Pizza delivery timing

In this task we are going to perform simple EDA operations to analyse the pizza delivery timings.

In [46]:
```python
#import packages

import numpy as np
import pandas as pd
import datetime as dt
from datetime import date
import calendar
import matplotlib.pyplot as plt
import seaborn as sb
```

In [3]:
```python
#getting the dataset
pdt=pd.read_csv("C:\\Users\\sujoydutta\\Desktop\\Data analysis\\Datasets for ML\\Additional Datasets\\diminos_d
pdt.head()
```

Out[3]:

	order_id	order_placed_at	order_delivered_at
0	1523111	2023-03-01 00:00:59	2023-03-01 00:18:07.443132
1	1523112	2023-03-01 00:03:59	2023-03-01 00:19:34.925241
2	1523113	2023-03-01 00:07:22	2023-03-01 00:22:28.291385
3	1523114	2023-03-01 00:07:47	2023-03-01 00:46:19.019399
4	1523115	2023-03-01 00:09:03	2023-03-01 00:25:13.619056

In [4]:
```python
#getting info
pdt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   order_id            15000 non-null  int64 
 1   order_placed_at     15000 non-null  object
 2   order_delivered_at  15000 non-null  object
dtypes: int64(1), object(2)
memory usage: 351.7+ KB
```

In [5]:
```python
# column name cleaning

pdt.columns = pdt.columns.str.replace(' ', '')
pdt.columns
```

Out[5]:
```
Index(['order_id', 'order_placed_at', 'order_delivered_at'], dtype='object')
```

In [6]:
```python
#splitting column order_placed_at
pdt[['Order day', 'Order_place_time']] = pdt.order_placed_at.str.split(" ", expand = True)
pdt
```

Out[6]:

	order_id	order_placed_at	order_delivered_at	Order day	Order_place_time
0	1523111	2023-03-01 00:00:59	2023-03-01 00:18:07.443132	2023-03-01	00:00:59
1	1523112	2023-03-01 00:03:59	2023-03-01 00:19:34.925241	2023-03-01	00:03:59
2	1523113	2023-03-01 00:07:22	2023-03-01 00:22:28.291385	2023-03-01	00:07:22
3	1523114	2023-03-01 00:07:47	2023-03-01 00:46:19.019399	2023-03-01	00:07:47
4	1523115	2023-03-01 00:09:03	2023-03-01 00:25:13.619056	2023-03-01	00:09:03
...	...	...	...	...	...
14995	1538106	2023-03-27 23:37:05	2023-03-27 23:52:37.409378	2023-03-27	23:37:05
14996	1538107	2023-03-27 23:47:38	2023-03-28 00:04:22.672912	2023-03-27	23:47:38
14997	1538108	2023-03-27 23:50:16	2023-03-28 00:05:40.676238	2023-03-27	23:50:16
14998	1538109	2023-03-27 23:52:44	2023-03-28 00:08:41.810358	2023-03-27	23:52:44
14999	1538110	2023-03-27 23:58:20	2023-03-28 00:13:42.499311	2023-03-27	23:58:20

15000 rows × 5 columns

In [13]:
```python
#splitting column order_delivered_at
pdt[['Order day', 'Order_deliver_time']] = pdt.order_delivered_at.str.split(" ", expand = True)
pdt
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In [13], line 2
      1 #splitting column order_delivered_at
----> 2 pdt[['Order day', 'Order_deliver_time']] = pdt.order_delivered_at.str.split(" ", expand = True)
      3 pdt

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:5583, in NDFrame.__getattr__(self, name)
   5576 if (
   5577     name not in self._internal_names_set
   5578     and name not in self._metadata
   5579     and name not in self._accessors
   5580     and self._info_axis._can_hold_identifiers_and_holds_name(name)
   5581 ):
   5582     return self[name]
-> 5583 return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'order_delivered_at'
```

In [8]:
```python
#dropping unnecessary columns
pdt=pdt.drop(['order_placed_at','order_delivered_at'],axis=1)
pdt.head(10)
```

