Table of contents

Infos tableau

Description

In [ ]:
```
# COMMENTAIRE
# Tableau des commandes et des livraison des avions Boeing, dans le temps.
```

Sources du tableau

In [ ]:
```
# COMMENTAIRE
# https://www.kaggle.com/datasets/nurielreuven/boeing-historical-airplane-orders-deliveries
# https://www.boeing.com/commercial
```

Infos générales sur les données brutes

In [ ]:
```
import pandas as pd

# Construisons une fonction permettant d'avoir des infos générales sur un tableau donné.
def generate_stats(dataframe, name):
```

```
    """
    Génère un tableau récapitulant les données principales du tableau.

    Arg:
        dataframe (panda dataframe): Nom de la variable contenant le tableau.
        name (string): Nom du tableau à spécifier.
    """
    # Récupération du nom donné au tableau
    dataframe.name = name

    # Création du tableau des infos par défaut
    dfDescribe = dataframe.describe(include = 'all')

    # Création info sur les dtypes de chaque colonne et stockage dans un tableau df_dType
    df_dType = (pd.DataFrame(dataframe.dtypes, columns=["dType"])).T

    # Option : ajout info de la fréquence en pourcentage, dans le cas où il y a des variables qualitatives
    if "freq" in dfDescribe.index:
        freqPourcent = ((dfDescribe.loc["freq"]*100/len(dataframe)).apply(pd.to_numeric, errors = "coerce")).round(1)
        df_freqPourcent = pd.DataFrame(freqPourcent).T
        df_freqPourcent = df_freqPourcent.rename(index={"freq": "freq%"}, inplace=False)

        # Insertion de la ligne df_freqPourcent entre les lignes freq et mean de dfDescribe, dans le cas où il existe des variables quantitatives
        if "mean" in dfDescribe.index:
            dfDescribeHaut = dfDescribe.loc[:"freq"]
            dfDescribeBas = dfDescribe.loc["mean":]
            dfDescribe = pd.concat([dfDescribeHaut, df_freqPourcent, dfDescribeBas], axis=0)
         # Insertion de la ligne df_freqpourcent à la fin du tableau (=> naturellement, après freq), dans le cas où il n'existe pas de variable quantitative
        else:
            dfDescribe = pd.concat([dfDescribe, df_freqPourcent], axis=0)

    # Création info sur le nombre de valeurs manquantes dans chaque colonne (+ pourcentage) et stockage dans un tableau
    MissingValuesCount = dataframe.isnull().sum()
    dfMissingValues = (pd.DataFrame(MissingValuesCount, columns = ["naCount"])).T

    # Option: ajout du pourcentage de valeurs manquantes, dans le cas où il existe des valeurs manquantes
    if dfMissingValues.loc["naCount"].sum() != 0:
        dfMissingValues.loc["naCount%"] = (dfMissingValues.loc["naCount"]*100/len(dataframe)).round(1)

    # Concaténation des tableaux dtype et valeurs manquantes avec le tableau des infos générales
    dataSummary = pd.concat([df_dType, dfMissingValues, dfDescribe], axis=0)

    # Transposons pour plus de lisibilité
    dataSummary = dataSummary.T

    # Trier le tableau par dType
    dataSummary_sort = (dataSummary).sort_values(by="dType")

    # Affichage du tableau infos générales
    print("---Infos générales sur les données du tableau:", dataframe.name, "---")
    print("Dimensions du tableau (nombre de lignes, nombre de colonnes):", dataframe.shape)
    display(dataSummary_sort)
```

In [ ]:

```
chemin_fichier = "OrdersandDeliveries.csv"

# import jeu de données .csv avec "," comme séparateur par défaut
data = pd.read_csv(chemin_fichier)
```

```
# Application de la fonction créée précédemment à data
generate_stats(data, "Data brutes")
```

```
---Infos générales sur les données du tableau: Data brutes ---
Dimensions du tableau (nombre de lignes, nombre de colonnes): (9073, 11)
```

	dType	naCount	naCount%	count	unique	top	freq	freq%
Country	object	0.0	0.0	9073	132	USA	3200	35.3
Customer Name	object	0.0	0.0	9073	570	United Airlines	339	3.7
Delivery Year	object	1025.0	11.3	8048	66	2018	210	2.3
Engine	object	0.0	0.0	9073	7	PW	3077	33.9
Model Series	object	0.0	0.0	9073	59	737-800	1208	13.3
Order Month	object	0.0	0.0	9073	13	Dec	1314	14.5
Order Year	object	0.0	0.0	9073	69	2007	349	3.8
Region	object	26.0	0.3	9047	14	North America	3420	37.7
Delivery Total	object	0.0	0.0	9073	34	1	3170	34.9
Order Total	object	0.0	0.0	9073	69	1	3410	37.6
Unfilled Orders	object	8735.0	96.3	338	60	1	54	0.6

In [ ]:
```
# COMMENTAIRE
# On remarque que toutes les colonnes sont en object. D'après le nom des colonnes, certaines devront être sous forme numérique pour être exploitables.
# C'est le cas pour les colonnes: "Delivery Year", "Order Year", "Delivery Total", "Order Total", "Unfilled Orders".
# On remarque également 96% de valeurs manquantes pour la colonne Unfilled Orders.
# Il manque également des valeurs pour les colonnes Region et Delivery Year.
```

5 premières lignes du tableau - 5 dernières lignes du tableau

In [ ]:
```
# Afficher les 5 premières lignes et les 5 dernières lignes du tableau
display(data.head())
display(data.tail())
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total	Unfilled Orders
0	Afghanistan	Ariana Afghan Airlines	1968	PW	727	Mar	1968	Central Asia	1	1	NaN
1	Afghanistan	Ariana Afghan Airlines	1970	PW	727	Apr	1969	Central Asia	1	1	NaN
2	Afghanistan	Ariana Afghan Airlines	1979	GE	DC-10	Sep	1978	Central Asia	1	1	NaN
3	Afghanistan	Ariana Afghan Airlines	NaN	CF	737-700	Nov	2005	Central Asia	0	4	NaN
4	Algeria	Air Algerie	1974	PW	727	Jan	1974	Africa	1	1	NaN

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total	Unfilled Orders
9068	Zimbabwe	Air Zimbabwe	1987	PW	737-200	Jan	1987	Africa	2	2	NaN
9069	Zimbabwe	Air Zimbabwe	1990	PW	767-200ER	Mar	1989	Africa	1	1	NaN
9070	Zimbabwe	Air Zimbabwe	1989	PW	767-200ER	Jul	1988	Africa	1	1	NaN
9071	Zimbabwe	Air Zimbabwe	1986	PW	737-200	Dec	1986	Africa	1	1	NaN
9072	All	All	All	All	All	All	All	All	24,025	33,587	5,163

In [ ]:
```
# COMMENTAIRE
# On remarque que la dernière ligne du tableau constitue un total des colonnes représentant des quantités.
```

Problématique

In [ ]:
```
# COMMENTAIRE
# À partir de ces données, on peut se demander par exemple:
# 1) Quels sont les modèles les plus vendus? (analyse univariée qualitative, histogramme)
# 2) Quels sont les modèles les plus vendus en fonction des pays (analyse bivariée qualitative, tableau de contingence)
# 3) Quelles ont été les périodes où Boeing a vendu le plus d'avion? (analyse bivariée) Avec quelles autres actualités peut-on corréler ces périodes?
# 4) Comment évolue les demandes de commandes (tout pays confondus) dans le temps (analyse bivariée, pourcentage d'augmentation/diminution)
# etc..
# Mais d'abord, nettoyons les données.
```

Nettoyage des données

Nettoyage 1 : Stockage et suppression dernière ligne

In [ ]:
```
# Stockage de la dernière ligne du tableau dans une variable
lastLine = data.iloc[[9072]]
display(lastLine)
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total	Unfilled Orders
9072	All	All	All	All	All	All	All	All	24,025	33,587	5,163

In [ ]:
```
# Suppression de la dernière ligne du tableau
data = data[:-1]
print("(Nombre de lignes, nombre de colonnes):", data.shape)
print("Les dernières lignes du tableau sont:")
display (data.tail(2))
```

```
(Nombre de lignes, nombre de colonnes): (9072, 11)
Les dernières lignes du tableau sont:
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total	Unfilled Orders
9070	Zimbabwe	Air Zimbabwe	1989	PW	767-200ER	Jul	1988	Africa	1	1	NaN
9071	Zimbabwe	Air Zimbabwe	1986	PW	737-200	Dec	1986	Africa	1	1	NaN

Nettoyage 2 : Renommage nom de colonne Delivery Year

In [ ]:
```
# COMMENTAIRE
# Remarque: suite à plusieurs bugs en voulant appeller la colonne Delivery Year, je me suis aperçue qu'il y avait un espace à la fin du nom, nous allons donc retirer cet espace pour éviter to
# Vérifions avant que d'autres colonnes n'ont pas d'espace à la fin ou autre incohérence.
```

In [ ]:
```
# Afficher le nom de toutes les colonnes
print(data.columns)
```

```
Index(['Country', 'Customer Name', 'Delivery Year ', 'Engine', 'Model Series',
       'Order Month', 'Order Year', 'Region', 'Delivery Total', 'Order Total',
       'Unfilled Orders'],
      dtype='object')
```

In [ ]:
```
# COMMENTAIRE
# Il n'y a que Delivery Year qui est concerné => simple rename sur la colonne concernée
```

In [ ]:
```
# Renommage de la colonne concernée
data.rename(columns={"Delivery Year ": "Delivery Year"}, inplace = True)
display(data["Delivery Year"].head())
```

