Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Отчет по ЛР1

«Разведочный анализ данных. Исследование и визуализация данных»

по дисциплине «Технологии машинного обучения»

Выполнил:
студент группы ИУ5-63Б
М.В. Дудник

Проверил:
Гапанюк Ю.Е.

2022 г.

Описание задания:

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь.

- Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных, например из Scikit-learn.

- Пример преобразования датасетов Scikit-learn в Pandas Dataframe можно посмотреть здесь.

Для лабораторных работ не рекомендуется выбирать датасеты большого размера.

- Создать ноутбук, который содержит следующие разделы:

1. Текстовое описание выбранного Вами набора данных.

2. Основные характеристики датасета.

3. Визуальное исследование датасета.

4. Информация о корреляции признаков.

- Сформировать отчет и разместить его в своем репозитории на github.

Лабораторная работа №1: "Разведочный анализ данных. Исследование и визуализация данных".

Текстовое описание набора данных

Датасет `Delivery-truck-trip-data.csv` содержит информацию о грузоперевозках.

Параметры:

- *BookingID* - уникальный идентификатор поездки,
- *Market/Regular* - тип поездки: Regular - поставщики, с которыми будет заключен контракт; Market - поставщик, с которым не будет контракта,
- *BookingIDDate* - дата бронирования,
- *vehicle_no* - номер транспортного средства,
- *OriginLocation* - место отправления,
- *DestinationLocation* - место назначения,
- *Org_lat_lon* - широта/долгота места отправления,
- *Des_lat_lon* - широта/долгота места назначения,
- *Planned_ETA* - планируемое расчетное время прибытия,
- *actual_eta* - фактическое время прибытия,
- *ontime* - рассчитано на основе Planned_ETA и actual_eta: True - если грузовик прибыл вовремя; False - если грузовик прибыл с задержкой,
- *trip_start_date* - дата/время начала поездки,
- *trip_end_date* - дата/время окончания поездки - на основании документации (не может учитываться при расчете задержки),
- *TRANSPORTATION_DISTANCE_IN_KM* - общее количество км пути,
- *vehicleType* - тип транспортного средства,
- *customerID* - сведения о клиенте (идентификатор),
- *customerNameCode* - сведения о клиенте (имя),
- *supplierID* - сведения о поставщике транспортного средства (идентификатор),
- *supplierNameCode* - сведения о поставщике транспортного средства (имя),
- *Material Shipped* - отгруженный материал.

Подключение библиотек для анализа данных

In [1]:
```
import datetime

import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
from sklearn.preprocessing import PolynomialFeatures, OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.impute import SimpleImputer

import re

warnings.simplefilter('ignore')
```

Загрузка датасета из файла `Delivery-truck-trip-data.csv`

In [2]:
```
data = pd.read_csv('Delivery-truck-trip-data.csv', encoding='windows-1251')
```

Основные характеристики датасета

Выведем первые 5 строк датасета для проверки корректного импорта данных:

In [3]:
```
data.head()
```

Out[3]:

	BookingID	Market/Regular	BookingID_Date	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon
0	MVCV0000927/082021	Market	8/17/2020	KA590408	TVSLSL-PUZHAL-HUB,CHENNAI,TAMIL NADU	ASHOK LEYLAND PLANT 1-HOSUR,HOSUR,KARNATAKA	13.1550,80.1960
1	VCV00014271/082021	Regular	8/27/2020	TN30BC5917	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
2	VCV00014382/082021	Regular	8/27/2020	TN22AR2748	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8710,79.7390
3	VCV00014743/082021	Regular	8/28/2020	TN28AQ0781	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
4	VCV00014744/082021	Regular	8/28/2020	TN68F1722	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8720,79.6320

Видим, что данные загружены корректно. Разбиения по строкам и столбцам произведены верно. Проблем с кодировкой не возникло.

Узнаем размер датасета:

In [4]:
```
print(f'Количество записей: {data.shape[0]}\nКоличество параметров: {data.shape[1]}')
```

```
Количество записей: 6878
Количество параметров: 20
```

Посмотрим краткую информацию обо всех параматрах датасета:

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6878 entries, 0 to 6877
Data columns (total 20 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   BookingID                     6878 non-null   object 
 1   Market/Regular                6878 non-null   object 
 2   BookingID_Date                6878 non-null   object 
 3   vehicle_no                    6878 non-null   object 
 4   Origin_Location               6878 non-null   object 
 5   Destination_Location          6878 non-null   object 
 6   Org_lat_lon                   6878 non-null   object 
 7   Des_lat_lon                   6878 non-null   object 
 8   Planned_ETA                   6878 non-null   object 
 9   actual_eta                    6878 non-null   object 
 10  ontime                        6878 non-null   bool   
 11  trip_start_date               6878 non-null   object 
 12  trip_end_date                 6878 non-null   object 
 13  TRANSPORTATION_DISTANCE_IN_KM  6878 non-null   float64
 14  vehicleType                   6878 non-null   object 
 15  customerID                    6878 non-null   object 
 16  customerNameCode              6878 non-null   object 
 17  supplierID                    6878 non-null   object 
 18  supplierNameCode              6878 non-null   object 
 19  Material Shipped              6878 non-null   object 
dtypes: bool(1), float64(1), object(18)
memory usage: 1.0+ MB
```

Видим, что в датасете присутствуют данные трёх типов: строковые (`object`), вещественные (`float64`) и логические (`bool`). Также узнаём, что во всех столбцах присутствует ровно `6878` значений, следовательно у нас отсутствуют пустые ячейки, что говорит об отсутствии явных пропусков данных в датасете.

Пропущенные данные

Убедимся еще раз, что в датасете нет пропущенных данных. Для этого выведем список параметров датасета и для каждого из них найдём количество `null` значений.

In [6]:
```
for column in data.columns:
    print(f'{column}: {data[column].isnull().sum()} null values')
```

```
BookingID: 0 null values
Market/Regular : 0 null values
BookingID_Date: 0 null values
vehicle_no: 0 null values
Origin_Location: 0 null values
Destination_Location: 0 null values
Org_lat_lon: 0 null values
Des_lat_lon: 0 null values
Planned_ETA: 0 null values
actual_eta: 0 null values
ontime: 0 null values
trip_start_date: 0 null values
trip_end_date: 0 null values
TRANSPORTATION_DISTANCE_IN_KM: 0 null values
vehicleType: 0 null values
customerID: 0 null values
customerNameCode: 0 null values
supplierID: 0 null values
supplierNameCode: 0 null values
Material Shipped: 0 null values
```

Видим, ни в одном столбце нет пустых ячеек, что говорит об отсутствии явных пропусков данных в датасете.

Неинформативные значения

Теперь проведём поиск неинформативных параметров, которые не пригодятся в дальнейшем при анализе. Неинформативными будем считать такие параметры, значения которых являются уникальными либо, наоборот, в абсолютном большинстве принимают одно и то же значние. Для поиска таких параметров посчитаем количество уникальных значений в каждом столбце. Тогда неинформативными будут параметры, количество уникальных значений которого равно 1 либо очень близко к количеству записей всего датасета.

Примечание: параметр `BookingID` служит для идентификации записей, поэтому хоть все его значения и являются уникальным, мы его не удаляем.

In [7]:
```
print(f'Всего записей: {data.shape[0]}')
print('--------------------')
for column in data.columns:
    print(f'{column}: {data[column].value_counts().count()} уникальных значений', end='\n\n')
```

```
Всего записей: 6878
--------------------
BookingID: 6873 уникальных значений

Market/Regular : 2 уникальных значений

BookingID_Date: 388 уникальных значений

vehicle_no: 2325 уникальных значений

Origin_Location: 180 уникальных значений

Destination_Location: 520 уникальных значений

Org_lat_lon: 173 уникальных значений

Des_lat_lon: 522 уникальных значений

Planned_ETA: 5284 уникальных значений

actual_eta: 5968 уникальных значений

ontime: 2 уникальных значений

trip_start_date: 5011 уникальных значений

trip_end_date: 4691 уникальных значений

TRANSPORTATION_DISTANCE_IN_KM: 564 уникальных значений

vehicleType: 44 уникальных значений

customerID: 39 уникальных значений

customerNameCode: 39 уникальных значений

supplierID: 321 уникальных значений

supplierNameCode: 309 уникальных значений

Material Shipped: 1406 уникальных значений
```

Видим, что неинформативными являются параметры `Planned_ETA`, `actual_eta`, `trip_start_date`, `trip_end_date`. Они не пригодятся нам при дальнейшем анализе, поэтому удалим их:

In [8]:
```
print(f"Параметры датасета: {data.columns}.")
del data['Planned_ETA']
del data['actual_eta']
del data['trip_start_date']
del data['trip_end_date']
print(f"Оставшиеся параметры датасета: {data.columns}.")
```

