

},

"metadata": {}

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Ploting number of null values with its variable\n",

"plt.figure(figsize=(7,7))\n",

"null\_df.plot(kind='bar')\n",

"plt.title('Graph of Number of null values with respect to its Variable')"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"metadata": {

"id": "lSHfrWwusyij",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"outputId": "e4ce2437-d616-4b14-a63f-21d16084d57f"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" percentage\_null\_values\n",

"hotel 0.0000\n",

"is\_canceled 0.0000\n",

"lead\_time 0.0000\n",

"arrival\_date\_year 0.0000\n",

"arrival\_date\_month 0.0000\n",

"arrival\_date\_week\_number 0.0000\n",

"arrival\_date\_day\_of\_month 0.0000\n",

"stays\_in\_weekend\_nights 0.0000\n",

"stays\_in\_week\_nights 0.0000\n",

"adults 0.0000\n",

"children 0.0046\n",

"babies 0.0000\n",

"meal 0.0000\n",

"country 0.5172\n",

"market\_segment 0.0000\n",

"distribution\_channel 0.0000\n",

"is\_repeated\_guest 0.0000\n",

"previous\_cancellations 0.0000\n",

"previous\_bookings\_not\_canceled 0.0000\n",

"reserved\_room\_type 0.0000\n",

"assigned\_room\_type 0.0000\n",

"booking\_changes 0.0000\n",

"deposit\_type 0.0000\n",

"agent 13.9514\n",

"company 93.9826\n",

"days\_in\_waiting\_list 0.0000\n",

"customer\_type 0.0000\n",

"adr 0.0000\n",

"required\_car\_parking\_spaces 0.0000\n",

"total\_of\_special\_requests 0.0000\n",

"reservation\_status 0.0000\n",

"reservation\_status\_date 0.0000"

],

"text/html": [

"\n",

" <div id=\"df-12b7f4cf-c4cf-45d5-be8b-e491d5d2077d\">\n",

" <div class=\"colab-df-container\">\n",

" <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>percentage\_null\_values</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>hotel</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>is\_canceled</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>lead\_time</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_year</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_month</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>adults</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>children</th>\n",

" <td>0.0046</td>\n",

" </tr>\n",

" <tr>\n",

" <th>babies</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>meal</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>country</th>\n",

" <td>0.5172</td>\n",

" </tr>\n",

" <tr>\n",

" <th>market\_segment</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>distribution\_channel</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>is\_repeated\_guest</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>previous\_cancellations</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>previous\_bookings\_not\_canceled</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>reserved\_room\_type</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>assigned\_room\_type</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>booking\_changes</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>deposit\_type</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>agent</th>\n",

" <td>13.9514</td>\n",

" </tr>\n",

" <tr>\n",

" <th>company</th>\n",

" <td>93.9826</td>\n",

" </tr>\n",

" <tr>\n",

" <th>days\_in\_waiting\_list</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>customer\_type</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>adr</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>total\_of\_special\_requests</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>reservation\_status</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>reservation\_status\_date</th>\n",

" <td>0.0000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-12b7f4cf-c4cf-45d5-be8b-e491d5d2077d')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-12b7f4cf-c4cf-45d5-be8b-e491d5d2077d button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-12b7f4cf-c4cf-45d5-be8b-e491d5d2077d');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 15

}

],

"source": [

"# Percentage of null values\n",

"percentage\_null\_df=pd.DataFrame(round(df.isna().sum()\*100/len(df),4)).rename(columns={0:'percentage\_null\_values'})\n",

"percentage\_null\_df"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "JwlEvmEPIgaS"

},

"source": [

"Children, country, agent, and company variables have null values of 0.0046%, 0.5172%, 13.9514%, and 93.9826%, respectively. Variable companies having more than 50% null values"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"metadata": {

"id": "D31jUzbtyADV",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "790df835-9b08-4135-acad-7588f74d6716"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"PRT 27453\n",

"GBR 10433\n",

"FRA 8837\n",

"ESP 7252\n",

"DEU 5387\n",

" ... \n",

"MMR 1\n",

"BFA 1\n",

"CYM 1\n",

"MLI 1\n",

"KHM 1\n",

"Name: country, Length: 177, dtype: int64"

]

},

"metadata": {},

"execution\_count": 16

}

],

"source": [

"#checking category of features whoes having null values\n",

"df.country.value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"metadata": {

"id": "zyZTwrlDfx87",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "9dd9b021-97bf-48b8-e927-3ebf9d03d2d3"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"9.0 28759\n",

"240.0 13028\n",

"14.0 3349\n",

"7.0 3300\n",

"250.0 2779\n",

" ... \n",

"497.0 1\n",

"337.0 1\n",

"510.0 1\n",

"165.0 1\n",

"449.0 1\n",

"Name: agent, Length: 333, dtype: int64"

]

},

"metadata": {},

"execution\_count": 17

}

],

"source": [

"df.agent.value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"metadata": {

"id": "vVpqjkXQjxnc",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "e23b299d-649a-45e4-bfbd-7a04daa89721"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0.0 79028\n",

"1.0 4695\n",

"2.0 3593\n",

"3.0 75\n",

"10.0 1\n",

"Name: children, dtype: int64"

]

},

"metadata": {},

"execution\_count": 18

}

],

"source": [

"df.children.value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 19,

"metadata": {

"id": "ZwNt8BD9f2a1",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "8887e7b7-a67d-4085-b8ff-fc6cba7086cd"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"40.0 851\n",

"223.0 503\n",

"45.0 238\n",

"153.0 206\n",

"154.0 133\n",

" ... \n",

"10.0 1\n",

"419.0 1\n",

"415.0 1\n",

"425.0 1\n",

"534.0 1\n",

"Name: company, Length: 352, dtype: int64"

]

},

"metadata": {},

"execution\_count": 19

}

],

"source": [

"df.company.value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"metadata": {

"id": "bTpxQLmUPDTP"

},

"outputs": [],

"source": [

"# Dropping variable having more than 50% null values\n",

"df.drop(columns='company', inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"metadata": {

"id": "NlxeYv6GgM0k"

},

"outputs": [],

"source": [

"# Replacing null values with the most frequent value in a variable\n",

"df['children']=df['children'].fillna(df['children'].mode()[0])\n",

"df['country']=df['country'].fillna(df['country'].mode()[0])\n",

"df['agent']=df['agent'].fillna(df['agent'].mode()[0])"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "xhg0F0EFmcrN"

},

"source": [

"From above, children, country, and agent are discrete numerical variables, so replaced null values with modes, and the variable company had null values greater than 50%, so removed it."

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"metadata": {

"id": "B2Tjm6VJhfX6",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "0f79fec7-5bb2-4163-9490-0ce584949d06"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"hotel 0\n",

"is\_canceled 0\n",

"lead\_time 0\n",

"arrival\_date\_year 0\n",

"arrival\_date\_month 0\n",

"arrival\_date\_week\_number 0\n",

"arrival\_date\_day\_of\_month 0\n",

"stays\_in\_weekend\_nights 0\n",

"stays\_in\_week\_nights 0\n",

"adults 0\n",

"children 0\n",

"babies 0\n",

"meal 0\n",

"country 0\n",

"market\_segment 0\n",

"distribution\_channel 0\n",

"is\_repeated\_guest 0\n",

"previous\_cancellations 0\n",

"previous\_bookings\_not\_canceled 0\n",

"reserved\_room\_type 0\n",

"assigned\_room\_type 0\n",

"booking\_changes 0\n",

"deposit\_type 0\n",

"agent 0\n",

"days\_in\_waiting\_list 0\n",

"customer\_type 0\n",

"adr 0\n",

"required\_car\_parking\_spaces 0\n",

"total\_of\_special\_requests 0\n",

"reservation\_status 0\n",

"reservation\_status\_date 0\n",

"dtype: int64"

]

},

"metadata": {},

"execution\_count": 22

}

],

"source": [

"# Check for null values are removed\n",

"df.isna().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"metadata": {

"id": "Bx2vIKSC3BCn",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "dcc978b0-7440-47c6-aa41-a040ea5f5921"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(87396, 31)"

]

},

"metadata": {},

"execution\_count": 23

}

],

"source": [

"df.shape"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "N-011GpRn91O"

},

"source": [

"All the null values have been successfully removed."

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"metadata": {

"id": "VCupbt6PFLod",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 364

},

"outputId": "b2df02e1-7fc9-44d7-f805-b74d9215ed37"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" is\_canceled lead\_time arrival\_date\_year \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 0.274898 79.891368 2016.210296 \n",

"std 0.446466 86.052325 0.686102 \n",

"min 0.000000 0.000000 2015.000000 \n",

"25% 0.000000 11.000000 2016.000000 \n",

"50% 0.000000 49.000000 2016.000000 \n",

"75% 1.000000 125.000000 2017.000000 \n",

"max 1.000000 737.000000 2017.000000 \n",

"\n",

" arrival\_date\_week\_number arrival\_date\_day\_of\_month \\\n",

"count 87396.000000 87396.000000 \n",

"mean 26.838334 15.815541 \n",

"std 13.674572 8.835146 \n",

"min 1.000000 1.000000 \n",

"25% 16.000000 8.000000 \n",

"50% 27.000000 16.000000 \n",

"75% 37.000000 23.000000 \n",

"max 53.000000 31.000000 \n",

"\n",

" stays\_in\_weekend\_nights stays\_in\_week\_nights adults \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 1.005263 2.625395 1.875795 \n",

"std 1.031921 2.053584 0.626500 \n",

"min 0.000000 0.000000 0.000000 \n",

"25% 0.000000 1.000000 2.000000 \n",

"50% 1.000000 2.000000 2.000000 \n",

"75% 2.000000 4.000000 2.000000 \n",

"max 19.000000 50.000000 55.000000 \n",

"\n",

" children babies is\_repeated\_guest previous\_cancellations \\\n",

"count 87396.000000 87396.000000 87396.000000 87396.000000 \n",

"mean 0.138633 0.010824 0.039075 0.030413 \n",

"std 0.455871 0.113597 0.193775 0.369145 \n",

"min 0.000000 0.000000 0.000000 0.000000 \n",

"25% 0.000000 0.000000 0.000000 0.000000 \n",

"50% 0.000000 0.000000 0.000000 0.000000 \n",

"75% 0.000000 0.000000 0.000000 0.000000 \n",

"max 10.000000 10.000000 1.000000 26.000000 \n",

"\n",

" previous\_bookings\_not\_canceled booking\_changes agent \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 0.183990 0.271603 82.260287 \n",

"std 1.731894 0.727245 109.061182 \n",

"min 0.000000 0.000000 1.000000 \n",

"25% 0.000000 0.000000 9.000000 \n",

"50% 0.000000 0.000000 9.000000 \n",

"75% 0.000000 0.000000 234.000000 \n",

"max 72.000000 21.000000 535.000000 \n",

"\n",

" days\_in\_waiting\_list adr required\_car\_parking\_spaces \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 0.749565 106.337246 0.084226 \n",

"std 10.015731 55.013953 0.281533 \n",

"min 0.000000 -6.380000 0.000000 \n",

"25% 0.000000 72.000000 0.000000 \n",

"50% 0.000000 98.100000 0.000000 \n",

"75% 0.000000 134.000000 0.000000 \n",

"max 391.000000 5400.000000 8.000000 \n",

"\n",

" total\_of\_special\_requests \n",

"count 87396.000000 \n",

"mean 0.698567 \n",

"std 0.831946 \n",

"min 0.000000 \n",

"25% 0.000000 \n",

"50% 0.000000 \n",

"75% 1.000000 \n",

"max 5.000000 "

],

"text/html": [

"\n",

" <div id=\"df-d0964095-bcdf-458f-bacc-7e711b22e464\">\n",

" <div class=\"colab-df-container\">\n",

" <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>is\_canceled</th>\n",

" <th>lead\_time</th>\n",

" <th>arrival\_date\_year</th>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <th>adults</th>\n",

" <th>children</th>\n",

" <th>babies</th>\n",

" <th>is\_repeated\_guest</th>\n",

" <th>previous\_cancellations</th>\n",

" <th>previous\_bookings\_not\_canceled</th>\n",

" <th>booking\_changes</th>\n",

" <th>agent</th>\n",

" <th>days\_in\_waiting\_list</th>\n",

" <th>adr</th>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <th>total\_of\_special\_requests</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>0.274898</td>\n",

" <td>79.891368</td>\n",

" <td>2016.210296</td>\n",

" <td>26.838334</td>\n",

" <td>15.815541</td>\n",

" <td>1.005263</td>\n",

" <td>2.625395</td>\n",

" <td>1.875795</td>\n",

" <td>0.138633</td>\n",

" <td>0.010824</td>\n",

" <td>0.039075</td>\n",

" <td>0.030413</td>\n",

" <td>0.183990</td>\n",

" <td>0.271603</td>\n",

" <td>82.260287</td>\n",

" <td>0.749565</td>\n",

" <td>106.337246</td>\n",

" <td>0.084226</td>\n",

" <td>0.698567</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>0.446466</td>\n",

" <td>86.052325</td>\n",

" <td>0.686102</td>\n",

" <td>13.674572</td>\n",

" <td>8.835146</td>\n",

" <td>1.031921</td>\n",

" <td>2.053584</td>\n",

" <td>0.626500</td>\n",

" <td>0.455871</td>\n",

" <td>0.113597</td>\n",

" <td>0.193775</td>\n",

" <td>0.369145</td>\n",

" <td>1.731894</td>\n",

" <td>0.727245</td>\n",

" <td>109.061182</td>\n",

" <td>10.015731</td>\n",

" <td>55.013953</td>\n",

" <td>0.281533</td>\n",

" <td>0.831946</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>2015.000000</td>\n",

" <td>1.000000</td>\n",

" <td>1.000000</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>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" <td>0.000000</td>\n",

" <td>-6.380000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>0.000000</td>\n",

" <td>11.000000</td>\n",

" <td>2016.000000</td>\n",

" <td>16.000000</td>\n",

" <td>8.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" <td>2.000000</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>0.000000</td>\n",

" <td>9.000000</td>\n",

" <td>0.000000</td>\n",

" <td>72.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>0.000000</td>\n",

" <td>49.000000</td>\n",

" <td>2016.000000</td>\n",

" <td>27.000000</td>\n",

" <td>16.000000</td>\n",

" <td>1.000000</td>\n",

" <td>2.000000</td>\n",

" <td>2.000000</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>0.000000</td>\n",

" <td>9.000000</td>\n",

" <td>0.000000</td>\n",

" <td>98.100000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>1.000000</td>\n",