Out[8]:

	order_id	Order day	Order_place_time	Order_deliver_time
0	1523111	2023-03-01	00:00:59	00:18:07.443132
1	1523112	2023-03-01	00:03:59	00:19:34.925241
2	1523113	2023-03-01	00:07:22	00:22:28.291385
3	1523114	2023-03-01	00:07:47	00:46:19.019399
4	1523115	2023-03-01	00:09:03	00:25:13.619056
5	1523116	2023-03-01	00:09:44	00:25:17.088349
6	1523117	2023-03-01	00:11:09	00:41:54.245295
7	1523118	2023-03-01	00:13:25	00:30:38.556491
8	1523119	2023-03-01	00:15:20	00:30:28.234367
9	1523120	2023-03-01	00:15:34	00:31:53.026195

In [9]:
```python
#changing column type
pdt["Order day"]= pd.to_datetime(pdt["Order day"])
pdt["Order_deliver_time"]= pd.to_datetime(pdt["Order_deliver_time"])
pdt["Order_place_time"]= pd.to_datetime(pdt["Order_place_time"])
pdt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   order_id            15000 non-null  int64         
 1   Order day           15000 non-null  datetime64[ns]
 2   Order_place_time    15000 non-null  datetime64[ns]
 3   Order_deliver_time  15000 non-null  datetime64[ns]
dtypes: datetime64[ns](3), int64(1)
memory usage: 468.9 KB
```

In [10]:
```python
#feature engineering by creating a new feature
pdt["deli_time"]=pdt["Order_deliver_time"]-pdt["Order_place_time"]

pdt["deli_time"]=pdt["deli_time"].dt.seconds
```

Out[10]:

	order_id	Order day	Order_place_time	Order_deliver_time	deli_time
0	1523111	2023-03-01	2023-03-30 00:00:59	2023-03-30 00:18:07.443132	1028
1	1523112	2023-03-01	2023-03-30 00:03:59	2023-03-30 00:19:34.925241	935
2	1523113	2023-03-01	2023-03-30 00:07:22	2023-03-30 00:22:28.291385	906
3	1523114	2023-03-01	2023-03-30 00:07:47	2023-03-30 00:46:19.019399	2312
4	1523115	2023-03-01	2023-03-30 00:09:03	2023-03-30 00:25:13.619056	970

In [11]:
```python
#Converting to get minutes
pdt["deli_time"]=pdt["deli_time"]// 60
pdt["deli_time"]
```

Out[11]:
```
0        17
1        15
2        15
3        38
4        16
         ..
14995    15
14996    16
14997    15
14998    15
14999    15
Name: deli_time, Length: 15000, dtype: int64
```

In [12]:
```python
pdt.head(10)
```

Out[12]:

	order_id	Order day	Order_place_time	Order_deliver_time	deli_time
0	1523111	2023-03-01	2023-03-30 00:00:59	2023-03-30 00:18:07.443132	17
1	1523112	2023-03-01	2023-03-30 00:03:59	2023-03-30 00:19:34.925241	15
2	1523113	2023-03-01	2023-03-30 00:07:22	2023-03-30 00:22:28.291385	15
3	1523114	2023-03-01	2023-03-30 00:07:47	2023-03-30 00:46:19.019399	38
4	1523115	2023-03-01	2023-03-30 00:09:03	2023-03-30 00:25:13.619056	16
5	1523116	2023-03-01	2023-03-30 00:09:44	2023-03-30 00:25:17.088349	15
6	1523117	2023-03-01	2023-03-30 00:11:09	2023-03-30 00:41:54.245295	30
7	1523118	2023-03-01	2023-03-30 00:13:25	2023-03-30 00:30:38.556491	17
8	1523119	2023-03-01	2023-03-30 00:15:20	2023-03-30 00:30:28.234367	15
9	1523120	2023-03-01	2023-03-30 00:15:34	2023-03-30 00:31:53.026195	16

