Proyek Analisis Data: E-Commerce Public Dataset

- Nama: Farrel Rasyad
- Email: farrelrasyad.frr@gmail.com (mailto:farrelrasyad.frr@gmail.com)
- Id Dicoding: farrel_rasyad_eypa
- github repo: https://github.com/FrL1902/python-data-analysis-project-dicoding (https://github.com/FrL1902/python-data-analysis-project-dicoding)

Menentukan Pertanyaan Bisnis

- Apa saja 5 kategori barang terlaris dalam jangka waktu tertentu?
- Apa saja 5 kategori dengan skor review (rata rata) terbagus dan terburuk?

Import semua library yang dibutuhkan

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import streamlit as st
from babel.numbers import format_currency
sns.set(style='dark')
```

Data Wrangling

Gathering Data

Di bagian gathering data, saya load dataset yang didownload dari dicoding

Customer

In [5]:
```
customers_df = pd.read_csv("data/customers_dataset.csv")
customers_df.head()
```

Out[5]:

	customer_id	customer_unique_id	customer_zip_code_pre
0	06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14
1	18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	9
2	4e7b3e00288586ebd08712fdd0374a03	060e732b5b29e8181a18229c7b0b2b5e	1
3	b2b6027bc5c5109e529d4dc6358b12c3	259dac757896d24d7702b9acbbff3f3c	8
4	4f2d8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d066	13

Geolocation

In [6]:
```
geolocation_df = pd.read_csv("data/geolocation_dataset.csv")
geolocation_df.head()
```

Out[6]:

	geolocation_zip_code_prefix	geolocation_lat	geolocation_lng	geolocation_city	geolocation_stat
0	1037	-23.545621	-46.639292	sao paulo	S
1	1046	-23.546081	-46.644820	sao paulo	S
2	1046	-23.546129	-46.642951	sao paulo	S
3	1041	-23.544392	-46.639499	sao paulo	S
4	1035	-23.541578	-46.641607	sao paulo	S

Order Items

In [7]:
```
order_items_df = pd.read_csv("data/order_items_dataset.csv")
order_items_df.head()
```

Out[7]:

	order_id	order_item_id	product_id	
0	00010242fe8c5a6d1ba2dd792cb16214	1	4244733e06e7ecb4970a6e2683c13e61	48436d
1	00018f77f2f0320c557190d7a144bdd3	1	e5f2d52b802189ee658865ca93d83a8f	dd7ddc
2	000229ec398224ef6ca0657da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b5103
3	00024acbcdf0a6daa1e931b038114c75	1	7634da152a4610f1595efa32f14722fc	9d7a1d3
4	00042b26cf59d7ce69dfabb4e55b4fd9	1	ac6c3623068f30de03045865e4e10089	df5603

Order Payments

In [8]:
```
order_payments_df = pd.read_csv("data/order_payments_dataset.csv")
order_payments_df.head()
```

Out[8]:

	order_id	payment_sequential	payment_type	payment_installments
0	b81ef226f3fe1789b1e8b2acac839d17	1	credit_card	8
1	a9810da82917af2d9aefd1278f1dcfa0	1	credit_card	1
2	25e8ea4e93396b6fa0d3dd708e76c1bd	1	credit_card	1
3	ba78997921bbcdc1373bb41e913ab953	1	credit_card	8
4	42fdf880ba16b47b59251dd489d4441a	1	credit_card	2

Order Reviews

In [9]:
```
order_reviews_df = pd.read_csv("data/order_reviews_dataset.csv")
order_reviews_df.head()
```

Out[9]:

	review_id	order_id	review_score	review_c
0	7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4	
1	80e641a11e56f04c1ad469d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5	
2	228ce5500dc1d8e020d8d1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5	
3	e64fb393e7b32834bb789ff8bb30750e	658677c97b385a9be170737859d3511b	5	
4	f7c4243c7fe1938f181bec41a392bdeb	8e6bfb81e283fa7e4f11123a3fb894f1	5	

Orders

In [10]:
```
orders_df = pd.read_csv("data/orders_dataset.csv")
orders_df.head()
```

Out[10]:

	order_id	customer_id	order_status	order_pur
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	20
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	20
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	20
3	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	20
4	ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	20

Product Category Name Translation

In [11]:
```
product_category_name_translation_df = pd.read_csv("data/product_category_name_
product_category_name_translation_df.head()
```

Out[11]:

	product_category_name	product_category_name_english
0	beleza_saude	health_beauty
1	informatica_acessorios	computers_accessories
2	automotivo	auto
3	cama_mesa_banho	bed_bath_table
4	moveis_decoracao	furniture_decor

Products

In [12]:
```
products_df = pd.read_csv("data/products_dataset.csv")
products_df.head()
```

Out[12]:

	product_id	product_category_name	product_name_lenght	product_des
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	40.0	
1	3aa071139cb16b67ca9e5dea641aaa2f	artes	44.0	
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer	46.0	
3	cef67bcfe19066a932b7673e239eb23d	bebes	27.0	
4	9dc1a7de274444849c219cff195d0b71	utilidades_domesticas	37.0	

Sellers

In [13]:
```
sellers_df = pd.read_csv("data/sellers_dataset.csv")
sellers_df.head()
```

Out[13]:

	seller_id	seller_zip_code_prefix	seller_city	seller_state
0	3442f8959a84dea7ee197c632cb2df15	13023	campinas	SP
1	d1b65fc7debc3361ea86b5f14c68d2e2	13844	mogi guacu	SP
2	ce3ad9de960102d0677a81f5d0bb7b2d	20031	rio de janeiro	RJ
3	c0f3eea2e14555b6faeea3dd58c1b1c3	4195	sao paulo	SP
4	51a04a8a6bdcb23deccc82b0b80742cf	12914	braganca paulista	SP

Assessing Data

Di bagian assessing data, saya mengecek data seperti:

- dari tipe data yang digunakan apakah sudah sesuai atau tidak,
- apakah ada data yang null?
- apakah ada data yang duplikat?
- apakah ada data yang abnormal dalam dataset?

Customer

In [14]:
```
customers_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 5 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   customer_id               99441 non-null  object
 1   customer_unique_id        99441 non-null  object
 2   customer_zip_code_prefix  99441 non-null  int64 
 3   customer_city             99441 non-null  object
 4   customer_state            99441 non-null  object
dtypes: int64(1), object(4)
memory usage: 3.8+ MB
```

In [15]:
```
customers_df.isna().sum()
```

Out[15]:
```
customer_id                 0
customer_unique_id          0
customer_zip_code_prefix    0
customer_city               0
customer_state              0
dtype: int64
```

In [16]:
```
print("Duplicates: ", customers_df.duplicated().sum())
```

```
Duplicates:  0
```

In [17]:
```
customers_df.describe()
```

Out[17]:

	customer_zip_code_prefix
count	99441.000000
mean	35137.474583
std	29797.938996
min	1003.000000
25%	11347.000000
50%	24416.000000
75%	58900.000000
max	99990.000000

Tabel ini berisikan data tamabahn customer, yaotu lokasi customer. Sudah di cek bahwa customer_zip_code_prefix memang ada yang nilainya 99990 di dunia nyata, tidak ada kelainan data yang abnormal. Tidak ada data yang null dan tidak ada data yang duplikat. Tabel Customer sudah bisa digunakan

Geolocation

In [18]:
```
geolocation_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000163 entries, 0 to 1000162
Data columns (total 5 columns):
 #   Column                       Non-Null Count    Dtype  
---  ------                       --------------    -----  
 0   geolocation_zip_code_prefix  1000163 non-null  int64  
 1   geolocation_lat              1000163 non-null  float64
 2   geolocation_lng              1000163 non-null  float64
 3   geolocation_city             1000163 non-null  object 
 4   geolocation_state            1000163 non-null  object 
dtypes: float64(2), int64(1), object(2)
memory usage: 38.2+ MB
```

In [19]:
```
geolocation_df.isna().sum()
```

Out[19]:
```
geolocation_zip_code_prefix    0
geolocation_lat                0
geolocation_lng                0
geolocation_city               0
geolocation_state              0
dtype: int64
```

In [20]:
```
print("Duplicates: ", geolocation_df.duplicated().sum())
```

```
Duplicates:  261831
```

In [21]:
```
geolocation_df.describe()
```

Out[21]:

	geolocation_zip_code_prefix	geolocation_lat	geolocation_lng
count	1.000163e+06	1.000163e+06	1.000163e+06
mean	3.657417e+04	-2.117615e+01	-4.639054e+01
std	3.054934e+04	5.715866e+00	4.269748e+00
min	1.001000e+03	-3.660537e+01	-1.014668e+02
25%	1.107500e+04	-2.360355e+01	-4.857317e+01
50%	2.653000e+04	-2.291938e+01	-4.663788e+01
75%	6.350400e+04	-1.997962e+01	-4.376771e+01
max	9.999000e+04	4.506593e+01	1.211054e+02

Tidak ada kelainan data yang abnormal. Tidak ada data yang null. Akan tetapi, ada banyak data yang duplikat. Harus di drop terlebih dahulu duplicate data nya

Order Items

In [22]:
```
order_items_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 7 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   order_id             112650 non-null  object 
 1   order_item_id        112650 non-null  int64  
 2   product_id           112650 non-null  object 
 3   seller_id            112650 non-null  object 
 4   shipping_limit_date  112650 non-null  object 
 5   price                112650 non-null  float64
 6   freight_value        112650 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 6.0+ MB
```

In [23]:
```
order_items_df.isna().sum()
```

Out[23]:
```
order_id              0
order_item_id         0
product_id            0
seller_id             0
shipping_limit_date   0
price                 0
freight_value         0
dtype: int64
```

In [24]:
```
print("Duplicates: ", order_items_df.duplicated().sum())
```

```
Duplicates:  0
```

In [25]:
```
order_items_df.describe()
```

Out[25]:

	order_item_id	price	freight_value
count	112650.000000	112650.000000	112650.000000
mean	1.197834	120.653739	19.990320
std	0.705124	183.633928	15.806405
min	1.000000	0.850000	0.000000
25%	1.000000	39.900000	13.080000
50%	1.000000	74.990000	16.260000
75%	1.000000	134.900000	21.150000
max	21.000000	6735.000000	409.680000

Untuk tabel order items ini tidak ada data yang duplikat atau null. Untuk price ini lebih bergantung terhadap sellernya ingin menjual dengan harga seberapa tinggi/rendah. Jadi perbedaan value di max dan min yang jauh ini normal. Akan tetapi, untuk kolom shipping_limit_date seharusnya menggunakan tipe data datetime. Selain itu, seharusnya data sudah bisa langsung digunakan

Order Payments

In [26]:
```
order_payments_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103886 entries, 0 to 103885
Data columns (total 5 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   order_id              103886 non-null  object 
 1   payment_sequential    103886 non-null  int64  
 2   payment_type          103886 non-null  object 
 3   payment_installments  103886 non-null  int64  
 4   payment_value         103886 non-null  float64
dtypes: float64(1), int64(2), object(2)
memory usage: 4.0+ MB
```

In [27]:
```
order_payments_df.isna().sum()
```

Out[27]:
```
order_id                0
payment_sequential      0
payment_type            0
payment_installments    0
payment_value           0
dtype: int64
```

In [28]:
```
print("Duplicates: ", order_payments_df.duplicated().sum())
```