```
0    1968
1    1970
2    1979
3     NaN
4    1974
Name: Delivery Year, dtype: object
```

Nettoyage 3 : Conversion des dtypes et création sous-échantillon dfOrders

In [ ]:
```
# Pour rappel, dtypes des données brutes, sous forme de tableau (pour pouvoir faire un comparatif avant/après à la fin)
df_TypeBefore = pd.DataFrame(data.dtypes, columns=["dTypeBefore"])
df_TypeBefore
```

Out[ ]:

	dTypeBefore
Country	object
Customer Name	object
Delivery Year	object
Engine	object
Model Series	object
Order Month	object
Order Year	object
Region	object
Delivery Total	object
Order Total	object
Unfilled Orders	object

In [ ]:
```
# Nous allons convertir les colonnes de quantité et d'année en numérique afin de pouvoir mieux les exploiter
colonneVersNum = ["Delivery Total", "Order Total", "Unfilled Orders", "Delivery Year", "Order Year"]
data[colonneVersNum] = data[colonneVersNum].apply(pd.to_numeric, errors = "coerce")
data.dtypes
```

Out[ ]:
```
Country             object
Customer Name       object
Delivery Year      float64
Engine              object
Model Series        object
Order Month         object
Order Year           int64
Region              object
Delivery Total       int64
Order Total          int64
Unfilled Orders    float64
dtype: object
```

In [ ]:
```
# COMMENTAIRE
# Après conversion, on remarque qu'"Unfilled orders" et "Delivery Year" sont en float64 au lieu de int.
# Ceci est dû au fait que ces colonnes contiennent des valeurs manquantes.
# D'après la doc de pandas, des valeurs manquantes en numérique ne peuvent pas être des int.
# Pour notre analyse, on se passera de la colonne Unfilled Orders, qui contient 96% de valeurs manquantes d'après le tableau des infos générales, généré au debut.
```

In [ ]:
```
# Création d'un sous-échantillon dfOrders sans les lignes où il y des valeurs manquantes dans Delivery Year, et sans la colonne Unfilled Orders.
dfOrders = data.dropna(subset=["Delivery Year"])
dfOrders = dfOrders.drop("Unfilled Orders", axis=1)

# Conversion de Delivery Year: float to int
dfOrders["Delivery Year"] = dfOrders["Delivery Year"].astype(int)

# Regrouper les infos des dtypes avant/apres modif dans un tableau
df_Type = pd.DataFrame(dfOrders.dtypes, columns=["dType"])                  # màj des dtypes de chaque colonne et stockage dans un tableau
df_TypeAvantApres = pd.concat([df_TypeBefore, df_Type], axis = 1)           # génère et stock le tableau des dtypes avant/après
df_TypeAvantApres = df_TypeAvantApres.sort_values(by ="dType")              # trier le tableau suivant la colonne dType actuelle

# Afficher le tableau avant/après modif des dtypes
print("---Type Avant/Apres---")
display(df_TypeAvantApres)
```

```
---Type Avant/Apres---
```

	dTypeBefore	dType
Delivery Year	object	int32
Order Year	object	int64
Delivery Total	object	int64
Order Total	object	int64
Country	object	object
Customer Name	object	object
Engine	object	object
Model Series	object	object
Order Month	object	object
Region	object	object
Unfilled Orders	object	NaN

In [ ]:
```
# COMMENTAIRE
# Delivery Year a bien été pris en compte comme étant un numérique int
```

Nettoyage 4 : Valeurs manquantes

Nettoyage 4.1 : Valeurs manquantes - colonne(s) concernée(s)

In [ ]:
```
# COMMENTAIRE
# On a nettoyé une partie des valeurs manquantes dans la section précédente afin de pouvoir convertir certaines colonnes en numérique.
# Voyons à présent quelle(s) colonne(s) il reste avec des valeurs manquantes, en utilisant notre nouvelle échantillon dfOrders.
```

In [ ]:
```
# Pour rappel
countNaN = dfOrders.isnull().sum()
countNaN = countNaN[countNaN>0]
countNaN
```

Out[ ]:
```
Region    24
dtype: int64
```

Nettoyage 4.2 : Valeurs manquantes dans la colonne "Region"

In [ ]:
```
# Quelles sont les valeurs prises par Region?
dfOrders["Region"].unique()
```

Out[ ]:
```
array(['Central Asia', 'Africa', 'South America', 'Oceania', 'Europe',
       'Caribbean', 'Middle East', 'South Asia', 'Southeast Asia',
       'North America', 'East Asia', 'Central America and Mexico', nan,
       'Unidentified'], dtype=object)
```

In [ ]:
```
# COMMENTAIRE
# On remarque deux valeurs inhabituelles: NaN et Unidentified
```

In [ ]:
```
# Afficher les lignes du tableau où les régions sont vides
condition = (dfOrders["Region"].isnull())
display(dfOrders.loc[condition])
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total
2310	Iran, Islamic Republic of	Airline of the Islamic Republi	1966	PW	727	Apr	1965	NaN	2	2
2311	Iran, Islamic Republic of	Airline of the Islamic Republi	1976	PW	747-200	Apr	1975	NaN	1	1
2312	Iran, Islamic Republic of	Airline of the Islamic Republi	1977	PW	747-200	Apr	1975	NaN	1	1
2313	Iran, Islamic Republic of	Airline of the Islamic Republi	1967	PW	727	May	1967	NaN	1	1
2314	Iran, Islamic Republic of	Airline of the Islamic Republi	1968	PW	727	May	1967	NaN	1	1
2316	Iran, Islamic Republic of	Airline of the Islamic Republi	1977	PW	747-100	Jun	1974	NaN	1	1
2317	Iran, Islamic Republic of	Airline of the Islamic Republi	1979	PW	747-100	Jun	1978	NaN	2	2
2318	Iran, Islamic Republic of	Airline of the Islamic Republi	1971	PW	737-200	Jul	1970	NaN	3	3
2319	Iran, Islamic Republic of	Airline of the Islamic Republi	1973	PW	707/720	Aug	1972	NaN	1	1
2320	Iran, Islamic Republic of	Airline of the Islamic Republi	1973	PW	737-200	Aug	1972	NaN	1	1
2321	Iran, Islamic Republic of	Airline of the Islamic Republi	1975	PW	727	Aug	1974	NaN	2	2
2322	Iran, Islamic Republic of	Airline of the Islamic Republi	1974	PW	727	Oct	1973	NaN	3	3
2323	Iran, Islamic Republic of	Airline of the Islamic Republi	1976	PW	747-100	Oct	1973	NaN	2	2
2324	Iran, Islamic Republic of	Airline of the Islamic Republi	1969	PW	707/720	Dec	1968	NaN	1	1
2325	Iran, Islamic Republic of	Airline of the Islamic Republi	1970	PW	707/720	Dec	1968	NaN	1	1
2326	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1977	PW	737-200	Feb	1977	NaN	1	1
2327	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1974	PW	707/720	May	1974	NaN	3	3
2328	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1977	PW	747-200	Jun	1977	NaN	1	1
2329	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1978	PW	707/720	Jun	1977	NaN	2	2
2330	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1978	PW	747-200	Jun	1977	NaN	3	3
2331	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1974	PW	707/720	Sep	1974	NaN	1	1
2332	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1974	PW	707/720	Nov	1974	NaN	2	2
2333	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1976	PW	707/720	Nov	1974	NaN	5	5
2334	Iran, Islamic Republic of	Islamic Republic of Iran Air F	1976	PW	707/720	Dec	1974	NaN	2	2

In [ ]:
```
# COMMENTAIRE
# On constate que les valeurs manquantes ne concernent que l'Iran
# On va vérifier si on peut récupérer la valeur de "Région" en regardant les autres lignes du tableau concernant l'Iran
```

In [ ]:
```
# Stockons toutes les lignes du tableau qui contiennent le mot "Iran", dans la variable lignes_filtrees
colonne_cible = "Country"
mot_specifique = "Iran"
lignes_filtrees = dfOrders[dfOrders[colonne_cible].str.contains(mot_specifique)]
```

```
# Affiche le nombre de lignes et de colonnes du tableau
print("(nombre de ligne, nombre de colonnes):", lignes_filtrees.shape)
display(lignes_filtrees.head())
```

```
(nombre de ligne, nombre de colonnes): (24, 10)
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total
2310	Iran, Islamic Republic of	Airline of the Islamic Republi	1966	PW	727	Apr	1965	NaN	2	2
2311	Iran, Islamic Republic of	Airline of the Islamic Republi	1976	PW	747-200	Apr	1975	NaN	1	1
2312	Iran, Islamic Republic of	Airline of the Islamic Republi	1977	PW	747-200	Apr	1975	NaN	1	1
2313	Iran, Islamic Republic of	Airline of the Islamic Republi	1967	PW	727	May	1967	NaN	1	1
2314	Iran, Islamic Republic of	Airline of the Islamic Republi	1968	PW	727	May	1967	NaN	1	1

In [ ]:
```
# COMMENTAIRE:
# Remarque: on avait 24 valeurs manquantes dans Region (24 lignes concernées).
# Le tableau avec la condition "Colonne Country contient le mot Iran" a également 24 lignes.
# On en déduit qu'il n'y a pas d'autres lignes contenant le pays de l'Iran et il n'y a que l'Iran dont la région est manquante.
```

In [ ]:
```
# La liste de régions ne contient pas de terme "West Asia" ou similaire, donc on va l'ajouter nous-même
dfOrders["Region"] = dfOrders['Region'].fillna("West Asia")

# Verification que le tableau a bien été updaté et qu'on n'a plus de NaN dans la colonne région et autres colonnes
display(dfOrders[dfOrders["Region"].str.contains("West Asia")].head())
dfOrders.isnull().sum()
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total
2310	Iran, Islamic Republic of	Airline of the Islamic Republi	1966	PW	727	Apr	1965	West Asia	2	2
2311	Iran, Islamic Republic of	Airline of the Islamic Republi	1976	PW	747-200	Apr	1975	West Asia	1	1
2312	Iran, Islamic Republic of	Airline of the Islamic Republi	1977	PW	747-200	Apr	1975	West Asia	1	1
2313	Iran, Islamic Republic of	Airline of the Islamic Republi	1967	PW	727	May	1967	West Asia	1	1
2314	Iran, Islamic Republic of	Airline of the Islamic Republi	1968	PW	727	May	1967	West Asia	1	1

Out[ ]:
```
Country           0
Customer Name     0
Delivery Year     0
Engine            0
Model Series      0
Order Month       0
Order Year        0
Region            0
Delivery Total    0
Order Total       0
dtype: int64
```

In [ ]:
```
# COMMENTAIRE
# Il n'y a plus de valeurs manquantes dans le sous-échantillon dfOrders.
# Passons à l'examination des valeurs "Unidentified" dans Region.
```

Nettoyage 4.3 : Stockage des lignes contenant comme valeurs "Unidentified"