In [16]:
```python
#dropping unnecessary columns
pdt1=pdt.drop(['Order_place_time','Order_deliver_time'],axis=1)
pdt1.head()
```

Out[16]:

	order_id	Order day	deli_time
0	1523111	2023-03-01	17
1	1523112	2023-03-01	15
2	1523113	2023-03-01	15
3	1523114	2023-03-01	38
4	1523115	2023-03-01	16

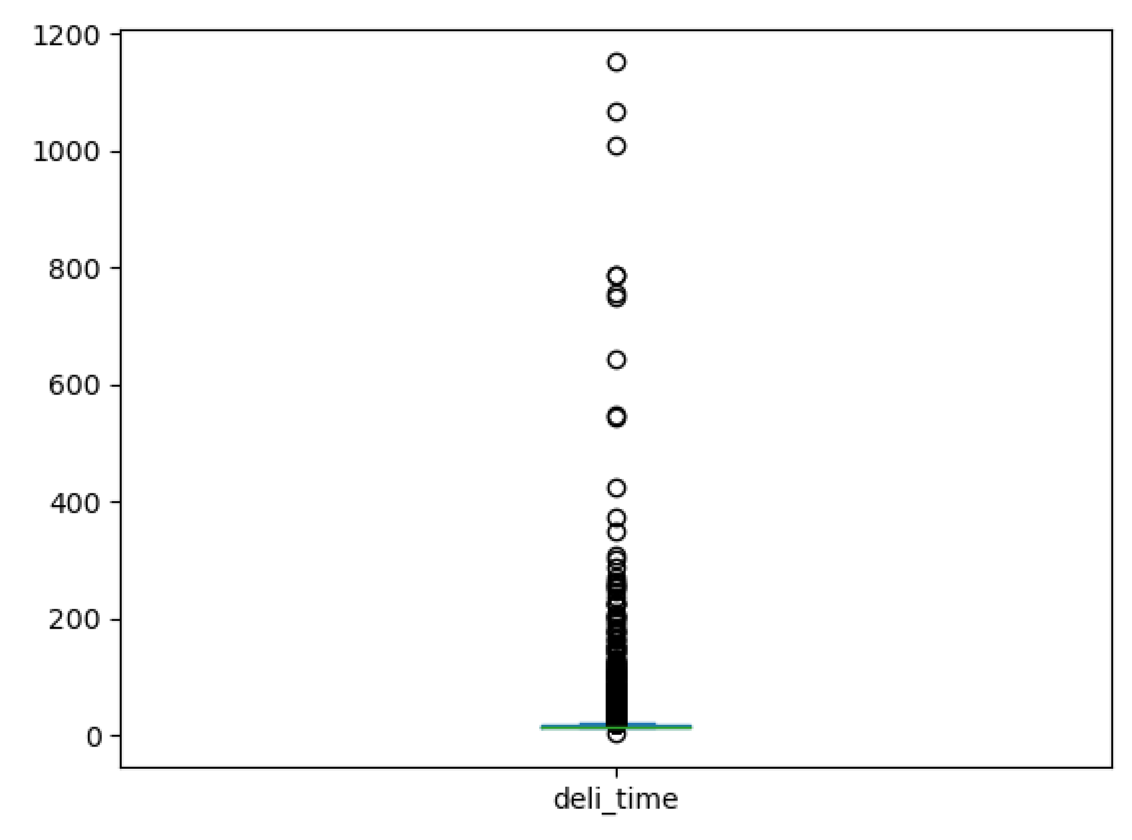
In [18]:
```python
#seeing the distribution
pdt1['deli_time'].describe()
```

Out[18]:
```
count    15000.000000
mean        18.552667
std         25.015547
min          3.000000
25%         15.000000
50%         15.000000
75%         17.000000
max       1152.000000
Name: deli_time, dtype: float64
```

In [19]:
```python
# Seeing extreme values


pdt1['deli_time'].plot(kind='box')
```

Out[19]:
```
<AxesSubplot: >
```



In [20]:
```python
#using percentile method

pdt1['deli_time'].quantile([0.1, 0.25, 0.5, 0.70, 0.9, 0.95, 0.99])
```