" <td>125.000000</td>\n",

" <td>2017.000000</td>\n",

" <td>37.000000</td>\n",

" <td>23.000000</td>\n",

" <td>2.000000</td>\n",

" <td>4.000000</td>\n",

" <td>2.000000</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>0.000000</td>\n",

" <td>234.000000</td>\n",

" <td>0.000000</td>\n",

" <td>134.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>1.000000</td>\n",

" <td>737.000000</td>\n",

" <td>2017.000000</td>\n",

" <td>53.000000</td>\n",

" <td>31.000000</td>\n",

" <td>19.000000</td>\n",

" <td>50.000000</td>\n",

" <td>55.000000</td>\n",

" <td>10.000000</td>\n",

" <td>10.000000</td>\n",

" <td>1.000000</td>\n",

" <td>26.000000</td>\n",

" <td>72.000000</td>\n",

" <td>21.000000</td>\n",

" <td>535.000000</td>\n",

" <td>391.000000</td>\n",

" <td>5400.000000</td>\n",

" <td>8.000000</td>\n",

" <td>5.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-d0964095-bcdf-458f-bacc-7e711b22e464')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-d0964095-bcdf-458f-bacc-7e711b22e464 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-d0964095-bcdf-458f-bacc-7e711b22e464');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 24

}

],

"source": [

"# Besic statistical description fo Dataset\n",

"df.describe()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "wiQLiml-FLoe"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ZAFSnB3ooDcs"

},

"source": [

"### <u>\*\*Handling Outliers\*\*<u>"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "k6kQWURV0w82"

},

"source": [

"\*\*\*Categorical veriables\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"metadata": {

"id": "WczkiYrLy91I",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "d5f1789d-fb10-410f-cfa2-ba82275925da"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Dataset having 12 categorical variables\n",

"------------------------------------------------------------------------------\n",

"['hotel', 'arrival\_date\_month', 'meal', 'country', 'market\_segment', 'distribution\_channel', 'reserved\_room\_type', 'assigned\_room\_type', 'deposit\_type', 'customer\_type', 'reservation\_status', 'reservation\_status\_date']\n"

]

}

],

"source": [

"# Obtaining categorical veriables\n",

"categorical\_veriables=[i for i in df.columns if df[i].dtypes=='O']\n",

"print(f'Dataset having {len(categorical\_veriables)} categorical variables')\n",

"print('--'\*39)\n",

"print(categorical\_veriables)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "OItT9GQVFLof"

},

"source": [

"Dataset having 12 categorical variables."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "PWN-ZGNf08rx"

},

"source": [

"\*\*\*Numerical variables\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"metadata": {

"id": "ZgDrrKMPhxUP",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "680f504c-ba00-41a4-8ebc-bce0757c3f96"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"There are 19 numerical variables.\n",

"------------------------------------------------------------------------------\n",

"['is\_canceled', 'lead\_time', 'arrival\_date\_year', 'arrival\_date\_week\_number', 'arrival\_date\_day\_of\_month', 'stays\_in\_weekend\_nights', 'stays\_in\_week\_nights', 'adults', 'children', 'babies', 'is\_repeated\_guest', 'previous\_cancellations', 'previous\_bookings\_not\_canceled', 'booking\_changes', 'agent', 'days\_in\_waiting\_list', 'adr', 'required\_car\_parking\_spaces', 'total\_of\_special\_requests']\n"

]

}

],

"source": [

"# Obtaining Numerical varibles\n",

"numerical\_variables=[i for i in df.columns if df[i].dtypes!='O']\n",

"print(f'There are {len(numerical\_variables)} numerical variables.')\n",

"print('--'\*39)\n",

"print(numerical\_variables)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "gp56V8o1FLoh"

},

"source": [

"There are 19 numerical variables."

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"metadata": {

"id": "l6zBkC0jcgMA",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "0e58f1b4-ed14-4f3f-95a5-cd4ec1fc0701"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"is\_canceled : [0 1]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"arrival\_date\_year : [2015 2016 2017]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"arrival\_date\_week\_number : [27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50\n",

" 51 52 53 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21\n",

" 22 23 24 25 26]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"arrival\_date\_day\_of\_month : [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24\n",

" 25 26 27 28 29 30 31]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"stays\_in\_weekend\_nights : [ 0 1 2 4 3 6 13 8 5 7 12 9 16 18 19 10 14]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"stays\_in\_week\_nights : [ 0 1 2 3 4 5 10 11 8 6 7 15 9 12 33 20 14 16 21 13 30 19 24 40\n",

" 22 42 50 25 17 32 26 18 34 35 41]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"adults : [ 2 1 3 4 40 26 50 27 55 0 20 6 5 10]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"children : [ 0. 1. 2. 10. 3.]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"babies : [ 0 1 2 10 9]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"is\_repeated\_guest : [0 1]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"previous\_cancellations : [ 0 1 2 3 26 25 14 4 24 19 5 21 6 13 11]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"previous\_bookings\_not\_canceled : [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 21 22 23 24\n",

" 25 27 28 29 30 19 26 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47\n",

" 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71\n",

" 72]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"booking\_changes : [ 3 4 0 1 2 5 17 6 8 7 10 16 9 13 12 20 14 15 11 21 18]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"days\_in\_waiting\_list : [ 0 50 47 65 122 75 101 150 125 14 60 34 100 22 121 61 39 5\n",

" 1 8 107 43 52 2 11 142 116 13 44 97 83 4 113 18 20 185\n",

" 93 109 6 37 105 154 64 99 38 48 33 77 21 80 59 40 58 89\n",

" 53 49 69 87 91 57 111 79 98 85 63 15 3 41 224 31 56 187\n",

" 176 71 55 96 236 259 207 215 160 120 30 32 27 62 24 108 147 379\n",

" 70 35 178 330 223 174 162 391 68 193 10 76 16 28 9 165 17 25\n",

" 46 7 84 175 183 23 117 12 54 26 73 45 19 42 72 81 92 74\n",

" 167 36]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"required\_car\_parking\_spaces : [0 1 2 8 3]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"total\_of\_special\_requests : [0 1 3 2 4 5]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"Dataset having 16 descrete variables\n"

]

}

],

"source": [

"# Obtaining Descrete varibles from Numerical varibles\n",

"# Variables having less than 150 categories are consider as descrete variable\n",

"descrete\_variavles=[]\n",

"for i in numerical\_variables:\n",

" if len(df[i].value\_counts())<=150:\n",

" descrete\_variavles.append(i)\n",

" print(i,':',df[i].unique())\n",

" print('\_\_'\*39)\n",

" else:\n",

" pass\n",

"\n",

"print(f'Dataset having {len(descrete\_variavles)} descrete variables')"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "M5El-N2k599f"

},

"source": [

"Dataset having 16 descrete variables. Discrete varibales are numerical but actually they are categorical."

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"metadata": {

"id": "2MD1rGilhtsG",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "3b30d431-51f8-44c5-8736-ef9452306c0a"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Dataset having 3 contineous variables\n",

"------------------------------------------------------------------------------\n",

"['lead\_time', 'agent', 'adr']\n"

]

}

],

"source": [

"# Obtaining contineous variables from numerical variables\n",

"contineous\_variables=[i for i in numerical\_variables if i not in descrete\_variavles]\n",

"print(f'Dataset having {len(contineous\_variables)} contineous variables')\n",

"print('--'\*39)\n",

"print(contineous\_variables)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "EBGyUZyaFLok"

},

"source": [

"Dataset having 3 contineous variables."

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"id": "FnZYT-ZZfTfi",

"outputId": "53b0c5e9-a4fa-48b4-a3e8-7d857a15e8e0"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x432 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x432 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x432 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Checking for outliers for contineous variables\n",

"\n",

"from scipy.stats import norm\n",

"for i in contineous\_variables:\n",

" plt.figure(figsize=(15,6))\n",

" plt.subplot(1,2,1)\n",

" ax=sns.boxplot(data=df[i])\n",

" ax.set\_title(f'{i}')\n",

" ax.set\_ylabel(i)\n",

"\n",

" plt.subplot(1,2,2)\n",

" ax=sns.distplot(df[i], fit=norm)\n",

" ax.set\_title(f'skewness of {i} : {df[i].skew()}')\n",

" ax.set\_xlabel(i)\n",

" print('\_\_'\*39)\n",

" plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "whmjpIALpOAl"

},

"source": [

"Outliers were found in the variables lead\_time and adr, but not in the variable agent."

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"metadata": {

"id": "wPA9JCD6jv6h"

},

"outputs": [],

"source": [

"# Using Inter Quartile range in skew symmetric curve for removing outliers\n",

"\n",

"# Outlier columns\n",

"outliers\_columns=['lead\_time','adr']\n",

"\n",

"# Copy dataset as new dataset\n",

"new\_df=df.copy()\n",

"\n",

"# Capping dataset\n",

"for i in outliers\_columns:\n",

" #Findng IQR\n",

" Q1=new\_df[i].quantile(0.25)\n",

" Q3=new\_df[i].quantile(0.75)\n",

" IQR=Q3-Q1\n",

" \n",

" # Defining lower and upper limit\n",

" lower\_limit =new\_df[i].quantile(0.25)-1.5\*IQR\n",

" upper\_limit =new\_df[i].quantile(0.75)+1.5\*IQR\n",

" \n",

" # Applying lower and upper limit to each variables\n",

" new\_df.loc[(new\_df[i] > upper\_limit),i] = upper\_limit\n",

" new\_df.loc[(new\_df[i] < lower\_limit),i] = lower\_limit"

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"metadata": {

"id": "6woAfWXv0dXt",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 827

},

"outputId": "5962c618-f86a-4f27-9d3f-e1200da25e98"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x432 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x432 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Checking for outliers for contineous variables\n",

"from scipy.stats import norm\n",

"for i in outliers\_columns:\n",

" plt.figure(figsize=(15,6))\n",

" plt.subplot(1,2,1)\n",

" ax=sns.boxplot(data=new\_df[i])\n",

" ax.set\_title(f'{i}')\n",

" ax.set\_ylabel(i)\n",

"\n",

" plt.subplot(1,2,2)\n",

" ax=sns.distplot(new\_df[i], fit=norm)\n",

" ax.set\_title(f'skewness of {i} : {new\_df[i].skew()}')\n",

" ax.set\_xlabel(i)\n",

" print('\_\_'\*50)\n",

" plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Ahx5KbUqFnkN"

},

"source": [

"Ouliers in the lead\_time and adr variables were removed. "

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"metadata": {

"id": "uezrZIV\_FLoo",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 364

},

"outputId": "24ff43e0-b110-46c1-d555-21a8470744e5"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" is\_canceled lead\_time arrival\_date\_year \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 0.274898 78.363987 2016.210296 \n",

"std 0.446466 81.047312 0.686102 \n",

"min 0.000000 0.000000 2015.000000 \n",

"25% 0.000000 11.000000 2016.000000 \n",

"50% 0.000000 49.000000 2016.000000 \n",

"75% 1.000000 125.000000 2017.000000 \n",

"max 1.000000 296.000000 2017.000000 \n",

"\n",

" arrival\_date\_week\_number arrival\_date\_day\_of\_month \\\n",

"count 87396.000000 87396.000000 \n",

"mean 26.838334 15.815541 \n",

"std 13.674572 8.835146 \n",

"min 1.000000 1.000000 \n",

"25% 16.000000 8.000000 \n",

"50% 27.000000 16.000000 \n",

"75% 37.000000 23.000000 \n",

"max 53.000000 31.000000 \n",

"\n",

" stays\_in\_weekend\_nights stays\_in\_week\_nights adults \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 1.005263 2.625395 1.875795 \n",

"std 1.031921 2.053584 0.626500 \n",

"min 0.000000 0.000000 0.000000 \n",

"25% 0.000000 1.000000 2.000000 \n",

"50% 1.000000 2.000000 2.000000 \n",

"75% 2.000000 4.000000 2.000000 \n",

"max 19.000000 50.000000 55.000000 \n",

"\n",

" children babies is\_repeated\_guest previous\_cancellations \\\n",

"count 87396.000000 87396.000000 87396.000000 87396.000000 \n",

"mean 0.138633 0.010824 0.039075 0.030413 \n",

"std 0.455871 0.113597 0.193775 0.369145 \n",

"min 0.000000 0.000000 0.000000 0.000000 \n",

"25% 0.000000 0.000000 0.000000 0.000000 \n",

"50% 0.000000 0.000000 0.000000 0.000000 \n",

"75% 0.000000 0.000000 0.000000 0.000000 \n",

"max 10.000000 10.000000 1.000000 26.000000 \n",

"\n",

" previous\_bookings\_not\_canceled booking\_changes agent \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 0.183990 0.271603 82.260287 \n",

"std 1.731894 0.727245 109.061182 \n",

"min 0.000000 0.000000 1.000000 \n",

"25% 0.000000 0.000000 9.000000 \n",

"50% 0.000000 0.000000 9.000000 \n",

"75% 0.000000 0.000000 234.000000 \n",

"max 72.000000 21.000000 535.000000 \n",

"\n",

" days\_in\_waiting\_list adr required\_car\_parking\_spaces \\\n",

"count 87396.000000 87396.000000 87396.000000 \n",

"mean 0.749565 105.344502 0.084226 \n",

"std 10.015731 49.190333 0.281533 \n",

"min 0.000000 -6.380000 0.000000 \n",

"25% 0.000000 72.000000 0.000000 \n",

"50% 0.000000 98.100000 0.000000 \n",

"75% 0.000000 134.000000 0.000000 \n",

"max 391.000000 227.000000 8.000000 \n",

"\n",

" total\_of\_special\_requests \n",

"count 87396.000000 \n",

"mean 0.698567 \n",

"std 0.831946 \n",

"min 0.000000 \n",

"25% 0.000000 \n",

"50% 0.000000 \n",

"75% 1.000000 \n",

"max 5.000000 "

],

"text/html": [

"\n",

" <div id=\"df-81c36ee5-9fda-45d1-bad4-6baf9ebcc0cb\">\n",

" <div class=\"colab-df-container\">\n",

" <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>is\_canceled</th>\n",

" <th>lead\_time</th>\n",

" <th>arrival\_date\_year</th>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <th>adults</th>\n",

" <th>children</th>\n",

" <th>babies</th>\n",

" <th>is\_repeated\_guest</th>\n",

" <th>previous\_cancellations</th>\n",

" <th>previous\_bookings\_not\_canceled</th>\n",

" <th>booking\_changes</th>\n",

" <th>agent</th>\n",

" <th>days\_in\_waiting\_list</th>\n",

" <th>adr</th>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <th>total\_of\_special\_requests</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" <td>87396.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>0.274898</td>\n",

