Liquiloans Junior_Analyst Assignment By HARIOM RAI

In [1]:

```
    import pandas as pd
    import numpy as np
    import seaborn as sns
    import matplotlib.pyplot as plt
```

In [2]:

```
dataset = pd.read_csv(r"C:\Users\Dell\Desktop\liquiloans\GSPE_DE_dataset\GSPE_DE_dataset.csv")
dataset
```

Out[2]:

	asst_id	product_type	manufacture_date	contract_start	contract_end	contact_date	contact_type	issue_type	topic_category	parts
0	0	Laptops	6/26/2017	6/26/2017	6/21/2021	10/1/2018	Voice	NaN	NaN	Hard Documenta
1	1	Laptops	12/12/2016	12/12/2016	12/2/2019	10/1/2018	Voice	NaN	NaN	Speaker, C
2	2	Laptops	5/21/2018	5/21/2018	5/30/2022	10/1/2018	Voice	NaN	NaN	(
3	3	Laptops	2/22/2016	2/22/2016	2/18/2019	10/1/2018	VOICE	Hard Drive	Booting	Harc
4	4	Laptops	8/7/2017	8/7/2017	8/3/2020	10/1/2018	Voice	NaN	NaN	E (Note
...	...	...	...	...	...	...	...	...	...	
99995	16660	Laptops	7/25/2016	7/25/2016	7/22/2019	10/1/2018	VOICE	Information Status	Booting	
99996	16661	Laptops	5/1/2017	5/1/2017	5/31/2021	10/1/2018	Voice	NaN	NaN	LCD Cha
99997	16662	Desktops	2/15/2016	2/15/2016	2/15/2021	10/1/2018	VOICE	Information Status	Power	
99998	16663	Laptops	12/21/2015	12/21/2015	12/25/2023	10/1/2018	CHAT	Audio, Video, Speakers, TV Tuner	LCD/Monitor	LCD (
99999	16664	Desktops	5/8/2017	5/8/2017	4/27/2020	10/1/2018	Voice	NaN	NaN	Mother LCD

100000 rows × 19 columns

Makeing a copy of dataset

In [3]:

```
dataset_copy = dataset.copy()
```

In [4]:

```
dataset_copy
```

Out[4]:

	asst_id	product_type	manufacture_date	contract_start	contract_end	contact_date	contact_type	issue_type	topic_category	parts
0	0	Laptops	6/26/2017	6/26/2017	6/21/2021	10/1/2018	Voice	NaN	NaN	Hard Documenta
1	1	Laptops	12/12/2016	12/12/2016	12/2/2019	10/1/2018	Voice	NaN	NaN	Speaker, C
2	2	Laptops	5/21/2018	5/21/2018	5/30/2022	10/1/2018	Voice	NaN	NaN	(
3	3	Laptops	2/22/2016	2/22/2016	2/18/2019	10/1/2018	VOICE	Hard Drive	Booting	Harc
4	4	Laptops	8/7/2017	8/7/2017	8/3/2020	10/1/2018	Voice	NaN	NaN	E (Note
...	...	...	...	...	...	...	...	...	...	

99995	16660	Laptops	7/25/2016	7/25/2016	7/22/2019	10/1/2018	VOICE	Information Status	Booting	
99996	16661	Laptops	5/1/2017	5/1/2017	5/31/2021	10/1/2018	Voice	NaN	NaN	LCD Cha
99997	16662	Desktops	2/15/2016	2/15/2016	2/15/2021	10/1/2018	VOICE	Information Status	Power	
99998	16663	Laptops	12/21/2015	12/21/2015	12/25/2023	10/1/2018	CHAT	Audio, Video, Speakers, TV Tuner	LCD/Monitor	LCD (
99999	16664	Desktops	5/8/2017	5/8/2017	4/27/2020	10/1/2018	Voice	NaN	NaN	Mother LCD

100000 rows × 19 columns

Droping duplicate values

In [5]:

```
dataset.duplicated().value_counts()
```

Out[5]:

```
False    86536
True     13464
dtype: int64
```

In [6]:

```
dataset.drop_duplicates(inplace = True)
```

In [7]:

```
dataset.shape
```

Out[7]:

```
(86536, 19)
```

Counting null values and droping columnsin which 30% or more null values are present

In [8]:

```
dataset.isnull().sum()
```

Out[8]:

```
asst_id                   0
product_type             20
manufacture_date          0
contract_start            0
contract_end              0
contact_date              0
contact_type             10
issue_type            12162
topic_category        28621
parts_sent            39339
repair_type               0
repeat_ct                 0
parts_ct                  0
agent_tenure_indays      10
contact_manager_flg       0
diagnostics               0
repeat_parts_sent     74973
region                    3
country               10002
dtype: int64
```

In [9]:

```
null_var = dataset.isnull().sum()*100/dataset.shape[0]
null_var
```

Out[9]:

```
asst_id                0.000000
product_type           0.023112
manufacture_date       0.000000
```

```
contract_start          0.000000
contract_end            0.000000
contact_date            0.000000
contact_type            0.011556
issue_type             14.054266
topic_category         33.074096
parts_sent             45.459693
repair_type             0.000000
repeat_ct               0.000000
parts_ct                0.000000
agent_tenure_indays     0.011556
contact_manager_flg     0.000000
diagnostics             0.000000
repeat_parts_sent      86.637931
region                  0.003467
country                11.558195
dtype: float64
```

In [10]:

```
drop_columns = null_var[null_var > 30].keys()
drop_columns
```

Out[10]:

```
Index(['topic_category', 'parts_sent', 'repeat_parts_sent'], dtype='object')
```

In [11]:

```
dataset.drop(columns = drop_columns, inplace = True)
```

In [12]:

```
dataset
```

Out[12]:

	asst_id	product_type	manufacture_date	contract_start	contract_end	contact_date	contact_type	issue_type	repair_type	repeat_ct	pa
0	0	Laptops	6/26/2017	6/26/2017	6/21/2021	10/1/2018	Voice	NaN	Hard	0	
1	1	Laptops	12/12/2016	12/12/2016	12/2/2019	10/1/2018	Voice	NaN	Hard	0	
2	2	Laptops	5/21/2018	5/21/2018	5/30/2022	10/1/2018	Voice	NaN	Hard	0	
3	3	Laptops	2/22/2016	2/22/2016	2/18/2019	10/1/2018	VOICE	Hard Drive	Hard	0	
4	4	Laptops	8/7/2017	8/7/2017	8/3/2020	10/1/2018	Voice	NaN	Hard	0	
...	...	...	...	...	...	...	...	...	...	...	
99972	16637	Laptops	6/5/2017	6/5/2017	6/1/2020	10/1/2018	VOICE	Hard Drive	Hard	0	
99980	16645	Laptops	4/11/2016	4/11/2016	4/8/2019	10/1/2018	CHAT	Mechanical Chassis / Rack	Hard	0	
99982	16647	Laptops	1/1/2018	1/1/2018	1/4/2021	10/1/2018	VOICE	System Board Components	Hard	0	
99990	16655	Laptops	1/18/2016	12/21/2015	12/24/2018	10/1/2018	VOICE	Power Subsystem / Cables / AC Adapter	Hard	0	
99992	16657	Laptops	9/5/2016	9/5/2016	9/6/2021	10/1/2018	Voice	NaN	Hard	0	