Out[20]:
```
0.10    15.00
0.25    15.00
0.50    15.00
0.70    16.00
0.90    21.00
0.95    27.00
0.99    64.01
Name: deli_time, dtype: float64
```

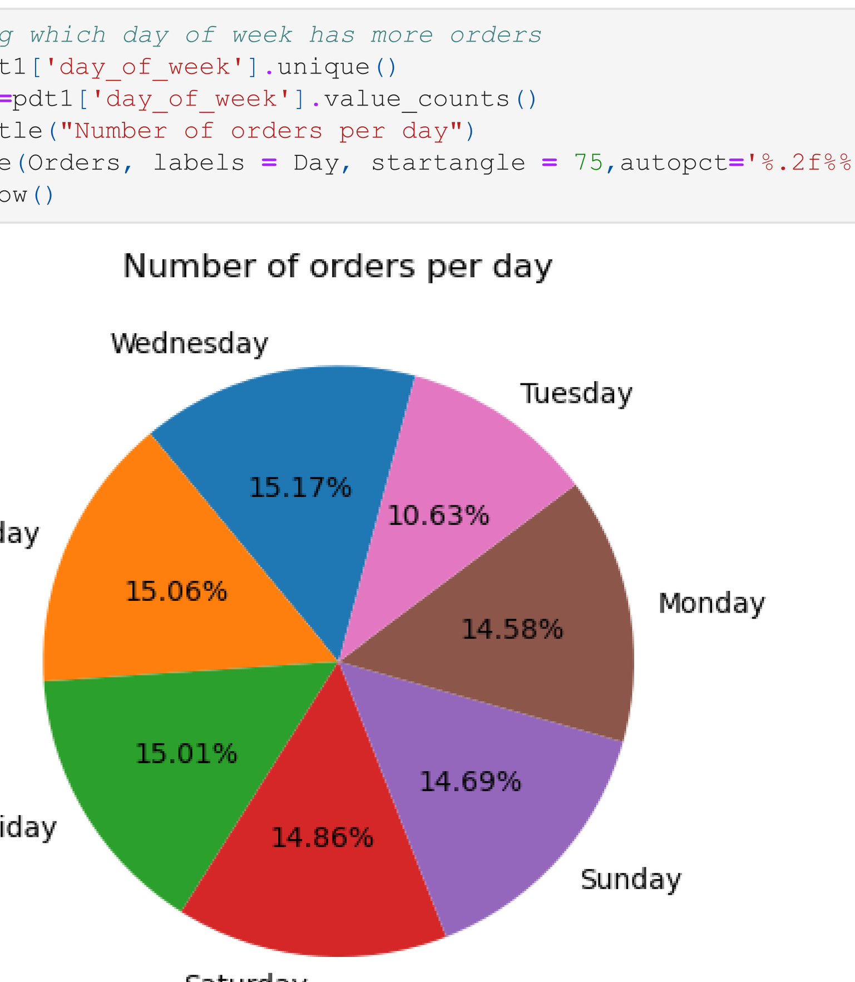
In [36]:
```python
#getting day of week
pdt1['day_of_week'] = pdt1['Order day'].dt.day_name()
pdt1['day_of_week']
```

Out[36]:
```
0        Wednesday
1        Wednesday
2        Wednesday
3        Wednesday
4        Wednesday
           ...
14995       Monday
14996      Tuesday
14997      Tuesday
14998      Tuesday
14999      Tuesday
Name: day_of_week, Length: 15000, dtype: object
```