" <td>78.363987</td>\n",

" <td>2016.210296</td>\n",

" <td>26.838334</td>\n",

" <td>15.815541</td>\n",

" <td>1.005263</td>\n",

" <td>2.625395</td>\n",

" <td>1.875795</td>\n",

" <td>0.138633</td>\n",

" <td>0.010824</td>\n",

" <td>0.039075</td>\n",

" <td>0.030413</td>\n",

" <td>0.183990</td>\n",

" <td>0.271603</td>\n",

" <td>82.260287</td>\n",

" <td>0.749565</td>\n",

" <td>105.344502</td>\n",

" <td>0.084226</td>\n",

" <td>0.698567</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>0.446466</td>\n",

" <td>81.047312</td>\n",

" <td>0.686102</td>\n",

" <td>13.674572</td>\n",

" <td>8.835146</td>\n",

" <td>1.031921</td>\n",

" <td>2.053584</td>\n",

" <td>0.626500</td>\n",

" <td>0.455871</td>\n",

" <td>0.113597</td>\n",

" <td>0.193775</td>\n",

" <td>0.369145</td>\n",

" <td>1.731894</td>\n",

" <td>0.727245</td>\n",

" <td>109.061182</td>\n",

" <td>10.015731</td>\n",

" <td>49.190333</td>\n",

" <td>0.281533</td>\n",

" <td>0.831946</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>2015.000000</td>\n",

" <td>1.000000</td>\n",

" <td>1.000000</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>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" <td>0.000000</td>\n",

" <td>-6.380000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>0.000000</td>\n",

" <td>11.000000</td>\n",

" <td>2016.000000</td>\n",

" <td>16.000000</td>\n",

" <td>8.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" <td>2.000000</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>0.000000</td>\n",

" <td>9.000000</td>\n",

" <td>0.000000</td>\n",

" <td>72.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>0.000000</td>\n",

" <td>49.000000</td>\n",

" <td>2016.000000</td>\n",

" <td>27.000000</td>\n",

" <td>16.000000</td>\n",

" <td>1.000000</td>\n",

" <td>2.000000</td>\n",

" <td>2.000000</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>0.000000</td>\n",

" <td>9.000000</td>\n",

" <td>0.000000</td>\n",

" <td>98.100000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>1.000000</td>\n",

" <td>125.000000</td>\n",

" <td>2017.000000</td>\n",

" <td>37.000000</td>\n",

" <td>23.000000</td>\n",

" <td>2.000000</td>\n",

" <td>4.000000</td>\n",

" <td>2.000000</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>0.000000</td>\n",

" <td>234.000000</td>\n",

" <td>0.000000</td>\n",

" <td>134.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>1.000000</td>\n",

" <td>296.000000</td>\n",

" <td>2017.000000</td>\n",

" <td>53.000000</td>\n",

" <td>31.000000</td>\n",

" <td>19.000000</td>\n",

" <td>50.000000</td>\n",

" <td>55.000000</td>\n",

" <td>10.000000</td>\n",

" <td>10.000000</td>\n",

" <td>1.000000</td>\n",

" <td>26.000000</td>\n",

" <td>72.000000</td>\n",

" <td>21.000000</td>\n",

" <td>535.000000</td>\n",

" <td>391.000000</td>\n",

" <td>227.000000</td>\n",

" <td>8.000000</td>\n",

" <td>5.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-81c36ee5-9fda-45d1-bad4-6baf9ebcc0cb')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-81c36ee5-9fda-45d1-bad4-6baf9ebcc0cb button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-81c36ee5-9fda-45d1-bad4-6baf9ebcc0cb');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 32

}

],

"source": [

"# Describe outlier free new\_df \n",

"new\_df.describe()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "IukQb6XoFLoo"

},

"source": [

"#### \*\*<u>Variables Information<u>\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zu\_jPLhPFLop"

},

"source": [

"- hotel:- Name of hotel ( City or Resort)\n",

"- is\_canceled:- Whether the booking is canceled or not (0 for no canceled and 1 for canceled)\n",

"- lead\_time:- time (in days) between booking transaction and actual arrival.\n",

"- arrival\_date\_year:- Year of arrival\n",

"- arrival\_date\_month:- month of arrival\n",

"- arrival\_date\_week\_number:- week number of arrival date.\n",

"- arrival\_date\_day\_of\_month:- Day of month of arrival date\n",

"- stays\_in\_weekend\_nights:- No. of weekend nights spent in a hotel\n",

"- stays\_in\_week\_nights:- No. of weeknights spent in a hotel\n",

"- adults:- No. of adults in single booking record.\n",

"- children:- No. of children in single booking record.\n",

"- babies:- No. of babies in single booking record. \n",

"- meal:- Type of meal chosen \n",

"- country:- Country of origin of customers (as mentioned by them)\n",

"- market\_segment:- What segment via booking was made and for what purpose.\n",

"- distribution\_channel:- Via which medium booking was made.\n",

"- is\_repeated\_guest:- Whether the customer has made any booking before(0 for No and 1 for Yes)\n",

"- previous\_cancellations:- No. of previous canceled bookings.\n",

"- previous\_bookings\_not\_canceled:- No. of previous non-canceled bookings.\n",

"- reserved\_room\_type:- Room type reserved by a customer.\n",

"- assigned\_room\_type:- Room type assigned to the customer.\n",

"- booking\_changes:- No. of booking changes done by customers\n",

"- deposit\_type:- Type of deposit at the time of making a booking (No deposit/ Refundable/ No refund)\n",

"- agent:- Id of agent for booking\n",

"- company:- Id of the company making a booking\n",

"- days\_in\_waiting\_list:- No. of days on waiting list.\n",

"- customer\_type:- Type of customer(Transient, Group, etc.)\n",

"- adr:- Average Daily Rate.\n",

"- required\_car\_parking\_spaces:- No. of car parking asked in booking\n",

"- total\_of\_special\_requests:- total no. of special request.\n",

"- reservation\_status:- Whether a customer has checked out or canceled,or not showed \n",

"- reservation\_status\_date:- Date of making reservation status."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "tKe32PtqFLoq"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ULwkuAaXFLoq"

},

"source": [

"### <u> \*\*Data Wrangling and Feature Engineering\*\* <u>"

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"metadata": {

"id": "V3HHtcf\_FLoq",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 661

},

"outputId": "03243e57-38d1-4615-82d9-bb919202731c"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" hotel is\_canceled lead\_time arrival\_date\_year \\\n",

"0 Resort Hotel 0 296 2015 \n",

"1 Resort Hotel 0 296 2015 \n",

"2 Resort Hotel 0 7 2015 \n",

"3 Resort Hotel 0 13 2015 \n",

"4 Resort Hotel 0 14 2015 \n",

"... ... ... ... ... \n",

"119385 City Hotel 0 23 2017 \n",

"119386 City Hotel 0 102 2017 \n",

"119387 City Hotel 0 34 2017 \n",

"119388 City Hotel 0 109 2017 \n",

"119389 City Hotel 0 205 2017 \n",

"\n",

" arrival\_date\_month arrival\_date\_week\_number \\\n",

"0 July 27 \n",

"1 July 27 \n",

"2 July 27 \n",

"3 July 27 \n",

"4 July 27 \n",

"... ... ... \n",

"119385 August 35 \n",

"119386 August 35 \n",

"119387 August 35 \n",

"119388 August 35 \n",

"119389 August 35 \n",

"\n",

" arrival\_date\_day\_of\_month stays\_in\_weekend\_nights \\\n",

"0 1 0 \n",

"1 1 0 \n",

"2 1 0 \n",

"3 1 0 \n",

"4 1 0 \n",

"... ... ... \n",

"119385 30 2 \n",

"119386 31 2 \n",

"119387 31 2 \n",

"119388 31 2 \n",

"119389 29 2 \n",

"\n",

" stays\_in\_week\_nights adults ... booking\_changes deposit\_type \\\n",

"0 0 2 ... 3 No Deposit \n",

"1 0 2 ... 4 No Deposit \n",

"2 1 1 ... 0 No Deposit \n",

"3 1 1 ... 0 No Deposit \n",

"4 2 2 ... 0 No Deposit \n",

"... ... ... ... ... ... \n",

"119385 5 2 ... 0 No Deposit \n",

"119386 5 3 ... 0 No Deposit \n",

"119387 5 2 ... 0 No Deposit \n",

"119388 5 2 ... 0 No Deposit \n",

"119389 7 2 ... 0 No Deposit \n",

"\n",

" agent days\_in\_waiting\_list customer\_type adr \\\n",

"0 9.0 0 Transient 0.00 \n",

"1 9.0 0 Transient 0.00 \n",

"2 9.0 0 Transient 75.00 \n",

"3 304.0 0 Transient 75.00 \n",

"4 240.0 0 Transient 98.00 \n",

"... ... ... ... ... \n",

"119385 394.0 0 Transient 96.14 \n",

"119386 9.0 0 Transient 225.43 \n",

"119387 9.0 0 Transient 157.71 \n",

"119388 89.0 0 Transient 104.40 \n",

"119389 9.0 0 Transient 151.20 \n",

"\n",

" required\_car\_parking\_spaces total\_of\_special\_requests \\\n",

"0 0 0 \n",

"1 0 0 \n",

"2 0 0 \n",

"3 0 0 \n",

"4 0 1 \n",

"... ... ... \n",

"119385 0 0 \n",

"119386 0 2 \n",

"119387 0 4 \n",

"119388 0 0 \n",

"119389 0 2 \n",

"\n",

" reservation\_status reservation\_status\_date \n",

"0 Check-Out 2015-07-01 \n",

"1 Check-Out 2015-07-01 \n",

"2 Check-Out 2015-07-02 \n",

"3 Check-Out 2015-07-02 \n",

"4 Check-Out 2015-07-03 \n",

"... ... ... \n",

"119385 Check-Out 2017-09-06 \n",

"119386 Check-Out 2017-09-07 \n",

"119387 Check-Out 2017-09-07 \n",

"119388 Check-Out 2017-09-07 \n",

"119389 Check-Out 2017-09-07 \n",

"\n",

"[87396 rows x 31 columns]"

],

"text/html": [

"\n",

" <div id=\"df-a13b9c63-4d52-46ce-9943-b14ee3ac05a6\">\n",

" <div class=\"colab-df-container\">\n",

" <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>hotel</th>\n",

" <th>is\_canceled</th>\n",

" <th>lead\_time</th>\n",

" <th>arrival\_date\_year</th>\n",

" <th>arrival\_date\_month</th>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <th>adults</th>\n",

" <th>...</th>\n",

" <th>booking\_changes</th>\n",

" <th>deposit\_type</th>\n",

" <th>agent</th>\n",

" <th>days\_in\_waiting\_list</th>\n",

" <th>customer\_type</th>\n",

" <th>adr</th>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <th>total\_of\_special\_requests</th>\n",

" <th>reservation\_status</th>\n",

" <th>reservation\_status\_date</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>296</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>3</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>0.00</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-01</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>296</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>4</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>0.00</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-01</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>7</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>75.00</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-02</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>13</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>304.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>75.00</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-02</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>14</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>240.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>98.00</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-03</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",

" <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",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119385</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>23</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>30</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>394.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>96.14</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-06</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119386</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>102</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>31</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>3</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>225.43</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119387</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>34</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>31</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>157.71</td>\n",

" <td>0</td>\n",

" <td>4</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119388</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>109</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>31</td>\n",

" <td>2</td>\n",

" <td>5</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>89.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>104.40</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" <tr>\n",

" <th>119389</th>\n",

" <td>City Hotel</td>\n",

" <td>0</td>\n",

" <td>205</td>\n",

" <td>2017</td>\n",

" <td>August</td>\n",

" <td>35</td>\n",

" <td>29</td>\n",

" <td>2</td>\n",

" <td>7</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>No Deposit</td>\n",

" <td>9.0</td>\n",

" <td>0</td>\n",

" <td>Transient</td>\n",

" <td>151.20</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>Check-Out</td>\n",

" <td>2017-09-07</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>87396 rows × 31 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-a13b9c63-4d52-46ce-9943-b14ee3ac05a6')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-a13b9c63-4d52-46ce-9943-b14ee3ac05a6 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-a13b9c63-4d52-46ce-9943-b14ee3ac05a6');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 33

}

],

"source": [

"# Dataset\n",

"new\_df"

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"metadata": {

"id": "tzJ-qIL7FLor",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "7125017f-b45f-4ee6-cec7-49333448ed29"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"hotel:['Resort Hotel' 'City Hotel']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"is\_canceled:[0 1]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"lead\_time:[296 7 13 14 0 9 85 75 23 35 68 18 37 12 72 127 78 48\n",

" 60 77 99 118 95 96 69 45 40 15 36 43 70 16 107 47 113 90\n",

" 50 93 76 3 1 10 5 17 51 71 63 62 101 2 81 79 21 109\n",

" 102 4 98 92 26 73 115 86 52 29 30 33 32 8 100 44 80 97\n",

" 64 39 34 27 82 94 110 111 84 66 104 28 258 112 65 67 55 88\n",

" 54 292 83 105 280 24 103 249 22 91 11 108 106 31 87 41 117 59\n",

" 53 58 116 42 38 56 49 6 57 19 25 123 46 89 61 130 74 119\n",

" 20 286 136 129 124 131 140 114 139 122 137 126 120 128 135 150 143 151\n",

" 132 125 157 147 138 156 164 159 160 161 149 154 163 155 142 144 248 175\n",

" 146 170 166 167 148 165 172 171 145 121 178 173 152 158 185 183 177 200\n",

" 192 207 174 134 283 153 197 133 241 193 235 194 261 260 216 169 209 238\n",

" 215 141 189 187 223 284 214 202 211 168 230 203 188 232 219 162 196 190\n",

" 259 228 176 250 201 186 199 180 206 205 224 222 182 210 275 212 229 218\n",

" 208 191 181 179 246 255 226 288 253 252 262 236 256 234 254 213 237 198\n",

" 195 239 263 265 274 217 220 221 233 257 227 276 225 264 277 204 290 266\n",

" 270 294 282 251 291 269 240 271 184 231 268 247 273 267 244 293 272 242\n",

" 295 285 243 245 279 281 278 289 287]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"arrival\_date\_year:[2015 2016 2017]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"arrival\_date\_month:['July' 'August' 'September' 'October' 'November' 'December' 'January'\n",