```
Параметры датасета: Index(['BookingID', 'Market/Regular ', 'BookingID_Date', 'vehicle_no',
       'Origin_Location', 'Destination_Location', 'Org_lat_lon', 'Des_lat_lon',
       'Planned_ETA', 'actual_eta', 'ontime', 'trip_start_date',
       'trip_end_date', 'TRANSPORTATION_DISTANCE_IN_KM', 'vehicleType',
       'customerID', 'customerNameCode', 'supplierID', 'supplierNameCode',
       'Material Shipped'],
      dtype='object').
Оставшиеся параметры датасета: Index(['BookingID', 'Market/Regular ', 'BookingID_Date', 'vehicle_no',
       'Origin_Location', 'Destination_Location', 'Org_lat_lon', 'Des_lat_lon',
       'ontime', 'TRANSPORTATION_DISTANCE_IN_KM', 'vehicleType', 'customerID',
       'customerNameCode', 'supplierID', 'supplierNameCode',
       'Material Shipped'],
      dtype='object').
```

Преобразование данных

Ещё раз посмотрим на наши данные:

In [9]:
```
data.head()
```

Out[9]:

	BookingID	Market/Regular	BookingID_Date	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon
0	MVCV0000927/082021	Market	8/17/2020	KA590408	TVSLSL-PUZHAL-HUB,CHENNAI,TAMIL NADU	ASHOK LEYLAND PLANT 1-HOSUR,HOSUR,KARNATAKA	13.1550,80.1960
1	VCV00014271/082021	Regular	8/27/2020	TN30BC5917	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
2	VCV00014382/082021	Regular	8/27/2020	TN22AR2748	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8710,79.7390
3	VCV00014743/082021	Regular	8/28/2020	TN28AQ0781	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
4	VCV00014744/082021	Regular	8/28/2020	TN68F1722	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8720,79.6320

Выведем типы данных для всех столбцов:

In [10]:
```
data.dtypes
```

Out[10]:

```
BookingID                        object
Market/Regular                   object
BookingID_Date                   object
vehicle_no                       object
Origin_Location                  object
Destination_Location             object
Org_lat_lon                      object
Des_lat_lon                      object
ontime                             bool
TRANSPORTATION_DISTANCE_IN_KM     float64
vehicleType                      object
customerID                       object
customerNameCode                 object
supplierID                       object
supplierNameCode                 object
Material Shipped                 object
dtype: object
```

Признак `BookingID_Date`

Признак `BookingID_Date` можно сделать датой.

In [11]:
```
data['BookingID_Date'] = data['BookingID_Date'].map(lambda x:pd.to_datetime(x, format='%m/%d/%Y'))
data.head()
```

Out[11]:

	BookingID	Market/Regular	BookingID_Date	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon
0	MVCV0000927/082021	Market	2020-08-17	KA590408	TVSLSL-PUZHAL-HUB,CHENNAI,TAMIL NADU	ASHOK LEYLAND PLANT 1-HOSUR,HOSUR,KARNATAKA	13.1550,80.1960
1	VCV00014271/082021	Regular	2020-08-27	TN30BC5917	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
2	VCV00014382/082021	Regular	2020-08-27	TN22AR2748	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8710,79.7390
3	VCV00014743/082021	Regular	2020-08-28	TN28AQ0781	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
4	VCV00014744/082021	Regular	2020-08-28	TN68F1722	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8720,79.6320

Дубликаты

Проверим данные на наличие дубликатов. Для начала посмотрим, все ли значения параметра `BookingID` уникальны.

In [12]:
```
print(f"Уникальных значений параметра 'BookingID': {data['BookingID'].unique().size}.")
print(f"Количество записей в датасете: {data.shape[0]}.")
```

```
Уникальных значений параметра 'BookingID': 6873.
Количество записей в датасете: 6878.
```

Видим, что количество уникальных значений параметра не совпадает с количеством записей в датасете. Следовательно в данном столбце есть дубликаты. Удалим дубликаты из датасета, воспользуясь методом `pd.duplicated`:

In [13]:
```
duplicate_flags = data.duplicated(subset=data.columns[0])
print('Количество найденных дубликатов:', duplicate_flags.sum())
print(f'Исходное количество записей: {data.shape[0]}')
data.drop(data[duplicate_flags].index, inplace=True)
print(f'Оставшееся количество записей: {data.shape[0]}')
```

```
Количество найденных дубликатов: 5
Исходное количество записей: 6878
Оставшееся количество записей: 6873
```

Другие параметры могут содержать неуникальные значения и это не будет являться признаком наличия дубликатов, так как параметры могут совпадать у разных доставок. Поэтому проверить на уникальность целиковые записи, то есть абсолютное совпадение всех параметров за исключением `BookingID`, который уже был проверен и очищен ранее. Для этого переведём все строковые данные в нижний регистр и затем воспользуемся методом `pd.duplicated`.

In [14]:
```
str_columns = data.dtypes[data.dtypes == object].index
data_lower = data.copy()
for column in str_columns:
    data_lower[column] = data[column].apply(lambda x:x.lower())
data_lower.head()
```

Out[14]:

	BookingID	Market/Regular	BookingID_Date	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon	Des_lat_
0	mvcv0000927/082021	market	2020-08-17	ka590408	tvslsl-puzhal-hub,chennai,tamil nadu	ashok leyland plant 1-hosur,hosur,karnataka	13.1550,80.1960	12.7400,77.8
1	vcv00014271/082021	regular	2020-08-27	tn30bc5917	daimler india commercial vehicles,kanchipuram,...	daimler india commercial vehicles,kanchipuram,...	12.8390,79.9540	12.8390,79.9
2	vcv00014382/082021	regular	2020-08-27	tn22ar2748	lucas tvs ltd-pondy,pondy,pondicherry	lucas tvs ltd-pondy,pondy,pondicherry	11.8710,79.7390	11.8710,79.7
3	vcv00014743/082021	regular	2020-08-28	tn28aq0781	daimler india commercial vehicles,kanchipuram,...	daimler india commercial vehicles,kanchipuram,...	12.8390,79.9540	12.8390,79.9
4	vcv00014744/082021	regular	2020-08-28	tn68f1722	lucas tvs ltd-pondy,pondy,pondicherry	lucas tvs ltd-pondy,pondy,pondicherry	11.8720,79.6320	11.8720,79.6

In [15]:
```
duplicate_flags = data_lower.duplicated(subset=data_lower.columns[1:])
print('Количество найденных дубликатов:', duplicate_flags.sum())
```