In [ ]:
```
# On a pris connaissance de la valeur "Unidentified" en traitant les valeurs manquantes de Region.

# Stockons les lignes du tableau qui respectent la condition que le mot "Unidentified" se trouve dans la colonne Region:
regions_unidentified = dfOrders.loc[dfOrders["Region"] == "Unidentified", :]
display(regions_unidentified.head())
print("(Nombre de lignes, nombre de colonnes:)", regions_unidentified.shape)
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total
5260	Unidentified	Unidentified Customer(s)	2015	GE	777F	Jan	2014	Unidentified	4	4
5270	Unidentified	Unidentified Customer(s)	2021	GE	747-8	Feb	2021	Unidentified	1	1
5284	Unidentified	Unidentified Customer(s)	2017	GE	747-8	Mar	2017	Unidentified	1	1
5285	Unidentified	Unidentified Customer(s)	2022	CF	737 MAX	Mar	2021	Unidentified	1	1
5291	Unidentified	Unidentified Customer(s)	2003	GE	747-400	Apr	2002	Unidentified	1	1

```
(Nombre de lignes, nombre de colonnes:) (21, 10)
```

In [ ]:
```
# COMMENTAIRE:
# On remarque que d'autres colonnes prennent aussi comme valeurs "unidentified"
# Vérifions s'il y a d'autres lignes qui contiennent Unidentified.
```

In [ ]:
```
# Mot à rechercher
mot = 'Unidentified'

# Fonction pour vérifier la présence du mot dans toutes les colonnes
def verif_mot(row):
  """
  Pour une ligne donnée dans un tableau, la fonction vérifie si les colonnes, dont le dtype est object, contiennent le mot "Unidentified".
  Si oui, alors return True.
  Arg:
    row : ligne d'un tableau.
  """
  for col in dfOrders.columns:
    if dfOrders[col].dtypes == "object":
        if mot in row[col]:
            return True
        return False

# Appliquer la fonction à chaque ligne et créer une nouvelle colonne dans dfOrders, contenant le résultat de la fonction (bool)
dfOrders['contient_mot'] = dfOrders.apply(verif_mot, axis=1)

# Créer un nouveau DataFrame filtré contenant les lignes avec le mot
df_unidentified = dfOrders.loc[dfOrders['contient_mot'] == True]

# Supprimer la colonne précédemment créée dans dfOrders
dfOrders = dfOrders.drop("contient_mot", axis=1)

# Afficher le nouveau DataFrame filtré
print("--- Tableau avec toutes les lignes contenant le mot Unidentified, quelque soit la colonne ---")
display(df_unidentified)
print("(Nombre de lignes, nombre de colonnes:", df_unidentified.shape)
```