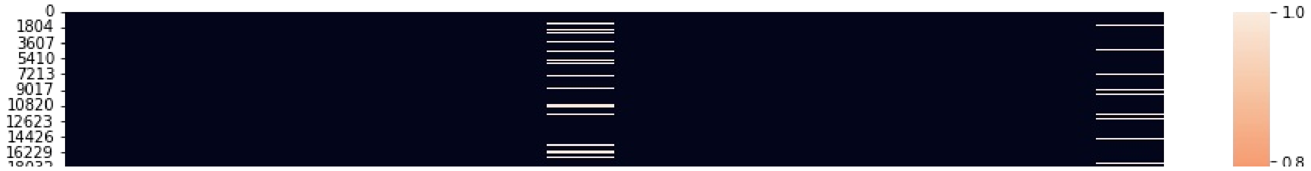
86536 rows × 16 columns

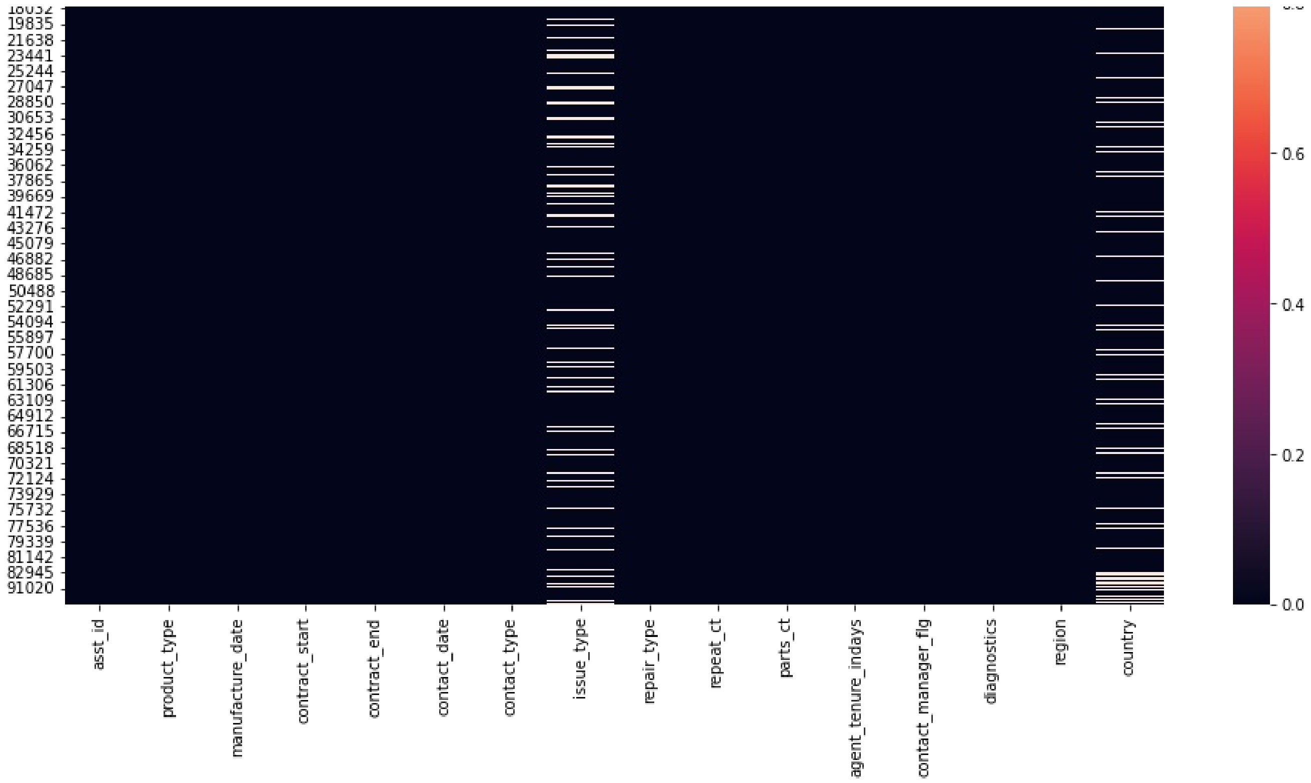
In [13]:

```
plt.figure(figsize = (16,9))
sns.heatmap(dataset.isnull())
```

Out[13]:

```
<AxesSubplot:>
```





Filling Null values of product_type column

In [14]:

```
dataset.isnull().sum()
```

Out[14]:

```
asst_id                    0
product_type              20
manufacture_date           0
contract_start             0
contract_end               0
contact_date               0
contact_type              10
issue_type             12162
repair_type                0
repeat_ct                  0
parts_ct                   0
agent_tenure_indays       10
contact_manager_flg        0
diagnostics                0
region                     3
country                10002
dtype: int64
```

In [15]:

```
dataset.product_type.value_counts()
```

Out[15]:

```
Laptops              63153
Desktops             20461
Other Electronics     2902
Name: product_type, dtype: int64
```

In [16]:

```
dataset.product_type.fillna(dataset.product_type.mode()[0], inplace = True)
```

In [17]:

```
dataset.isnull().sum()
```

Out[17]:

```
asst_id                    0
product_type               0
manufacture_date           0
contract_start             0
contract_end               0
contact_date               0
```

```
contact_type               10
issue_type              12162
repair_type                 0
repeat_ct                   0
parts_ct                    0
agent_tenure_indays        10
contact_manager_flg         0
diagnostics                 0
region                      3
country                 10002
dtype: int64
```

Changing datatype of contact_type column

In [18]:

```
dataset.contact_type.value_counts()
```

Out[18]:

```
VOICE      60360
CHAT       12478
Voice      12465
EMAIL       1124
Unknown       99
Name: contact_type, dtype: int64
```

In [19]:

```
dataset["contact_type"] = dataset["contact_type"].str.title()
```

In [20]:

```
dataset['contact_type']
```

Out[20]:

```
0        Voice
1        Voice
2        Voice
3        Voice
4        Voice
         ...  
99972    Voice
99980     Chat
99982    Voice
99990    Voice
99992    Voice
Name: contact_type, Length: 86536, dtype: object
```

In [21]:

```
dataset.isnull().sum()
```

Out[21]:

```
asst_id                     0
product_type                0
manufacture_date            0
contract_start              0
contract_end                0
contact_date                0
contact_type               10
issue_type              12162
repair_type                 0
repeat_ct                   0
parts_ct                    0
agent_tenure_indays        10
contact_manager_flg         0
diagnostics                 0
region                      3
country                 10002
dtype: int64
```

Filling Null values of region column

In [22]:

```
dataset['region'].value_counts()
```

Out[22]:

```
Hogwarts        60538
```

```
Middle Earth    20904
Milky Way        5091
Name: region, dtype: int64
```

In [23]:
```
dataset['region'].fillna(dataset['region'].mode()[0], inplace=True)
```

In [24]:
```
dataset['region'].value_counts()
```

Out[24]:
```
Hogwarts        60541
Middle Earth    20904
Milky Way        5091
Name: region, dtype: int64
```

Filling Null values of agent_tenure_indays column

In [25]:
```
dataset.agent_tenure_indays.fillna(dataset.agent_tenure_indays.median(), inplace = True)
```

In [26]:
```
dataset.isnull().sum()
```

Out[26]:
```
asst_id                    0
product_type               0
manufacture_date           0
contract_start             0
contract_end               0
contact_date               0
contact_type              10
issue_type             12162
repair_type                0
repeat_ct                  0
parts_ct                   0
agent_tenure_indays        0
contact_manager_flg        0
diagnostics                0
region                     0
country                10002
dtype: int64
```

In [27]:
```
dataset.contact_type.value_counts()
```

Out[27]:
```
Voice      72825
Chat       12478
Email       1124
Unknown       99
Name: contact_type, dtype: int64
```

Filling Null values of contact_type column

In [28]:
```
dataset.contact_type.fillna(dataset.contact_type.mode()[0], inplace = True)
```

In [29]:
```
dataset.contact_type.value_counts()
```

Out[29]:
```
Voice      72835
Chat       12478
Email       1124
Unknown       99
Name: contact_type, dtype: int64
```

In [30]:
```
dataset.country.value_counts()
```

Out[30]:
```
Zonko's Joke Shop    53306
```

```
Out[30]: Zonko's Joke Shop
         Lorien                 15694
         Merope                  3829
         Mordor                  2597
         Pollux                   347
         The Shire                166
         Shrieking Shack           77
         Vega                      61
         Ravenclaw                 53
         Hufflepuff                45
         Rohan                     38
         Polaris                   31
         Becrux                    24
         Honeyduke's               21
         Antares                   21
         Arcturus                  19
         Capella                   15
         Gryffindor                14
         Fangorn                   13
         Acrux                     13
         Canopus                   11
         Sirius                    11
         Spica                     11
         Procyon                   10
         Three Broomsticks         10
         Regulus                    9
         Gondor                     8
         Hobbiton                   6
         Castor                     6
         Isengard                   5
         Pleione                    5
         Minas Tirith               5
         Alcor                      4
         Helm's Deep                4
         Mirkwood                   4
         Rigel                      4
         Mintaka                    3
         Sabik                      3
         Erebor                     3
         Slytherin                  3
         Diagon Alley               3
         Fomalhaut                  3
         Hog's Head Inn             3
         Altair                     3
         Aldebaran                  3
         Bree                       2
         Muscida                    2
         Moria                      1
         Rivendell                  1
         Betelgeuse                 1
         Mizar                      1
         Rukbat                     1
         Bellatrix                  1
         Name: country, dtype: int64
```

Filling Null values of country column

In [31]:

```python
dataset['country'] = dataset.groupby('region')['country'].apply(lambda x: x.fillna(x.mode()[0]))
```

In [32]:

```python
dataset.country.value_counts()
```

```
Out[32]: Zonko's Joke Shop      60312
         Lorien                 18051
         Merope                  4468
         Mordor                  2597
         Pollux                   347
         The Shire                166
         Shrieking Shack           77
         Vega                      61
         Ravenclaw                 53
         Hufflepuff                45
         Rohan                     38
         Polaris                   31
         Becrux                    24
         Honeyduke's               21
         Antares                   21
         Arcturus                  19
```

```
Capella                15
Gryffindor             14
Fangorn                13
Acrux                  13
Canopus                11
Sirius                 11
Spica                  11
Procyon                10
Three Broomsticks      10
Regulus                 9
Gondor                  8
Hobbiton                6
Castor                  6
Isengard                5
Pleione                 5
Minas Tirith            5
Alcor                   4
Helm's Deep             4
Mirkwood                4
Rigel                   4
Mintaka                 3
Sabik                   3
Erebor                  3
Slytherin               3
Diagon Alley            3
Fomalhaut               3
Hog's Head Inn          3
Altair                  3
Aldebaran               3
Bree                    2
Muscida                 2
Moria                   1
Rivendell               1
Betelgeuse              1
Mizar                   1
Rukbat                  1
Bellatrix               1
Name: country, dtype: int64
```