In [37]:
```python
#seeing data types
pdt1.info()
```

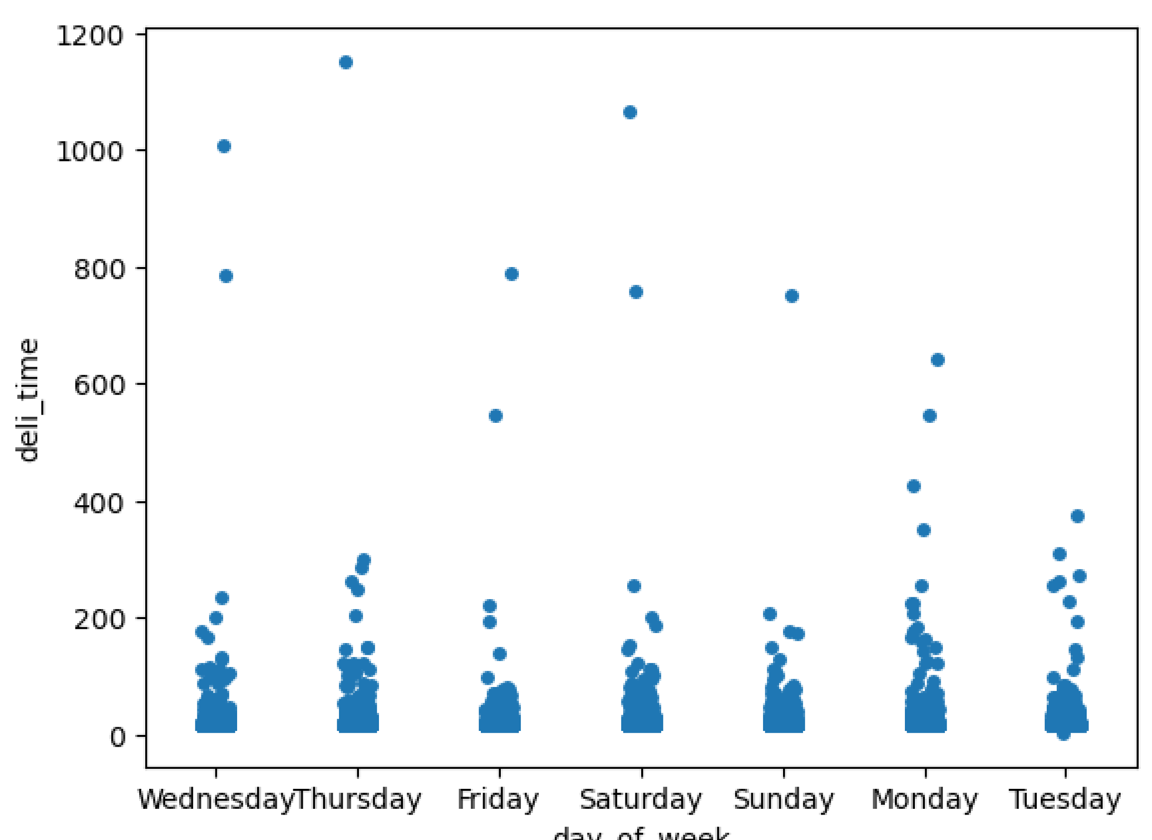
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype         
---  ------       --------------  -----         
 0   order_id     15000 non-null  int64         
 1   Order day    15000 non-null  datetime64[ns]
 2   deli_time    15000 non-null  int64         
 3   day_of_week  15000 non-null  object        
dtypes: datetime64[ns](1), int64(2), object(1)
memory usage: 468.9+ KB
```

In [43]:
```python
#seeing which day of week has more orders
Day=pdt1['day_of_week'].unique()
Orders=pdt1['day_of_week'].value_counts()
plt.title("Number of orders per day")
plt.pie(Orders, labels = Day, startangle = 75,autopct='%.2f%%')
plt.show()
```



In [48]:
```python
#seeing delivery time per day of the week
sb.stripplot(x="day_of_week", y="deli_time", data=pdt1)
```

Out[48]:
```
<AxesSubplot: xlabel='day_of_week', ylabel='deli_time'>
```



Insights Gathered

The following points have been noticed:

1.Delivery is the slowest on Monday.

2.The least orders are on Tuesdays.

3.The average delivery time is around 18 minutes.

4.The delivery time is 27 minutes in 95th percentile.

5.Wedenesday,Thursday and Friday are comparatively busier than other days.