```
Количество найденных дубликатов: 50
```

Убедимся, что эти данные на самом деле являются дубликатами. Для этого выведем несколько примеров повторяющихся записей.

In [16]:
```
data[(data_lower.duplicated(subset=data.columns[1:], keep=False))].sort_values('TRANSPORTATION_DISTANCE_
```

Out[16]:

	BookingID	Market/Regular	BookingID_Date	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon	
1002	VCV00008376/082021	Regular	2020-08-18	TN88D4133	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540	1
975	VCV00008633/082021	Regular	2020-08-18	TN88D4133	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540	1
286	VCV00013291/082021	Regular	2020-08-26	TN30BB1036	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540	1
338	VCV00013105/082021	Regular	2020-08-26	TN30BB1036	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540	1
394	VCV00012401/082021	Regular	2020-08-25	TN28AQ0975	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540	1

Удалим дубликаты из датасета:

In [17]:
```
print(f'Исходное количество записей: {data.shape[0]}')
data.drop(data[duplicate_flags].index, inplace=True)
data_lower.drop(data_lower[duplicate_flags].index, inplace=True)
print(f'Оставшееся количество записей: {data.shape[0]}')
```

```
Исходное количество записей: 6873
Оставшееся количество записей: 6823
```

Устранение ошибок

Для того, чтобы найти самые явные ошибки, рассмотрим для некоторых строковых параметров самые редко встречающиеся значения. Так мы сможем обнаружить возможные опечатки в данных.

In [18]:
```
columns = ['vehicle_no', 'Origin_Location', 'Destination_Location', 'Org_lat_lon', 'Des_lat_lon', 'Mater
for column in columns:
    freq_count = data[column].value_counts().sort_values(ascending=True)
    print(freq_count[freq_count <= 2], end='\n')
    print('\n---------------\n')
```

```
TN52D5668     1
TN29AL1774    1
TN31F8538     1
TN16C9838     1
TN52D2889     1
             ..
UP17AT8988    2
HR55AA6702    2
GJ36T6515     2
HR47D5476     2
HR47D4729     2
Name: vehicle_no, Length: 1702, dtype: int64

---------------

Sedarapet, Pondicherry, India                                1
Bebedhol, Pune, Maharashtra                                  1
Gurgaon, Gurgaon, Haryana                                    1
TVS LOGISTICS SERVICES LIMITED,LUCKNOW,UTTAR PRADESH         1
Sakchi Court, East Singhbhum, Jharkhand                      1
Guruvoyal, Tiruvallur, Tamil Nadu                            1
Burnpur Mkt, Bardhaman, West Bengal                          1
Pondicherry, Puducherry, India                               1
Ongole Bazar, Prakasam, Andhra Pradesh                       1
Craw Ford Colony, Tiruchirappalli, Tamil Nadu                1
Alampatti, Tuticorin, Tamil Nadu                             1
Madhapur, Hyderabad, Telangana                               1
Chaupati, Mumbai, Maharashtra                                1
Markapur, Prakasam, Andhra Pradesh                           1
Shitla, Bankura, West Bengal                                 1
Tatiparthy, Rangareddy, Telangana                            1
Kommadi, Visakhapatnam, Andhra Pradesh                       1
```