```
Duplicates:  0
```

In [29]:
```
order_payments_df.describe()
```

Out[29]:

	payment_sequential	payment_installments	payment_value
count	103886.000000	103886.000000	103886.000000
mean	1.092679	2.853349	154.100380
std	0.706584	2.687051	217.494064
min	1.000000	0.000000	0.000000
25%	1.000000	1.000000	56.790000
50%	1.000000	1.000000	100.000000
75%	1.000000	4.000000	171.837500
max	29.000000	24.000000	13664.080000

Di tabel Order Payments ini seharusnya sudah bisa digunakan karena tidak ada data yang null dan duplicate.

Order Reviews

In [30]:
```
order_reviews_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   review_id                99224 non-null  object
 1   order_id                 99224 non-null  object
 2   review_score             99224 non-null  int64 
 3   review_comment_title     11568 non-null  object
 4   review_comment_message   40977 non-null  object
 5   review_creation_date     99224 non-null  object
 6   review_answer_timestamp  99224 non-null  object
dtypes: int64(1), object(6)
memory usage: 5.3+ MB
```

In [31]:
```
order_reviews_df.isna().sum()
```

Out[31]:
```
review_id                     0
order_id                      0
review_score                  0
review_comment_title      87656
review_comment_message    58247
review_creation_date          0
review_answer_timestamp       0
dtype: int64
```

In [32]:
```
print("Duplicates: ", order_reviews_df.duplicated().sum())
```

```
Duplicates:  0
```

In [33]:
```
order_reviews_df.describe()
```

Out[33]:

	review_score
count	99224.000000
mean	4.086421
std	1.347579
min	1.000000
25%	4.000000
50%	5.000000
75%	5.000000
max	5.000000

Di tabel ini ada lumayan banyak kolom yang null, tetapi sebenarnya kolom tersebut tidak berkaitan dengan data pertanyaan bisnis. Lalu, sebenarnya untuk review, memang banyak orang yang hanya menilai dengan memberikan beberapa bintang saja. Bnayak orang yang tidak mengkomen/review karena memang customer tidak terlalu peduli. Ada juga kesalahan di tipe data. Seharusnya kolom review_answer_timestamp dan review_creation_date menggunakan datetime. Sebenarnya Kolom ini tidak digunakan sama sekali karena tidak akan digunakan dalam proses menjawab pertanyaan bisnis yang saya ajukan. Jadi kemungkinan besar data dari tabel ini tidak akan saya gunakan.

Orders

In [34]:
```
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   order_id                       99441 non-null  object
 1   customer_id                    99441 non-null  object
 2   order_status                   99441 non-null  object
 3   order_purchase_timestamp       99441 non-null  object
 4   order_approved_at              99281 non-null  object
 5   order_delivered_carrier_date   97658 non-null  object
 6   order_delivered_customer_date  96476 non-null  object
 7   order_estimated_delivery_date  99441 non-null  object
dtypes: object(8)
memory usage: 6.1+ MB
```

In [35]:
```
orders_df.isna().sum()
```

Out[35]:
```
order_id                            0
customer_id                         0
order_status                        0
order_purchase_timestamp            0
order_approved_at                 160
order_delivered_carrier_date     1783
order_delivered_customer_date    2965
order_estimated_delivery_date       0
dtype: int64
```

In [36]:
```
print("Duplicates: ", orders_df.duplicated().sum())
```

```
Duplicates:  0
```

In [37]:
```
orders_df.describe()
```

Out[37]:

	order_id	customer_id	order_status	order
count	99441	99441	99441	
unique	99441	99441	8	
top	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	
freq	1	1	96478	

Untuk tabel ini, lumayan banyak data yang null, semua data kolom waktu juga menggunakan tipe data yang bukan Datetime

Product Category Name Translation

In [38]:
```
product_category_name_translation_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 2 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   product_category_name          71 non-null     object
 1   product_category_name_english  71 non-null     object
dtypes: object(2)
memory usage: 1.2+ KB
```

In [39]:
```
product_category_name_translation_df.isna().sum()
```

Out[39]:
```
product_category_name            0
product_category_name_english    0
dtype: int64
```

In [40]:
```
print("Duplicates: ", product_category_name_translation_df.duplicated().sum())
```

```
Duplicates:  0
```

In [41]:
```
product_category_name_translation_df.describe()
```

Out[41]:

	product_category_name	product_category_name_english
count	71	71
unique	71	71
top	beleza_saude	health_beauty
freq	1	1

Untuk tabel ini tidak ada yang aneh, duplikat, null. Karena tabel ini hanya tabel untuk translasi data saja, atau seperti master table, karena tidak ada yang duplikat/null, tabel ini sudah bisa digunakan

Products

In [42]:
```
products_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32951 entries, 0 to 32950
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   product_id                  32951 non-null  object 
 1   product_category_name       32341 non-null  object 
 2   product_name_lenght         32341 non-null  float64
 3   product_description_lenght  32341 non-null  float64
 4   product_photos_qty          32341 non-null  float64
 5   product_weight_g            32949 non-null  float64
 6   product_length_cm           32949 non-null  float64
 7   product_height_cm           32949 non-null  float64
 8   product_width_cm            32949 non-null  float64
dtypes: float64(7), object(2)
memory usage: 2.3+ MB
```

In [43]:
```
products_df.isna().sum()
```

Out[43]:
```
product_id                    0
product_category_name       610
product_name_lenght         610
product_description_lenght  610
product_photos_qty          610
product_weight_g              2
product_length_cm             2
product_height_cm             2
product_width_cm              2
dtype: int64
```

In [44]:
```
print("Duplicates: ", products_df.duplicated().sum())
```

```
Duplicates:  0
```

In [45]:
```
products_df.describe()
```

Out[45]:

	product_name_lenght	product_description_lenght	product_photos_qty	product_weight_g	p
count	32341.000000	32341.000000	32341.000000	32949.000000	
mean	48.476949	771.495285	2.188986	2276.472488	
std	10.245741	635.115225	1.736766	4282.038731	
min	5.000000	4.000000	1.000000	0.000000	
25%	42.000000	339.000000	1.000000	300.000000	
50%	51.000000	595.000000	1.000000	700.000000	
75%	57.000000	972.000000	3.000000	1900.000000	
max	76.000000	3992.000000	20.000000	40425.000000	

Untuk tabel ini karena berisi tentang informasi tambahan suatu produk. Kecuali saya membutuhkan suatu informasi spesifik ke suatu produk, tabel ini tidak terlalu berguna. Tabel ini juga memiliki null values di semua kolom kecuali product_id

Sellers

In [46]:
```
sellers_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3095 entries, 0 to 3094
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   seller_id               3095 non-null   object
 1   seller_zip_code_prefix  3095 non-null   int64 
 2   seller_city             3095 non-null   object
 3   seller_state            3095 non-null   object
dtypes: int64(1), object(3)
memory usage: 96.8+ KB
```

In [47]:
```
sellers_df.isna().sum()
```

Out[47]:
```
seller_id                 0
seller_zip_code_prefix    0
seller_city               0
seller_state              0
dtype: int64
```

In [48]:
```
print("Duplicates: ", sellers_df.duplicated().sum())
```

```
Duplicates:  0
```

In [49]:
```
sellers_df.describe()
```

Out[49]:

	seller_zip_code_prefix
count	3095.000000
mean	32291.059451
std	32713.453830
min	1001.000000
25%	7093.500000
50%	14940.000000
75%	64552.500000
max	99730.000000

tabel ini hanya berisikan informasi tambahan seller yaitu lokasinya. Tidak ada data yang null dan duplikat. Akan tetapi, kecuali saya menginginkan suatu informasi spesifik ke suatu seller, data dari tabel ini tidak berguna

Berikut adalah chart hasil proses Assessing Data

	Tipe Data	Missing Value	Duplicate Data	Inaccurate value
Customer	-	-	-	-
Geolocation	-	-	261831	-
Order Items	Shipping_limit_date harusnya jadi datetime	-	-	-
Order Payments	-	-	-	-
Order Reviews	Review_creation_date, review_answer_timestamp harusnya pake datetime	di kolom review_comment_title, review_comment_message	-	-
Orders	Order_purchase_timestamp, Order_approved_at, order_delivered_carrier_date, Order_delivered_customer_date, Order_estimated_delivery_date Semua diatas harus pake datetime	di kolom order_approved_at, order_delivered_carrier_date, order_delivered_customer_date	-	-
Product Category Name Translation	-	-	-	-
Products	-	Semuanya ada missing values kecuali di product_id	-	-
Sellers	-	-	-	-

Cleaning Data

Geolocation

In [50]:
```
print("Duplicates: ", geolocation_df.duplicated().sum())
```

```
Duplicates:  261831
```

In [51]:
```
geolocation_df.drop_duplicates(inplace=True)
```

In [52]:
```
print("Duplicates: ", geolocation_df.duplicated().sum())
```

```
Duplicates:  0
```

Duplicate data di tabel geolocation_df sudah hilang dan tabel sudah bisa digunakan

Order Items

In [53]:
```
order_items_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 7 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   order_id             112650 non-null  object 
 1   order_item_id        112650 non-null  int64  
 2   product_id           112650 non-null  object 
 3   seller_id            112650 non-null  object 
 4   shipping_limit_date  112650 non-null  object 
 5   price                112650 non-null  float64
 6   freight_value        112650 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 6.0+ MB
```

In [54]:
```
datetime_columns = ["shipping_limit_date"]

for column in datetime_columns:
  order_items_df[column] = pd.to_datetime(order_items_df[column])
```

In [55]:
```
order_items_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 7 columns):
 #   Column               Non-Null Count   Dtype         
---  ------               --------------   -----         
 0   order_id             112650 non-null  object        
 1   order_item_id        112650 non-null  int64         
 2   product_id           112650 non-null  object        
 3   seller_id            112650 non-null  object        
 4   shipping_limit_date  112650 non-null  datetime64[ns]
 5   price                112650 non-null  float64       
 6   freight_value        112650 non-null  float64       
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 6.0+ MB
```

kolom shipping_limit_date sudah terubah menjadi tipe data datetime. Tabel sudah bisa digunakan

Order Reviews

In [56]:
```
order_reviews_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   review_id                99224 non-null  object
 1   order_id                 99224 non-null  object
 2   review_score             99224 non-null  int64 
 3   review_comment_title     11568 non-null  object
 4   review_comment_message   40977 non-null  object
 5   review_creation_date     99224 non-null  object
 6   review_answer_timestamp  99224 non-null  object
dtypes: int64(1), object(6)
memory usage: 5.3+ MB
```

untuk tabel Order Reviews, saya hanya akan menggunakan bagian pentingnya saja. untuk review, sebenarnya review_score itu sudah cukup dalam mengolah data. Untuk review_comment_title dan review_comment_message bisa diwakili oleh review_score

In [57]:
```
# drop kolom data yang tidak digunakan

order_reviews_df = order_reviews_df.drop(['review_comment_title','review_commer
```

In [58]:
```
# Mengubah kolom "review_creation_date" dan "review_answer_timestamp" ke tipe d

datetime_columns = ['review_creation_date','review_answer_timestamp']