```
--- Tableau avec toutes les lignes contenant le mot Unidentified, quelque soit la colonne ---
```

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total	contient_mot
5260	Unidentified	Unidentified Customer(s)	2015	GE	777F	Jan	2014	Unidentified	4	4	True
5270	Unidentified	Unidentified Customer(s)	2021	GE	747-8	Feb	2021	Unidentified	1	1	True
5284	Unidentified	Unidentified Customer(s)	2017	GE	747-8	Mar	2017	Unidentified	1	1	True
5285	Unidentified	Unidentified Customer(s)	2022	CF	737 MAX	Mar	2021	Unidentified	1	1	True
5291	Unidentified	Unidentified Customer(s)	2003	GE	747-400	Apr	2002	Unidentified	1	1	True
5292	Unidentified	Unidentified Customer(s)	2003	GE	767-200ER	Apr	2002	Unidentified	1	1	True
5293	Unidentified	Unidentified Customer(s)	2017	GE	777-300ER	Apr	2015	Unidentified	5	5	True
5294	Unidentified	Unidentified Customer(s)	2022	CF	737 MAX	Apr	2021	Unidentified	3	3	True
5302	Unidentified	Unidentified Customer(s)	2014	CF	737-800	May	2013	Unidentified	3	3	True
5303	Unidentified	Unidentified Customer(s)	2015	CF	737-800	May	2013	Unidentified	9	9	True
5304	Unidentified	Unidentified Customer(s)	2016	CF	737-800	May	2013	Unidentified	2	2	True
5305	Unidentified	Unidentified Customer(s)	2022	CF	737 MAX	May	2013	Unidentified	1	1	True
5312	Unidentified	Unidentified Customer(s)	2016	CF	737-800	Jun	2014	Unidentified	1	1	True
5324	Unidentified	Unidentified Customer(s)	2017	GE	747-8F	Jul	2017	Unidentified	2	2	True
5333	Unidentified	Unidentified Customer(s)	2016	GE	777-300ER	Aug	2014	Unidentified	10	10	True
5344	Unidentified	Unidentified Customer(s)	2002	PW	767-300ER	Sep	2001	Unidentified	2	2	True
5345	Unidentified	Unidentified Customer(s)	2003	PW	767-300ER	Sep	2001	Unidentified	1	1	True
5361	Unidentified	Unidentified Customer(s)	2013	GE	747-8F	Nov	2012	Unidentified	2	2	True
5381	Unidentified	Unidentified Customer(s)	2016	GE	777F	Dec	2014	Unidentified	2	2	True
5382	Unidentified	Unidentified Customer(s)	2017	GE	787-9	Dec	2008	Unidentified	3	3	True
5383	Unidentified	Unidentified Customer(s)	2018	GE	787-9	Dec	2008	Unidentified	2	2	True

(Nombre de lignes, nombre de colonnes: (21, 11)

In [ ]:
```
# COMMENTAIRE:
# Le tableau contient 21 lignes, comme le tableau region_unidentified.
# Il n'y a donc pas d'autres lignes contenant le mot "Unidentified".
# Pour notre analyse, faisons le choix de garder ces lignes dans notre sous-échantillon dfOrders pour le moment.
```

Autres colonnes: cohérence des valeurs

In [ ]:
```
dfOrders["Customer Name"].unique()
```

Out[ ]:
```
array(['Ariana Afghan Airlines', 'Air Algerie', 'Tassili Airlines',
       'SonAir', 'TAAG Angola Airlines', 'Aerolineas Argentinas',
       'AUSTRAL - Cielos del Sur', 'Estado Nacional Argentino ENA',
       'Ansett Australia', 'Ansett Worldwide Av. Equipment',
       'Australian Airlines Limited', 'Australian Wedgetail',
       'BDS Australia P-8', 'Jetstar - Duplicated (See JSA)',
       'Qantas Airways', 'Trans-Australia Airlines', 'Virgin Australia',
       'Virgin Australia Airlines', 'Austrian Airlines', 'Lauda Air',
       'Azerbaijan Airlines', 'Silk Way Airlines', 'Bahamasair',
       'Gulf Air', 'Biman Bangladesh Airlines',
       'Belavia Belarusian Airlines', 'Air Belgium',
       'Brussels Airlines SA/NV', 'City Bird', 'DHL International',
       'NATO', 'SABENA Aerospace', 'Sobelair', 'Trans European Airways',
       'AOF Leasing', 'Novel Leasing', 'LAB', 'Cruzeiro',
       'Federative Republic of Brazil', 'GOL Linhas Aereas',
       'LATAM Airlines Brasil', 'Panair do Brasil', 'Transbrasil',
       'Varig Airlines', 'VASP Airlines', 'Royal Brunei Airlines',
       'Cameroon Airlines', 'Republic of Cameroon', 'Air Canada',
       'Canadian Air Force', 'Canadian Airlines', 'CP Air',
       'Dome Petroleum', 'Eastern Provincial Airways',
       'Eldorado Aviation', 'Nordair', 'Pacific Western Airlines',
       'Quebecair', 'Transair (Inactive)', 'Wardair', 'WestJet',
       'Bain Capital Griffin Internati', 'Chilean Air Force Mission',
       'LADECO', 'LATAM Airlines Group', '9 Air', 'Air China',
       'Air China Cargo', 'Air China Southwest Branch',
       'Bank of Communications Leasing', 'CAAC-Civil Aviation of China',
       'Cathay Pacific Airways', 'CATIC', 'CES Leasing Corporation',
       'China Cargo', 'China Eastern Airlines',
       'China Eastern Airlines Wuhan', 'China Eastern Yunnan Airlines',
       'China Northern Airlines', 'China Southern Airlines',
       'China Xinhua Airlines', 'China Xinjiang Airlines',
       'Donghai Airlines', 'Dragon Aviation Leasing',
       'Everbright Financial Leasing', 'Hainan Airlines Holding',
       'Hebei Airlines Company, LTD', 'ICBC Leasing',
       'Jade Cargo International', 'Juneyao Air Co., LTD.',
       'Kunming Airlines', 'Okay Airways Company Limited',
       'Ruili Airlines', 'Shandong Airlines', 'Shanghai Airlines',
       'Shenzhen Airlines', 'Silk Road Leasing', 'Xiamen Airlines',
       'Zhongyuan Airlines', 'Avianca', 'Lignes Aeriennes de Congolaise',
       'Avianca Costa Rica', 'Air Afrique', 'Skyways Leasing', 'CSA',
       'Smartwings, a.s.', 'Maersk Air', 'Sterling Air A/S',
       'Dominicana Airlines', 'TAME', 'AMC Airlines', 'EgyptAir',
       'Government of Egypt', 'Avianca El Salvador',
       'Ceiba Intercontinental', 'Ethiopian Airlines Group',
       'Fiji Airways', 'Finnair', 'Aeromaritime', 'Air Charter',
       'Air France', 'Air-Lib', 'AOM French Airlines',
       'Continent Air Paris', 'Euralair', 'Republic of France',
       'Transavia Company SAS', 'Union Aeromaritime Transport', 'UTA',
       'Air Tahiti Nui', 'Air Gabon', 'Aero Lloyd', 'Air Berlin',
       'Bavaria', 'Condor Flugdienst', 'dba',
       'Deucalion Capital VII Limited', 'German Air Force', 'Germania',
       'Germanwings GmbH', 'LTU', 'Lufthansa', 'Lufthansa Cargo',
       'Suedflug', 'TUI Group', 'Ghana Airways', 'GB Airways',
       'Olympic Airlines', 'Air Guinee', 'SAHSA', 'CDB Aviation',
       'Malev Hungarian Airlines Ltd.', 'Icelandair', 'Air India',
       'BDS Indian Navy P-8I', 'Jet Airways', 'Jet Lite', 'SpiceJet',
       'Vistara', 'Garuda Indonesia', 'Indonesian Air Force-DUPLICATE',
       'Lion Air', 'Pelita Air Service', 'Sriwijaya Air',
       'Airline of the Islamic Republi', 'Islamic Republic of Iran Air F',
       'Iraqi Airways', 'Republic of Iraq', 'Aer Lingus', 'AerCap',
```

'Avia Capital Leasing', 'Avolon - Ireland',
'CDB Financial Leasing', 'DAE 4 Ireland Limited',
'DAE Aerospace Enterprise', 'GPA Group', 'Irish Aerospace',
'Pembroke Capital Limited', 'Ryanair', 'SMBC Aviation Capital',
'Standard Chartered Aviation Fi', 'Timaero Ireland Limited',
'Arkia Israeli Airlines', 'EL AL Israel Airlines',
'MG Aviation Limited', 'Aero Trasporti Italiani',
'AIR ITALY S.p.A.', 'Air Trading 92', 'Alitalia',
'Italian Air Force', 'Italian Air Force Tanker',
'Air Jamaica-See CBL', 'Air Nippon',
'All Nippon Airways Co., Ltd.', 'ANA Holdings Inc.',
'Atlantis Aviation Corporation', 'BDS Japan International Tanker',
'Government of Japan', 'ITOCHU Corporation', 'Japan Airlines',
'Japan Airlines Co., Ltd.', 'Japan ASDF Tanker',
'Japan Asia Airways', 'Japan Domestic Airlines',
'Japan Transocean Air', 'MCAP Japan01 Limited', 'Mitsui & Co.',
'Nippon Cargo Airlines', 'Sumitomo Corporation',
'Ardennes Epsilon', 'Hashemite Kingdom of Jordan',
'Royal Jordanian', 'Air Astana', 'Kazakhstan Airlines',
'SCAT Airlines', 'East African Airways', 'Kenya Airways', 'ALAFCO',
'Kuwait Air Force', 'Kuwait Airways', 'LoadAir Cargo',
'Middle East Airlines', 'Republic of Liberia',
'Cargolux Airlines Internationa', 'Luxair', 'Air Madagascar',
'CALC Aircraft Assets Limited', 'Malaysia Airlines', 'Air Malta',
'Mauritania Airlines', 'Air Mauritius', 'Aeromexico',
'Government of Mexico', 'Lineas Aereas Azteca', 'Mexicana',
'VivaAerobus', 'MIAT Mongolian Airlines', 'Royal Air Maroc',
'Royal Moroccan Air Force', 'LAM', 'Air Namibia', 'Nauru Airlines',
'Nepal Airlines Corporation', 'Air France-KLM Group',
'Air Holland Leasing', 'Dutch Caribbean Airlines',
'ITOCHU AirLease B.V.', 'KLM Royal Dutch Airlines',
'Martinair Cargo', 'Sojitz', 'Transavia Airlines',
'Air New Zealand', 'The Republic of Niger', 'Arik Air',
'Federal Government of Nigeria', 'Midway Airlines',
'Nigeria Airways', 'BDS Norway P-8', 'Busy Bee', 'Mey-Air',
'Norwegian Air Shuttle', 'SAS Norway', 'Oman Air (SAOC)',
'Pakistan International Airline', 'Copa Airlines', 'FAUCETT',
'FUERZA AEREA DEL PERU', 'Philippine Airlines',
'Enter Air Sp. z o.o.', 'LOT Polish Airlines', 'Air Atlantis',
'Republic of Portugal', 'TAP Portugal', 'Qatar Airways',
'Air Austral', 'Socialist Republic of Romania',
'TAROM Romanian Air Transport', 'Aeroflot - Russian Airlines',
'AirBridgeCargo Airlines', 'BLF Ltd.', 'S7 Group',
'Sberbank Leasing', 'UTair Aviation', 'Volga-Dnepr UK Ltd',
'RwandAir', 'Samoa Airways', 'Albaraka', 'Dream Aviation Ltd.',
'Kingdom of Saudi Arabia', 'Mid East Jet', 'Rafic B. Hariri',
'Saudi Arabian Airlines', 'Air Senegal International',
'Republic of Senegal', 'Air Serbia', 'AVIOGENEX', 'Pan Adria',
'BOC Aviation Limited', 'MSA-Malaysia/Singapore A/L',
'Scoot PTE LTD', 'SilkAir', 'Singapore Airlines', 'SkyEurope',
'Adria Airways d.d.', 'Comair Limited', 'South African Airways',
'Asiana Airlines', 'Government of Korea', 'Jeju Air', 'Korean Air',
'Republic of Korea Air Force', 'Air Europa', 'AVIACO',
'Iberia Airlines', 'L.T.E. International Airways',
'Quantum Air S.A.', 'Vingreiser AS', 'Buraq Air',
'Libyan Airlines', 'Sudan Airways', 'Finans AB Balans',
'KB Flygplanet', 'Linjeflyg', 'Malmo Aviation',
'Scandinavian Airlines', 'Transair Sweden', 'Balair/CTA',
'Flightlease', 'Maritime Investment & Shipping',
'Noga Export Import', 'PrivatAir', 'Swiss International Air Lines',

'Swissair', 'Syrianair', 'Air Asia Company', 'China Airlines',
'EVA Air', 'Far Eastern Air Transport', 'Great China Airlines',
'Hwa-Hsia Leasing', 'Mandarin Airlines',
'Republic of China Air Force', 'UNI Airways', 'Somon Air',
'Air Tanzania', 'Tanzania', 'Air Siam', 'Nok Air',
'Royal Thai Air Force', 'Thai Airways International', 'BWIA',
'TunisAir', 'Cyprus Turkish Airlines (KTHY)', 'Onur Air',
'Pegasus Airlines', 'Sky Airlines', 'SunExpress Airlines',
'Turkish Air Force Peace Eagle', 'Turkish Airlines',
'Turkmenhowayollary Agency', 'Goiania',
'Ukraine International Airlines', 'Unidentified Customer(s)',
'Dubai Aerospace Enterprise', 'Dubai Air Wing', 'Emirates',
'Etihad Airways', 'flydubai', 'National Airlines',
'Novus Aviation Capital', 'Presidential Flight', 'Air Europe',
'BDS UK P-8', 'British Airtours', 'British Airways',
'British Caledonian Airways', 'British Eagle International',
'British Royal Air Force', 'buzz', 'easyJet', 'Flightpath',
'Jet2.com Ltd', 'Midland Montagu Leasing', 'Monarch Airlines',
'MyTravel Airways', 'Novair International Airways',
'Orion Airways', 'Paramount Airways - (Inactive)',
'Rolls-Royce A/C Management', 'Security Pacific EuroFinance',
'Thomas Cook Airlines', 'TUI Travel PLC', 'U-Land Airlines',
'Virgin Atlantic Airways', 'West Coast Airlines - UK (Inac',
'PLUNA', '777 Partners', 'Air Florida', 'Air Lease Corporation',
'AirCal', 'Airlift International',
'AirTran Airways (Merged with S', 'Alaska Airlines',
'Aloha Airlines', 'Altavair LLC', 'America West Airlines',
'American Airlines', 'American Flyers Airlines',
'ARAMCO Associated Company', 'ATA Airlines, Inc.',
'ATAircraft One', 'Atlas Air, Inc.', 'Aviation Capital Group',
'Aviation Service & Support', 'BBAM Aircraft Management LP',
'BDS U.S. Navy (P-8A Poseidon)', 'BDS USAF Tanker Program',
'Boeing Capital Corporation', 'Bonanza Airlines',
'Boullioun Aviation Services', 'Braniff Airlines',
'Business Aerotech', 'Business Jet / VIP Customer(s)',
'Capitol Air', 'CIT Aerospace LLC', 'Delta Air Lines', 'DHL',
'DHL Aviation Americas', 'Eastern Air Lines', 'Eastwind Airlines',
'Essex Wire Corp.', 'Executive Jet Aviation', 'FAA',
'FedEx Express', 'Flying Tiger', 'Frontier Airlines',
'GATX Financial Corporation', 'GATX Jet Partners Ltd.',
'Handlingair Establishment', 'Hawaiian Airlines',
'Heller Financial', 'Hughes Airwest, Inc.',
'International Lease Finance Co', 'Itel Air',
'Jackson Square Aviation', 'Jet America Airlines',
'KE Aircraft Leasing', 'Korean Air Force Peace Eye',
'Laker Airways (Bahamas)', 'M&T Aviation USA Inc', 'MarkAir',
'McDonnell Douglas', 'Miami Air', 'Midwest Airlines, Inc.',
'NAS Aviation Services LLC', 'NASA', 'NEA Holdings, Inc.',
'North Central Airlines', 'Northwest-Merged w Delta', 'Oak Hill',
'Ozark Air Lines', 'PACE Airlines', 'Pacific Air Lines',
'Pacific Northern', 'Pacific Southwest Airlines',
'Pan Am World Airways', 'Panagra Cargo', 'Playboy',
'Polaris Aircraft Leasing', 'Pro Air', 'Purdue Airlines',
'Reno Air', 'Republic Airlines', 'Saturn Airways',
'Seaboard World Airlines', 'Southern Air Transport',
'Southern Airways - USA (Inacti', 'Southwest Airlines',
'Standard Airways', 'Starflite Corporation',
'Texas Air Corporation', 'Tigerair', 'Tracinda Corp.',
'Trans Carib Air', 'Transamerica Airlines',
'Transtar Airlines Corporation', 'TWA', 'U.S. Air Force',

```
       'United Airlines', 'United States Navy', 'Universal Airlines',
       'UPS', 'US Airways, Inc.', 'USAF PAR Program',
       'Voyager Aviation Aircraft Leas', 'WEDGE Group',
       'Western Airlines', 'Western Pacific Airlines', 'Wien Air Alaska',
       'World Airways, Inc.', 'Uzbekistan Airways', 'Aeropostal',
       'AVENSA', 'Republic of Venezuela', 'VIASA', 'Vietnam Airlines',
       'Alyemen Airlines', 'Yemenia', 'Zambia Airways', 'Air Zimbabwe'],
      dtype=object)
```

In [ ]:
```
# COMMENTAIRE
# R.A.S
```

In [ ]:
```
print(dfOrders["Engine"].unique())
```