```
Kommadi, Visakhapatnam, Andhra Pradesh                          1
Seetharampet, Hyderabad, Telangana                              1
ASHOK LEYLAND PARTS-HOSUR,HOSUR,TAMIL NADU                      1
Goraj, Ahmedabad, Gujarat                                       1
Panvel City, Raigarh, Maharashtra                               1
TVS SUPPLY CHAIN SOLUTIONS LIMITED,Coimbatore,TAMIL NADU        1
Doma, Patna, Bihar                                              1
Nanajipur, Rangareddy, Telangana                                1
MIDC Nagpur, Nagpur, Maharashtra                                1
Barrackpore Govt.Housing, North 24 Parganas, West Bengal        1
Nariman Point, Mumbai, Maharashtra                              1
Jawahar Market Bhilai, Durg, Chattisgarh                        1
Indrapur, Kamrup, Assam                                         1
Powai Iit, Mumbai, Maharashtra                                  1
Angadpur, Bardhaman, West Bengal                                1
Gholkunda, Bankura, West Bengal                                 1
Rangapani, Darjiling, West Bengal                               1
Rambilli, Visakhapatnam, Andhra Pradesh                         1
Mekhali, Pune, Maharashtra                                      1
Pardih, East Singhbhum, Jharkhand                               1
Sarar, Vadodara, Gujarat                                        1
Bidhan Nagar CK Market, North 24 Parganas, West Bengal          1
Chimbipada, Thane, Maharashtra                                  1
HERO MOTOCORP LTD,ALWAR,RAJASTHAN                               1
Dhepargaon, Kamrup, Assam                                       1
Bharat Nagar, Mumbai, Maharashtra                               2
Visakhapatnam Port, Visakhapatnam, Andhra Pradesh               2
Madras Electricity System, Chennai, Tamil Nadu                  2
Singanapudi, Krishna, Andhra Pradesh                            2
Devalapura, Mysore, Karnataka                                   2
Jonnalagadda, Krishna, Andhra Pradesh                           2
Jainpur I A, Kanpur Dehat, Uttar Pradesh                        2
Gobindpur Housing Colony, East Singhbhum, Jharkhand             2
Karnataka 562114, India                                         2
Mhc Manimajra, Chandigarh, Chandigarh                           2
Baghthala, Rewari, Haryana                                      2
Golmuri, East Singhbhum, Jharkhand                              2
Mulund West, Mumbai, Maharashtra                                2
Russel Street, Kolkata, West Bengal                             2
Nasratpura ED, Ghaziabad, Uttar Pradesh                         2
Dhirenpara, Kamrup, Assam                                       2
Sunguvarchatram, Kanchipuram, Tamil Nadu                        2
Name: Origin_Location, dtype: int64

---------------

Nalanda, Bihar, India                           1
Gohe BK, Pune, Maharashtra                      1
Hewai, Hazaribagh, Jharkhand                    1
Geeta Peeth, Kangra, Himachal Pradesh           1
Kewda Khurd, Udaipur, Rajasthan                 1
                                               ..
Kudasan, Gandhinagar, Gujarat                   2
Supaul, Bihar, India                            2
Matar, Sirmaur, Himachal Pradesh                2
Bairia, Patna, Bihar                            2
Tiruchirappalli, Tiruchirappalli, Tamil Nadu    2
Name: Destination_Location, Length: 242, dtype: int64

---------------

12.0001,79.7483995                      1
13.202214,80.131693                     1
22.232742,84.885455                     1
18.685788,73.665394                     1
16.522635118740617,80.691347925000017   1
9.159925,77.836013                      1
15.740921,79.271895                     1
10.716218,78.66824                      1
28.4240,76.9990                         1
12.5000,79.5600                         1
17.829632,83.389622                     1
28.430086,77.017841                     1
17.389673,78.478036                     1
16.996676,78.679584                     1
26.8500,80.9200                         1
12.837284,79.7041744000001              1
11.9415915,79.8083133                   1
```

```
11.9415915,79.8083153                    1
17.441148,78.391069                      1
17.444099,78.252598                      1
19.1253,72.907667                        1
17.464871,82.929988                      1
23.081873,87.03741                       1
22.250953,84.850277                      1
19.000914,73.20576                       1
28.3540,76.9390                          1
17.248088,78.395599                      1
21.109305,78.991035                      1
18.6300,73.7540                          1
11.0700,77.0970                          1
22.76074,88.387051                       1
22.969075,72.325933                      1
19.377771,73.006582                      1
22.585401,88.428272                      1
22.105618,73.152148                      1
22.660545,86.386965                      1
25.609688,85.118112                      1
18.068686,74.609489                      1
26.586896,88.353075                      1
21.178079,81.38943                       1
18.924943,72.822246                      1
19.178201,72.943314                      2
23.508352,87.326459                      2
28.3180,76.8930                          2
13.06919,80.266978                       2
28.644514,77.425732                      2
30.000345,76.732209                      2
12.7400,77.8200                          2
28.1290,76.8180                          2
23.525309955369213,87.264364980245       2
18.963397,72.820749                      2
22.749591,86.281875                      2
28.3730,76.8970                          2
22.601673,88.137679                      2
12.924586,79.878994                      2
12.223062,76.690357                      2
16.80061,80.299025                       2
Name: Org_lat_lon, dtype: int64

---------------

25.1240603,85.45947489999999             1
12.930964,77.604922                      1
28.528459,77.214648                      1
22.102931,82.075027                      1
12.141186,78.13934                       1
                                        ..
23.4760,71.9740                          2
26.878998,75.801096                      2
28.685897,77.211773                      2
24.391350439053529,86.570766574899267    2
12.938565,80.237708                      2
Name: Des_lat_lon, Length: 244, dtype: int64