for column in datetime_columns:
  order_reviews_df[column] = pd.to_datetime(order_reviews_df[column])
```

In [59]:
```
order_reviews_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 5 columns):
 #   Column                   Non-Null Count  Dtype         
---  ------                   --------------  -----         
 0   review_id                99224 non-null  object        
 1   order_id                 99224 non-null  object        
 2   review_score             99224 non-null  int64         
 3   review_creation_date     99224 non-null  datetime64[ns]
 4   review_answer_timestamp  99224 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(2)
memory usage: 3.8+ MB
```

Tabel order reviews sudah bisa digunakan

Orders

In [60]:
```
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   order_id                       99441 non-null  object
 1   customer_id                    99441 non-null  object
 2   order_status                   99441 non-null  object
 3   order_purchase_timestamp       99441 non-null  object
 4   order_approved_at              99281 non-null  object
 5   order_delivered_carrier_date   97658 non-null  object
 6   order_delivered_customer_date  96476 non-null  object
 7   order_estimated_delivery_date  99441 non-null  object
dtypes: object(8)
memory usage: 6.1+ MB
```

In [61]:
```
# Mengubah kolom banyak kolom yang memiliki tanggal ke tipe data datetime

datetime_columns = ['order_purchase_timestamp',
                    'order_approved_at',
                    'order_delivered_carrier_date',
                    'order_delivered_customer_date',
                    'order_estimated_delivery_date']

for column in datetime_columns:
  orders_df[column] = pd.to_datetime(orders_df[column])
```

In [62]:
```
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       99441 non-null  object        
 1   customer_id                    99441 non-null  object        
 2   order_status                   99441 non-null  object        
 3   order_purchase_timestamp       99441 non-null  datetime64[ns]
 4   order_approved_at              99281 non-null  datetime64[ns]
 5   order_delivered_carrier_date   97658 non-null  datetime64[ns]
 6   order_delivered_customer_date  96476 non-null  datetime64[ns]
 7   order_estimated_delivery_date  99441 non-null  datetime64[ns]
dtypes: datetime64[ns](5), object(3)
memory usage: 6.1+ MB
```

In [63]:
```
orders_df.head(5)
```

Out[63]:

	order_id	customer_id	order_status	order_pur
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2(
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2(
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2(
3	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2(
4	ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2(

In [64]:
```
# karena saya ingin mengolah data berdasarkan 2 pertanyaan diatas,
# berarti saya harus mempunyai informasi semua barang yang berhasil dijual
# berarti saya hanya membutuhkan data order yang statusnya "delivered" di kolom

orders_df = orders_df[orders_df['order_status'] == 'delivered']
```

In [65]:
```
orders_df.order_status.value_counts()
```

Out[65]:
```
delivered    96478
Name: order_status, dtype: int64
```

In [66]:
```
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96478 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   order_id                       96478 non-null  object
 1   customer_id                    96478 non-null  object
 2   order_status                   96478 non-null  object
 3   order_purchase_timestamp       96478 non-null  datetime64[ns]
 4   order_approved_at              96464 non-null  datetime64[ns]
 5   order_delivered_carrier_date   96476 non-null  datetime64[ns]
 6   order_delivered_customer_date  96470 non-null  datetime64[ns]
 7   order_estimated_delivery_date  96478 non-null  datetime64[ns]
dtypes: datetime64[ns](5), object(3)
memory usage: 6.6+ MB
```

In [67]:
```
orders_df.isna().sum()
```

Out[67]:
```
order_id                        0
customer_id                     0
order_status                    0
order_purchase_timestamp        0
order_approved_at              14
order_delivered_carrier_date    2
order_delivered_customer_date   8
order_estimated_delivery_date   0
dtype: int64
```

karena masih ada yang null, tetapi data yang null tersebut sedikit, saya drop data tersebut

In [68]:
```
orders_df = orders_df.dropna()
```

In [69]:
```
orders_df.isna().sum()
```

Out[69]:
```
order_id                        0
customer_id                     0
order_status                    0
order_purchase_timestamp        0
order_approved_at               0
order_delivered_carrier_date    0
order_delivered_customer_date   0
order_estimated_delivery_date   0
dtype: int64
```

In [70]:
```
orders_df.sort_values(by='order_delivered_customer_date', ascending=False)
```

Out[70]:

	order_id	customer_id	order_status	orde
18731	7e708aed151d6a8601ce8f2eaa712bf4	033fab69968b0d69099d64423831a236	delivered	
56635	450cb96c63e1e5b49d34f223f67976d2	27ae7c8a8fc20ce80d96f01b6f19961b	delivered	
92319	b2997e1d7061605e9285496c581d1fbd	9e83d47684eb1a58b1c31830f5de10ac	delivered	
43810	a2b4be96b53022618030c17ed437604d	ffa87b4246c4848711afb512bd51f161	delivered	
21098	7d09831e67caa193da82cfea3bee7aa5	1409b2945191b7aff1975ba2ce9918c5	delivered	
...	...	...	...	
699	ac2b7c522d811acba0aa270ed3e112e4	ef21aebbb093a6db29ccc6aa0b89c347	delivered	
52382	92b44b87f1f7670b8911c5f0e642435e	e561a3f61440b031d3be286a696d06eb	delivered	
56143	d1eb8e4e276a4eea13a5c462c0765e60	9031f9dcde5860b34e6c65ac5c796d30	delivered	
1384	7033745709b7cf1bac7d2533663592de	7f0ca17bb33b230b47459437cf0682c7	delivered	
59102	36989eb07a0de2d3d3129eea35553875	aadd27185177fc7ac9b364898ac09343	delivered	

96455 rows × 8 columns

untuk tabel orders seharusnya sudah bisa digunakan

Products

In [71]:
```
products_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32951 entries, 0 to 32950
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   product_id                  32951 non-null  object 
 1   product_category_name       32341 non-null  object 
 2   product_name_lenght         32341 non-null  float64
 3   product_description_lenght  32341 non-null  float64
 4   product_photos_qty          32341 non-null  float64
 5   product_weight_g            32949 non-null  float64
 6   product_length_cm           32949 non-null  float64
 7   product_height_cm           32949 non-null  float64
 8   product_width_cm            32949 non-null  float64
dtypes: float64(7), object(2)
memory usage: 2.3+ MB
```

In [72]:
```
products_df.isna().sum()
```

Out[72]:
```
product_id                    0
product_category_name       610
product_name_lenght         610
product_description_lenght  610
product_photos_qty          610
product_weight_g              2
product_length_cm             2
product_height_cm             2
product_width_cm              2
dtype: int64
```

tabel ini tidak terlalu berguna untuk menjawab pertanyaan diatas. jadi akan saya drop semua data null sebagai cleaning proses cleaning

In [73]:
```
products_df = products_df.dropna()
```

In [74]:
```
products_df.isna().sum()
```

Out[74]:
```
product_id                  0
product_category_name       0
product_name_lenght         0
product_description_lenght  0
product_photos_qty          0
product_weight_g            0
product_length_cm           0
product_height_cm           0
product_width_cm            0
dtype: int64
```

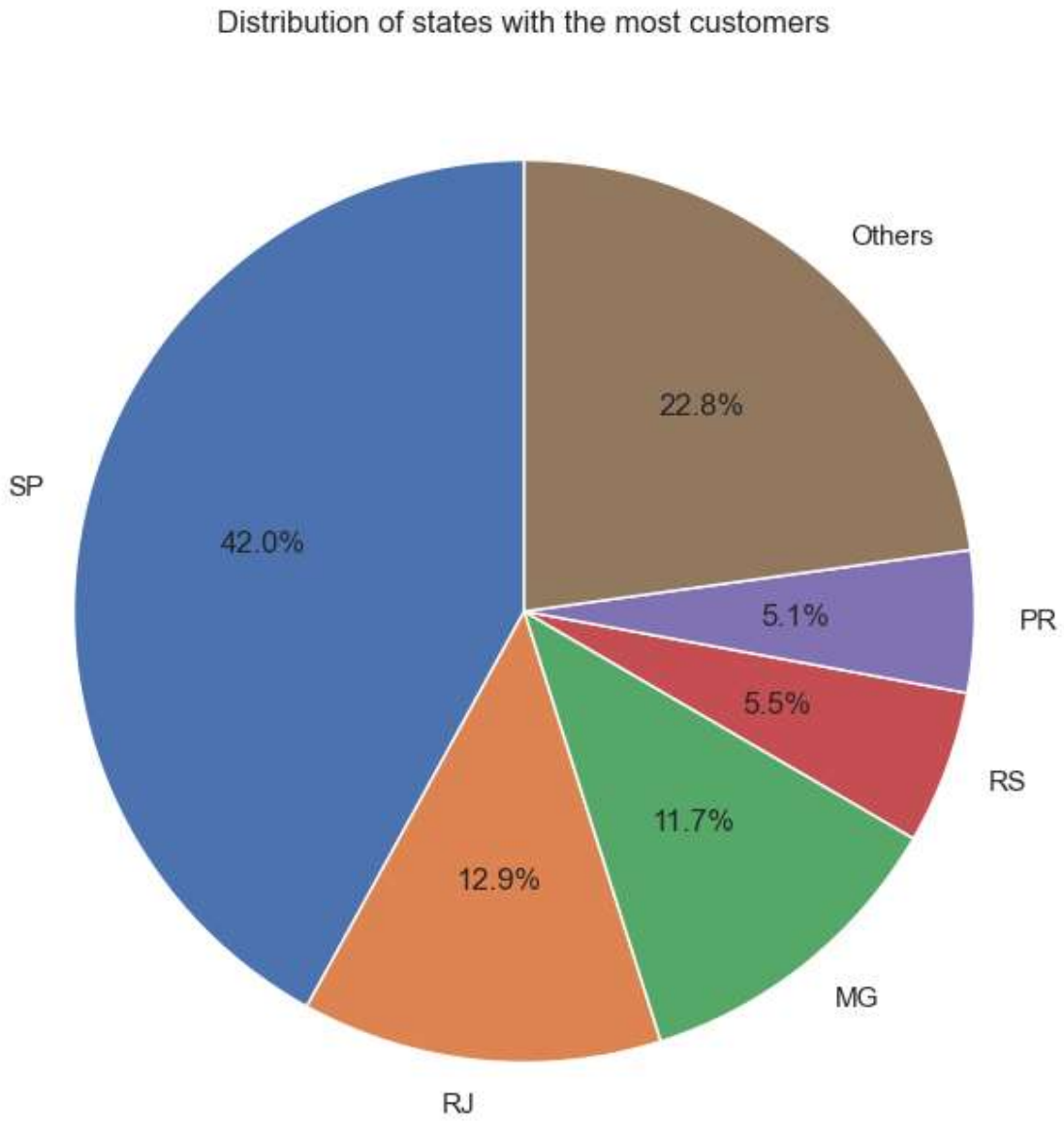
In [75]:
```
products_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32340 entries, 0 to 32950
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   product_id                  32340 non-null  object 
 1   product_category_name       32340 non-null  object 
 2   product_name_lenght         32340 non-null  float64
 3   product_description_lenght  32340 non-null  float64
 4   product_photos_qty          32340 non-null  float64
 5   product_weight_g            32340 non-null  float64
 6   product_length_cm           32340 non-null  float64
 7   product_height_cm           32340 non-null  float64
 8   product_width_cm            32340 non-null  float64
dtypes: float64(7), object(2)
memory usage: 2.5+ MB
```

Exploratory Data Analysis (EDA)

Customer

In [76]:
```
customer_state_counts = customers_df['customer_state'].value_counts()

top_cities = customer_state_counts.head(5)
top_cities['Others'] = customer_state_counts.iloc[5:].sum()