" 'February' 'March' 'April' 'May' 'June']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"arrival\_date\_week\_number:[27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50\n",

" 51 52 53 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21\n",

" 22 23 24 25 26]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"arrival\_date\_day\_of\_month:[ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24\n",

" 25 26 27 28 29 30 31]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"stays\_in\_weekend\_nights:[ 0 1 2 4 3 6 13 8 5 7 12 9 16 18 19 10 14]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"stays\_in\_week\_nights:[ 0 1 2 3 4 5 10 11 8 6 7 15 9 12 33 20 14 16 21 13 30 19 24 40\n",

" 22 42 50 25 17 32 26 18 34 35 41]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"adults:[ 2 1 3 4 40 26 50 27 55 0 20 6 5 10]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"children:[ 0. 1. 2. 10. 3.]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"babies:[ 0 1 2 10 9]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"meal:['BB' 'FB' 'HB' 'SC' 'Undefined']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"country:['PRT' 'GBR' 'USA' 'ESP' 'IRL' 'FRA' 'ROU' 'NOR' 'OMN' 'ARG' 'POL' 'DEU'\n",

" 'BEL' 'CHE' 'CN' 'GRC' 'ITA' 'NLD' 'DNK' 'RUS' 'SWE' 'AUS' 'EST' 'CZE'\n",

" 'BRA' 'FIN' 'MOZ' 'BWA' 'LUX' 'SVN' 'ALB' 'IND' 'CHN' 'MEX' 'MAR' 'UKR'\n",

" 'SMR' 'LVA' 'PRI' 'SRB' 'CHL' 'AUT' 'BLR' 'LTU' 'TUR' 'ZAF' 'AGO' 'ISR'\n",

" 'CYM' 'ZMB' 'CPV' 'ZWE' 'DZA' 'KOR' 'CRI' 'HUN' 'ARE' 'TUN' 'JAM' 'HRV'\n",

" 'HKG' 'IRN' 'GEO' 'AND' 'GIB' 'URY' 'JEY' 'CAF' 'CYP' 'COL' 'GGY' 'KWT'\n",

" 'NGA' 'MDV' 'VEN' 'SVK' 'FJI' 'KAZ' 'PAK' 'IDN' 'LBN' 'PHL' 'SEN' 'SYC'\n",

" 'AZE' 'BHR' 'NZL' 'THA' 'DOM' 'MKD' 'MYS' 'ARM' 'JPN' 'LKA' 'CUB' 'CMR'\n",

" 'BIH' 'MUS' 'COM' 'SUR' 'UGA' 'BGR' 'CIV' 'JOR' 'SYR' 'SGP' 'BDI' 'SAU'\n",

" 'VNM' 'PLW' 'QAT' 'EGY' 'PER' 'MLT' 'MWI' 'ECU' 'MDG' 'ISL' 'UZB' 'NPL'\n",

" 'BHS' 'MAC' 'TGO' 'TWN' 'DJI' 'STP' 'KNA' 'ETH' 'IRQ' 'HND' 'RWA' 'KHM'\n",

" 'MCO' 'BGD' 'IMN' 'TJK' 'NIC' 'BEN' 'VGB' 'TZA' 'GAB' 'GHA' 'TMP' 'GLP'\n",

" 'KEN' 'LIE' 'GNB' 'MNE' 'UMI' 'MYT' 'FRO' 'MMR' 'PAN' 'BFA' 'LBY' 'MLI'\n",

" 'NAM' 'BOL' 'PRY' 'BRB' 'ABW' 'AIA' 'SLV' 'DMA' 'PYF' 'GUY' 'LCA' 'ATA'\n",

" 'GTM' 'ASM' 'MRT' 'NCL' 'KIR' 'SDN' 'ATF' 'SLE' 'LAO']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"market\_segment:['Direct' 'Corporate' 'Online TA' 'Offline TA/TO' 'Complementary' 'Groups'\n",

" 'Undefined' 'Aviation']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"distribution\_channel:['Direct' 'Corporate' 'TA/TO' 'Undefined' 'GDS']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"is\_repeated\_guest:[0 1]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"previous\_cancellations:[ 0 1 2 3 26 25 14 4 24 19 5 21 6 13 11]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"previous\_bookings\_not\_canceled:[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 21 22 23 24\n",

" 25 27 28 29 30 19 26 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47\n",

" 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71\n",

" 72]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"reserved\_room\_type:['C' 'A' 'D' 'E' 'G' 'F' 'H' 'L' 'P' 'B']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"assigned\_room\_type:['C' 'A' 'D' 'E' 'G' 'F' 'I' 'B' 'H' 'P' 'L' 'K']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"booking\_changes:[ 3 4 0 1 2 5 17 6 8 7 10 16 9 13 12 20 14 15 11 21 18]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"deposit\_type:['No Deposit' 'Refundable' 'Non Refund']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"agent:[ 9. 304. 240. 303. 15. 241. 8. 250. 115. 5. 175. 134. 156. 243.\n",

" 242. 3. 105. 40. 147. 306. 184. 96. 2. 127. 95. 146. 177. 6.\n",

" 143. 244. 149. 167. 300. 171. 305. 67. 196. 152. 142. 261. 104. 36.\n",

" 26. 29. 258. 110. 71. 181. 88. 251. 275. 69. 248. 208. 256. 314.\n",

" 126. 281. 273. 253. 185. 330. 334. 328. 326. 321. 324. 313. 38. 155.\n",

" 68. 335. 308. 332. 94. 348. 310. 339. 375. 66. 327. 387. 298. 91.\n",

" 245. 385. 257. 393. 168. 405. 249. 315. 75. 128. 307. 11. 436. 1.\n",

" 201. 183. 223. 368. 336. 291. 464. 411. 481. 10. 154. 468. 410. 390.\n",

" 440. 495. 492. 493. 434. 57. 531. 420. 483. 526. 472. 429. 16. 446.\n",

" 34. 78. 139. 252. 270. 47. 114. 301. 193. 182. 135. 350. 195. 352.\n",

" 355. 159. 363. 384. 360. 331. 367. 64. 406. 163. 414. 333. 427. 431.\n",

" 430. 426. 438. 433. 418. 441. 282. 432. 72. 450. 180. 454. 455. 59.\n",

" 451. 254. 358. 469. 165. 467. 510. 337. 476. 502. 527. 479. 508. 535.\n",

" 302. 497. 187. 13. 7. 27. 14. 22. 17. 28. 42. 20. 19. 45.\n",

" 37. 61. 39. 21. 24. 41. 50. 30. 54. 52. 12. 44. 31. 83.\n",

" 32. 63. 60. 55. 56. 89. 87. 118. 86. 85. 210. 214. 129. 179.\n",

" 138. 174. 170. 153. 93. 151. 119. 35. 173. 58. 53. 133. 79. 235.\n",

" 192. 191. 236. 162. 215. 157. 287. 132. 234. 98. 77. 103. 107. 262.\n",

" 220. 121. 205. 378. 23. 296. 290. 229. 33. 286. 276. 425. 484. 323.\n",

" 403. 219. 394. 509. 111. 423. 4. 70. 82. 81. 74. 92. 99. 90.\n",

" 112. 117. 106. 148. 158. 144. 211. 213. 216. 232. 150. 267. 227. 247.\n",

" 278. 280. 285. 289. 269. 295. 265. 288. 122. 294. 325. 341. 344. 346.\n",

" 359. 283. 364. 370. 371. 25. 141. 391. 397. 416. 404. 299. 197. 73.\n",

" 354. 444. 408. 461. 388. 453. 459. 474. 475. 480. 449.]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"days\_in\_waiting\_list:[ 0 50 47 65 122 75 101 150 125 14 60 34 100 22 121 61 39 5\n",

" 1 8 107 43 52 2 11 142 116 13 44 97 83 4 113 18 20 185\n",

" 93 109 6 37 105 154 64 99 38 48 33 77 21 80 59 40 58 89\n",

" 53 49 69 87 91 57 111 79 98 85 63 15 3 41 224 31 56 187\n",

" 176 71 55 96 236 259 207 215 160 120 30 32 27 62 24 108 147 379\n",

" 70 35 178 330 223 174 162 391 68 193 10 76 16 28 9 165 17 25\n",

" 46 7 84 175 183 23 117 12 54 26 73 45 19 42 72 81 92 74\n",

" 167 36]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"customer\_type:['Transient' 'Contract' 'Transient-Party' 'Group']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"adr:[ 0. 75. 98. ... 91.02 209.25 157.71]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"required\_car\_parking\_spaces:[0 1 2 8 3]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"total\_of\_special\_requests:[0 1 3 2 4 5]\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"reservation\_status:['Check-Out' 'Canceled' 'No-Show']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"reservation\_status\_date:['2015-07-01' '2015-07-02' '2015-07-03' '2015-05-06' '2015-04-22'\n",