---------------

CONTROL LEVER ASSY                   1
STICKER / LABEL/SIDE DOOR 6X4        1
INTERMEDIATE FLANGE / FRONT PLATE    1
ZB BRACKET ROOF FRT CTR              1
STICKER / WRAPPER                    1
                                   ..
PLAIN NUT,M8X1.25                    2
LAMP REAR LH                         2
SPRING WASHER,M6                     2
X0483110                             2
SHIELD OIL PAN                       2
Name: Material Shipped, Length: 984, dtype: int64

---------------
```

Видим, что явных ошибок в значениях нет.

Агрегирование данных

Посмотрим на итоговый вид набора данных после всех сделанных преобразований.

In [19]:
```
pd.set_option('display.max_columns', 22)
data.head()
```

Out[19]:

	BookingID	Market/Regular	BookingID_Date	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon
0	MVCV0000927/082021	Market	2020-08-17	KA590408	TVSLSL-PUZHAL-HUB,CHENNAI,TAMIL NADU	ASHOK LEYLAND PLANT 1-HOSUR,HOSUR,KARNATAKA	13.1550,80.1960
1	VCV00014271/082021	Regular	2020-08-27	TN30BC5917	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
2	VCV00014382/082021	Regular	2020-08-27	TN22AR2748	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8710,79.7390
3	VCV00014743/082021	Regular	2020-08-28	TN28AQ0781	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	12.8390,79.9540
4	VCV00014744/082021	Regular	2020-08-28	TN68F1722	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY	11.8720,79.6320

Воспользуемся методом `describe` для получения основных численных характеристик по каждому из признаков. Выведем показатели отдельно для числовых и отдельно для строковых признаков.

In [20]:
```
data.describe()
```

Out[20]:

	TRANSPORTATION_DISTANCE_IN_KM
count	6823.000000
mean	502.005943
std	738.371434
min	0.000000
25%	30.000000
50%	109.000000
75%	667.000000
max	2954.700000

In [21]:
```
data.describe(include=['object'])
```

Out[21]:

	BookingID	Market/Regular	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon	De
count	6823	6823	6823	6823	6823	6823	
unique	6823	2	2325	180	520	173	
top	MVCV0000927/082021	Regular	TS15UC9341	Mugabala, Bangalore Rural, Karnataka	DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,...	16.560192249175344,80.792293091599547	12.839
freq	1	6757	36	565	339	1171	

Полученные характеристики убеждают нас в отсутствии явных выбросов и ошибок в данных (так как нет, например, отрицательных значений количества км пути).

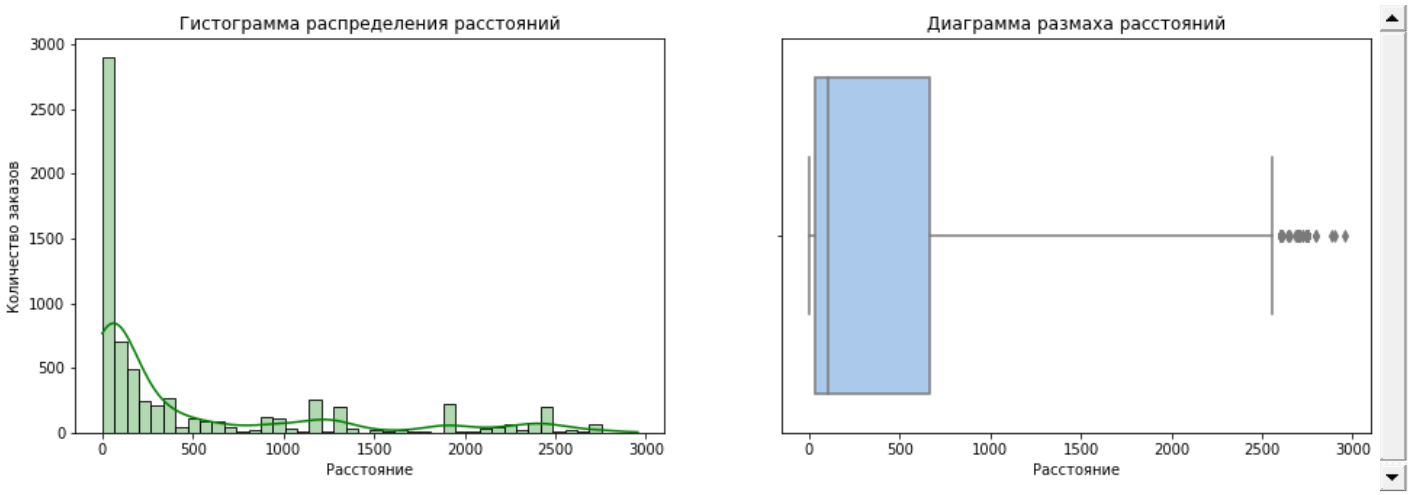
Несколько интересных выводов, которые уже можно сделать на основании агрегированных данных:

- основная часть грузоперевозок осуществляется на расстояние от 30 до 700 км;
- медианное расстояние грузоперевозки составляет около 500 км, самая протяженная грузоперевозка осущевлялась на расстояние примерно в 6 раз больше;
- в подавялющем большинстве грузоперевозки осуществляются для заказчиков, с которыми заключен контракт;
- самая популярная марка грузовика - 40 FT 3XL Trailer 35MT.

Визуальное исследование датасета, корреляция признаков

Расстояние

In [22]:
```
fig = plt.figure(figsize=(16, 5))
axes = fig.subplots(1 ,2)
sns.histplot(data['TRANSPORTATION_DISTANCE_IN_KM'], kde=True, color='green', alpha=0.3, ax=axes[0])
axes[0].title.set_text(f"Гистограмма распределения расстояний")
axes[0].set_xlabel('Расстояние')
axes[0].set_ylabel('Количество заказов')
axes[1].title.set_text('Диаграмма размаха расстояний')
sns.boxplot(x=data['TRANSPORTATION_DISTANCE_IN_KM'], ax=axes[1], whis=3, palette='pastel');
axes[1].set_xlabel('Расстояние')
plt.show();
```

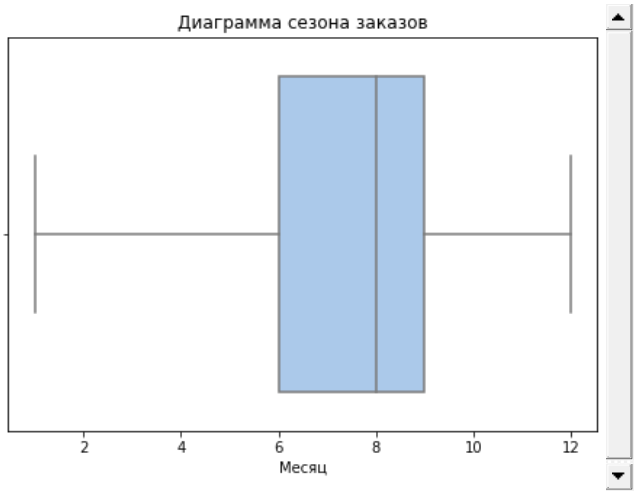


Видим, что наблюдается обратная зависимость количества заказов и рассстояния. Однако данные распределены неравномерно, так что нельзя говорить о строгой зависимости этих параметров.

Сезонность заказов

In [23]:
```
months = data['BookingID_Date'].map(lambda x:x.month)