plt.figure(figsize=(8, 8))
plt.pie(top_cities, labels=top_cities.index, autopct='%1.1f%%', startangle=90)
plt.title('Distribution of states with the most customers')
plt.show()
```



Dari pie chart ini kita bisa tahu bahwa customer paling banyak berasal dari state dengan inisial SP

In [77]:
```
customers_df.customer_state.value_counts()
```

Out[77]:
```
SP    41746
RJ    12852
MG    11635
RS     5466
PR     5045
SC     3637
BA     3380
DF     2140
ES     2033
GO     2020
PE     1652
CE     1336
PA      975
MT      907
MA      747
MS      715
PB      536
PI      495
RN      485
AL      413
SE      350
TO      280
RO      253
AM      148
AC       81
AP       68
RR       46
Name: customer_state, dtype: int64
```

Bisa dilihat dari sini bahwa sebagian besar customer yang terdaftar berasal dari state yang berinisial SP

Customers dan Orders

Karena jumlah customernya banyak sekali, melebihi jumlah ordernya bahkan, berarti ada banyak customer yang hanya terdaftar saja di database dan tidak/belum melakukan pembelian sama sekali. Seberapa banyak tapi?

In [78]:
```
customer_id_in_orders_df =  orders_df.customer_id.tolist()
customers_df["status"] = customers_df["customer_id"].apply(lambda x: "Active" i
customers_df.sample(5)
```

Out[78]:

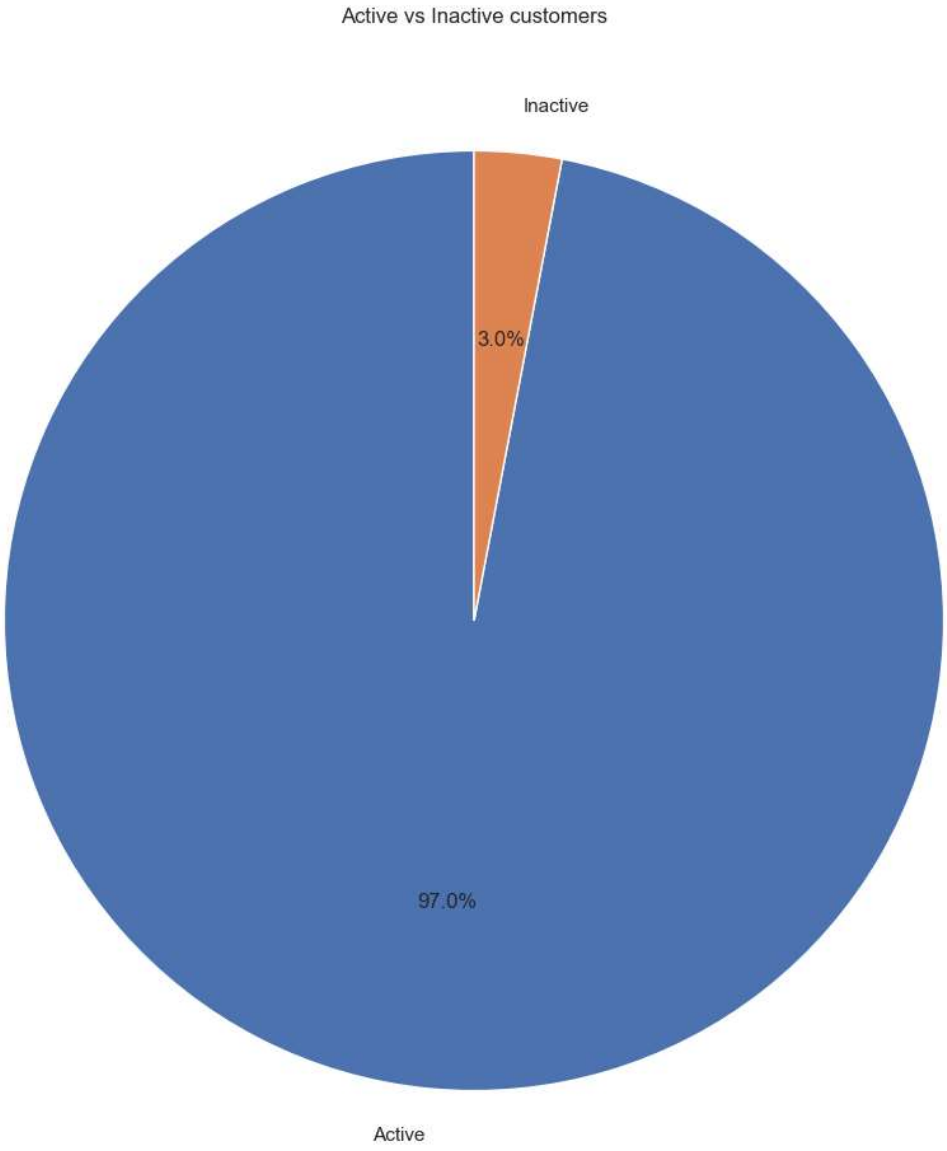
	customer_id	customer_unique_id	customer_zip_code
41218	847913f5809f26d34e7787f48094b934	03e8a26864627ee0c7fb470b46cbaee7	
87811	1392bf4ea4156b566c869594d1ce5a59	6b8b906b71b6bfd0b17eae63ea0dbeba	
62764	c98bc4064c2da2e8f87a2476bc07a5b0	c89e2005ddcd889c687343e2a7d7a115	
93845	800512d0215633d567abe5ce2591f01d	e3c95cff4b8cca496061b369eaa006f7	
94820	7ad257dbeec09b715a8c65fc2fe83a33	4f4d2fe2987efa0b9e4e61a466e8ed8b	

In [79]:
```
customers_df.groupby(by="status").customer_id.count()
```

Out[79]:
```
status
Active      96455
Inactive     2986
Name: customer_id, dtype: int64
```

benar, berarti ada lumayan banyak customer yang terdaftar, tetapi tidak melakukan transaksi

In [80]:
```
customers_active = customers_df.groupby(by="status").customer_id.count()

plt.figure(figsize=(12, 12))
plt.pie(customers_active, labels=customers_active.index, autopct='%1.1f%%', sta
plt.title('Active vs Inactive customers')
plt.show()
```



Dari pie chart ini kita bisa melihat bahwa memang banyak sekali customer yang terdaftar dan aktif(97%) dibandingkan customer yang terdaftar tetapi tidak aktif. Maksud dari aktif disini adalah pernah membeli suatu barang. dan

In [81]:
```
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96455 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       96455 non-null  object        
 1   customer_id                    96455 non-null  object        
 2   order_status                   96455 non-null  object        
 3   order_purchase_timestamp       96455 non-null  datetime64[ns]
 4   order_approved_at              96455 non-null  datetime64[ns]
 5   order_delivered_carrier_date   96455 non-null  datetime64[ns]
 6   order_delivered_customer_date  96455 non-null  datetime64[ns]
 7   order_estimated_delivery_date  96455 non-null  datetime64[ns]
dtypes: datetime64[ns](5), object(3)
memory usage: 6.6+ MB
```

In [82]:
```
orders_df.customer_id.value_counts()
```

Out[82]:
```
9ef432eb6251297304e76186b10a928d    1
110b79f06a0f49a38da99084706a382d    1
c840e43d3d57dbb4e99374570f2488cd    1
96c6a3143d7cc33bbebf14ab6abed0ce    1
eb4350b67a0264c67e5e06a038e4afbb    1
                                   ..
7db5bde0f0fcee8817a4c317cf05429a    1
0b8520e0d24d4e14482e01e73e5740c0    1
a33988fde632872a5c5458823b9c2d01    1
42f104a41e8e13c7eaaa9d8e11e7c7f5    1
edb027a75a1449115f6b43211ae02a24    1
Name: customer_id, Length: 96455, dtype: int64
```

ok dari hasil value_counts customer_id di tabel orders_df saya menemukan bahwa setelah setiap customer yang active hanya melakukan transaksi maksimal sekali

Orders dengan Payment (liat metode bayar paling populer)

In [83]:
```
orders_payment_df = pd.merge(
    left=orders_df,
    right=order_payments_df,
    how="left",
    left_on="order_id",
    right_on="order_id"
)
orders_payment_df.head()
```

Out[83]:

	order_id	customer_id	order_status	order_pur
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2(
1	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2(
2	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2(
3	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2(
4	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2(

In [84]:
```
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96455 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   order_id                       96455 non-null  object
 1   customer_id                    96455 non-null  object
 2   order_status                   96455 non-null  object
 3   order_purchase_timestamp       96455 non-null  datetime64[ns]
 4   order_approved_at              96455 non-null  datetime64[ns]
 5   order_delivered_carrier_date   96455 non-null  datetime64[ns]
 6   order_delivered_customer_date  96455 non-null  datetime64[ns]
 7   order_estimated_delivery_date  96455 non-null  datetime64[ns]
dtypes: datetime64[ns](5), object(3)
memory usage: 6.6+ MB
```

In [85]:
```
orders_payment_df.order_id.value_counts()
```

Out[85]:
```
ccf804e764ed5650cd8759557269dc13    26
285c2e15bebd4ac83635ccc563dc71f4    22
895ab968e7bb0d5659d16cd74cd1650c    21
ee9ca989fc93ba09a6eddc250ce01742    19
fedcd9f7ccdc8cba3a18defedd1a5547    19
                                    ..
bd9e8d4cec0e721d5ffebe153035eeea     1
0866001f9f2ff47df177b77ce6abecd4     1
baaed1de8d45224736f9660579d8cc97     1
df111b726761108ddc8bd410a4d54b09     1
66dea50a8b16d9b4dee7af250b4be1a5     1
Name: order_id, Length: 96455, dtype: int64
```

In [86]:
```
print("Duplicate Data: ", orders_payment_df.order_id.duplicated().sum())
```