" '2015-06-23' '2015-07-05' '2015-07-06' '2015-07-07' '2015-07-08'\n",

" '2015-05-11' '2015-07-15' '2015-07-16' '2015-05-29' '2015-05-19'\n",

" '2015-06-19' '2015-05-23' '2015-05-18' '2015-07-09' '2015-06-02'\n",

" '2015-07-13' '2015-07-04' '2015-06-29' '2015-06-16' '2015-06-18'\n",

" '2015-06-12' '2015-06-09' '2015-05-26' '2015-07-11' '2015-07-12'\n",

" '2015-07-17' '2015-04-15' '2015-05-13' '2015-07-10' '2015-05-20'\n",

" '2015-05-12' '2015-07-14' '2015-06-17' '2015-05-01' '2015-03-30'\n",

" '2015-07-19' '2015-06-03' '2015-06-26' '2015-05-14' '2015-07-20'\n",

" '2015-05-07' '2015-05-28' '2015-04-13' '2015-03-25' '2015-07-21'\n",

" '2015-06-27' '2015-07-18' '2015-07-23' '2015-06-08' '2015-06-22'\n",

" '2015-06-24' '2015-03-05' '2015-06-01' '2015-04-24' '2015-07-22'\n",

" '2015-05-27' '2015-04-06' '2015-04-11' '2015-07-25' '2015-07-28'\n",

" '2015-07-29' '2015-06-25' '2015-07-24' '2015-06-05' '2015-06-30'\n",

" '2015-06-13' '2015-06-11' '2015-07-30' '2015-07-27' '2015-04-29'\n",

" '2015-06-04' '2015-07-26' '2015-08-01' '2015-08-02' '2015-06-15'\n",

" '2015-04-23' '2015-07-31' '2015-05-25' '2015-08-03' '2015-04-17'\n",

" '2015-08-04' '2015-08-06' '2015-05-15' '2015-05-09' '2015-03-17'\n",

" '2015-05-22' '2015-08-07' '2015-04-04' '2015-08-05' '2015-08-08'\n",

" '2015-08-10' '2015-05-04' '2015-06-06' '2015-08-09' '2015-08-15'\n",

" '2015-08-11' '2015-03-28' '2015-08-14' '2015-08-12' '2015-08-16'\n",

" '2015-05-16' '2015-08-21' '2015-08-13' '2015-08-17' '2015-04-20'\n",

" '2015-08-18' '2015-08-23' '2015-08-22' '2015-08-19' '2015-08-20'\n",

" '2015-08-29' '2015-03-31' '2015-05-30' '2015-08-25' '2015-04-14'\n",

" '2015-08-24' '2015-03-24' '2015-05-21' '2015-08-28' '2015-08-26'\n",

" '2015-08-27' '2015-08-30' '2015-08-31' '2015-09-06' '2015-09-03'\n",

" '2015-09-04' '2015-09-02' '2015-09-01' '2015-09-05' '2015-06-20'\n",

" '2015-09-07' '2015-09-10' '2015-09-11' '2015-09-08' '2015-09-09'\n",

" '2015-09-13' '2015-09-15' '2015-04-10' '2015-01-02' '2014-11-18'\n",

" '2015-09-12' '2015-09-17' '2015-09-14' '2015-04-07' '2015-09-19'\n",

" '2015-09-16' '2015-09-20' '2015-01-18' '2015-10-23' '2015-01-22'\n",

" '2015-01-01' '2015-09-22' '2015-09-24' '2015-09-18' '2015-09-21'\n",

" '2015-09-30' '2015-09-25' '2015-09-27' '2015-09-28' '2015-10-12'\n",

" '2015-09-29' '2015-09-23' '2015-10-01' '2015-09-26' '2015-04-18'\n",

" '2015-10-02' '2015-10-04' '2015-10-08' '2015-10-03' '2015-10-07'\n",

" '2015-10-09' '2015-10-11' '2015-10-05' '2015-10-06' '2015-10-10'\n",

" '2015-10-14' '2015-10-15' '2015-10-18' '2015-10-13' '2015-10-20'\n",

" '2015-10-19' '2015-10-31' '2015-10-16' '2015-10-21' '2015-10-22'\n",

" '2015-10-17' '2015-10-24' '2015-10-25' '2015-10-28' '2015-10-27'\n",

" '2015-10-26' '2015-10-30' '2015-11-05' '2015-10-29' '2015-11-03'\n",

" '2015-11-07' '2015-11-04' '2015-11-01' '2015-11-02' '2015-11-17'\n",

" '2015-11-06' '2015-11-10' '2015-11-08' '2015-11-09' '2015-11-15'\n",

" '2015-11-16' '2015-11-11' '2015-11-12' '2015-11-14' '2015-11-13'\n",

" '2015-11-18' '2015-11-22' '2015-11-19' '2015-11-21' '2015-11-20'\n",

" '2015-11-24' '2015-11-25' '2015-11-23' '2015-11-28' '2015-11-26'\n",

" '2015-11-27' '2015-11-29' '2015-12-04' '2015-12-01' '2015-12-06'\n",

" '2015-12-08' '2015-12-02' '2015-12-03' '2015-12-31' '2015-12-05'\n",

" '2015-12-10' '2015-12-17' '2015-11-30' '2015-12-12' '2015-12-07'\n",

" '2016-01-05' '2015-12-11' '2015-12-13' '2015-12-15' '2015-12-16'\n",

" '2015-12-19' '2015-12-18' '2015-12-26' '2015-12-27' '2015-12-22'\n",

" '2015-12-23' '2015-12-24' '2015-12-29' '2015-12-28' '2015-12-20'\n",

" '2015-12-30' '2016-01-02' '2016-01-01' '2015-12-25' '2016-01-03'\n",

" '2016-01-04' '2016-01-11' '2016-01-07' '2015-12-21' '2016-01-09'\n",

" '2016-01-10' '2016-01-08' '2016-01-06' '2016-01-12' '2016-01-13'\n",

" '2016-01-23' '2016-02-09' '2016-01-15' '2016-01-16' '2016-01-17'\n",

" '2016-01-19' '2016-01-18' '2016-01-21' '2016-01-24' '2016-01-22'\n",

" '2016-01-29' '2016-01-27' '2016-01-25' '2016-03-08' '2016-01-26'\n",

" '2016-01-20' '2016-01-30' '2016-02-01' '2016-02-02' '2016-02-08'\n",

" '2016-02-07' '2016-01-28' '2016-02-05' '2016-02-03' '2016-02-13'\n",

" '2016-02-10' '2016-02-04' '2016-02-12' '2016-02-11' '2016-02-16'\n",

" '2016-02-14' '2016-02-15' '2016-02-20' '2016-02-06' '2016-01-14'\n",

" '2016-02-17' '2016-02-21' '2016-02-24' '2016-02-25' '2016-02-19'\n",

" '2016-02-18' '2016-02-26' '2016-02-23' '2016-03-05' '2016-02-22'\n",

" '2016-02-27' '2016-03-03' '2016-03-24' '2016-03-04' '2016-02-29'\n",

" '2016-03-01' '2016-03-02' '2016-03-30' '2016-03-07' '2016-03-14'\n",

" '2016-03-21' '2016-03-09' '2016-03-12' '2016-03-22' '2016-03-10'\n",

" '2016-03-11' '2016-03-20' '2016-03-15' '2016-03-17' '2016-03-16'\n",

" '2016-03-19' '2016-03-27' '2016-03-18' '2016-03-26' '2016-03-31'\n",

" '2016-03-28' '2016-03-29' '2016-04-01' '2016-03-23' '2016-04-02'\n",

" '2016-03-25' '2016-03-13' '2016-04-04' '2016-04-03' '2016-04-05'\n",

" '2016-04-08' '2016-04-06' '2016-04-09' '2016-04-12' '2016-04-16'\n",

" '2016-04-17' '2016-04-27' '2016-04-14' '2016-04-18' '2016-04-21'\n",

" '2016-04-19' '2016-04-20' '2016-04-10' '2016-04-13' '2016-04-11'\n",

" '2016-04-07' '2016-04-15' '2016-04-22' '2016-04-23' '2016-04-26'\n",

" '2016-04-28' '2016-04-24' '2016-04-25' '2016-04-29' '2016-04-30'\n",

" '2016-05-01' '2016-05-10' '2016-05-02' '2016-05-07' '2016-05-08'\n",

" '2016-05-12' '2016-05-04' '2016-05-06' '2016-05-03' '2016-05-09'\n",

" '2016-05-05' '2016-05-13' '2016-05-14' '2016-05-18' '2016-05-19'\n",

" '2016-05-15' '2016-05-16' '2016-05-11' '2016-05-21' '2016-05-22'\n",

" '2016-05-20' '2016-05-24' '2016-05-25' '2016-05-26' '2016-05-23'\n",

" '2016-05-27' '2016-05-17' '2016-05-29' '2016-05-28' '2016-05-30'\n",

" '2016-05-31' '2016-06-01' '2016-06-03' '2016-06-08' '2016-06-02'\n",

" '2016-06-05' '2016-06-06' '2016-06-13' '2016-06-07' '2016-06-10'\n",

" '2016-06-11' '2016-06-16' '2016-06-12' '2016-06-14' '2016-06-17'\n",

" '2016-06-04' '2016-06-18' '2016-06-21' '2016-06-09' '2016-06-24'\n",

" '2016-06-20' '2016-06-25' '2016-06-22' '2016-06-26' '2016-06-23'\n",

" '2016-07-01' '2016-06-15' '2016-06-28' '2016-07-02' '2016-06-19'\n",

" '2016-06-27' '2016-07-04' '2016-06-30' '2016-07-05' '2016-07-08'\n",

" '2016-07-09' '2016-07-07' '2016-07-12' '2016-06-29' '2016-07-10'\n",

" '2016-07-15' '2016-07-03' '2016-07-16' '2016-07-14' '2016-07-18'\n",

" '2016-07-13' '2016-07-06' '2016-07-20' '2016-07-21' '2016-07-23'\n",

" '2016-07-19' '2016-07-11' '2016-07-28' '2016-07-17' '2016-07-25'\n",

" '2016-07-22' '2016-07-29' '2016-08-03' '2016-08-02' '2016-08-04'\n",

" '2016-08-08' '2016-08-10' '2016-08-01' '2016-08-06' '2016-03-06'\n",

" '2016-08-05' '2016-07-26' '2016-08-07' '2016-07-30' '2016-07-24'\n",

" '2016-08-12' '2016-07-27' '2016-08-13' '2016-08-18' '2016-08-16'\n",

" '2016-08-15' '2016-08-17' '2016-08-11' '2016-07-31' '2016-08-19'\n",

" '2016-09-01' '2016-08-23' '2016-08-26' '2016-08-20' '2016-08-21'\n",

" '2016-09-04' '2016-08-22' '2016-08-27' '2016-08-25' '2016-08-09'\n",

" '2016-09-05' '2016-08-24' '2016-09-10' '2016-08-29' '2016-09-09'\n",

" '2016-08-30' '2016-09-13' '2016-08-31' '2016-09-14' '2016-09-12'\n",

" '2016-09-15' '2016-08-14' '2016-09-02' '2016-09-08' '2016-09-19'\n",

" '2016-09-16' '2016-09-07' '2016-09-21' '2016-09-06' '2016-09-22'\n",

" '2016-09-17' '2016-09-20' '2016-09-03' '2016-09-26' '2016-09-23'\n",

" '2016-09-18' '2016-09-29' '2016-10-02' '2016-10-01' '2016-09-27'\n",

" '2016-09-25' '2016-10-05' '2016-09-11' '2016-09-30' '2016-10-09'\n",

" '2016-10-03' '2016-10-06' '2016-10-11' '2016-09-24' '2016-10-13'\n",

" '2016-09-28' '2016-10-08' '2016-10-07' '2016-10-16' '2016-08-28'\n",

" '2016-10-17' '2016-10-18' '2016-10-10' '2016-10-04' '2016-10-15'\n",

" '2016-10-19' '2016-10-21' '2016-10-12' '2016-10-24' '2016-10-26'\n",

" '2016-10-23' '2016-10-20' '2016-10-25' '2016-10-27' '2016-10-28'\n",

" '2016-10-30' '2016-10-29' '2016-11-01' '2016-11-04' '2016-10-14'\n",

" '2016-11-07' '2016-11-03' '2016-11-10' '2016-11-14' '2016-11-02'\n",

" '2016-10-31' '2016-11-11' '2016-11-08' '2016-11-05' '2016-11-25'\n",

" '2016-11-09' '2016-11-20' '2016-11-21' '2016-10-22' '2016-11-22'\n",

" '2016-11-16' '2016-11-23' '2016-11-17' '2016-11-06' '2016-11-15'\n",

" '2016-11-13' '2016-11-12' '2016-11-27' '2016-11-19' '2016-11-30'\n",

" '2016-11-18' '2016-12-02' '2016-12-04' '2016-11-29' '2016-12-07'\n",

" '2016-11-28' '2016-12-03' '2016-12-06' '2016-11-24' '2016-12-08'\n",

" '2016-12-05' '2016-12-10' '2016-12-13' '2016-12-14' '2016-12-16'\n",

" '2016-12-15' '2016-12-17' '2016-12-19' '2016-12-21' '2016-12-20'\n",

" '2016-12-22' '2016-12-23' '2016-12-24' '2016-12-01' '2016-12-27'\n",

" '2016-12-29' '2016-12-30' '2016-12-12' '2017-01-02' '2016-12-11'\n",

" '2017-01-03' '2017-01-04' '2017-01-01' '2016-12-26' '2017-01-06'\n",

" '2016-12-28' '2016-12-18' '2017-01-10' '2017-01-11' '2017-01-07'\n",

" '2017-01-12' '2017-01-16' '2017-01-14' '2017-01-13' '2017-01-05'\n",

" '2017-01-17' '2017-01-20' '2016-12-09' '2017-01-26' '2016-12-31'\n",

" '2017-01-23' '2017-01-27' '2017-01-28' '2017-01-19' '2017-01-25'\n",

" '2017-01-24' '2017-01-29' '2017-01-18' '2016-12-25' '2017-01-15'\n",

" '2017-01-21' '2017-02-01' '2017-02-02' '2017-01-31' '2017-02-03'\n",

" '2017-02-04' '2017-02-06' '2017-02-07' '2017-02-08' '2017-01-30'\n",

" '2017-02-09' '2017-01-09' '2017-02-11' '2017-02-10' '2017-02-12'\n",

" '2017-02-13' '2017-02-14' '2017-02-16' '2017-02-17' '2017-02-18'\n",

" '2017-02-19' '2017-02-20' '2017-02-15' '2017-02-21' '2017-02-22'\n",

" '2017-02-26' '2017-02-23' '2017-02-24' '2017-02-25' '2017-02-28'\n",

" '2017-03-05' '2017-02-27' '2017-03-03' '2017-03-06' '2017-03-02'\n",

" '2017-03-08' '2017-03-09' '2017-03-10' '2017-03-07' '2017-03-12'\n",

" '2017-03-13' '2017-03-14' '2017-03-01' '2017-03-18' '2017-03-17'\n",

" '2017-03-24' '2017-03-22' '2017-03-26' '2017-03-27' '2017-03-11'\n",

" '2017-03-28' '2017-03-29' '2017-03-30' '2017-03-31' '2017-03-19'\n",

" '2017-01-22' '2017-04-02' '2017-03-20' '2017-04-03' '2017-01-08'\n",

" '2017-03-23' '2017-04-05' '2017-02-05' '2017-04-04' '2017-03-15'\n",

" '2017-04-07' '2017-03-25' '2017-04-08' '2017-04-06' '2017-03-21'\n",

" '2017-04-10' '2017-04-01' '2017-04-11' '2017-04-13' '2017-04-15'\n",

" '2017-04-12' '2017-03-04' '2017-04-19' '2017-04-22' '2017-04-20'\n",

" '2017-05-02' '2017-04-09' '2017-04-23' '2017-04-24' '2017-04-16'\n",

" '2017-04-28' '2017-04-18' '2017-04-26' '2017-04-25' '2017-04-17'\n",

" '2017-04-21' '2017-05-03' '2017-05-04' '2017-03-16' '2017-05-05'\n",

" '2017-04-29' '2017-04-14' '2017-05-08' '2017-04-27' '2017-05-11'\n",

" '2017-05-01' '2017-05-10' '2017-05-13' '2017-05-06' '2017-05-14'\n",

" '2017-05-16' '2017-04-30' '2017-05-15' '2017-05-07' '2017-05-09'\n",

" '2017-05-17' '2017-05-21' '2017-05-12' '2017-05-22' '2017-05-24'\n",

" '2017-05-23' '2017-05-25' '2017-05-26' '2017-05-28' '2017-05-27'\n",

" '2017-05-29' '2017-05-19' '2017-05-31' '2017-05-20' '2017-06-01'\n",

" '2017-05-30' '2017-06-02' '2016-11-26' '2017-06-04' '2017-06-05'\n",

" '2017-06-06' '2017-06-07' '2017-05-18' '2017-06-09' '2017-06-10'\n",

" '2017-06-11' '2017-06-12' '2017-06-14' '2017-06-08' '2017-06-16'\n",

" '2017-06-13' '2017-06-03' '2017-06-24' '2017-06-20' '2017-06-19'\n",

" '2017-06-21' '2017-06-26' '2017-06-27' '2017-06-22' '2017-06-28'\n",

" '2017-06-15' '2017-06-29' '2017-06-30' '2017-06-18' '2017-07-04'\n",

" '2017-07-08' '2017-07-05' '2017-07-03' '2017-07-07' '2017-07-01'\n",

" '2017-07-06' '2017-07-11' '2017-07-12' '2017-06-23' '2017-07-13'\n",

" '2017-07-02' '2017-07-10' '2017-07-14' '2017-07-15' '2017-07-16'\n",

" '2017-07-18' '2017-07-17' '2017-07-19' '2017-07-20' '2017-07-21'\n",

" '2017-06-25' '2017-06-17' '2017-07-24' '2017-07-26' '2017-07-09'\n",

" '2017-07-27' '2017-07-28' '2017-07-31' '2017-07-29' '2017-07-22'\n",

" '2017-08-02' '2017-08-01' '2017-08-03' '2017-08-04' '2017-07-25'\n",

" '2017-07-23' '2017-08-09' '2017-08-10' '2017-07-30' '2017-08-07'\n",

" '2017-08-13' '2017-08-05' '2017-08-14' '2017-08-08' '2017-08-16'\n",

" '2017-08-17' '2017-08-15' '2017-08-18' '2017-08-20' '2017-08-22'\n",

" '2017-08-06' '2017-08-25' '2017-08-26' '2017-08-23' '2017-08-11'\n",

" '2017-08-27' '2017-08-21' '2017-08-29' '2017-08-31' '2017-08-12'\n",

" '2017-08-19' '2016-01-31' '2017-09-01' '2017-08-28' '2015-04-03'\n",

" '2015-01-21' '2015-01-28' '2015-01-29' '2015-01-30' '2015-02-02'\n",

" '2015-02-05' '2015-02-06' '2015-02-09' '2015-02-10' '2015-02-11'\n",

" '2015-02-12' '2015-02-19' '2015-02-20' '2015-02-23' '2015-02-24'\n",

" '2015-02-25' '2015-02-26' '2015-02-27' '2015-03-03' '2015-03-04'\n",

" '2015-03-06' '2015-03-09' '2015-03-11' '2015-03-12' '2015-03-18'\n",

" '2015-04-02' '2015-06-14' '2015-04-08' '2015-04-16' '2015-04-25'\n",

" '2015-04-28' '2015-05-08' '2017-09-06' '2016-02-28' '2015-12-09'\n",

" '2015-12-14' '2017-09-09' '2017-09-02' '2017-08-24' '2017-08-30'\n",

" '2017-09-03' '2017-09-04' '2017-09-05' '2017-09-07' '2017-09-08'\n",

" '2017-09-10' '2017-09-12' '2017-09-14' '2015-04-30' '2015-04-21'\n",

" '2015-04-05' '2015-03-13' '2015-05-05' '2015-03-29' '2015-06-10'\n",

" '2015-04-27' '2014-10-17' '2015-01-20' '2015-02-17' '2015-03-10'\n",

" '2015-03-23']\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

}

],

"source": [

"#checking unique values in each variable\n",

"for i in new\_df.columns:\n",

" print(f'{i}:{new\_df[i].unique()}')\n",

" print('\_\_'\*50)"