fig = plt.figure(figsize=(16, 5))
axes = fig.subplots(1 ,2)
sns.histplot(months, kde=True, color='green', alpha=0.3, ax=axes[0])
axes[0].title.set_text(f"Гистограмма распределения заказов по месяцам")
axes[0].set_xlabel('Месяц')
axes[0].set_ylabel('Количество заказов')
axes[1].title.set_text('Диаграмма сезона заказов')
sns.boxplot(x=months, ax=axes[1], whis=3, palette='pastel');
axes[1].set_xlabel('Месяц')
plt.show();
```



Видим, что самые высокие показатели количества заказов наблюдаются все летние месяцы, всплеск заказов прихдится на август.

Марки грузовиков

In [24]:
```
target_name = 'TRANSPORTATION_DISTANCE_IN_KM'
feature_names = data.columns[data.columns != target_name]
X = data[feature_names]
y = data[target_name]
```

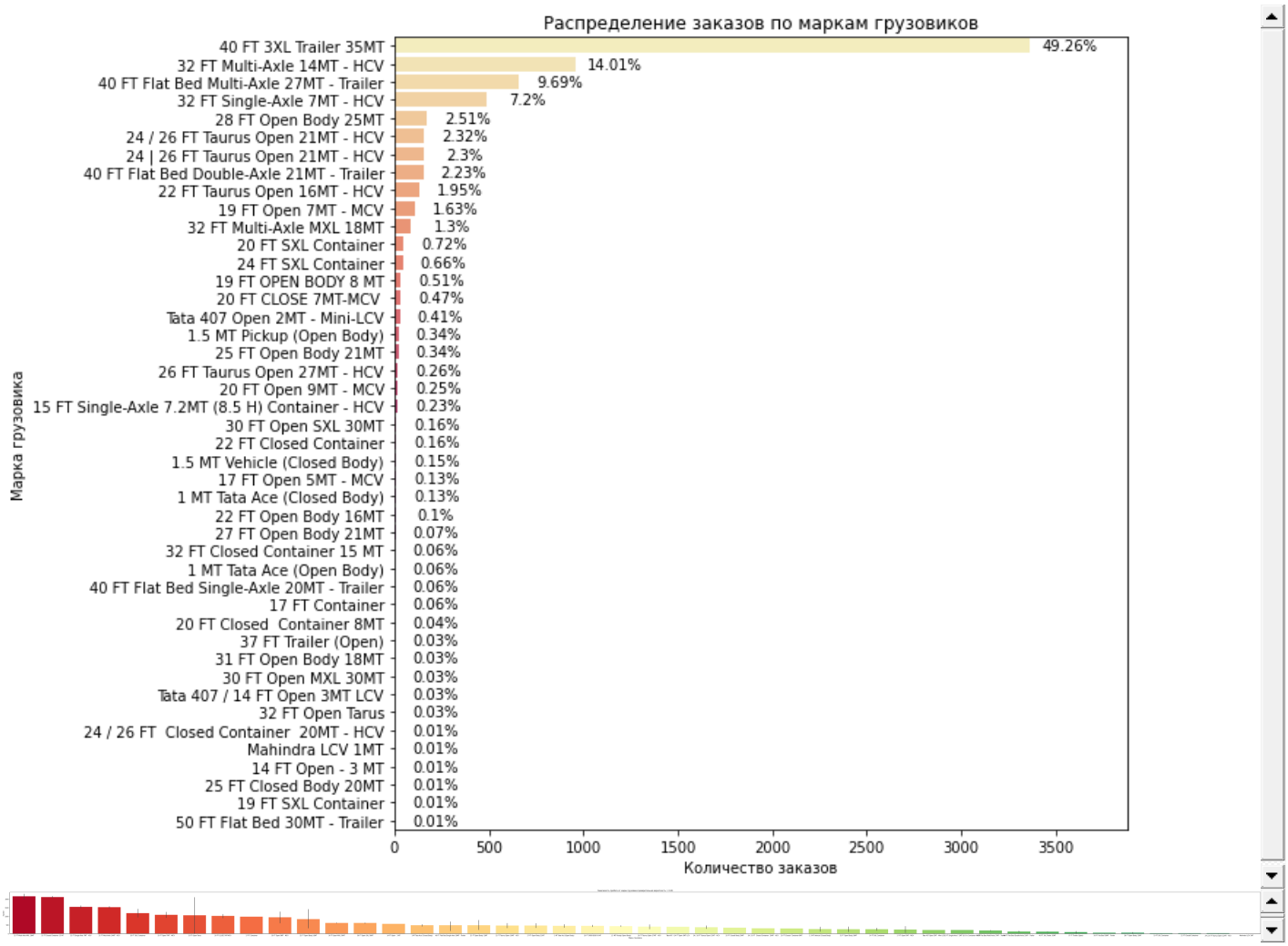
In [25]:
```
def show_marks(ax, percent=False, vert=False):
    if vert:
        ax.set_xlim(0, ax.get_xlim()[1] * 1.1)
    else:
        ax.set_ylim(0, ax.get_ylim()[1] * 1.1)