```
Duplicate Data:  4279
```

In [87]:
```
orders_payment_df = orders_payment_df.drop_duplicates(subset='order_id', keep='
```

In [88]:
```
orders_payment_df.drop_duplicates(inplace=True)
```

In [89]:
```
orders_payment_df.info()
```

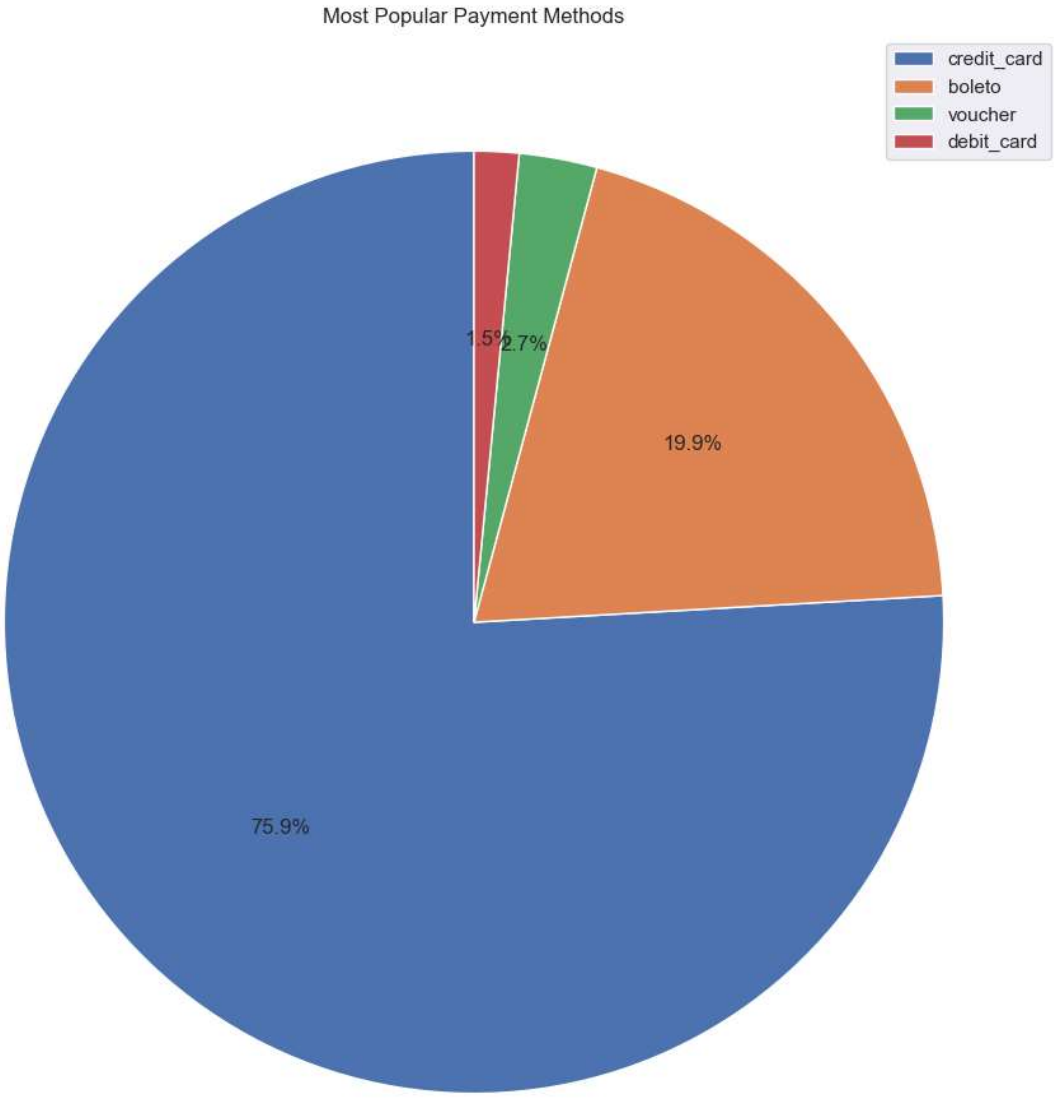
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96455 entries, 0 to 100733
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       96455 non-null  object        
 1   customer_id                    96455 non-null  object        
 2   order_status                   96455 non-null  object        
 3   order_purchase_timestamp       96455 non-null  datetime64[ns]
 4   order_approved_at              96455 non-null  datetime64[ns]
 5   order_delivered_carrier_date   96455 non-null  datetime64[ns]
 6   order_delivered_customer_date  96455 non-null  datetime64[ns]
 7   order_estimated_delivery_date  96455 non-null  datetime64[ns]
 8   payment_sequential             96454 non-null  float64       
 9   payment_type                   96454 non-null  object        
 10  payment_installments           96454 non-null  float64       
 11  payment_value                  96454 non-null  float64       
dtypes: datetime64[ns](5), float64(3), object(4)
memory usage: 9.6+ MB
```

OK. yang saya baru lakukan adalah join tabel orders_df dan order_payments_df. Tujuannya adalah untuk mencari tahu metode pembayaran yang paling populer dipakai oleh customer Akan tetapi, ada masalah yaitu beberapa order_idnya menjadi terulang, karena itu saya buang data order id yang duplikat. Sekarang saya bisa mencari tahu metode pembayaran paling populer dan memvisualisasikannya dengan pie chart

In [90]:
```
orders_payment_df['payment_type'].value_counts()
```

Out[90]:
```
credit_card     73212
boleto          19177
voucher          2582
debit_card       1483
Name: payment_type, dtype: int64
```

In [91]:
```
payment_methods = orders_payment_df['payment_type'].value_counts()

payment_methods = payment_methods.head(4)

plt.figure(figsize=(12, 12))
plt.pie(payment_methods, autopct='%1.1f%%', startangle=90)
plt.legend(payment_methods.index, loc="best")
plt.title('Most Popular Payment Methods')
plt.show()
```



Jadi, berdasarkan piechart tersebut, lebih dari 75% transaksi sukses memakai metode pembayaran "credit_card". Kedua adalah "boleto", Boleto sendiri adalah salah satu tipe pembayaran yang bisa ditemui di Brazil dan berbentuk semacam invoice. Bisa dilihat disini https://en.wikipedia.org/wiki/Boleto (https://en.wikipedia.org/wiki/Boleto). Tipe pembayaran Voucher dan Debit ternyata kurang populer

Orders dengan Order_Reviews (liat rating paling banyak)

Jadi, saya ingin tahu frekuensi tiap skor yang muncul untuk tiap barang di tabel orders

In [92]:
```
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96455 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       96455 non-null  object        
 1   customer_id                    96455 non-null  object        
 2   order_status                   96455 non-null  object        
 3   order_purchase_timestamp       96455 non-null  datetime64[ns]
 4   order_approved_at              96455 non-null  datetime64[ns]
 5   order_delivered_carrier_date   96455 non-null  datetime64[ns]
 6   order_delivered_customer_date  96455 non-null  datetime64[ns]
 7   order_estimated_delivery_date  96455 non-null  datetime64[ns]
dtypes: datetime64[ns](5), object(3)
memory usage: 6.6+ MB
```

In [93]:
```
order_reviews_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 5 columns):
 #   Column                   Non-Null Count  Dtype         
---  ------                   --------------  -----         
 0   review_id                99224 non-null  object        
 1   order_id                 99224 non-null  object        
 2   review_score             99224 non-null  int64         
 3   review_creation_date     99224 non-null  datetime64[ns]
 4   review_answer_timestamp  99224 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(2)
memory usage: 3.8+ MB
```

In [94]:
```
orders_ratings_df = pd.merge(
    left=orders_df,
    right=order_reviews_df,
    how="left",
    left_on="order_id",
    right_on="order_id"
)
orders_ratings_df.head()
```

Out[94]:

	order_id	customer_id	order_status	order_pur
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2(
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2(
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2(
3	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2(
4	ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2(

In [95]:
```
orders_ratings_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 96984 entries, 0 to 96983
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   order_id                       96984 non-null  object
 1   customer_id                    96984 non-null  object
 2   order_status                   96984 non-null  object
 3   order_purchase_timestamp       96984 non-null  datetime64[ns]
 4   order_approved_at              96984 non-null  datetime64[ns]
 5   order_delivered_carrier_date   96984 non-null  datetime64[ns]
 6   order_delivered_customer_date  96984 non-null  datetime64[ns]
 7   order_estimated_delivery_date  96984 non-null  datetime64[ns]
 8   review_id                      96338 non-null  object
 9   review_score                   96338 non-null  float64
 10  review_creation_date           96338 non-null  datetime64[ns]
 11  review_answer_timestamp        96338 non-null  datetime64[ns]
dtypes: datetime64[ns](7), float64(1), object(4)
memory usage: 9.6+ MB
```

In [96]:
```
orders_ratings_df = orders_ratings_df.drop_duplicates(subset='order_id', keep='
```

In [97]:
```
print("Duplicate Data: ", orders_ratings_df.order_id.duplicated().sum())
```

```
Duplicate Data:  0
```

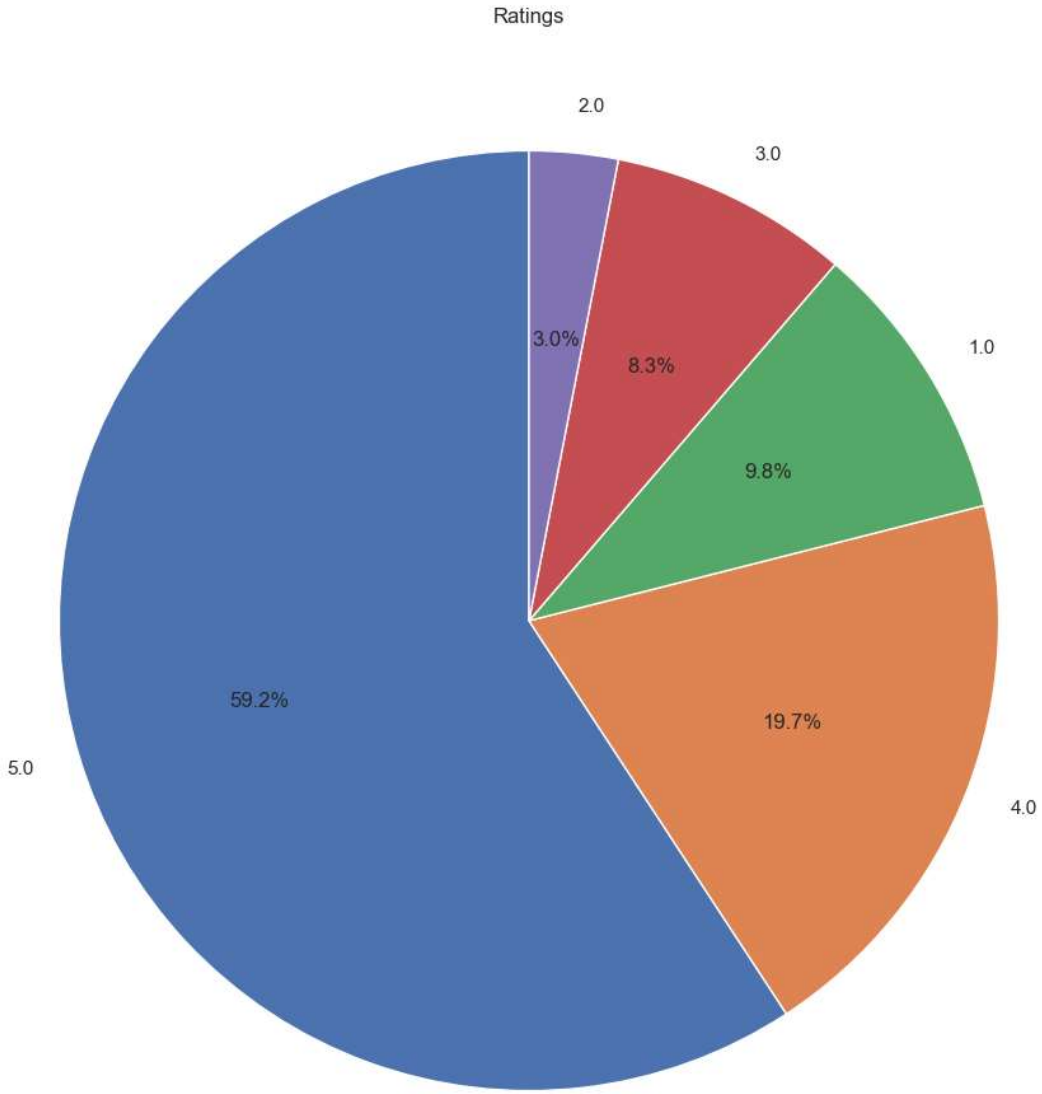
In [98]:
```
orders_ratings_df = orders_ratings_df.dropna()
```