```
['PW' 'GE' 'CF' 'RR' 'NS' 'BR']
```

In [ ]:
```
# COMMENTAIRE
# R.A.S
```

In [ ]:
```
print(dfOrders["Model Series"].unique())
```

```
['727' 'DC-10' '737-200' '737-800' '767-300' '737-700C' '737-600'
 '737-700' '777-300ER' '777-200ER' '747-200' 'MD-80' '707/720' '737 MAX'
 'DC-9' '767-200' '737-300' '737-500' '737-400' '737-700W' '737-800A'
 '717-200' '747-100' '767-300ER' '747-400' '747-300' '787-8' '787-9'
 '767-200ER' '747-400ER' '767-300F' '777-200LR' '757-200' '747-8F' 'MD-11'
 'DC-8' '747-400M' '777F' '777-200' '747-8' '747-400F' '777-300'
 '747-400ERF' 'MD-90' '737-900ER' '737-900' '737-100' '757-200PF'
 '757-300' '747-400D' '787-10' '767-2C' '757-200M' 'BBJ' 'BBJ2' 'BBJ3'
 '767-400ER']
```

In [ ]:
```
# COMMENTAIRE
# R.A.S
```

Ajout colonne "type" et "utilisation" d'avion

In [ ]:
```
# COMMENTAIRE
# Afin de mieux appréhender les données, on peut classer les modèles d'avions par différents critères.
# Pour notre analyse, on utilisera le type de vol (court-courrier, moyen-courrier, long-courrier) ainsi que l'utilisation principale (commercial, cargo, avions privés...)
# Pour se faire, j'ai demandé à plusieurs IA de me classer les modèles de BOEING selon les critères ci-dessus.
# Attention, les avions peuvent être par exemple à la fois court et moyen-courrier. C'est supposé être le type de vol et l'utilisation principaux.
```

In [ ]:
```
# Source du dictionnaire: open.ai - chatgpt 4.0
avions_boeing = {
    'Modele': [
        '727', 'DC-10', '737-200', '737-800', '767-300', '737-700C', '737-600',
        '737-700', '777-300ER', '777-200ER', '747-200', 'MD-80', '707/720', '737 MAX',
        'DC-9', '767-200', '737-300', '737-500', '737-400', '737-700W', '737-800A',
        '717-200', '747-100', '767-300ER', '747-400', '747-300', '787-8', '787-9',
        '767-200ER', '747-400ER', '767-300F', '777-200LR', '757-200', '747-8F', 'MD-11',
        'DC-8', '747-400M', '777F', '777-200', '747-8', '747-400F', '777-300',
        '747-400ERF', 'MD-90', '737-900ER', '737-900', '737-100', '757-200PF',
        '757-300', '747-400D', '787-10', '767-2C', '757-200M', 'BBJ', 'BBJ2', 'BBJ3',
        '767-400ER'
    ],
    'Type': [
        'moyen-courrier', 'long-courrier', 'court-courrier', 'moyen-courrier', 'long-courrier',
        'moyen-courrier', 'court-courrier', 'moyen-courrier', 'long-courrier', 'long-courrier',
```

```
        'long-courrier', 'moyen-courrier', 'long-courrier', 'moyen-courrier', 'court-courrier',
        'long-courrier', 'moyen-courrier', 'court-courrier', 'moyen-courrier', 'moyen-courrier',
        'moyen-courrier', 'court-courrier', 'long-courrier', 'long-courrier', 'long-courrier',
        'long-courrier', 'long-courrier', 'long-courrier', 'long-courrier', 'long-courrier',
        'long-courrier', 'long-courrier', 'moyen-courrier', 'long-courrier', 'long-courrier',
        'long-courrier', 'long-courrier', 'long-courrier', 'long-courrier', 'long-courrier',
        'long-courrier', 'long-courrier', 'long-courrier', 'moyen-courrier', 'moyen-courrier',
        'moyen-courrier', 'court-courrier', 'moyen-courrier', 'moyen-courrier', 'long-courrier',
        'long-courrier', 'long-courrier', 'moyen-courrier', 'long-courrier', 'long-courrier',
        'long-courrier', 'long-courrier'
    ],
    'Utilisation': [
        'avion commercial', 'avion commercial', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion cargo', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion commercial', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion commercial', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion commercial', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion commercial', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion commercial', 'avion cargo', 'avion commercial',
        'avion commercial', 'avion cargo', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion cargo', 'avion commercial', 'avion commercial',
        'avion cargo', 'avion commercial', 'avion cargo', 'avion commercial', 'avion commercial',
        'avion commercial', 'avion commercial', 'avion cargo', 'avion commercial',
        'avion commercial', 'avion commercial', 'avion cargo', 'avion cargo',
        'avion privé et affaires', 'avion privé et affaires', 'avion privé et affaires',
        'avion commercial'
    ]
}
```

```
df_avions_boeing = pd.DataFrame(avions_boeing)
display(df_avions_boeing.head())
display(df_avions_boeing.tail())
```

	Modele	Type	Utilisation
0	727	moyen-courrier	avion commercial
1	DC-10	long-courrier	avion commercial
2	737-200	court-courrier	avion commercial
3	737-800	moyen-courrier	avion commercial
4	767-300	long-courrier	avion commercial

	Modele	Type	Utilisation
52	757-200M	moyen-courrier	avion cargo
53	BBJ	long-courrier	avion privé et affaires
54	BBJ2	long-courrier	avion privé et affaires
55	BBJ3	long-courrier	avion privé et affaires
56	767-400ER	long-courrier	avion commercial

In [ ]:
```
# COMMENTAIRE
# Vérif rapide de la pertinence des données:
# D'après techno-science.net,
```

```
    # 727: moyen courrier
    # DC-10: long courrier
# D'après menkoraviation.com
    # 737-200 court/moyen courrier
# D'après kenya-airways.com
    # 737-800 court/moyen courrier
# BBJ = Boeing Business Jet, ce sont bien des jets privés
# On suppose que ces données ont une certaine pertinence.
```

In [ ]:
```
# Ajout des colonnes Types et Utilisation à dfOrders:
types = []
utilisation = []

for value in dfOrders["Model Series"]:
   ligne_cible = df_avions_boeing[df_avions_boeing["Modele"].str.contains(value)]
   if not ligne_cible.empty:
      type_avion = (ligne_cible["Type"].iloc[0])
      utilisation_avion = (ligne_cible["Utilisation"].iloc[0])
   else:
      type_avion = "Unidentified"
      utilisation_avion = "Unidentified"
   types.append(type_avion)
   utilisation.append(utilisation_avion)

dfOrders["Type"] = types
dfOrders["Utilisation"] = utilisation

dfOrders.head()
```

Out[ ]:

	Country	Customer Name	Delivery Year	Engine	Model Series	Order Month	Order Year	Region	Delivery Total	Order Total	Type	Utilisation
0	Afghanistan	Ariana Afghan Airlines	1968	PW	727	Mar	1968	Central Asia	1	1	moyen-courrier	avion commercial
1	Afghanistan	Ariana Afghan Airlines	1970	PW	727	Apr	1969	Central Asia	1	1	moyen-courrier	avion commercial
2	Afghanistan	Ariana Afghan Airlines	1979	GE	DC-10	Sep	1978	Central Asia	1	1	long-courrier	avion commercial
4	Algeria	Air Algerie	1974	PW	727	Jan	1974	Africa	1	1	moyen-courrier	avion commercial
5	Algeria	Air Algerie	1974	PW	737-200	Jan	1974	Africa	1	1	court-courrier	avion commercial

Récap info générales du sous-échantillon dfOrders suite à nettoyage

In [ ]:
```
# Affichons les stats de notre sous-échantillon dfOrders suite au nettoyage et aux ajouts de colonne.
generate_stats(dfOrders, "dfOrders")
```

```
---Infos générales sur les données du tableau: dfOrders ---
Dimensions du tableau (nombre de lignes, nombre de colonnes): (8047, 12)
```

	dType	naCount	count	unique	top	freq	freq%	mean	std	min	25%	50%	75%	max
Delivery Year	int32	0	8047.0	NaN	NaN	NaN	NaN	1993.957748	16.924698	1958.0	1980.0	1995.0	2009.0	2022.0
Order Year	int64	0	8047.0	NaN	NaN	NaN	NaN	1991.264571	15.962678	1955.0	1978.0	1992.0	2005.0	2022.0
Delivery Total	int64	0	8047.0	NaN	NaN	NaN	NaN	2.985585	3.336395	1.0	1.0	2.0	3.0	50.0
Order Total	int64	0	8047.0	NaN	NaN	NaN	NaN	2.983224	3.337036	0.0	1.0	2.0	3.0	50.0
Country	object	0	8047	129	USA	2855	35.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Customer Name	object	0	8047	510	International Lease Finance Co	315	3.9	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Engine	object	0	8047	6	PW	2995	37.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Model Series	object	0	8047	57	737-800	1126	14.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Order Month	object	0	8047	12	Dec	1160	14.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Region	object	0	8047	14	North America	3058	38.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	object	0	8047	3	moyen-courrier	3590	44.6	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Utilisation	object	0	8047	3	avion commercial	7199	89.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [ ]:
```
# COMMENTAIRE
# On a maintenant beaucoup plus d'infos apparaissant dans le tableau des infos générales
# Plus de valeurs manquantes mais attention, on a gardé les lignes Unidentified. On n'a plus la colonne Unfilled Orders et on n'a plus les lignes où Delivery Year était manquant.
# Les colonnes passées en numérique nous apprennent que dans notre sous-échantillon, les ordres pris en compte ont été passés entre 1955 et 2022, pour des livraisons entre 1958 et 2022.
```

Analyse des données

Import des librairies utiles

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Seaborn theme
sns.set_theme(style="darkgrid")
```