    for i, bar in enumerate(ax.patches):
        if vert:
            h = bar.get_width()
            ax.text(h+ax.get_xlim()[1]*0.055, i, f'{round(h * (100 / X.shape[0] if percent else 1), 2)}'
                    ha='center', va='center')
        else:
            h = bar.get_height()
            ax.text(i, h+ax.get_ylim()[1]*0.04, f'{round(h * (100 / X.shape[0] if percent else 1), 2)}'
                    ha='center', va='center')

def my_countplot(feature, figsize, title, xlabel, ylabel, vert=False, sort=False):
    fig = plt.figure(figsize=figsize)
    order = (X[feature].value_counts().index if sort else None)
    plot = sns.countplot(y=X[feature] if vert else None, x=None if vert else X[feature], order=order, pa
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    show_marks(plot.axes, True, vert)
    plt.show();

def my_barplot(feature_name, x_label, title, figsiz, hue_feature=None, legend_title=None):
    plt.figure(figsize=figsiz)
    if hue_feature:
        my_plot = sns.barplot(x=X[feature_name], y=y, saturation=1, hue=X[hue_feature])
        my_plot.legend(title=legend_title);
    else:
        order = data.groupby(feature_name)[target_name].mean().sort_values(ascending=False).index
        sns.barplot(x=X[feature_name], y=y, order=order, palette='RdYlGn', saturation=1)
    plt.title(f'Зависимость пробега от {title} (доверительная вероятность = 0.95)')
    plt.ylabel('Пробег');
    plt.xlabel(x_label)
```

In [26]:
```
my_countplot('vehicleType', (9, 10), 'Распределение заказов по маркам грузовиков',
                                    'Количество заказов', 'Марка грузовика', vert=True, sort=True)
my_barplot('vehicleType', 'Марка грузовика', 'марки грузовика', (150, 5), )
```



Видим, что у грузовика с самым большим числом заказов - не самый большой пробег.

Выводы

1. Количество заказов имеет отрицательную зависимость от расстояния доставки. Однако не в каждом случае возможно применить эту закономерность.

2. Количество заказов зависит от сезона. Анализ выявил, что самые высокие показатели количества заказов наблюдаются все летние месяцы, всплеск заказов прихдится на август.

3. Количество заказов, выполненных на определенном транспортном средстве, никак не зависит от его пробега. У транспортного средства с самым большим количеством заказов не обязательно самый большой пробег.