In [99]:
```
orders_ratings_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 95809 entries, 0 to 96983
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       95809 non-null  object        
 1   customer_id                    95809 non-null  object        
 2   order_status                   95809 non-null  object        
 3   order_purchase_timestamp       95809 non-null  datetime64[ns]
 4   order_approved_at              95809 non-null  datetime64[ns]
 5   order_delivered_carrier_date   95809 non-null  datetime64[ns]
 6   order_delivered_customer_date  95809 non-null  datetime64[ns]
 7   order_estimated_delivery_date  95809 non-null  datetime64[ns]
 8   review_id                      95809 non-null  object        
 9   review_score                   95809 non-null  float64       
 10  review_creation_date           95809 non-null  datetime64[ns]
 11  review_answer_timestamp        95809 non-null  datetime64[ns]
dtypes: datetime64[ns](7), float64(1), object(4)
memory usage: 9.5+ MB
```

sekarang kedua data dari tabel orders dan order_reviews sudah di left join. Sebenarnya bisa saja langsung melihat value counts dari tabel order_reviews_df. tetapi saya ingin agar datanya lebih akurat, yaitu data yang sukses terkirim saja

In [100]:
```
scores = orders_ratings_df['review_score'].value_counts()

plt.figure(figsize=(12, 12))
plt.pie(scores, labels=scores.index, autopct='%1.1f%%', startangle=90)
plt.title('Ratings')
plt.show()
```

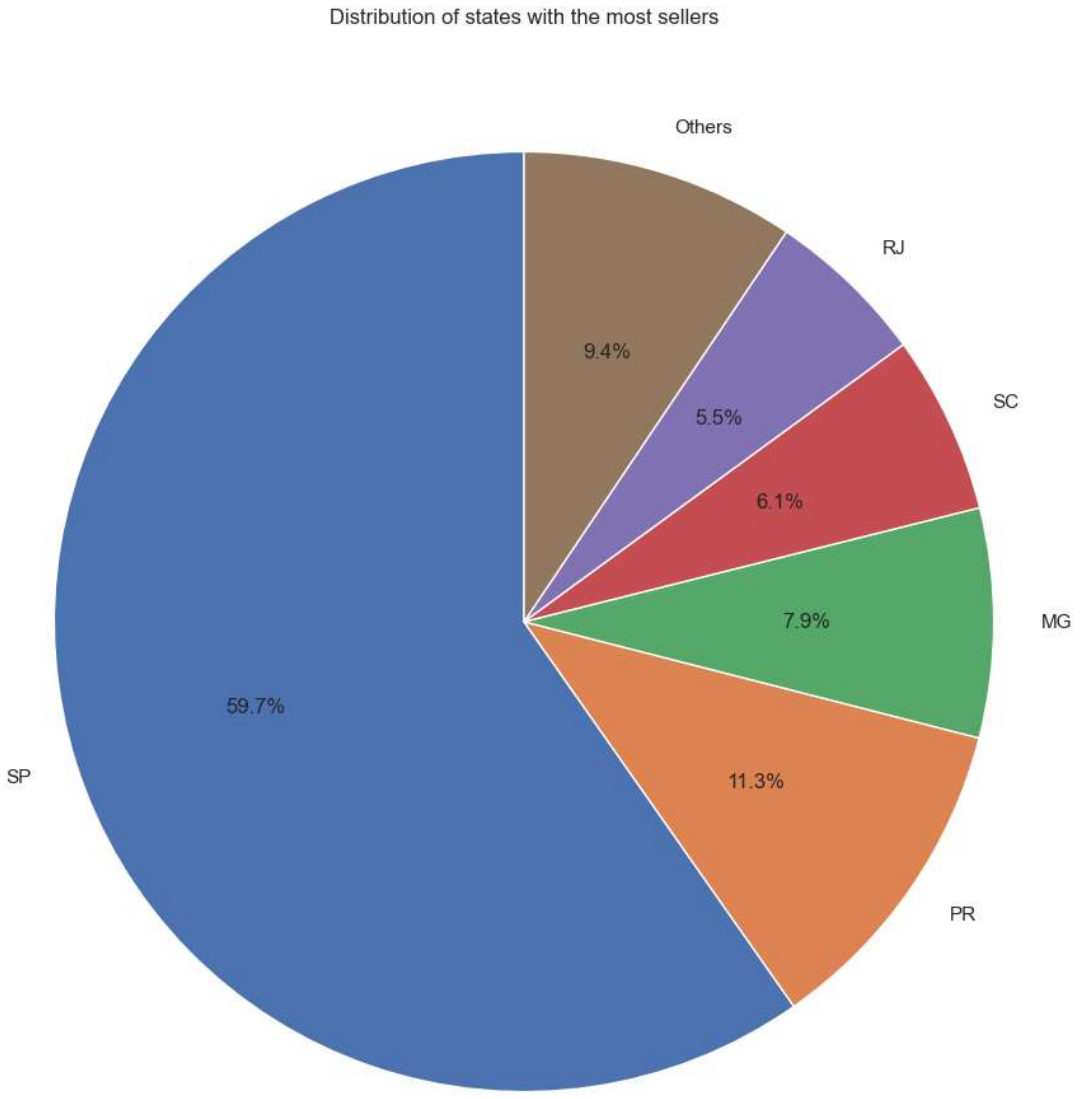


Dari piechart diatas, saya mengetahui bahwa banyak customer sebagian besar sangat puas dengan barang yang diterimanya dengan memberikan skor review 5 atau 4. Ada banyak customer yang memberikan skor 1. Customer jarang memberikan skor 3 dan 2

Seller

Saya ingin tahu 5 lokasi dengan seller terbanyak

In [101]:
```
sellerr_state_counts = sellers_df['seller_state'].value_counts()

top_cities = sellerr_state_counts.head(5)
top_cities['Others'] = sellerr_state_counts.iloc[5:].sum()

plt.figure(figsize=(12, 12))
plt.pie(top_cities, labels=top_cities.index, autopct='%1.1f%%', startangle=90)
plt.title('Distribution of states with the most sellers')
plt.show()
```



Jadi seperti customer, lokasi seller banyak terletak di state berinisial SP

Visualization & Explanatory Analysis

Interval data yang akan dipakai untuk menjawap pertanyaan adalah data tahun 2017 karena tahun 2017 ada lengkap dari bulan Januari hingga Desember

In [102]:
```
most_recent_date = orders_df['order_purchase_timestamp'].max()

oldest_date = orders_df['order_purchase_timestamp'].min()

print('most_recent_date: ' + str(most_recent_date))
print('oldest_date: ' + str(oldest_date))
```

```
most_recent_date: 2018-08-29 15:00:37
oldest_date: 2016-09-15 12:16:38
```

In [103]:
```
orders_ratings_df_2017 = orders_ratings_df[orders_ratings_df['order_purchase_ti
```

In [104]:
```
orders_ratings_df_2017.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43083 entries, 0 to 96981
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       43083 non-null  object        
 1   customer_id                    43083 non-null  object        
 2   order_status                   43083 non-null  object        
 3   order_purchase_timestamp       43083 non-null  datetime64[ns]
 4   order_approved_at              43083 non-null  datetime64[ns]
 5   order_delivered_carrier_date   43083 non-null  datetime64[ns]
 6   order_delivered_customer_date  43083 non-null  datetime64[ns]
 7   order_estimated_delivery_date  43083 non-null  datetime64[ns]
 8   review_id                      43083 non-null  object        
 9   review_score                   43083 non-null  float64       
 10  review_creation_date           43083 non-null  datetime64[ns]
 11  review_answer_timestamp        43083 non-null  datetime64[ns]
dtypes: datetime64[ns](7), float64(1), object(4)
memory usage: 4.3+ MB
```

In [105]:
```
most_recent_date = orders_ratings_df_2017['order_purchase_timestamp'].max()

oldest_date = orders_ratings_df_2017['order_purchase_timestamp'].min()

print('most_recent_date: ' + str(most_recent_date))
print('oldest_date: ' + str(oldest_date))
```

```
most_recent_date: 2017-12-31 23:29:31
oldest_date: 2017-01-05 11:56:06
```

In [106]:
```
order_items_for_processing=order_items_df
```

In [107]:
```
order_items_for_processing = order_items_for_processing.drop(['order_item_id'],
```

In [108]:
```
order_items_for_processing = order_items_for_processing.drop(['price'], axis=1)
```

In [109]:
```
order_items_for_processing = order_items_for_processing.drop(['freight_value'],
```

In [110]:
```
order_items_for_processing = order_items_for_processing.drop(['shipping_limit_d
```

In [111]:
```
order_items_for_processing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 3 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   order_id    112650 non-null  object
 1   product_id  112650 non-null  object
 2   seller_id   112650 non-null  object
dtypes: object(3)
memory usage: 2.6+ MB
```

In [112]:
```
order_items_for_processing.drop_duplicates(inplace=True)
```

In [113]:
```
order_items_for_processing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 102425 entries, 0 to 112649
Data columns (total 3 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   order_id    102425 non-null  object
 1   product_id  102425 non-null  object
 2   seller_id   102425 non-null  object
dtypes: object(3)
memory usage: 3.1+ MB
```

In [114]:
```
part_df = pd.merge(orders_ratings_df_2017, order_items_for_processing, on='orde
```

In [115]:
```
part_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 44773 entries, 0 to 44772
Data columns (total 14 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       44773 non-null  object        
 1   customer_id                    44773 non-null  object        
 2   order_status                   44773 non-null  object        
 3   order_purchase_timestamp       44773 non-null  datetime64[ns]
 4   order_approved_at              44773 non-null  datetime64[ns]
 5   order_delivered_carrier_date   44773 non-null  datetime64[ns]
 6   order_delivered_customer_date  44773 non-null  datetime64[ns]
 7   order_estimated_delivery_date  44773 non-null  datetime64[ns]
 8   review_id                      44773 non-null  object        
 9   review_score                   44773 non-null  float64       
 10  review_creation_date           44773 non-null  datetime64[ns]
 11  review_answer_timestamp        44773 non-null  datetime64[ns]
 12  product_id                     44773 non-null  object        
 13  seller_id                      44773 non-null  object        
dtypes: datetime64[ns](7), float64(1), object(6)
memory usage: 5.1+ MB
```

In [116]:
```
print("Duplicates: ", part_df.order_id.duplicated().sum())
```

```
Duplicates:  1690
```

In [117]:
```
part_df = part_df.drop_duplicates(subset='order_id', keep='first')
```

In [118]:
```
print("Duplicates: ", part_df.order_id.duplicated().sum())
```

```
Duplicates:  0
```

In [119]:
```
part_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43083 entries, 0 to 44772
Data columns (total 14 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       43083 non-null  object        
 1   customer_id                    43083 non-null  object        
 2   order_status                   43083 non-null  object        
 3   order_purchase_timestamp       43083 non-null  datetime64[ns]
 4   order_approved_at              43083 non-null  datetime64[ns]
 5   order_delivered_carrier_date   43083 non-null  datetime64[ns]
 6   order_delivered_customer_date  43083 non-null  datetime64[ns]
 7   order_estimated_delivery_date  43083 non-null  datetime64[ns]
 8   review_id                      43083 non-null  object        
 9   review_score                   43083 non-null  float64       
 10  review_creation_date           43083 non-null  datetime64[ns]
 11  review_answer_timestamp        43083 non-null  datetime64[ns]
 12  product_id                     43083 non-null  object        
 13  seller_id                      43083 non-null  object        
dtypes: datetime64[ns](7), float64(1), object(6)
memory usage: 4.9+ MB
```

In [120]:
```
part_df.head()
```

Out[120]:

	order_id	customer_id	order_status	order_pu
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2
1	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2
2	a4591c265e18cb1dcee52889e2d8acc3	503740e9ca751ccdda7ba28e9ab8f608	delivered	2
3	6514b8ad8028c9f2cc2374ded245783f	9bdf08b4b3b52b5526ff42d37d47f222	delivered	2
4	76c6e866289321a7c93b82b54852dc33	f54a9f0e6b351c431402b8461ea51999	delivered	2