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"metadata": {

"id": "HUpaWL2EFLos",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "2df8d867-89c5-4c6d-ff3b-698e29d5bc7a"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"Int64Index: 87396 entries, 0 to 119389\n",

"Data columns (total 31 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 hotel 87396 non-null object \n",

" 1 is\_canceled 87396 non-null int64 \n",

" 2 lead\_time 87396 non-null int64 \n",

" 3 arrival\_date\_year 87396 non-null int64 \n",

" 4 arrival\_date\_month 87396 non-null object \n",

" 5 arrival\_date\_week\_number 87396 non-null int64 \n",

" 6 arrival\_date\_day\_of\_month 87396 non-null int64 \n",

" 7 stays\_in\_weekend\_nights 87396 non-null int64 \n",

" 8 stays\_in\_week\_nights 87396 non-null int64 \n",

" 9 adults 87396 non-null int64 \n",

" 10 children 87396 non-null float64\n",

" 11 babies 87396 non-null int64 \n",

" 12 meal 87396 non-null object \n",

" 13 country 87396 non-null object \n",

" 14 market\_segment 87396 non-null object \n",

" 15 distribution\_channel 87396 non-null object \n",

" 16 is\_repeated\_guest 87396 non-null int64 \n",

" 17 previous\_cancellations 87396 non-null int64 \n",

" 18 previous\_bookings\_not\_canceled 87396 non-null int64 \n",

" 19 reserved\_room\_type 87396 non-null object \n",

" 20 assigned\_room\_type 87396 non-null object \n",

" 21 booking\_changes 87396 non-null int64 \n",

" 22 deposit\_type 87396 non-null object \n",

" 23 agent 87396 non-null float64\n",

" 24 days\_in\_waiting\_list 87396 non-null int64 \n",

" 25 customer\_type 87396 non-null object \n",

" 26 adr 87396 non-null float64\n",

" 27 required\_car\_parking\_spaces 87396 non-null int64 \n",

" 28 total\_of\_special\_requests 87396 non-null int64 \n",

" 29 reservation\_status 87396 non-null object \n",

" 30 reservation\_status\_date 87396 non-null object \n",

"dtypes: float64(3), int64(16), object(12)\n",

"memory usage: 23.4+ MB\n"

]

}

],

"source": [

"# Checking info of newly formed dataset after removed outliers\n",

"new\_df.info()"

]

},

{

"cell\_type": "code",

"execution\_count": 36,

"metadata": {

"id": "VE1-WgNCFLos"

},

"outputs": [],

"source": [

"# Change datatype of variables children and agent to correct format from float64 to int64\n",

"new\_df[['children','agent']]=new\_df[['children','agent']].astype('int64')\n",

"\n",

"# Change datatype of variable reservation\_status\_date to correct format from object to datetime64\n",

"new\_df['reservation\_status\_date']=pd.to\_datetime(new\_df['reservation\_status\_date'], format='%Y-%m-%d')"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"metadata": {

"id": "X6OqW4KzFLot",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "2fc83a65-4f67-4829-d209-5591c63f0e5b"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"Int64Index: 87396 entries, 0 to 119389\n",

"Data columns (total 3 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 children 87396 non-null int64 \n",

" 1 agent 87396 non-null int64 \n",

" 2 reservation\_status\_date 87396 non-null datetime64[ns]\n",

"dtypes: datetime64[ns](1), int64(2)\n",

"memory usage: 4.7 MB\n"

]

}

],

"source": [

"# Checking datatype\n",

"new\_df[['children','agent','reservation\_status\_date']].info()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "95xPpnsEFLot"

},

"source": [

"Variables children and agent are in float64 format, so converted them to the appropriate datatype, int64. The reservation\_status\_date variable is in object format, so converted it to datetime64 format."

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"metadata": {

"id": "zg1CTSHmFLou",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 386

},

"outputId": "4a133d48-7de7-4cd1-adac-1908dc165254"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" hotel is\_canceled lead\_time arrival\_date\_year arrival\_date\_month \\\n",

"0 Resort Hotel 0 296 2015 July \n",

"1 Resort Hotel 0 296 2015 July \n",

"2 Resort Hotel 0 7 2015 July \n",

"3 Resort Hotel 0 13 2015 July \n",

"4 Resort Hotel 0 14 2015 July \n",

"\n",

" arrival\_date\_week\_number arrival\_date\_day\_of\_month \\\n",

"0 27 1 \n",

"1 27 1 \n",

"2 27 1 \n",

"3 27 1 \n",

"4 27 1 \n",

"\n",

" stays\_in\_weekend\_nights stays\_in\_week\_nights adults ... \\\n",

"0 0 0 2 ... \n",

"1 0 0 2 ... \n",

"2 0 1 1 ... \n",

"3 0 1 1 ... \n",

"4 0 2 2 ... \n",

"\n",

" required\_car\_parking\_spaces total\_of\_special\_requests reservation\_status \\\n",

"0 0 0 Check-Out \n",

"1 0 0 Check-Out \n",

"2 0 0 Check-Out \n",

"3 0 0 Check-Out \n",

"4 0 1 Check-Out \n",

"\n",

" reservation\_status\_date total\_stays total\_people total\_childrens \\\n",

"0 2015-07-01 0 2.0 0.0 \n",

"1 2015-07-01 0 2.0 0.0 \n",

"2 2015-07-02 1 1.0 0.0 \n",

"3 2015-07-02 1 1.0 0.0 \n",

"4 2015-07-03 2 2.0 0.0 \n",

"\n",

" reserved\_room\_assigned guest\_category lead\_time\_category \n",

"0 yes couple high \n",

"1 yes couple high \n",

"2 no single low \n",

"3 yes single low \n",

"4 yes couple low \n",

"\n",

"[5 rows x 37 columns]"

],

"text/html": [

"\n",

" <div id=\"df-13f175a3-9bd5-410d-afc7-8d8182fa5df7\">\n",

" <div class=\"colab-df-container\">\n",

" <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>hotel</th>\n",

" <th>is\_canceled</th>\n",

" <th>lead\_time</th>\n",

" <th>arrival\_date\_year</th>\n",

" <th>arrival\_date\_month</th>\n",

" <th>arrival\_date\_week\_number</th>\n",

" <th>arrival\_date\_day\_of\_month</th>\n",

" <th>stays\_in\_weekend\_nights</th>\n",

" <th>stays\_in\_week\_nights</th>\n",

" <th>adults</th>\n",

" <th>...</th>\n",

" <th>required\_car\_parking\_spaces</th>\n",

" <th>total\_of\_special\_requests</th>\n",

" <th>reservation\_status</th>\n",

" <th>reservation\_status\_date</th>\n",

" <th>total\_stays</th>\n",

" <th>total\_people</th>\n",

" <th>total\_childrens</th>\n",

" <th>reserved\_room\_assigned</th>\n",

" <th>guest\_category</th>\n",

" <th>lead\_time\_category</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>296</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-01</td>\n",

" <td>0</td>\n",

" <td>2.0</td>\n",

" <td>0.0</td>\n",

" <td>yes</td>\n",

" <td>couple</td>\n",

" <td>high</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>296</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-01</td>\n",

" <td>0</td>\n",

" <td>2.0</td>\n",

" <td>0.0</td>\n",

" <td>yes</td>\n",

" <td>couple</td>\n",

" <td>high</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>7</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-02</td>\n",

" <td>1</td>\n",

" <td>1.0</td>\n",

" <td>0.0</td>\n",

" <td>no</td>\n",

" <td>single</td>\n",

" <td>low</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>13</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-02</td>\n",

" <td>1</td>\n",

" <td>1.0</td>\n",

" <td>0.0</td>\n",

" <td>yes</td>\n",

" <td>single</td>\n",

" <td>low</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Resort Hotel</td>\n",

" <td>0</td>\n",

" <td>14</td>\n",

" <td>2015</td>\n",

" <td>July</td>\n",

" <td>27</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>...</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>Check-Out</td>\n",

" <td>2015-07-03</td>\n",

" <td>2</td>\n",

" <td>2.0</td>\n",

" <td>0.0</td>\n",

" <td>yes</td>\n",

" <td>couple</td>\n",

" <td>low</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 37 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-13f175a3-9bd5-410d-afc7-8d8182fa5df7')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-13f175a3-9bd5-410d-afc7-8d8182fa5df7 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-13f175a3-9bd5-410d-afc7-8d8182fa5df7');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 38

}

],

"source": [

"# Adding night stays on week night and weekend night in one variable to 'total\_stays'\n",

"new\_df['total\_stays']=new\_df['stays\_in\_weekend\_nights']+ new\_df['stays\_in\_week\_nights']\n",

"\n",

"# Converting \"adults,\" \"children,\" and \"babies\" to total\_people by adding it\n",

"new\_df['total\_people']= new\_df['adults']+ df['children']+ new\_df['babies']\n",

"\n",

"# Creating 'total\_childrens' variable by adding 'chldrens' and 'babies' variables\n",

"new\_df['total\_childrens']= df['children']+ new\_df['babies']\n",

"\n",

"# Creating 'reserved\_room\_assigned' variable which describe same room assigned or not \n",

"new\_df['reserved\_room\_assigned']=np.where(new\_df['reserved\_room\_type']==new\_df['assigned\_room\_type'], 'yes', 'no')\n",

"\n",

"# Creating 'guest\_category' from variable 'total\_people'\n",

"new\_df['guest\_category']=np.where(new\_df['total\_people']==1, 'single',\n",

" np.where(new\_df['total\_people']==2, 'couple', 'family'))\n",

"\n",

"# Creating 'lead\_time\_category' from 'lead\_time' variale to display category\n",

"new\_df['lead\_time\_category']=np.where(new\_df['lead\_time']<=15, 'low',\n",

" np.where((new\_df['lead\_time']>15) & (new\_df['lead\_time']<90), 'medium', 'high'))\n",

"\n",

"#checking dataset\n",

"new\_df.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"metadata": {

"id": "wLwkv7qLFLov",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "f464d78c-224f-4d7e-d3d8-b39aa722411a"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(87396, 37)"

]

},

"metadata": {},

"execution\_count": 39

}

],

"source": [

"new\_df.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 40,

"metadata": {

"id": "zWnLjaDUFLow"

},

"outputs": [],

"source": [

"# Remove observations having value 0 in total\_people variable\n",

"new\_df.drop(new\_df[new\_df['total\_people']==0].index, inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 41,

"metadata": {

"id": "lsSROAtfFLow",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "c3f52188-e795-4190-9e71-d9f35e770757"

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(87230, 37)"

]

},

"metadata": {},

"execution\_count": 41

}

],

"source": [

"new\_df.shape"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "dbY4sdV4FLox"

},

"source": [

"Because observations of the variable total\_people cannot be zero, observations with 0 values are removed, reducing the number of observations to 87230 from 87396. "

]

},

{

"cell\_type": "code",

"execution\_count": 42,

"metadata": {

"id": "rN8HDsfkFLox",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "b10f9d6d-3d91-40f7-e9ed-e770bca96237"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"Int64Index: 87230 entries, 0 to 119389\n",

"Data columns (total 37 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 hotel 87230 non-null object \n",

" 1 is\_canceled 87230 non-null int64 \n",

" 2 lead\_time 87230 non-null int64 \n",

" 3 arrival\_date\_year 87230 non-null int64 \n",

" 4 arrival\_date\_month 87230 non-null object \n",

" 5 arrival\_date\_week\_number 87230 non-null int64 \n",

" 6 arrival\_date\_day\_of\_month 87230 non-null int64 \n",

" 7 stays\_in\_weekend\_nights 87230 non-null int64 \n",

" 8 stays\_in\_week\_nights 87230 non-null int64 \n",

" 9 adults 87230 non-null int64 \n",

" 10 children 87230 non-null int64 \n",

" 11 babies 87230 non-null int64 \n",

" 12 meal 87230 non-null object \n",

" 13 country 87230 non-null object \n",

" 14 market\_segment 87230 non-null object \n",

" 15 distribution\_channel 87230 non-null object \n",

" 16 is\_repeated\_guest 87230 non-null int64 \n",

" 17 previous\_cancellations 87230 non-null int64 \n",

" 18 previous\_bookings\_not\_canceled 87230 non-null int64 \n",

" 19 reserved\_room\_type 87230 non-null object \n",

" 20 assigned\_room\_type 87230 non-null object \n",

" 21 booking\_changes 87230 non-null int64 \n",

" 22 deposit\_type 87230 non-null object \n",

" 23 agent 87230 non-null int64 \n",

" 24 days\_in\_waiting\_list 87230 non-null int64 \n",

" 25 customer\_type 87230 non-null object \n",

" 26 adr 87230 non-null float64 \n",

" 27 required\_car\_parking\_spaces 87230 non-null int64 \n",

" 28 total\_of\_special\_requests 87230 non-null int64 \n",

" 29 reservation\_status 87230 non-null object \n",

" 30 reservation\_status\_date 87230 non-null datetime64[ns]\n",

" 31 total\_stays 87230 non-null int64 \n",

" 32 total\_people 87230 non-null float64 \n",

" 33 total\_childrens 87230 non-null float64 \n",

" 34 reserved\_room\_assigned 87230 non-null object \n",

" 35 guest\_category 87230 non-null object \n",

" 36 lead\_time\_category 87230 non-null object \n",

"dtypes: datetime64[ns](1), float64(3), int64(19), object(14)\n",

"memory usage: 25.3+ MB\n"

]

}

],

"source": [

"# Checking info of new dataset\n",

"new\_df.info()"

]

},

{

"cell\_type": "code",

"execution\_count": 43,

"metadata": {

"id": "u8B7PlqfFLoy",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "f202fbf7-09a2-4f55-df86-1de43b43e8fa"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"Int64Index: 87230 entries, 0 to 119389\n",

"Data columns (total 2 columns):\n",

" # Column Non-Null Count Dtype\n",

"--- ------ -------------- -----\n",

" 0 total\_people 87230 non-null int64\n",

" 1 total\_childrens 87230 non-null int64\n",

"dtypes: int64(2)\n",

"memory usage: 2.0 MB\n"

]

}

],

"source": [

"# Coverting datatype of variables total\_people and total\_childrens to int64 from float64\n",