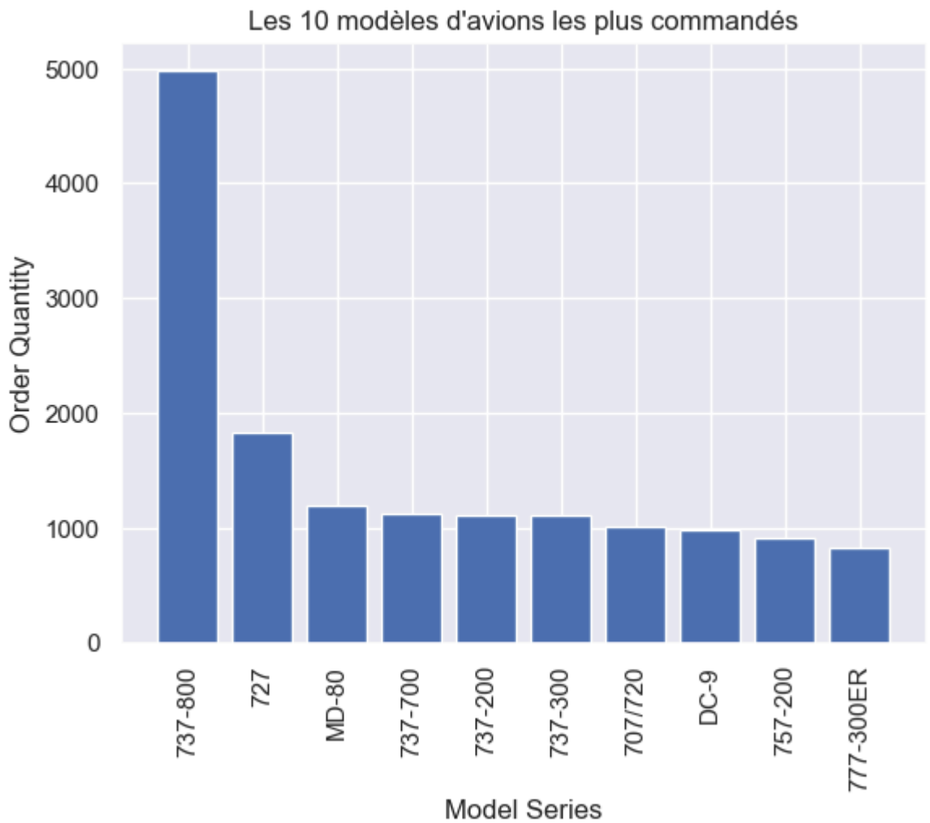
Modèles les plus commandés (histogramme)

In [ ]:
```
# Sous-échantillon utilisé: dfOrders
# Création sous-échantillon avec uniquement Model Series et Order Total
model_orders = dfOrders.groupby('Model Series')['Order Total'].sum().reset_index()

# Tri du nouveau sous-échantillon par modèle les plus commandés
model_orders = (model_orders.sort_values(by="Order Total", ascending=False))
display(model_orders.head())
```

	Model Series	Order Total
13	737-800	4974
2	727	1831
55	MD-80	1191
10	737-700	1128
5	737-200	1114

In [ ]:
```
# COMMENTAIRE
# Ainsi, on connaît le nombre total de commandes passées pour chaque modèle.
```

In [ ]:
```
# Comme il y a 57 modèles d'avions, on s'intéresse aux avions les plus commandés
range = 10

X = "Model Series"
Y = "Order Total"
plt.bar(height=model_orders[Y][:range], x=model_orders[X][:range])
plt.xticks(rotation=90)
plt.xlabel("Model Series")
plt.ylabel("Order Quantity")
plt.title(f"Les {range} modèles d'avions les plus commandés")
plt.show()
```



In [ ]:
```
# COMMENTAIRE
# On constate que le 737-800 se démarque fortement des autres modèles.

# Photo du 737-800
from IPython.display import Image
Image(url="boeing_737_800_generic_white_0000_turbosquid.jpg", width=600)
```

Out[ ]:

% de l'ensemble des clients ayant déjà commandé le top modèle (valeur)

In [ ]:
```
# On peut se demander quel pourcentage de l'ensemble des clients de Boeing ont déjà commandé une 737-800?

customer_737800 = dfOrders["Customer Name"][dfOrders["Model Series"] == "737-800"].nunique()*100/dfOrders["Customer Name"].nunique()
customer_737800 = round(customer_737800)
print(f"Ainsi, environ {customer_737800}% des clients de Boeing ont déjà commandé un 737-800")
```

```
Ainsi, environ 25% des clients de Boeing ont déjà commandé un 737-800
```

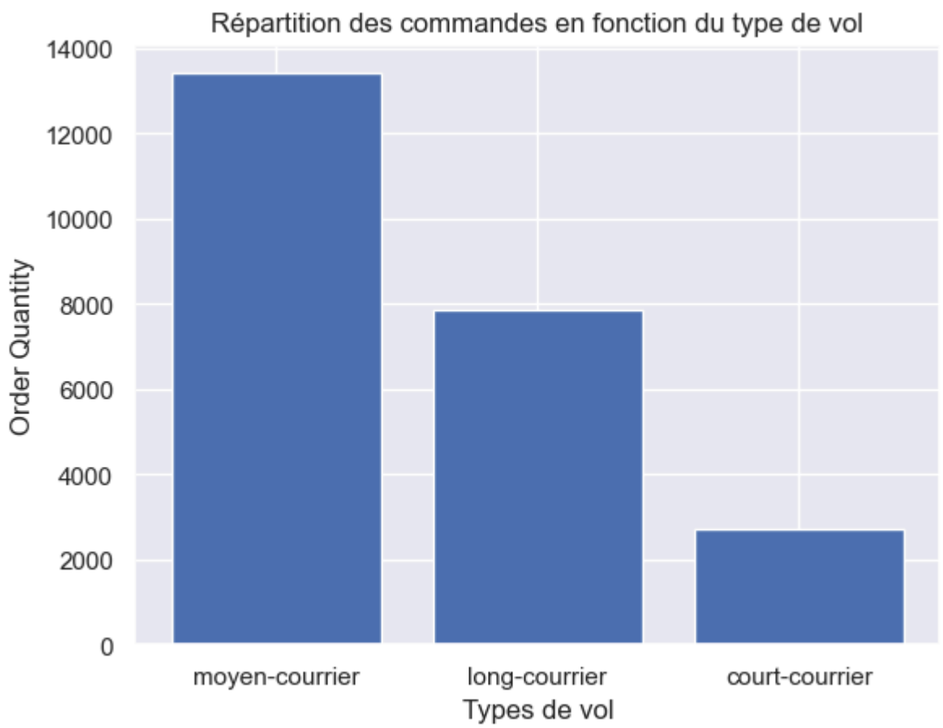
Type le plus commandé (histogramme)

In [ ]:
```
# Sous-échantillon utilisé: dfOrders
# Création sous-échantillon avec uniquement Type et Order Total

type_orders = dfOrders.groupby('Type')['Order Total'].sum().reset_index()

# Tri du nouveau sous-échantillon par modèle les plus commandés
type_orders = (type_orders.sort_values(by="Order Total", ascending=False))
display(type_orders.head())
```

	Type	Order Total
2	moyen-courrier	13410
1	long-courrier	7863
0	court-courrier	2733

In [ ]:
```
# COMMENTAIRE
# Ainsi, on connaît le nombre total de commandes passées pour chaque type de vol.
```

In [ ]:
```
range = len(type_orders)

X = "Type"
Y = "Order Total"

plt.bar(height=type_orders[Y][:range], x=type_orders[X][:range])
plt.xticks(rotation=0)
plt.xlabel("Types de vol")
plt.ylabel("Order Quantity")
plt.title("Répartition des commandes en fonction du type de vol")
plt.show()
```



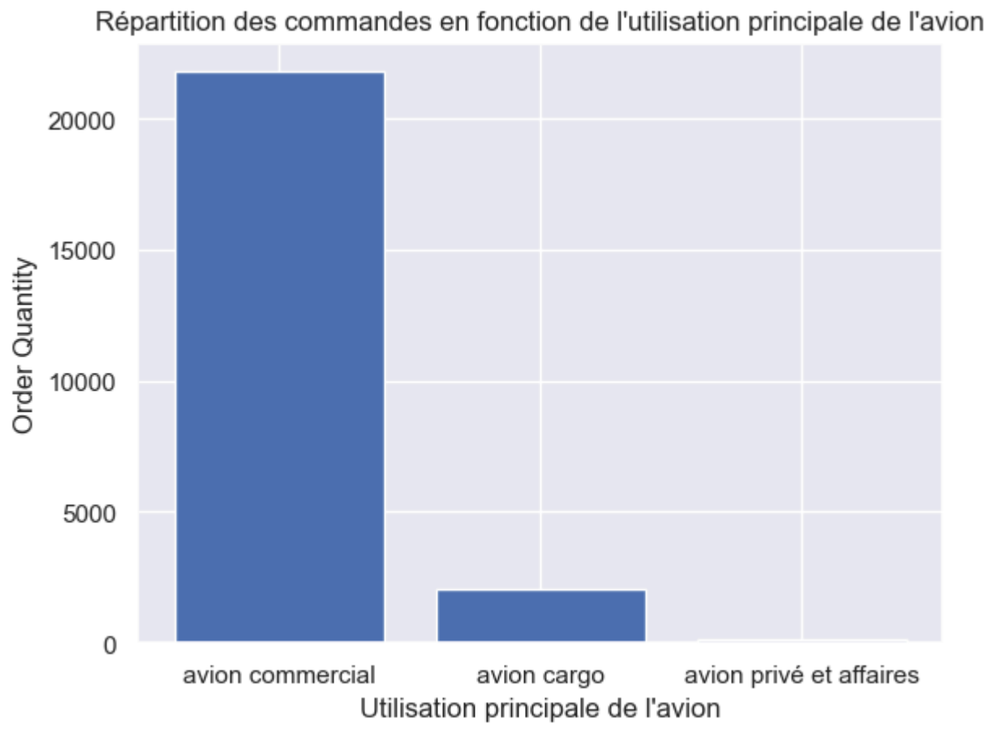
Utilisation la plus commune (histogramme)

In [ ]:
```
# Sous-échantillon utilisé: dfOrders
# Création sous-échantillon avec uniquement Utilisation et Order Total

utilisation_orders = dfOrders.groupby('Utilisation')['Order Total'].sum().reset_index()

# Tri du nouveau sous-échantillon par modèle les plus commandés
utilisation_orders = (utilisation_orders.sort_values(by="Order Total", ascending=False))
display(utilisation_orders.head())
```

	Utilisation	Order Total
1	avion commercial	21815
0	avion cargo	2039
2	avion privé et affaires	152