In [121]:
```
final_df = pd.merge(part_df, order_payments_df, on='order_id')
```

In [122]:
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 45367 entries, 0 to 45366
Data columns (total 18 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       45367 non-null  object        
 1   customer_id                    45367 non-null  object        
 2   order_status                   45367 non-null  object        
 3   order_purchase_timestamp       45367 non-null  datetime64[ns]
 4   order_approved_at              45367 non-null  datetime64[ns]
 5   order_delivered_carrier_date   45367 non-null  datetime64[ns]
 6   order_delivered_customer_date  45367 non-null  datetime64[ns]
 7   order_estimated_delivery_date  45367 non-null  datetime64[ns]
 8   review_id                      45367 non-null  object        
 9   review_score                   45367 non-null  float64       
 10  review_creation_date           45367 non-null  datetime64[ns]
 11  review_answer_timestamp        45367 non-null  datetime64[ns]
 12  product_id                     45367 non-null  object        
 13  seller_id                      45367 non-null  object        
 14  payment_sequential             45367 non-null  int64         
 15  payment_type                   45367 non-null  object        
 16  payment_installments           45367 non-null  int64         
 17  payment_value                  45367 non-null  float64       
dtypes: datetime64[ns](7), float64(2), int64(2), object(7)
memory usage: 6.6+ MB
```

In [123]:
```
final_df = final_df.drop_duplicates(subset='order_id', keep='first')
```

In [124]:
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43083 entries, 0 to 45366
Data columns (total 18 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   order_id                       43083 non-null  object        
 1   customer_id                    43083 non-null  object        
 2   order_status                   43083 non-null  object        
 3   order_purchase_timestamp       43083 non-null  datetime64[ns]
 4   order_approved_at              43083 non-null  datetime64[ns]
 5   order_delivered_carrier_date   43083 non-null  datetime64[ns]
 6   order_delivered_customer_date  43083 non-null  datetime64[ns]
 7   order_estimated_delivery_date  43083 non-null  datetime64[ns]
 8   review_id                      43083 non-null  object        
 9   review_score                   43083 non-null  float64       
 10  review_creation_date           43083 non-null  datetime64[ns]
 11  review_answer_timestamp        43083 non-null  datetime64[ns]
 12  product_id                     43083 non-null  object        
 13  seller_id                      43083 non-null  object        
 14  payment_sequential             43083 non-null  int64         
 15  payment_type                   43083 non-null  object        
 16  payment_installments           43083 non-null  int64         
 17  payment_value                  43083 non-null  float64       
dtypes: datetime64[ns](7), float64(2), int64(2), object(7)
memory usage: 6.2+ MB
```

In [125]:
```
final_df.head()
```

Out[125]:

	order_id	customer_id	order_status	order_pu
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2
3	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2
4	a4591c265e18cb1dcee52889e2d8acc3	503740e9ca751ccdda7ba28e9ab8f608	delivered	2
5	6514b8ad8028c9f2cc2374ded245783f	9bdf08b4b3b52b5526ff42d37d47f222	delivered	2
6	76c6e866289321a7c93b82b54852dc33	f54a9f0e6b351c431402b8461ea51999	delivered	2

Penjelasan. Saya menggabungkan order_items_df, order_reviews_df, orders_df, order_payments_df agar bisa mendapatkan data utuhnya. Mengapa saya memilih orders_df sebagai tabel/dataframe utama? karena orders_df merupakan data paling unik dari semuanya. Mengapa saya tidak memilih order_items_df? ternyata order_id di dataframe tersebut tidak unik. Jadinya ada data seperti pembelian barang yang sama sebanyak 21 kali. Lebih mudah jika saya mencari dataframe yang unik, baru menggabungkan total semua pembeliannya dari tabel order_payments_df. Dengan ini, seharusnya data di final_df ini sudah cukup untuk menjawab pertanyaan yang ada.

Pertanyaan 1:

Apa saja 5 kategori barang terlaris di tahun 2017?

Cara saya untuk mendapatkan jawaban ini adalah pertama untuk menyambungkan dataframe products dengan product_category_name_translation saya pertama memilih products karena di tabel ini memiliki product_id yang nanti bisa disambungkan ke final_df lalu saya memilih product_category_name_translation untuk translasi data ini dari bahasa spanyol ke inggris

In [126]:
```
product_category_name_translation_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 2 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   product_category_name          71 non-null     object
 1   product_category_name_english  71 non-null     object
dtypes: object(2)
memory usage: 1.2+ KB
```

In [127]:
```
products_df.head()
```

Out[127]:

	product_id	product_category_name	product_name_lenght	product_des
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	40.0	
1	3aa071139cb16b67ca9e5dea641aaa2f	artes	44.0	
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer	46.0	
3	cef67bcfe19066a932b7673e239eb23d	bebes	27.0	
4	9dc1a7de274444849c219cff195d0b71	utilidades_domesticas	37.0	

In [128]:
```
product_id_names = products_df.iloc[:, :2]
```

In [129]:
```
product_id_names.head()
```

Out[129]:

	product_id	product_category_name
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria
1	3aa071139cb16b67ca9e5dea641aaa2f	artes
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer
3	cef67bcfe19066a932b7673e239eb23d	bebes
4	9dc1a7de274444849c219cff195d0b71	utilidades_domesticas

In [130]:
```
print("Duplicates: ", product_id_names.product_id.duplicated().sum())
```

```
Duplicates:  0
```

In [131]:
```
product_id_names.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32340 entries, 0 to 32950
Data columns (total 2 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   product_id            32340 non-null  object
 1   product_category_name  32340 non-null  object
dtypes: object(2)
memory usage: 758.0+ KB
```

In [132]:
```
product_id_names.product_category_name.value_counts()
```

Out[132]:
```
cama_mesa_banho                  3029
esporte_lazer                    2867
moveis_decoracao                 2657
beleza_saude                     2444
utilidades_domesticas            2335
                                 ... 
fashion_roupa_infanto_juvenil       5
casa_conforto_2                     5
pc_gamer                            3
seguros_e_servicos                  2
cds_dvds_musicais                   1
Name: product_category_name, Length: 73, dtype: int64
```

In [133]:
```
product_translation = pd.merge(
    left=product_id_names,
    right=product_category_name_translation_df,
    how="left",
    left_on="product_category_name",
    right_on="product_category_name"
)
product_translation.head()
```

Out[133]:

	product_id	product_category_name	product_category_name_english
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	perfumery
1	3aa071139cb16b67ca9e5dea641aaa2f	artes	art
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer	sports_leisure
3	cef67bcfe19066a932b7673e239eb23d	bebes	baby
4	9dc1a7de274444849c219cff195d0b71	utilidades_domesticas	housewares

In [134]:
```
product_translation.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32340 entries, 0 to 32339
Data columns (total 3 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   product_id                     32340 non-null  object
 1   product_category_name          32340 non-null  object
 2   product_category_name_english  32327 non-null  object
dtypes: object(3)
memory usage: 1010.6+ KB
```

In [135]:
```
product_translation = product_translation.dropna()
```

In [136]:
```
product_translation.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32327 entries, 0 to 32339
Data columns (total 3 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   product_id                     32327 non-null  object
 1   product_category_name          32327 non-null  object
 2   product_category_name_english  32327 non-null  object
dtypes: object(3)
memory usage: 1010.2+ KB
```

In [137]:
```
final_df = pd.merge(
    left=final_df,
    right=product_translation,
    how="left",
    left_on="product_id",
    right_on="product_id"
)
final_df.head()
```

Out[137]:

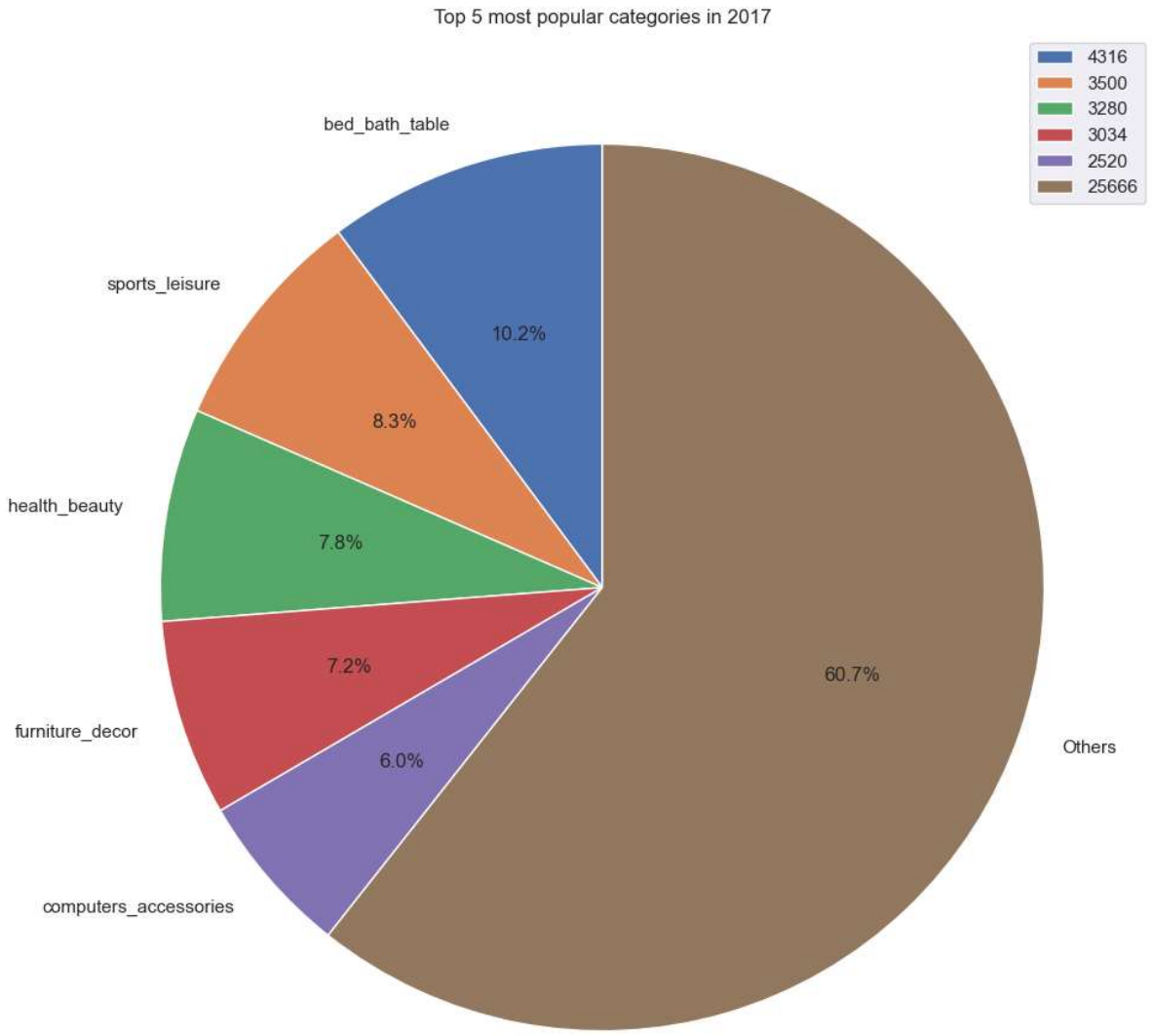
	order_id	customer_id	order_status	order_pu
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2
1	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2
2	a4591c265e18cb1dcee52889e2d8acc3	503740e9ca751ccdda7ba28e9ab8f608	delivered	2
3	6514b8ad8028c9f2cc2374ded245783f	9bdf08b4b3b52b5526ff42d37d47f222	delivered	2
4	76c6e866289321a7c93b82b54852dc33	f54a9f0e6b351c431402b8461ea51999	delivered	2

In [138]:
```
final_df = final_df.dropna()
```

In [139]:
```
# final_df.to_csv('all_df.csv')
```

In [140]:
```
popular_categories = final_df
```

In [141]:
```
popular_categories = final_df['product_category_name_english'].value_counts()

top_categories = popular_categories.head(5)
top_categories['Others'] = popular_categories.iloc[5:].sum()

plt.figure(figsize=(12, 12))
plt.pie(top_categories, labels=top_categories.index, autopct='%1.1f%%', startar
plt.title('Top 5 most popular categories in 2017')
plt.legend(top_categories, loc="best")
plt.show()
```



In [142]:
```
top_categories.head()
```

Out[142]:
```
bed_bath_table          4316
sports_leisure          3500
health_beauty           3280
furniture_decor         3034
computers_accessories   2520
Name: product_category_name_english, dtype: int64
```

Berdasarkan data yang sudah diolah dan ditampilkan oleh piechart diatas, kita bisa menyimpulkan bahwa kategori paling populer di tahun 2017 adalah kategori bed_bath_table. 4 kategori populer lainnya secara berurutan adalah sports_leisure, health_beauty, furniture_decor, computers_accessories

Apa saja 5 kategori dengan skor review (rata rata) terbagus dan terburuk?