"new\_df['total\_people']=new\_df['total\_people'].astype('int64')\n",

"new\_df['total\_childrens']=new\_df['total\_childrens'].astype('int64')\n",

"\n",

"\n",

"#Checking datatype of total\_column\n",

"new\_df[['total\_people','total\_childrens']].info()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "mkWtgJjBFLoz"

},

"source": [

"\*\*\*Insights\*\*\*\n",

"\n",

"- The variables \"children,\" \"agent,\" \"reservation\_status\_date,\" \"total\_people,\" and \"total\_children\" do not have the correct datatype format. So they are transformed from float64 datatypes to int64. The variable \"reservation\_status\_date\" was transformed from object datatype to datetime64.\n",

"- For more convenience in the dataset, \"total\_stays\" and \"total\_people\" variables are created. By adding the variables \"stays\_in\_weekend\_nights\" and \"stays\_in\_weeknights,\" the variable \"total\_stays\" is created. The variables \"adults,\" \"children,\" and \"babies\" are combined, the variable \"total\_people\" is created. \n",

"- The variable \"reserved\_room\_assigned\" is made up of the variables \"reserved\_room\_type\" and \"assigned\_room\_type,\" which describe whether or not a reserved room for a customer has been assigned. From variables \"children\" and \"babies,\" a new \"total\_children\" variable is created by adding both of them.\n",

"- The variable \"total\_people\" was used to create \"guest\_category,\" which describes bookings made for individuals, couples, or families. The variable 'lead\_time\_category' was created from the variable 'lead\_time' to display lead time as low, medium, and high. \n",

"- Variable \"total\_people\" cannot be 0, so observations in total\_people having 0 values are dropped.\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "fURO-kFgFLoz"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "mRM\_7TnIFLo0"

},

"source": [

"## <ins> \*\*Exploratory Data Analysis\*\* <ins>"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "E5HUAUShFLo0"

},

"source": [

"#### \*\*Univariate Analysis\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 44,

"metadata": {

"id": "gLo2epSrFLo1"

},

"outputs": [],

"source": [

"# Obtaing target variable\n",

"excluded\_variables=[var for var in new\_df.columns if len(new\_df[var].value\_counts()) > 15]\n",

"target\_variables=[var for var in new\_df.columns if var not in excluded\_variables]\n",

"\n",

"# Defining r to autofit the number and size of plots\n",

"r = int(len(target\_variables)/3 +1)"

]

},

{

"cell\_type": "code",

"execution\_count": 45,

"metadata": {

"id": "tbU8OguFFLo1"

},

"outputs": [],

"source": [

"# Defining a function to Notate the percent count of each value on the bars\n",

"def annot\_percent(axes):\n",

" '''Takes axes as input and labels the percent count of each bar in a countplot'''\n",

" for p in plot.patches:\n",

" total = sum(p.get\_height() for p in plot.patches)/100\n",

" percent = round((p.get\_height()/total),2)\n",

" x = p.get\_x() + p.get\_width()/2\n",

" y = p.get\_height()\n",

" plot.annotate(f'{percent}%', (x, y), ha='center', va='bottom')"

]

},

{

"cell\_type": "code",

"execution\_count": 46,

"metadata": {

"id": "q6mtA\_tlFLo2",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"outputId": "fd0e9928-4921-44fd-d450-abf42f7bfdd0"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1296x1944 with 24 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting the countplots for each variable in target\_variables\n",

"plt.figure(figsize=(18,r\*3))\n",

"for n,var in enumerate(target\_variables):\n",

" plot = plt.subplot(r,3,n+1)\n",

" sns.countplot(x=new\_df[var]).margins(y=0.15)\n",

" plt.title(f'{var.title()}',weight='bold')\n",

" plt.tight\_layout()\n",

" annot\_percent(plot)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Xn6jIVVCFLo3"

},

"source": [

"\*\*Obersvations\*\*\n",

"- A city hotel was most preferred by 61.07 percent of customers over a resort.\n",

"- 72.48% of bookings are not cancelled. Almost one-third of all reservations are canceled. \n",

"- Bookings increased by 33.28% in 2016 compared to 2015, but fell by 12.25% in 2017.\n",

"- Customers make the most reservations in August, followed by July. Customers made the fewest reservations in November, December, and January. So we can make offers to customers in November, December, and January to maximise booking.\n",

"- BB is the most requested food.\n",

"- Most of the bookings are made through the online platform.\n",

"- The top distribution channel is TA/TO, which is used to make most of the bookings.\n",

"- The majority of hotel bookings are made by new customers. Very few customers (3.86%) visited again.\n",

"- The customer's top preference is for Room A to be reserved.\n",

"- Customers do not want to pay a pre-deposit for a reservation.\n",

"- Most customers (80%) preferred to book a hotel for a short stay.\n",

"- 90% of people do not require parking spaces for their vehicles.\n",

"- 70% chance that bookings will not be cancelled by customers.\n",

"- Reserved rooms were not assigned to 15% of customers. Ensure that customers receive the rooms they have reserved.\n",

"- Reservations were often made for two people. 10% or so of guests brought their families. Few bring their families with them. Offer family-friendly discounts to encourage reservations for family and business events."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Hj9TAreWFLo6"

},

"source": [

"#### \*\*Bivariate Analysis\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ji1yhA\_BFLo6"

},

"source": [

"- \*\*\*Question 1: Is not having a reserved room assigned a reason for booking cancellations?\*\*\* "

]

},

{

"cell\_type": "code",

"execution\_count": 47,

"metadata": {

"id": "XqDKzRdHFLo7",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 351

},

"outputId": "a2198d88-23b8-4f2f-b964-b60d6a967648"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting count plot for is\_canceled in terms of reserved\_room\_assigned\n",

"plt.figure(figsize=(5,5))\n",

"plot=plt.subplot(111)\n",

"ax=sns.countplot(x=new\_df['is\_canceled'], hue=new\_df['reserved\_room\_assigned'])\n",

"ax.set\_title('is\_canceled v/s reserved\_room\_assigned')\n",

"annot\_percent(plot)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "FE70pQExFLo7"

},

"source": [

"The inability to assign a reserved room to a customer is not grounds for cancellation. Less than 1% of people who cancelled their booking when the reserved room was not assigned."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "V9\_SkG0jFLo8"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "GS7Tr6gLFLo8"

},

"source": [

"\*\*\*Question 2 : Is the high lead\_time a reason for booking cancellations?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 48,

"metadata": {

"id": "8CJ-eTrjFLo9",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 351

},

"outputId": "ac76fad7-37c8-4355-ff8b-eb5a5de1b0f2"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting count plot for lead\_time\_category in terms of is\_canceled\n",

"plt.figure(figsize=(5,5))\n",

"plot=plt.subplot(111)\n",

"ax=sns.countplot(x=new\_df['lead\_time\_category'], hue=new\_df['is\_canceled'])\n",

"ax.set\_title('lead\_time\_category v/s is\_canceled')\n",

"annot\_percent(plot)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "4WMjRr7ZFLo-"

},

"source": [

"Less lead time means fewer cancellations. Customers who book hotels more than 15 days in advance are more likely to cancel their booking. Booking cancellations are not caused by a longer Lead time."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zxx858q0FLo-"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "DP\_4s\_\_MFLo-"

},

"source": [

"\*\*\*Question 3: How many people are reservations made for?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 49,

"metadata": {

"id": "DFJwt21MFLo\_",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 350

},

"outputId": "97cb0d52-e4a0-4d8e-efa8-9052dd4cb83f"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting count plot for hotel\n",

"plt.figure(figsize=(5,5))\n",

"plot=plt.subplot(111)\n",

"ax=sns.countplot(x=new\_df['hotel'], hue=new\_df['guest\_category'])\n",

"ax.set\_title('hotel v/s guest\_category')\n",

"annot\_percent(plot)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "TsBP0rALFLpA"

},

"source": [

"Most customers book hotels for two people (couples). Customers prefer city hotels over resorts for family bookings. A city hotel is preferred when booking for a single person."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "vDQCZ74-FLpA"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ovO24B6XFLpB"

},

"source": [

"\*\*\*Question 4: Which hotel type has the most advanced reservations?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 50,

"metadata": {

"id": "Y5kQDUDXFLpC",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 334

},

"outputId": "69fdc6e8-9c85-449f-c132-e6a82a84a76d"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting violin plot for hotel against lead\_time\n",

"plt.figure(figsize=(5,5))\n",

"sns.violinplot(x=new\_df['hotel'], y=new\_df['lead\_time'])\n",

"ax.set\_title('is\_canceled v/s lead\_time')\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "jL63O\_rwFLpC"

},

"source": [

"In comparison to city hotels, guests book resort hotels a little bit in advance."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "HQ-BkxX7FLpD"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "J-ZYxPrmFLpD"

},

"source": [

"\*\*\*Question 5: Which distribution channels have the most cancellations of bookings?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 51,

"metadata": {

"id": "NnRlWJgUFLpE",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 351

},

"outputId": "4438629d-0c6b-4b63-f738-d452e5dcf0ec"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting a count plot for distribution channels in terms of the number of booking cancellations\n",

"plt.figure(figsize=(5,5))\n",

"plot=plt.subplot(111)\n",

"ax=sns.countplot(x=new\_df['distribution\_channel'], hue=new\_df['is\_canceled'])\n",

"ax.set\_title('% of booking cancellations per distribution channel')\n",

"annot\_percent(plot)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "bupbruxAFLpE"

},

"source": [

"The majority of canceled bookings were made through the TA/TO distribution channel. Bookings made through the Direct, Corporate, and GDS distribution channels are extremely unlikely to be canceled."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "bzGp871AFLpF"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "mhNDQKKJFLpF"

},

"source": [

"\*\*\*Question 6 : Which market segment is most used for booking hotels, and which market segment bookings are most canceled?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 52,

"metadata": {

"id": "4zOQPOvsFLpG",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 369

},

"outputId": "eaba48ca-9807-45ba-ed1b-2b15a626460d"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 720x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting count plot for market segment\n",

"plt.figure(figsize=(10,5))\n",

"ax=sns.countplot(x=new\_df['market\_segment'], hue=new\_df['is\_canceled'])\n",

"ax.set\_title('booking cancellations per market\_segment')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "eRk5L2piFLpG"

},

"source": [

"The majority of hotel reservations are made online, as are the majority of cancellations of reservations made by customers who made their reservations online. "

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "8qiS3DaVFLpH"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Y3KSD6KGFLpH"

},

"source": [

"\*\*\*Question 7 : Which room generates a higher ADR?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 53,

"metadata": {

"id": "IQSng1YJFLpI",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 514

},

"outputId": "67d76758-3ecf-409c-844b-85b136c32be7"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x576 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting box plot assigned\_room\_type against ADR\n",

"plt.figure(figsize=(15,8))\n",

"ax=sns.boxplot(x=new\_df['assigned\_room\_type'], y=new\_df['adr'])\n",

"ax.set\_title('Box plot of assigned\_room\_type v/s adr')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "L4l45BaqFLpI"

},

"source": [

"Room types G, followed by H, generate high ADR. Room I has a very low ADR. The most popular room is A, but it has a lower ADR than other room types that are less popular with customers for bookings. To maximize revenue, the hotel should increase the number of rooms in A, G and H."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "LKkXTHrBFLpJ"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "U0NKE6uKFLpN"

},

"source": [

"\*\*\*Question 8 : Which hotel type is the busiest?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 54,

"metadata": {

"id": "xTJ3Ulw6FLpN",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 350

},

"outputId": "8cc3fbda-f87b-4d2d-932a-00795f1a997a"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting bar plot for hotel against number of waiting days for bookings\n",

"plt.figure(figsize=(5,5))\n",

"plot=plt.subplot(111)\n",

"ax=sns.barplot(x=new\_df['hotel'], y=new\_df['days\_in\_waiting\_list'])\n",

"ax.set\_title('hotel v/s days\_in\_waiting\_list')\n",

"annot\_percent(plot)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "u-42rGTXFLpO"

},

"source": [

"A city hotel is busier than a resort."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "jl5prg0aFLpO"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ij-AZ--BFLpP"

},

"source": [

"\*\*\*Question 9: Which month is the busiest for hotels?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 55,

"metadata": {

"id": "45J1cujVFLpP",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 513

},

"outputId": "eb03d60c-7de1-43a1-98f3-da67c0dc94df"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 720x504 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Bar plot for arrival month of customer against waiting days for bookings\n",

"plt.figure(figsize=(10,7))\n",

"plot=plt.subplot(111)\n",

"ax=sns.barplot(x=new\_df['arrival\_date\_month'], y=new\_df['days\_in\_waiting\_list'], hue=new\_df['hotel'])\n",

"ax.set\_title('bar plot of arrival\_date\_month vs. days\_in\_waiting\_list in terms of hotel types')\n",

"annot\_percent(plot)\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "-Xua\_sYxFLpQ"

},

"source": [

"Customers prefer city hotels over resort hotels for the New Year. The city hotel is busiest in October and September , which means that this month has the most bookings. In the fourth quarter of the year, in December, resorts are busier than city hotels. Resort hotels are significantly less busy in the third quarter of the year than they are throughout the year. In July there is no longer waiting period for bookings. "

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "JzpX3T8aFLpQ"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "XNNa9AyPFLpQ"

},

"source": [

"\*\*\*Question 10 : Which customer type generates more revenue in terms of hotel types and customer types?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 56,

"metadata": {

"id": "kUSuY2C7FLpQ",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 369

},

"outputId": "43568907-200d-4a3d-e219-230e38a0d103"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting bar plot for diffrent customers type against ADR\n",

"plt.figure(figsize=(5,5))\n",

"plot=plt.subplot(1,1,1)\n",

"ax=sns.barplot(x=new\_df['customer\_type'], y=new\_df['adr'], hue=new\_df['hotel'])\n",

"ax.set\_title('bar plot of customer\_type vs. adr in terms of hotel types')\n",

"annot\_percent(plot)\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "PDV9gkf1FLpR"

},

"source": [

"City hotels generate more revenue (54.86%) than resort hotels (45.14%). Transient customers who book rooms for a short period of time generate more revenue than other types of customers."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "236R58NVFLpR"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "MFhfvrLrFLpR"

},

"source": [

"\*\*\*Question 11 : In terms of hotel types, how many parking spaces are most frequently requested by customers?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 57,

"metadata": {

"id": "fLMMXuiqFLpS",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 369

},

"outputId": "eee66767-cbac-42eb-b44e-385ebfd4c097"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 504x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting count plot for number of requested parking spaces in terms of hotel\n",