In [ ]:
```
# COMMENTAIRE
# Ainsi, on connaît le nombre total de commandes passées pour chaque utilisation d'avion.
```

In [ ]:
```
range = len(utilisation_orders)

X = "Utilisation"
Y = "Order Total"

plt.bar(height=utilisation_orders[Y][:range], x=utilisation_orders[X][:range])
plt.xticks(rotation=0)
plt.xlabel("Utilisation principale de l'avion")
plt.ylabel("Order Quantity")
plt.title("Répartition des commandes en fonction de l'utilisation principale de l'avion")
plt.show()
```



Total des commandes passées par Region (histogramme)

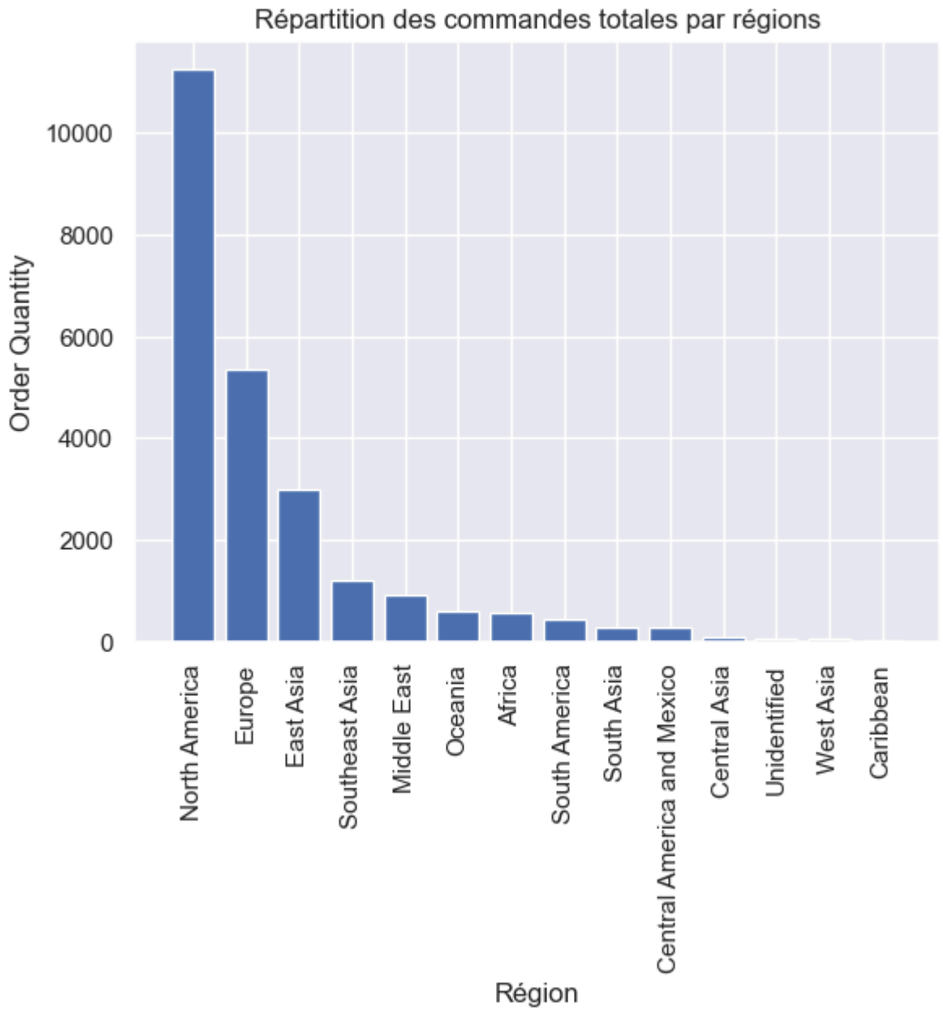
In [ ]:
```
# Sous-échantillon utilisé: dfOrders
# Création sous-échantillon avec uniquement Region et Order Total

region_orders = dfOrders.groupby('Region')['Order Total'].sum().reset_index()
```

```
# Tri du nouveau sous-échantillon par modèle les plus commandés
region_orders = (region_orders.sort_values(by="Order Total", ascending=False))
display(region_orders.head())
```

	Region	Order Total
7	North America	11241
5	Europe	5357
4	East Asia	2974
11	Southeast Asia	1207
6	Middle East	918

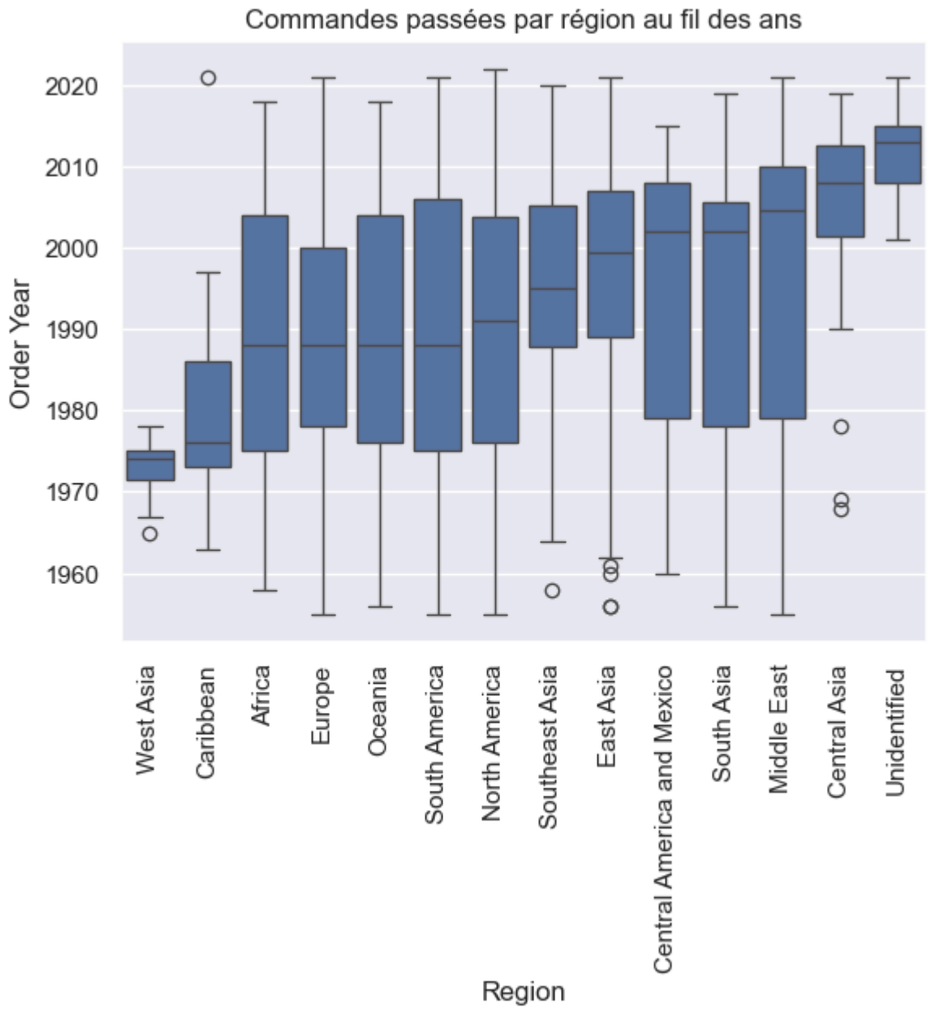
In [ ]:
```
# COMMENTAIRE
# Ainsi, on connaît le nombre total de commandes passées pour chaque Region du monde.
```

In [ ]:
```
range = len(region_orders)

X = "Region"
Y = "Order Total"

plt.bar(height=region_orders[Y][:range], x=region_orders[X][:range])
plt.xticks(rotation=90)
plt.xlabel("Région")
plt.ylabel("Order Quantity")
plt.title("Répartition des commandes totales par régions")
plt.show()
```



Commandes passées par Region au fil des ans (boxplot)

In [ ]:
```
region_year = dfOrders[["Region", "Order Year"]]

# Order
my_order = region_year.groupby(["Region"])["Order Year"].median().sort_values(ascending=True)

sns.boxplot(data = region_year, x = "Region", y = 'Order Year', order=my_order.index)
plt.xticks(rotation = 90)
plt.title("Commandes passées par région au fil des ans")
plt.show()
```



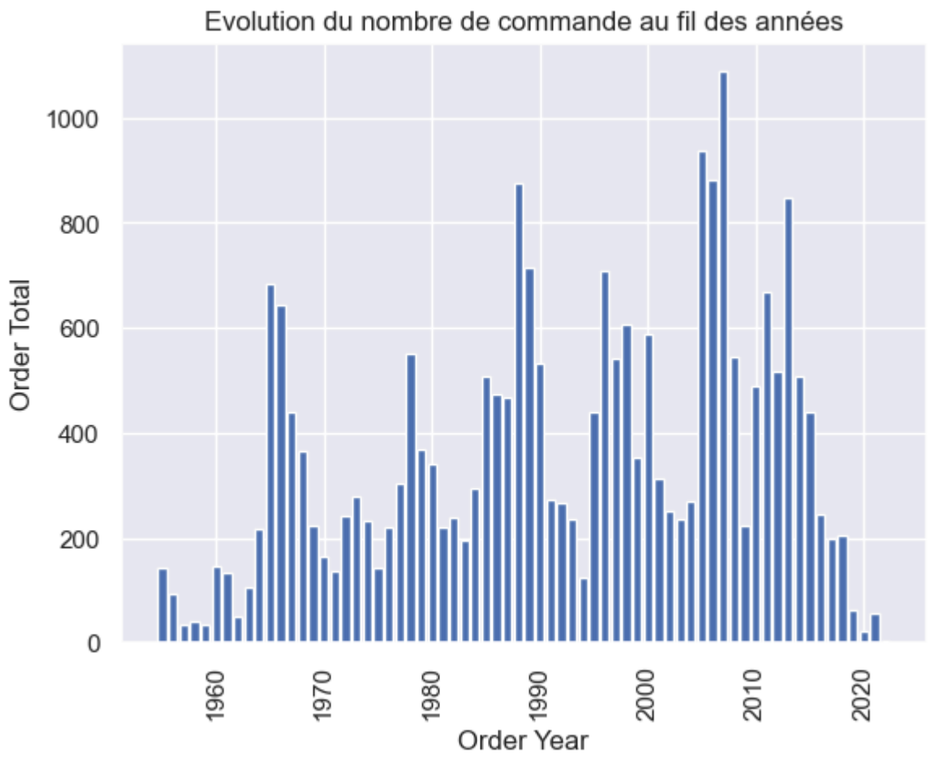
Evolution du nombre de commandes par année (histogramme)