Cara saya menjawab ini adalah dengan menggunakan final_df yang sudah dipakai di pertanyaan nomor 1, lalu saya potong bagian nama kategori dan skornya, saya baru lalu akan hitung rata ratanya

In [143]:
```
category_score = final_df[['review_score', 'product_category_name_english']]
category_score.head()
```

Out[143]:

	review_score	product_category_name_english
0	4.0	housewares
1	5.0	pet_shop
2	4.0	auto
3	5.0	auto
4	1.0	furniture_decor

In [144]:
```
category_score.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 42316 entries, 0 to 43082
Data columns (total 2 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   review_score                   42316 non-null  float64
 1   product_category_name_english  42316 non-null  object
dtypes: float64(1), object(1)
memory usage: 991.8+ KB
```

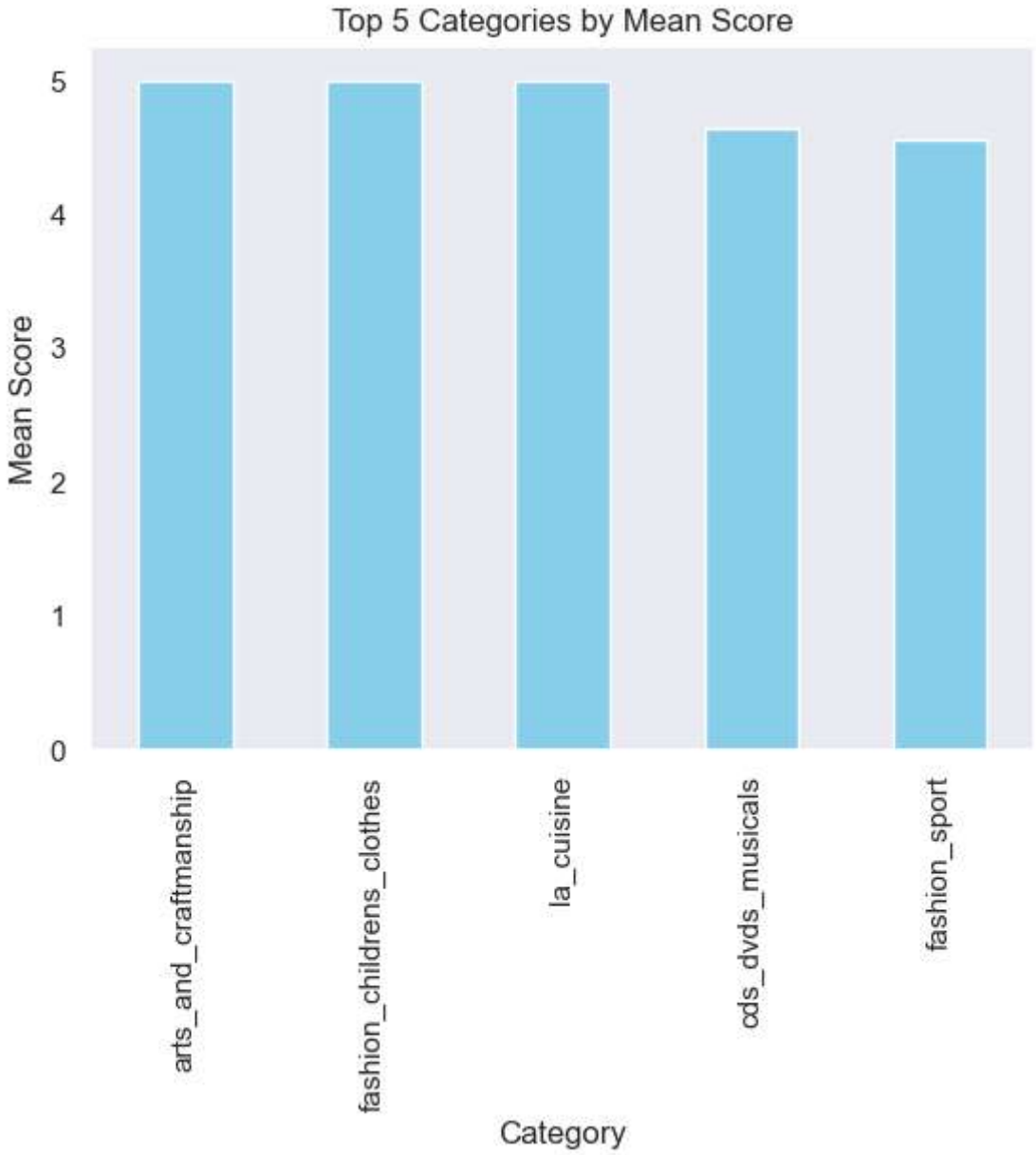
In [145]:
```
category_score.describe()
```

Out[145]:

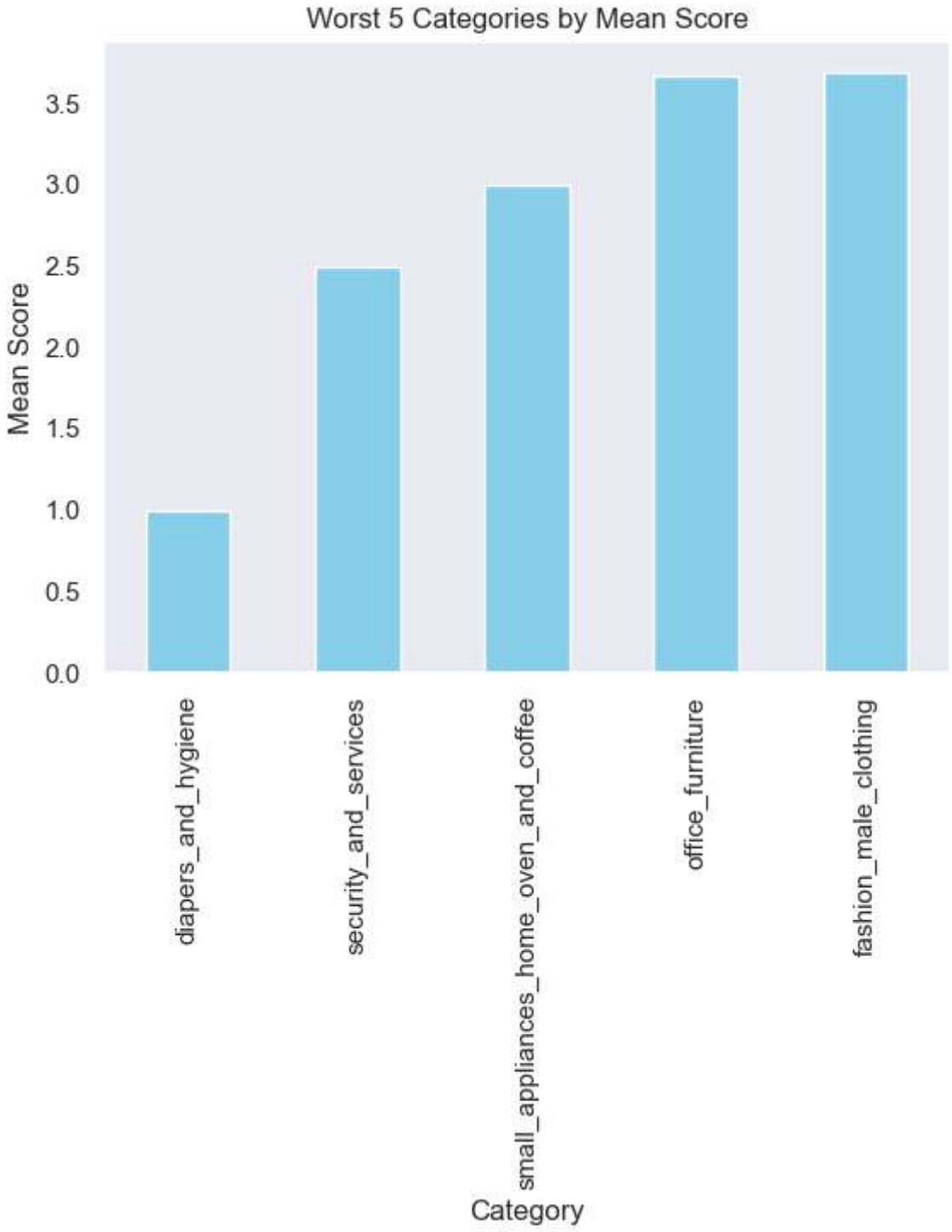
	review_score
count	42316.000000
mean	4.177096
std	1.254582
min	1.000000
25%	4.000000
50%	5.000000
75%	5.000000
max	5.000000

In [146]:
```
mean_category_scores = category_score.groupby(by="product_category_name_englis
    "review_score": "mean",
}).sort_values(by="review_score", ascending=False)
```

In [147]:
```
df = pd.DataFrame(mean_category_scores)

# Calculate the mean of each category
category_means = df.groupby('product_category_name_english')['review_score'].me

# Get the top 10 categories based on mean score
top_5_categories = category_means.nlargest(5)
bad_5_categories = category_means.nsmallest(5)

# Plotting a bar chart for best score
top_5_categories.plot(kind='bar', color='skyblue')
plt.title('Top 5 Categories by Mean Score')
plt.xlabel('Category')
plt.ylabel('Mean Score')
plt.show()
```



In [148]:
```
# Plotting a bar chart for best score
bad_5_categories.plot(kind='bar', color='skyblue')
plt.title('Worst 5 Categories by Mean Score')
plt.xlabel('Category')
plt.ylabel('Mean Score')
plt.show()
```



Berdasarkan bar chart diatas, ternyata ada 3 kategori yang memiliki nilai tertinggi yang sama. yaitu arts and craftmanship, fashion childrens clothes, dan la cuisine. Untuk kategori dengan skor terburuk, jatuh kepada diapers and hygiene.