In [33]:
```
dataset.isnull().sum()
```

Out[33]:
```
asst_id                    0
product_type               0
manufacture_date           0
contract_start             0
contract_end               0
contact_date               0
contact_type               0
issue_type             12162
repair_type                0
repeat_ct                  0
parts_ct                   0
agent_tenure_indays        0
contact_manager_flg        0
diagnostics                0
region                     0
country                    0
dtype: int64
```

In [34]:
```
dataset.issue_type.value_counts()
```

Out[34]:
```
System Board Components             11310
Fee Based Support                    8829
Operating System                     8420
Hard Drive                           7355
Audio, Video, Speakers, TV Tuner     7297
                                     ...
EQL - Hardware                          2
Compellent - Hardware                   1
MWD                                     1
Try & Buy                               1
Lost / Stolen in Transit                1
Name: issue_type, Length: 82, dtype: int64
```

Filling Null values of issue_type column

Filling Null values of issue_type column

In [35]:

```
dataset['issue_type'] = dataset.groupby(['product_type', 'region'])['issue_type'].apply(lambda x: x.fillna(x.mode
```

In [36]:

```
dataset.issue_type.value_counts()
```

Out[36]:

```
System Board Components            21218
Fee Based Support                  10351
Operating System                    8420
Audio, Video, Speakers, TV Tuner    7401
Hard Drive                          7355
                                    ...
EQL - Hardware                         2
Compellent - Hardware                  1
MWD                                    1
Try & Buy                              1
Lost / Stolen in Transit               1
Name: issue_type, Length: 82, dtype: int64
```

In [37]:

```
dataset.isnull().sum()
```

Out[37]:

```
asst_id                0
product_type           0
manufacture_date       0
contract_start         0
contract_end           0
contact_date           0
contact_type           0
issue_type             0
repair_type            0
repeat_ct              0
parts_ct               0
agent_tenure_indays    0
contact_manager_flg    0
diagnostics            0
region                 0
country                0
dtype: int64
```

In [38]:

```
dataset
```

Out[38]:

	asst_id	product_type	manufacture_date	contract_start	contract_end	contact_date	contact_type	issue_type	repair_type	repeat_ct	pa
0	0	Laptops	6/26/2017	6/26/2017	6/21/2021	10/1/2018	Voice	System Board Components	Hard	0	
1	1	Laptops	12/12/2016	12/12/2016	12/2/2019	10/1/2018	Voice	System Board Components	Hard	0	
2	2	Laptops	5/21/2018	5/21/2018	5/30/2022	10/1/2018	Voice	System Board Components	Hard	0	
3	3	Laptops	2/22/2016	2/22/2016	2/18/2019	10/1/2018	Voice	Hard Drive	Hard	0	
4	4	Laptops	8/7/2017	8/7/2017	8/3/2020	10/1/2018	Voice	System Board Components	Hard	0	
...	...	...	...	...	...	...	...	...	...	...	
99972	16637	Laptops	6/5/2017	6/5/2017	6/1/2020	10/1/2018	Voice	Hard Drive	Hard	0	
99980	16645	Laptops	4/11/2016	4/11/2016	4/8/2019	10/1/2018	Chat	Mechanical Chassis / Rack	Hard	0	
99982	16647	Laptops	1/1/2018	1/1/2018	1/4/2021	10/1/2018	Voice	System Board Components	Hard	0	
99990	16655	Laptops	1/18/2016	12/21/2015	12/24/2018	10/1/2018	Voice	Power Subsystem / Cables / AC Adapter	Hard	0	
								System			

99992	16657	Laptops	9/5/2016	9/5/2016	9/6/2021	10/1/2018	Voice	Board Components	Hard	0

86536 rows × 16 columns

droping unnecessary column

In [39]:

```
dataset.drop(columns = 'asst_id',inplace = True)
```

In [40]:

```
dataset
```

Out[40]:

	product_type	manufacture_date	contract_start	contract_end	contact_date	contact_type	issue_type	repair_type	repeat_ct	parts_ct	a
0	Laptops	6/26/2017	6/26/2017	6/21/2021	10/1/2018	Voice	System Board Components	Hard	0	3	
1	Laptops	12/12/2016	12/12/2016	12/2/2019	10/1/2018	Voice	System Board Components	Hard	0	2	
2	Laptops	5/21/2018	5/21/2018	5/30/2022	10/1/2018	Voice	System Board Components	Hard	0	1	
3	Laptops	2/22/2016	2/22/2016	2/18/2019	10/1/2018	Voice	Hard Drive	Hard	0	1	
4	Laptops	8/7/2017	8/7/2017	8/3/2020	10/1/2018	Voice	System Board Components	Hard	0	1	
...	...	...	...	...	...	...	...	...	...	...	
99972	Laptops	6/5/2017	6/5/2017	6/1/2020	10/1/2018	Voice	Hard Drive	Hard	0	3	
99980	Laptops	4/11/2016	4/11/2016	4/8/2019	10/1/2018	Chat	Mechanical Chassis / Rack	Hard	0	4	
99982	Laptops	1/1/2018	1/1/2018	1/4/2021	10/1/2018	Voice	System Board Components	Hard	0	2	
99990	Laptops	1/18/2016	12/21/2015	12/24/2018	10/1/2018	Voice	Power Subsystem / Cables / AC Adapter	Hard	0	2	
99992	Laptops	9/5/2016	9/5/2016	9/6/2021	10/1/2018	Voice	System Board Components	Hard	0	1	

86536 rows × 15 columns

Changing important dates columns in date datatype

In [41]:

```
dataset[["manufacture_date","contract_start","contract_end","contact_date"]] = dataset[["manufacture_date","contr
```

In [42]:

```
dataset['agent_tenure_indays'] = dataset['agent_tenure_indays'].astype("int64")
```

In [43]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 86536 entries, 0 to 99992
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype         
---  ------               --------------  -----         
 0   product_type         86536 non-null  object        
 1   manufacture_date     86536 non-null  datetime64[ns]
 2   contract_start       86536 non-null  datetime64[ns]
 3   contract_end         86536 non-null  datetime64[ns]
 4   contact_date         86536 non-null  datetime64[ns]
 5   contact_type         86536 non-null  object        
 6   issue_type           86536 non-null  object        
 7   repair_type          86536 non-null  object
 8   repeat_ct            86536 non-null  int64
 9   parts_ct             86536 non-null  int64
 10  agent_tenure_indays  86536 non-null  int64
 11  contact_manager_flg  86536 non-null  int64
 12  diagnostics          86536 non-null  object
 13  region               86536 non-null  object
 14  country              86536 non-null  object
dtypes: datetime64[ns](4), int64(4), object(7)
memory usage: 13.1+ MB
```

In [44]:

```
dataset
```

Out[44]:

	product_type	manufacture_date	contract_start	contract_end	contact_date	contact_type	issue_type	repair_type	repeat_ct	parts_ct	a
0	Laptops	2017-06-26	2017-06-26	2021-06-21	2018-10-01	Voice	System Board Components	Hard	0	3	
1	Laptops	2016-12-12	2016-12-12	2019-12-02	2018-10-01	Voice	System Board Components	Hard	0	2	
2	Laptops	2018-05-21	2018-05-21	2022-05-30	2018-10-01	Voice	System Board Components	Hard	0	1	
3	Laptops	2016-02-22	2016-02-22	2019-02-18	2018-10-01	Voice	Hard Drive	Hard	0	1	
4	Laptops	2017-08-07	2017-08-07	2020-08-03	2018-10-01	Voice	System Board Components	Hard	0	1	
...	...	...	...	...	...	...	...	...	...	...	
99972	Laptops	2017-06-05	2017-06-05	2020-06-01	2018-10-01	Voice	Hard Drive	Hard	0	3	
99980	Laptops	2016-04-11	2016-04-11	2019-04-08	2018-10-01	Chat	Mechanical Chassis / Rack	Hard	0	4	
99982	Laptops	2018-01-01	2018-01-01	2021-01-04	2018-10-01	Voice	System Board Components	Hard	0	2	
99990	Laptops	2016-01-18	2015-12-21	2018-12-24	2018-10-01	Voice	Power Subsystem / Cables / AC Adapter	Hard	0	2	
99992	Laptops	2016-09-05	2016-09-05	2021-09-06	2018-10-01	Voice	System Board Components	Hard	0	1	

86536 rows × 15 columns

Doing Some important Data Analysis which will be good to take some important decision

1. Which product type has the highest number of warranty claims?

In [45]:

```
# group the data by product_type and count the number of warranty claims for each product type
warranty_claims_by_product_type = dataset.groupby('product_type')['contract_start'].count().sort_values(ascending
warranty_claims_by_product_type
```

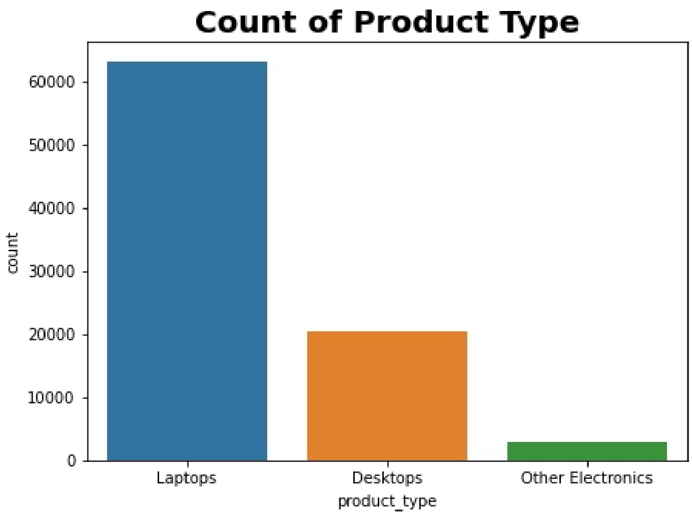
Out[45]:

```
product_type
Laptops              63173
Desktops             20461
Other Electronics     2902
Name: contract_start, dtype: int64
```

In [46]:

```
#Plot the graph for count of types of products used
plt.figure(figsize = (7,5))

sns.countplot(x ='product_type', data = dataset)
```

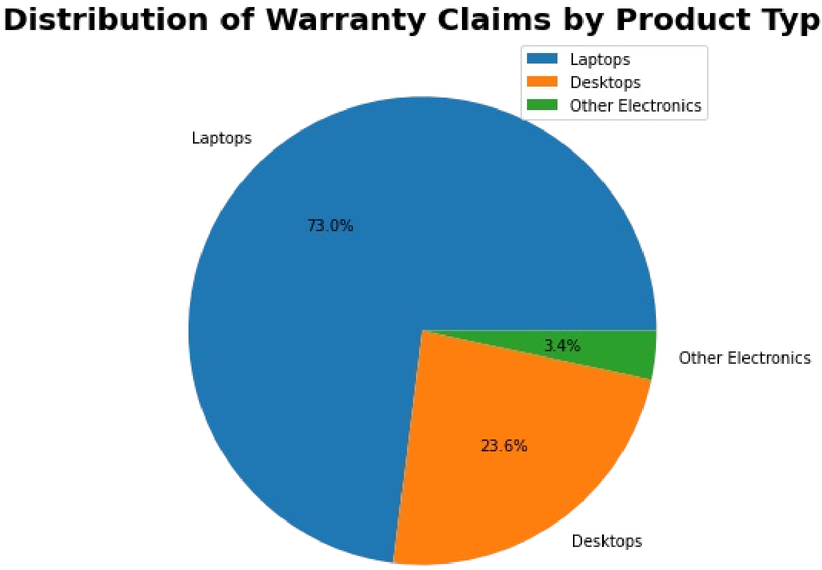
```
plt.title("Count of Product Type",color = "black",size = 20, fontweight = "bold")
plt.show()

print("Product type with the highest number of warranty claims: ", warranty_claims_by_product_type.index[0])
```



```
Product type with the highest number of warranty claims:  Laptops
```

In [48]:

```
#Plot the pie chart for count of types of products used
plt.figure(figsize = (10,7))
plt.pie(warranty_claims_by_product_type, labels=warranty_claims_by_product_type.index, autopct='%1.1f%%')
plt.title('Distribution of Warranty Claims by Product Type',color = "black",size = 20, fontweight = "bold")
plt.legend()
plt.show()
```



2. What is the most common issue reported by customers for each product type?

In [49]:

```
# find the mode of issue_type for each group
most_common_issue = dataset.groupby('product_type')['issue_type'].apply(lambda x: x.mode()[0])
most_common_issue
```

Out[49]:

```
product_type
Desktops                    Fee Based Support
Laptops                 System Board Components
Other Electronics                Imaging Device
Name: issue_type, dtype: object
```

In [50]:

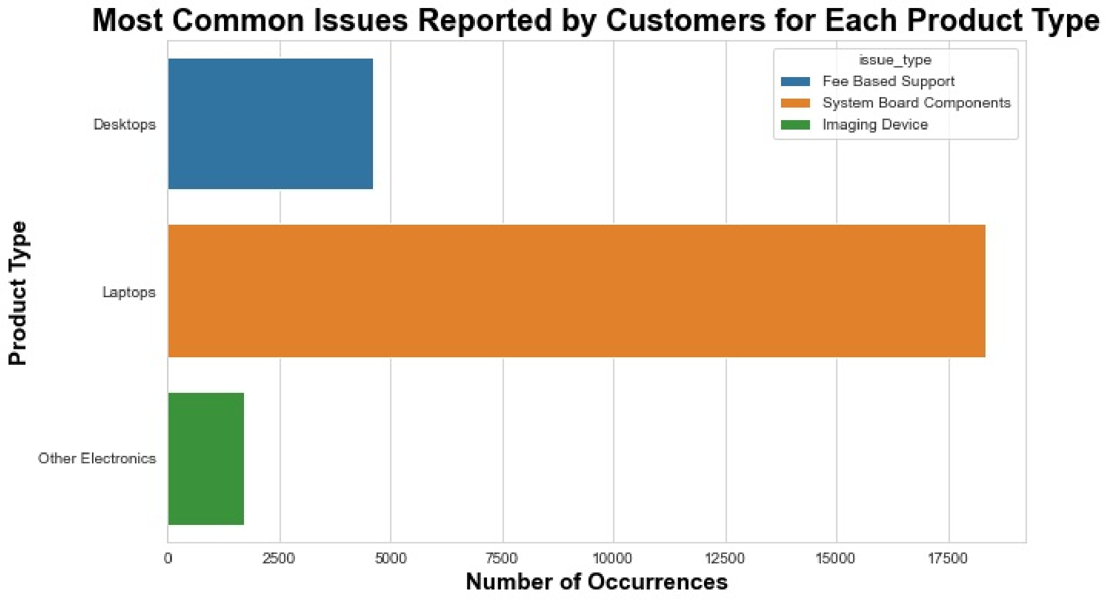
```
# grouping the data by product type and issue type
grouped_data = dataset.groupby(['product_type', 'issue_type'])

# counting the number of occurrences for each group
count_data = grouped_data.size().reset_index(name='count')

# get the most common issues for each product type
max_issue = count_data.groupby('product_type')['count'].idxmax()
max_issue_data = count_data.loc[max_issue]
max_issue_data
```

Out[50]:

	product_type	issue_type	count
33	Desktops	Fee Based Support	4599
143	Laptops	System Board Components	18329
171	Other Electronics	Imaging Device	1715

In [51]:

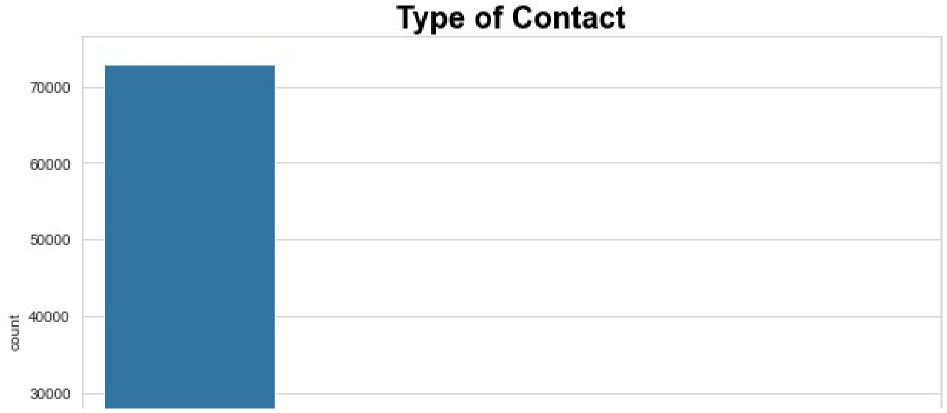
```
# plot the horizontal bar chart
sns.set_style("whitegrid")
plt.figure(figsize=(10, 6))
sns.barplot(data=max_issue_data, x='count', y='product_type', hue='issue_type', dodge=False)
plt.xlabel('Number of Occurrences',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Product Type',color = "black",size = 15, fontweight = "bold")
plt.title('Most Common Issues Reported by Customers for Each Product Type',color = "black",size = 20, fontweight
plt.show()
```

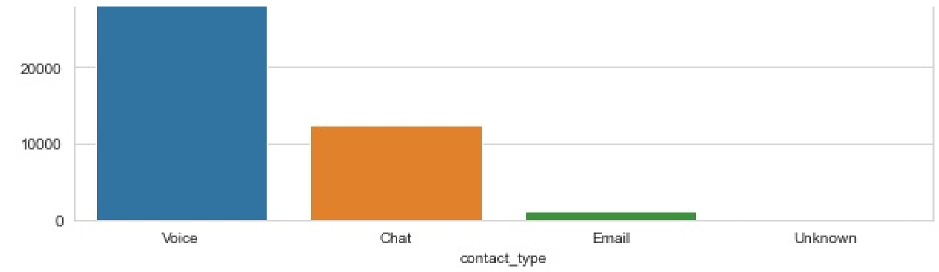


3. Plot the graph for medium of contact

In [52]:

```
#Plot the graph for medium of contact

plt.figure(figsize = (10,7))
sns.countplot(x ='contact_type', data = dataset)
plt.title("Type of Contact",color = "black",size = 20, fontweight = "bold")
plt.show()
```





4. Find any Corelation in the dataset

In [53]:

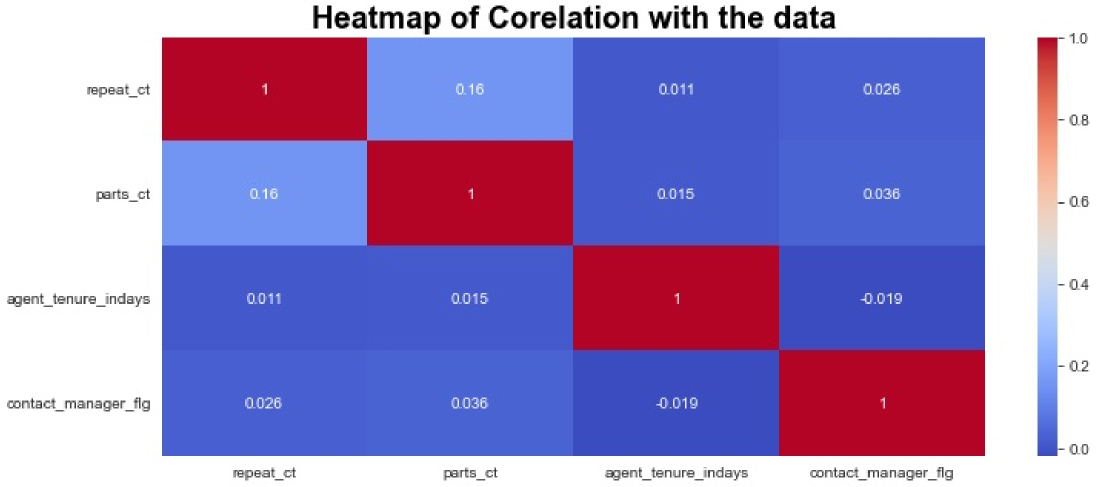
```
dataset.corr()
```

Out[53]:

	repeat_ct	parts_ct	agent_tenure_indays	contact_manager_flg
repeat_ct	1.000000	0.156486	0.010638	0.025665
parts_ct	0.156486	1.000000	0.015437	0.035753
agent_tenure_indays	0.010638	0.015437	1.000000	-0.018653
contact_manager_flg	0.025665	0.035753	-0.018653	1.000000

In [54]:

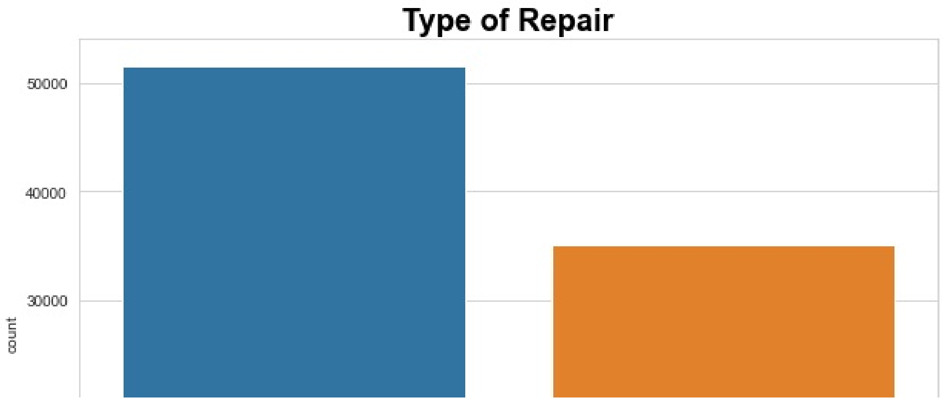
```
plt.figure(figsize = (12,5))
sns.heatmap(dataset.corr(),cmap='coolwarm',annot=True)
plt.title("Heatmap of Corelation with the data",color = "black",size = 20, fontweight = "bold")
plt.show()
```

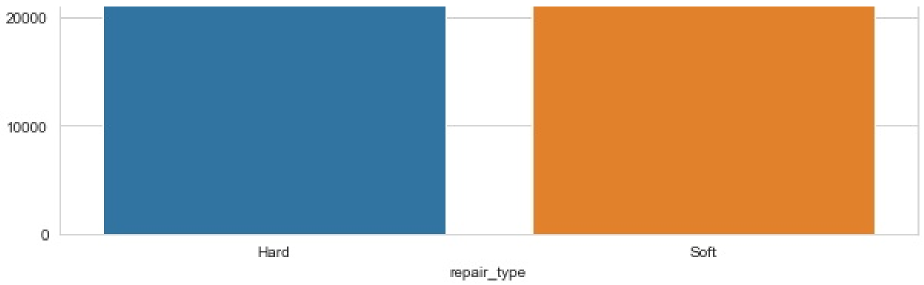


5. Plot the graph for types of repair

In [55]:

```
#Plot the graph for types of repair
plt.figure(figsize = (10,7))
sns.countplot(x ='repair_type', data = dataset)
plt.title("Type of Repair",color = "black",size = 20, fontweight = "bold")
plt.show()
```





6. What is the average agent tenure in days for each product type?

In [56]:

```
# group the data by product type and calculated the mean of agent tenure
average_agent_tenure = dataset.groupby('product_type')['agent_tenure_indays'].mean().sort_values(ascending = Fals
average_agent_tenure
```

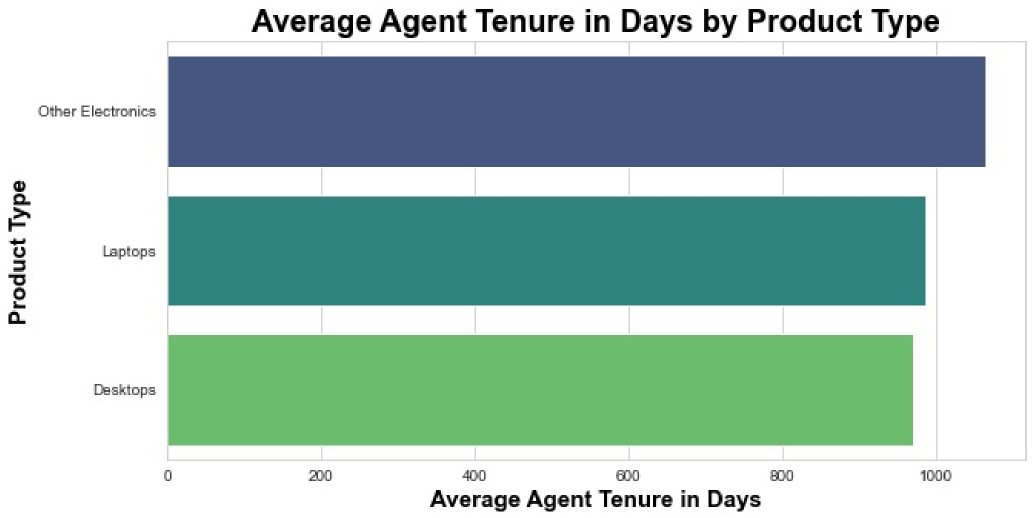
Out[56]:

```
product_type
Other Electronics    1063.136458
Laptops               984.898485
Desktops              970.131665
Name: agent_tenure_indays, dtype: float64
```

Horizontal Bar Chart

In [57]:

```
# create a horizontal bar chart
plt.figure(figsize=(10,5))
sns.barplot(y=average_agent_tenure.index, x=average_agent_tenure.values, palette='viridis')
plt.title('Average Agent Tenure in Days by Product Type',color = "black",size = 20, fontweight = "bold")
plt.xlabel('Average Agent Tenure in Days',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Product Type',color = "black",size = 15, fontweight = "bold")
plt.show()
```

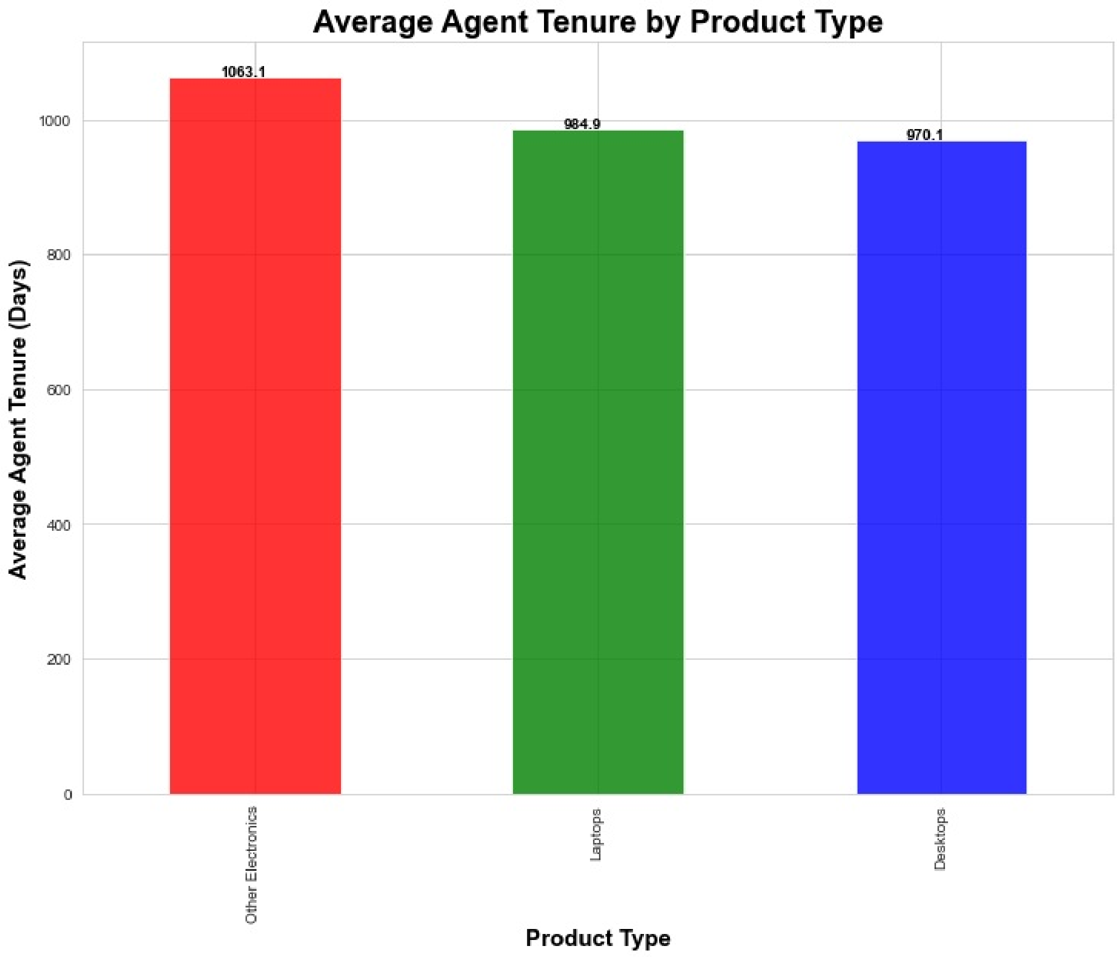


Vertical Bar chart

In [59]:

```
plt.figure(figsize = (12,9))

# plot a bar chart
ax = average_agent_tenure.plot(kind='bar', color=['red', 'green', 'blue'], alpha=0.8)

# add labels and title
ax.set_xlabel('Product Type',color = "black",size = 15, fontweight = "bold")
ax.set_ylabel('Average Agent Tenure (Days)',color = "black",size = 15, fontweight = "bold")
ax.set_title('Average Agent Tenure by Product Type',color = "black",size = 20, fontweight = "bold")

# add the average values as text inside the bars
for i, v in enumerate(average_agent_tenure):
    ax.text(i - 0.1, v + 1, str(round(v, 1)), color='black', fontweight='bold')
```

```
# display the plot
plt.show()
print("Maximum average agent tenure by product type :",average_agent_tenure.index[0])
```



```
Maximum average agent tenure by product type : Other Electronics
```

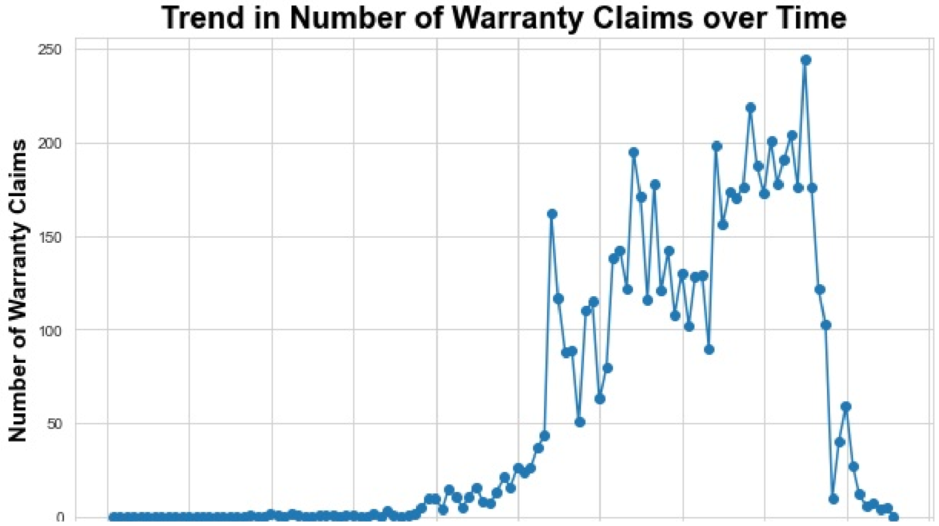
7. What is the trend in the number of warranty claims over time?

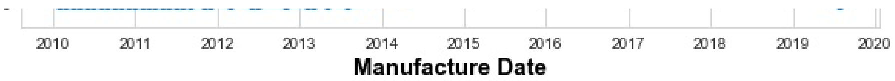
In [60]:

```
# group by month and sum the 'repeat_ct' column
warranty_claims_by_month = dataset.groupby(pd.Grouper(key='manufacture_date', freq='M')).agg({'repeat_ct': 'sum'}
```

In [61]:

```
# plot the line chart
plt.figure(figsize=(10,6))
plt.plot(warranty_claims_by_month.index, warranty_claims_by_month['repeat_ct'], marker='o')
plt.xlabel('Manufacture Date',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Number of Warranty Claims',color = "black",size = 15, fontweight = "bold")
plt.title('Trend in Number of Warranty Claims over Time',color = "black",size = 20, fontweight = "bold")
plt.show()
```





8. Are there any specific regions that have a higher number of warranty claims than others?

In [62]:

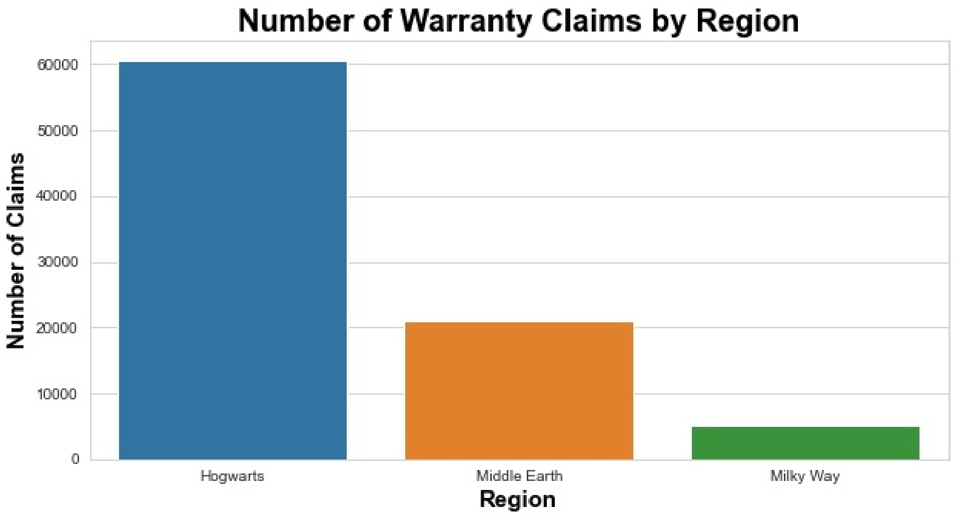
```
# group the data by region and calculate the total number of warranty claims in each region
claims_by_region = dataset.groupby('region')['product_type'].count()
claims_by_region
```

Out[62]:

```
region
Hogwarts        60541
Middle Earth    20904
Milky Way        5091
Name: product_type, dtype: int64
```

In [63]:

```
plt.figure(figsize = (10,5))

# plot the results using a bar chart
sns.barplot(x = claims_by_region.index, y = claims_by_region.values)

# add title and axis labels
plt.title('Number of Warranty Claims by Region',color = "black",size = 20, fontweight = "bold")
plt.xlabel('Region',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Number of Claims',color = "black",size = 15, fontweight = "bold")

# show the plot
plt.show()
print("Region where heighest number of warranty claimed is: ", claims_by_region.index[0])
```



```
Region where heighest number of warranty claimed is:  Hogwarts
```

9. Is there a relationship between the number of parts sent and the product type?

In [64]:

```
# group the data by product type and calculate the average number of parts sent for each product type
avg_parts_by_product_type = dataset.groupby('product_type')['parts_ct'].mean()
avg_parts_by_product_type
```

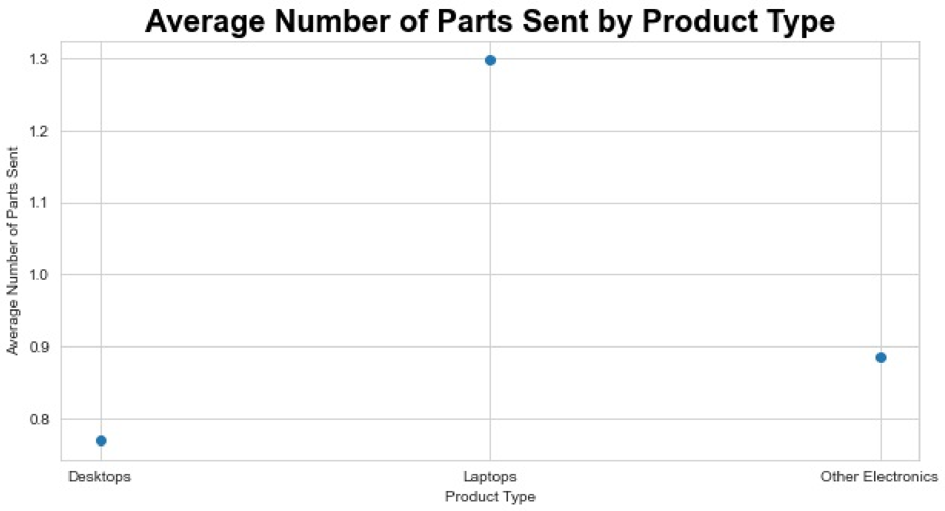
Out[64]:

```
product_type
Desktops             0.768975
Laptops              1.297659
Other Electronics    0.885252
Name: parts_ct, dtype: float64
```

In [65]:

In [65]:

```python
plt.figure(figsize = (10,5))

# plot the results using a scatter plot
plt.scatter(avg_parts_by_product_type.index, avg_parts_by_product_type.values)

# add title and axis labels
plt.title('Average Number of Parts Sent by Product Type',color = "black",size = 20, fontweight = "bold")
plt.xlabel('Product Type')
plt.ylabel('Average Number of Parts Sent',)

# show the plot
plt.show()
```



In [66]:

```python
# group the data by product type and calculate the total number of parts sent for each product type
parts_sent_by_product_type = dataset.groupby('product_type')['parts_ct'].sum()
parts_sent_by_product_type
```

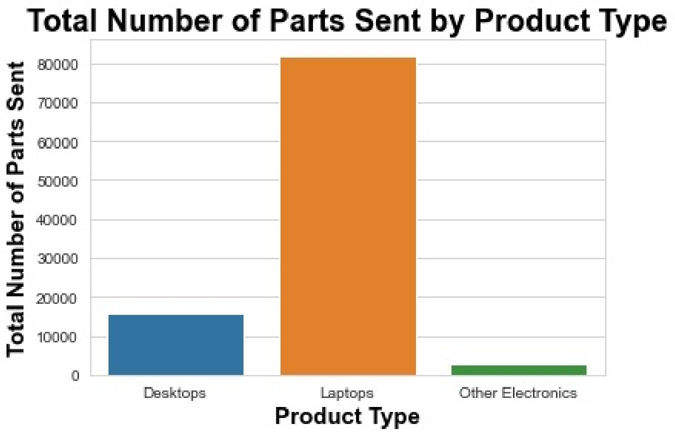
Out[66]:

```
product_type
Desktops             15734
Laptops              81977
Other Electronics     2569
Name: parts_ct, dtype: int64
```

In [67]:

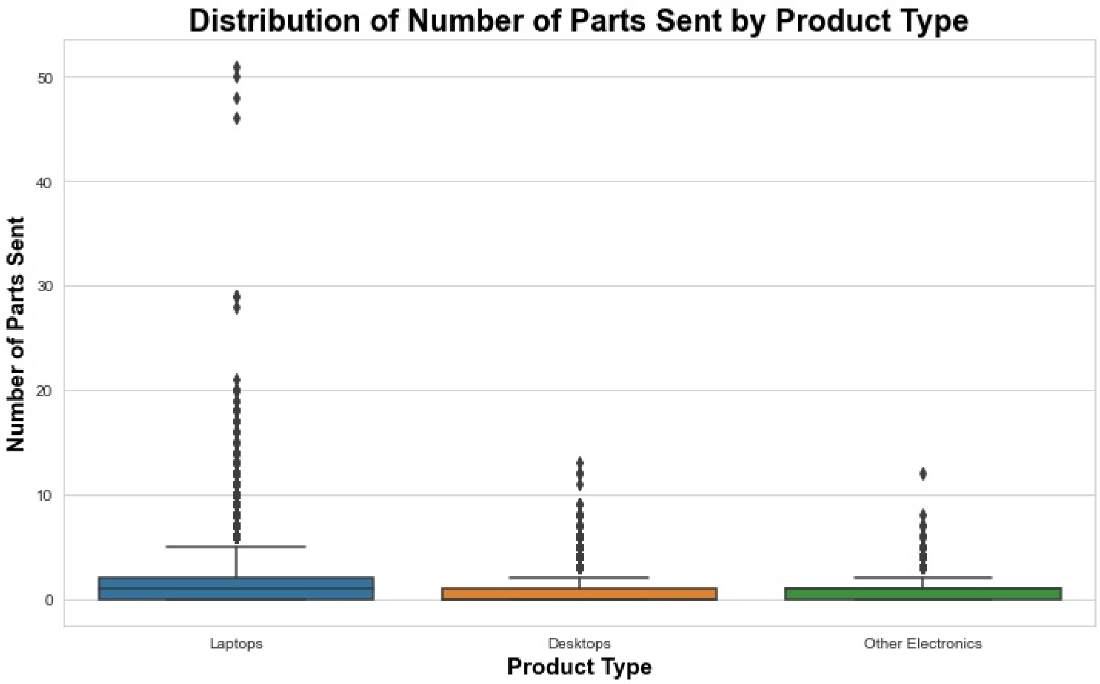
```python
# plot the results using a bar chart
sns.barplot(parts_sent_by_product_type.index, parts_sent_by_product_type.values)

# add title and axis labels
plt.title('Total Number of Parts Sent by Product Type', color='black', size=20, fontweight='bold')
plt.xlabel('Product Type',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Total Number of Parts Sent',color = "black",size = 15, fontweight = "bold")

# show the plot
plt.show()
```

```
C:\Users\Dell\AppData\Roaming\Python\Python36\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the fo
llowing variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, an
d passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



In [68]:

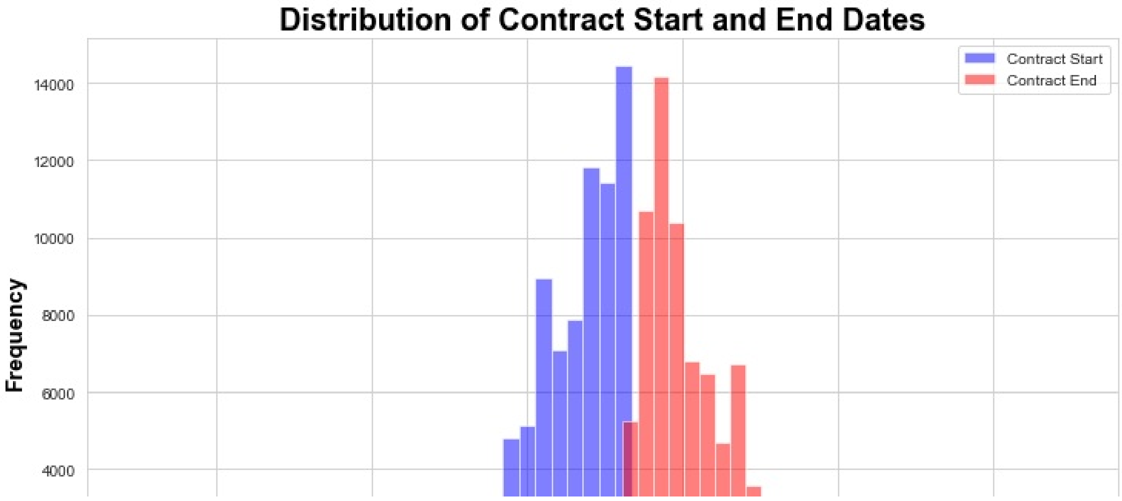
```
plt.figure(figsize = (12,7))
# create a box plot of the number of parts sent for each product type
sns.boxplot(x='product_type', y='parts_ct', data=dataset)

# add title and axis labels
plt.title('Distribution of Number of Parts Sent by Product Type',color = "black",size = 20, fontweight = "bold")
plt.xlabel('Product Type',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Number of Parts Sent',color = "black",size = 15, fontweight = "bold")

# show the plot
plt.show()
```