"plt.figure(figsize=(7,5))\n",

"ax=sns.countplot(x=new\_df['required\_car\_parking\_spaces'], hue=new\_df['hotel'])\n",

"ax.set\_title('number of parking spaces requested by customers in terms of hotel types')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "gvECDI3nFLpS"

},

"source": [

"Most of the customers (91.63%) did not request any parking spaces. Only a few customers (8.31%) requested parking. One parking space is most desirable to customers. The hotel can eliminate parking spaces 3 and 8 and concentrate on parking spaces 1 and 2, which will reduce the cost of providing parking spaces."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "4e8tfM8gFLpT"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "vv2DZuI5FLpT"

},

"source": [

"\*\*\*Question 12 : What is the most common number of nights booked by customers?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 58,

"metadata": {

"id": "du19\_wiDFLpT",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 515

},

"outputId": "b799790c-a092-4318-aaf0-c8e3075ad78e"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x576 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting count plot for total\_stays\n",

"plt.figure(figsize=(15,8))\n",

"plot=plt.subplot(111)\n",

"ax=sns.countplot(x=new\_df['total\_stays'])\n",

"ax.set\_title('number of nights preferred to stay by % of customers')\n",

"annot\_percent(plot)\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "xtukcDb5FLpT"

},

"source": [

"The majority of the guests are staying at the hotel for three nights. Customers who book hotels for more than one week are very rare. Most hotel bookings are for less than 7 nights. The hotel should pay more attention to keeping reservations available for less than 7 nights to not lose customers and to generate more revenue."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "LuARWZiHFLpU"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "7HmeS-NHFLpU"

},

"source": [

"\*\*\*Question 13 : What is the most common number of special requests made by customers, and what kind of customer are they?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 59,

"metadata": {

"id": "8apvbEExFLpU",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 369

},

"outputId": "dd7277b5-e734-4133-d5af-b34613ed51ef"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 720x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting count plot for total\_of\_special\_requests in terms of different guest type\n",

"plt.figure(figsize=(10,5))\n",

"plot=plt.subplot(111)\n",

"ax=sns.countplot(x=new\_df['total\_of\_special\_requests'], hue=new\_df['guest\_category'])\n",

"ax.set\_title('total\_of\_special\_requests by % of customers')\n",

"annot\_percent(plot)\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "d3QQsJ\_4FLpV"

},

"source": [

"Approximately half of all customers do not make special requests. Customers frequently make one special request. Couples make the majority of special requests. Bookings made for single people are not more requested for special requests than those made for couples, followed by family and business events."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Q8oaxW\_pFLpV"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Z-p6IMVNFLpV"

},

"source": [

"\*\*\*Question 14 : Is the ADR affected by the hotel not giving a reserved room?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 60,

"metadata": {

"id": "HwM6-gp2FLpV",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 369

},

"outputId": "3e04e3f1-e144-4658-92b7-d678bb351870"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 504x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting box plot for reserved\_room\_assigned \n",

"plt.figure(figsize=(7,5))\n",

"plot=plt.subplot(111)\n",

"ax=sns.boxplot(x=new\_df['reserved\_room\_assigned'], y=new\_df['adr'])\n",

"ax.set\_title('Box plot for reserved\_room\_assigned')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ayZcCWFAFLpW"

},

"source": [

"There is no significant difference in ADR between reserved rooms assigned  and not reserved rooms assigned. Not assigning a reserved room does not affect ADR."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "OsOAK3ydFLpX"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "TI0WDKo6FLpX"

},

"source": [

"\*\*\*Question 15: The majority of bookings were made for how many people, and the majority of cancellations of bookings were made for how many people?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 61,

"metadata": {

"id": "lB2t\_rRmFLpX",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 721

},

"outputId": "60946c29-49b2-4cc1-a7d6-108dbd35aec8"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# targeted variable\n",

"variables=['hotel','is\_canceled']\n",

"\n",

"#ploting graph using for each above variabe\n",

"n=0\n",

"for var in variables:\n",

" plt.figure(figsize=(5,5))\n",

" plot=plt.subplot(1,1,n+1)\n",

" ax=sns.countplot(x=new\_df['guest\_category'], hue=new\_df[var])\n",

" ax.set\_title(f'guest\_category v/s {var} in terms of % of customers')\n",

" annot\_percent(plot)\n",

" plt.tight\_layout()\n",

" plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "taoNKg8nFLpY"

},

"source": [

"The majority of hotel reservations are made for couples. Couples' reservations are more likely to be canceled than singles' or families' reservations. Bookings for single people are rarely canceled."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "mA4mQyY-FLpY"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "JNdDj33hFLpZ"

},

"source": [

"\*\*\*Question 16 : Which country makes the most reservations, and which agent makes the most bookings?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 62,

"metadata": {

"id": "oin\_I-t6FLpZ",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 350

},

"outputId": "f9ffd20e-2866-43d3-a64d-3d09b2a462e8"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 504x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Creating country\_df which counts number of bookings by each country\n",

"country\_df = pd.DataFrame(new\_df['country'].value\_counts()).reset\_index()\n",

"country\_df.rename(columns={'index': 'country','country': 'num\_of\_bookings'},inplace=True)\n",

"\n",

"# Plotting point plot for country with number of bookings\n",

"plt.figure(figsize=(7,5))\n",

"ax=sns.barplot(x=country\_df['country'].head(10), y=country\_df['num\_of\_bookings'])\n",

"ax.set\_title('Top 10 countries with number of bookings')\n",

"plt.show()"

]

},

{

"cell\_type": "code",

"execution\_count": 63,

"metadata": {

"id": "KUxU--r5FLpZ",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 350

},

"outputId": "c34dd216-0924-4439-9b27-e07c999a4f77"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 504x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Creating agent\_df which counts number of bookings by each agent\n",

"agent\_df = pd.DataFrame(new\_df['agent'].value\_counts()).reset\_index()\n",

"agent\_df.rename(columns={'index': 'agent','agent': 'num\_of\_bookings'},inplace=True)\n",

"\n",

"# Plotting point plot for agent with number of bookings\n",

"plt.figure(figsize=(7,5))\n",

"ax=sns.barplot(x=agent\_df['agent'].head(10), y=agent\_df['num\_of\_bookings'])\n",

"ax.set\_title('Top 10 agent with number of bookings')\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "DKLmoVAcFLpZ"

},

"source": [

"The majority of reservations are made through country PRT. Customers make the most bookings in the following top 5 countries: PRT, GBR, FRA, ESP, and DEU.\n",

"Agent nummber 9 made most number of bookings. 9, 240, 7, 14 and 250 are the top 5 agents by number of bookings made."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "w7zJOn-RFLpa"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Ibir0y46FLpa"

},

"source": [

"\*\*\*Question 17 : Does a longer waiting period cause the cancellation of bookings?\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 64,

"metadata": {

"id": "HX7mG6WXFLpb",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 369

},

"outputId": "4006e326-a850-462d-8abe-3e9dd7527515"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"#Selecting bookings with non zero waiting time\n",

"waiting\_time=new\_df[new\_df['days\_in\_waiting\_list']!=0]\n",

"\n",

"#ploting graph\n",

"\n",

"plt.figure(figsize=(5,5))\n",

"ax=sns.kdeplot(x=waiting\_time['days\_in\_waiting\_list'], hue=waiting\_time['is\_canceled'])\n",

"ax.set\_title('days\_in\_waiting\_list')\n",

"plt.tight\_layout()\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "PKfR3M2DFLpb"

},

"source": [

"The majority of canceled bookings have a waiting period of less than 150 days, but those that are not canceled bookings by customers have a waiting period of less than 150 days, which has a higher density than the canceled bookings. So a longer waiting period is not a reason for booking cancellation. "

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "-XuOlt0fFLpc"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "WSOEGMvkFLpc"

},

"source": [

"#### \*\*Multivariate Analysis\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "sVzb\_zc2FLpd"

},

"source": [

"##### \*\*\*1] Correlation Heatmap\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 65,

"metadata": {

"id": "olfarqx9FLpd",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 630

},

"outputId": "90cc9908-a987-4ead-f1e4-5550e9996030"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1080x576 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Plotting a correlation heatmap for the dataset\n",

"plt.figure(figsize=(15,8))\n",

"sns.heatmap(new\_df.corr(), vmin=-1, cmap='coolwarm', annot=True)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "OmCoq1n7FLpf"

},

"source": [

"- In the heatmap, we see some high correlation between a few variables because we created new variables total\_stays, total\_people, and total\_children from existing variables and did not drop old variables.\n",

"- The variables lead\_time and is\_canceled have weak relationships. The most likely reason for cancellation is a longer lead time."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "mjK2uWiyFLpf"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "wcaCdp5iFLpg"

},

"source": [

"##### \*\*\*2] Pair Plot\*\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 66,

"metadata": {

"id": "pFumxhs0FLpg",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 973

},

"outputId": "f20d92b5-27fb-4670-b6ac-68beaebd5ae6"

},

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 720x576 with 0 Axes>"

]

},

"metadata": {}

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1440x1440 with 72 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"# Creating new dataset\n",

"new\_df2=new\_df[['hotel','is\_canceled','lead\_time','arrival\_date\_year','arrival\_date\_month','meal','market\_segment','distribution\_channel','reserved\_room\_type',\n",

" 'assigned\_room\_type','deposit\_type','days\_in\_waiting\_list', 'customer\_type', 'adr','total\_stays',\n",

" 'total\_people', 'total\_childrens', 'reserved\_room\_assigned',\n",

" 'guest\_category', 'lead\_time\_category']]\n",

"\n",

"# Plotting pair plot for dataset\n",

"plt.figure(figsize=(10,8))\n",

"ax=sns.pairplot(new\_df2)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "g\_csQ51fFLpg"

},

"source": [

"- We can conclude from the relationship between lead\_time and is\_canceled that a longer lead time does not result in cancellations. \n",

"- We can conclude from the lead\_time and arrival\_date\_year graphs that people were consistently interested in booking rooms in advance in 2015, 2016, and 2017. \n",

"- According to the graph of arrival\_date\_year and days\_in\_waiting\_list, 2016 had the longest waiting period for room bookings.\n",

"- We can conclude from the graph of ADR and days\_in\_waiting\_list that a short minimum waiting period for bookings results in a high ADR. \n",

"- From the graph of \"is\_canceled\" and \"days\_in\_waiting\_list,\" we conclude that waiting for bookings is not a reason for booking cancellation."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "\_zq\_GEIDFLpg"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "5ZQiMIE\_FLph"

},

"source": [

"## \*\*<u>Solution to Business Objective<u>\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "9ZzsHiHJFLph"

},

"source": [

"- A city hotel has more bookings than a resort. Offer packages and promotions to promote bookings for the resort hotel.\n",

"- BB is the most requested food. The hotel should maintain food quality while also offering discounts on other foods to promote other food types, reducing the burden on kitchen management and keeping a variety of food options available to customers.\n",

"- Most of the bookings are made through the online platform. Hotels can cut costs by eliminating market segments such as complementary and aviation because bookings through these segments are very low.\n",

"- Because most bookings made through TA/TO distribution are followed by corporate distribution, hotels should invest in both TA/TO and corporate distribution channels. The GDS distribution channel can be eliminated by hoteliers because bookings made through it are extremely low.\n",

"- Very few customers (3.86%) visited again. So hotels can increase repeat bookings by offering the right repeat booking incentives, understanding the motivations behind repeat bookings, marketing to your guests’ past interests, and assessing past bookings to identify priority guests.\n",

"- Because rooms A and D are the most popular with customers, the hotel should maintain their quality. The hotel should promote rooms E, F, and G to increase demand by offering discounts. Because customers do not prefer to book room types B, C, H, and L, the hotel can eliminate them, lowering the cost of these rooms.\n",

"- Customers do not want to pay a pre-deposit for a reservation. Hotels should promote advance deposits because not only does an advance deposit allow you to recognize revenue faster, it also greatly decreases the risk of cancellations.\n",

"- Because 3 and 8 parking spaces were rarely requested by customers, hotels can only keep bookings for 1 and 2 parking spaces to save money.\n",

"- 15% of customers were not given reserved rooms. Make sure that guests get the rooms they have booked.\n",

"- Almost 25% of customers cancelled their bookings. Hotel should implement a cancellation policy, discount on confirmed bookings, and send booking reminders to guests to reduce booking cancellations.\n",

"- People typically book rooms for two people, so encourage family and group bookings. You can maximize revenue by promoting it with a discounted offer for group bookings."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "f9o3IoNyFLph"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zIf1Tn2bFLph"

},

"source": [

"## \*\*<u>Conclusion<u>\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "L6sDS0s2FLpi"

},

"source": [

"- The top country with the most number of bookings is PRT, and the number one agent with the most number of bookings is 9. \n",

"- Customers favored city hotels more than resort hotels by a margin of 61.07 percent.\n",

"- One of the four reservations is canceled.\n",

"- The most popular food is BB.\n",

"- The Online (internet) platform is used to make the majority of bookings.\n",

"- The majority of the bookings are made using TA/TO, the leading distribution channel.\n",

"- The vast majority of hotel bookings are made by new guests. Almost no consumers (3.86%) returned.\n",

"- The customer wants Room A to be reserved the most.\n",

"- Customers do not wish to make a bookings with a pre-deposit.\n",

"- Customers (80%) favored making a hotel reservation for a short visit.\n",

"- Only 10% of people require space to park their cars.\n",

"- Most visitors are couples.\n",

"- The inability to assign a reserved room to a customer is not grounds for cancellation.\n",

"- Booking cancellations are not caused by a longer Lead time.\n",

"- A city hotel is busier than a resort.\n",

"- The busiest months for hotels are October and September. There isn't a lengthy wait for reservations in July.\n",

"- Not assigning a reserved room does not affect ADR."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "vPp6XN6TFLpi"

},

"source": [

"."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "aQPn31aLFLpi"

},

"source": [

"## \*\*<u>Challenges<u>\*\*"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "QDe4jORLFLpj"

},

"source": [

"- The data contained a large number of duplicates.\n",

"- The improper data type format was used for the data.\n",

"- It was challenging to select the best visualization techniques.\n",

"- The dataset contained a large number of null values."

]

},

{

"cell\_type": "code",

"execution\_count": 66,

"metadata": {

"id": "X8b0oMCoFLpj"

},

"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": "3.9.15"

}

},

"nbformat": 4,

"nbformat\_minor": 0

}