In [ ]:
```
# Sous-échantillon utilisé: dfOrders
# Création sous-échantillon

df_filtered = dfOrders.groupby("Order Year")["Order Total"].sum().reset_index()
display(df_filtered)
```

	Order Year	Order Total
0	1955	143
1	1956	92
2	1957	35
3	1958	41
4	1959	35
...	...	...
63	2018	204
64	2019	64
65	2020	21
66	2021	55
67	2022	3

68 rows × 2 columns

In [ ]:
```
range = len(df_filtered)

X = "Order Year"
Y = "Order Total"

plt.bar(height=df_filtered[Y][:range], x=df_filtered[X][:range])
plt.xticks(rotation=90)
plt.xlabel(X)
plt.ylabel(Y)
plt.title("Evolution du nombre de commande au fil des années")
plt.show()
```



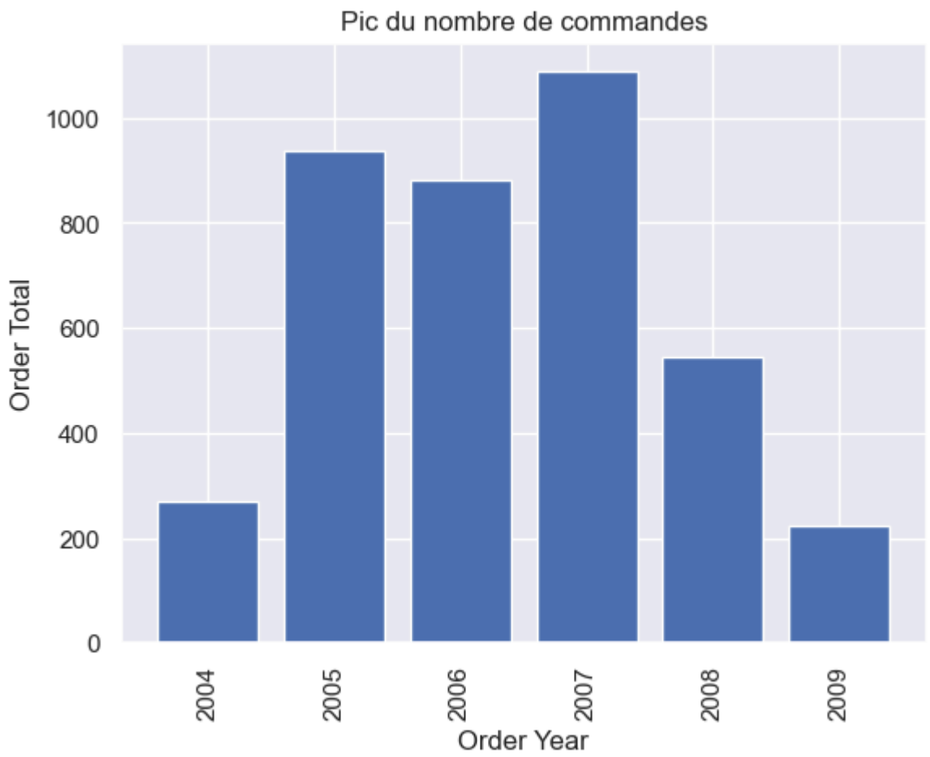
In [ ]:
```
# COMMENTAIRE
# Il y a eu un pic pas loin de 2010, on peut regarder de plus près
```

In [ ]:
```
# Création mask
condition = (df_filtered["Order Year"] >= 2004) & (df_filtered["Order Year"] < 2010)

# Création sous-échantillon filtré avec le mask
df_intervalle = df_filtered.loc[condition]
display(df_intervalle)
```

	Order Year	Order Total
49	2004	271
50	2005	936
51	2006	882
52	2007	1088
53	2008	544
54	2009	222

In [ ]:
```
range = len(df_intervalle)

X = "Order Year"
Y = "Order Total"

plt.bar(height=df_intervalle[Y][:range], x=df_intervalle[X][:range])
plt.xticks(rotation=90)
```

```
plt.xlabel(X)
plt.ylabel(Y)
plt.title("Pic du nombre de commandes")
plt.show()
```



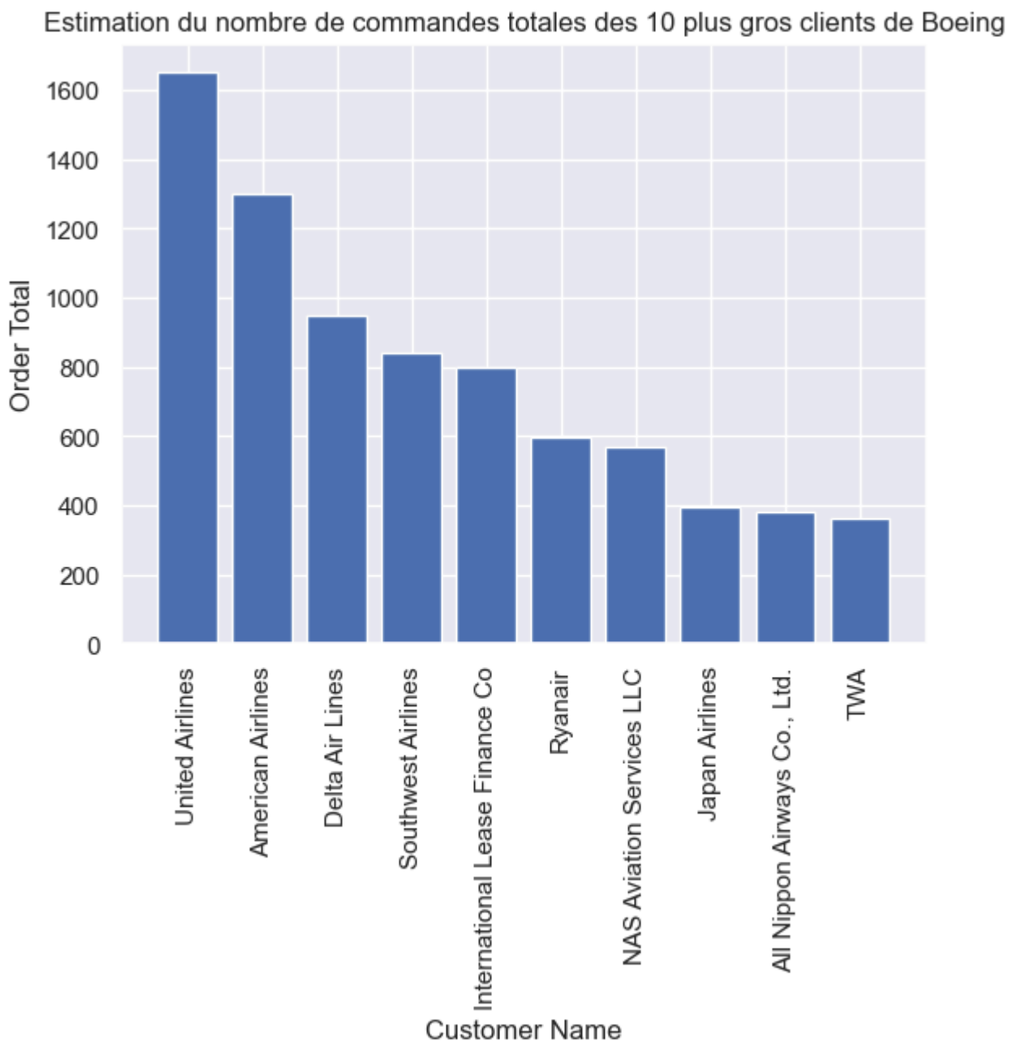
Clients ayant le plus commandé (histogramme)

In [ ]:
```
# Sous-échantillon utilisé: dfOrders
# Création sous-échantillon avec uniquement Customer Name et Order Total

customer_order = dfOrders.groupby("Customer Name")["Order Total"].sum().reset_index()

# Tri du nouveau sous-échantillon par nombre de commandes
customer_order = (customer_order.sort_values(by="Order Total", ascending=False))
display(customer_order.head())
```

	Customer Name	Order Total
478	United Airlines	1651
76	American Airlines	1299
168	Delta Air Lines	950
423	Southwest Airlines	841
240	International Lease Finance Co	799

In [ ]:
```
range = 10
```

```
X = "Customer Name"
Y = "Order Total"

plt.bar(height=customer_order[Y][:range], x=customer_order[X][:range])
plt.xticks(rotation=90)
plt.xlabel(X)
plt.ylabel(Y)
plt.title(f"Estimation du nombre de commandes totales des {range} plus gros clients de Boeing")
plt.show()
```



% de l'ensemble des clients ayant réalisé 5 commandes ou moins au total (valeur)

In [ ]:
```
customer_5max = customer_order["Customer Name"][customer_order["Order Total"] <= 5].nunique()*100/customer_order["Customer Name"].nunique()
customer_5max = round(customer_5max)
print(f"Ainsi, environ {customer_5max}% des clients de Boeing ont réalisé au maximum 5 commandes.")
```

```
Ainsi, environ 43% des clients de Boeing ont réalisé au maximum 5 commandes.
```

Conclusion

Synthèse des commentaires

In [ ]:
```
# COMMENTAIRE
# Ce dataframe nous a permis d'appréhender l'influence de Boeing dans le monde via le nombre de commandes réalisées depuis 1955.
# Les principaux clients de l'entreprise sont des compagnies américaines et ce sont les avions commerciaux qui sont le plus commandés.
```

Limitation des données

In [ ]:
```
# COMMENTAIRE
# Rappelons tout de même que les résultats représentent une estimation, sachant que le sous-échantillon dfOrders représente 89% du tableau brut (Commandes supposées non honorées non prises en
# Par ailleurs, le tableau brut ainsi que le sous-échantillon dfOrders contient des lignes avec des valeurs non-identifiées (sauf concernant le nombre de commandes). Toutefois, cela représent
```