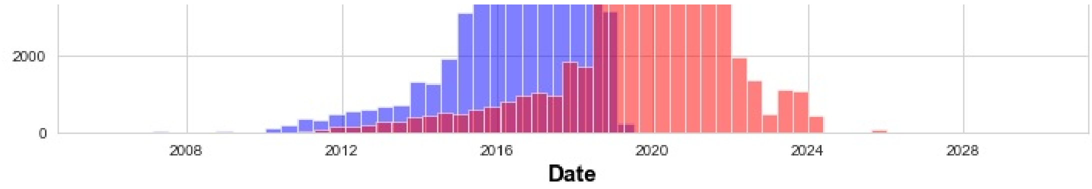


10. What is the distribution of contract start and end dates?

In [69]:

```
plt.figure(figsize = (12,7))

# create a histogram of contract start dates
plt.hist(dataset['contract_start'], bins=50, alpha=0.5, color='blue', label='Contract Start')

# create a histogram of contract end dates
plt.hist(dataset['contract_end'], bins=50, alpha=0.5, color='red', label='Contract End')

# add title and axis labels
plt.title('Distribution of Contract Start and End Dates',color = "black",size = 20, fontweight = "bold")
plt.xlabel('Date',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Frequency',color = "black",size = 15, fontweight = "bold")

# add legend
plt.legend()

# show the plot
plt.show()
```

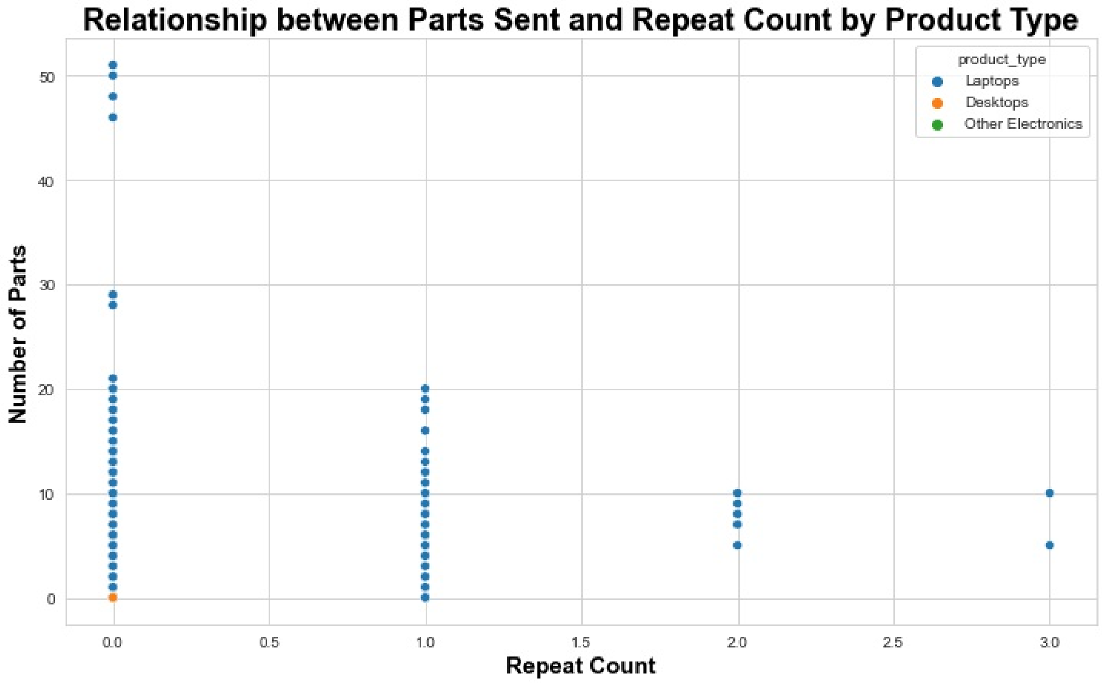




11. Is there a relationship between the number of parts sent and the repeat count for each product type?

In [70]:

```
# Plot scatter plot for parts vs repeat count, with different colors for each product type
plt.figure(figsize=(12,7))
sns.scatterplot(data=dataset, x='repeat_ct', y='parts_ct', hue='product_type')

# Add labels and title
plt.xlabel('Repeat Count',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Number of Parts',color = "black",size = 15, fontweight = "bold")
plt.title('Relationship between Parts Sent and Repeat Count by Product Type',color = "black",size = 20, fontweigh

# Show the plot
plt.show()
```



12. Is there a relationship between the number of parts sent and the repeat count for each product type?

In [71]:

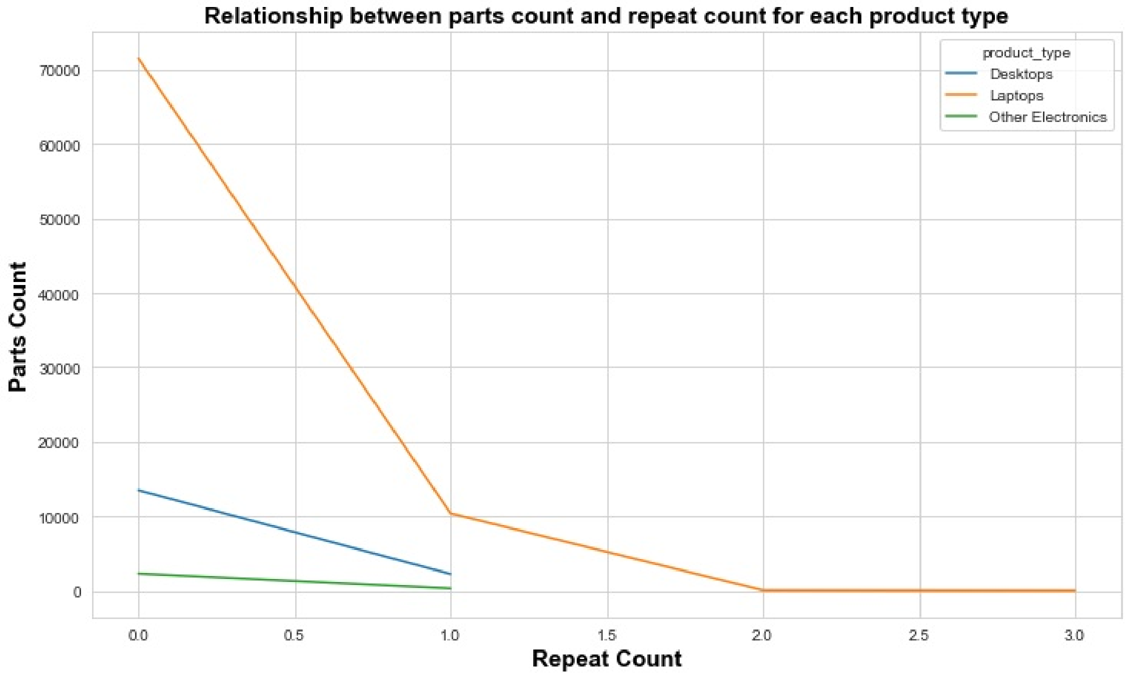
```
grouped_df = dataset.groupby(['product_type', 'repeat_ct'])['parts_ct'].sum().reset_index()
grouped_df
```

Out[71]:

	product_type	repeat_ct	parts_ct
0	Desktops	0	13502
1	Desktops	1	2232
2	Laptops	0	71530
3	Laptops	1	10386
4	Laptops	2	46
5	Laptops	3	15
6	Other Electronics	0	2290
7	Other Electronics	1	279

In [72]:

```
plt.figure(figsize = (12,7))
# Create a double line chart to visualize the relationship between parts count and repeat count for each product
sns.lineplot(x='repeat_ct', y='parts_ct', data=grouped_df, hue='product_type')
# Set the title and axis labels
plt.title('Relationship between parts count and repeat count for each product type',color = "black",size = 15, fo
plt.xlabel('Repeat Count',color = "black",size = 15, fontweight = "bold")
plt.ylabel('Parts Count',color = "black",size = 15, fontweight = "bold")

# Show the chart
plt.show()
```



Thank You

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js