PROBLEMS How was the sales trend over the months? • What are the most frequently purchased products? • How many products does the customer purchase in each transaction? · What are the most profitable segment customers? • Based on your findings, what strategy could you recommend to the business to gain more profit? Importing libraries import pandas as pd In [1]: import matplotlib.pyplot as plt import warnings warnings. filterwarnings('ignore') import cufflinks as cf from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot init_notebook_mode(connected=True) cf.go_offline() cf.set_config_file(theme='space', sharing='public', offline=True) **Data Loading** data=pd.read_csv("Sales Data.csv") In [2]: data.head() In [3]: TransactionNo Date ProductNo ProductName Price Quantity CustomerNo Country Out[3]: 581482 12/9/2019 22485 Set Of 2 Wooden Market Crates 21.47 12 17490.0 United Kingdom 581475 12/9/2019 22596 Christmas Star Wish List Chalkboard 10.65 36 13069.0 United Kingdom 2 581475 12/9/2019 23235 13069.0 United Kingdom Storage Tin Vintage Leaf 11.53 12 3 581475 12/9/2019 23272 Tree T-Light Holder Willie Winkie 10.65 12 13069.0 United Kingdom 581475 12/9/2019 23239 Set Of 4 Knick Knack Tins Poppies 11.94 13069.0 United Kingdom data.tail() In [4] **TransactionNo** Date ProductNo ProductName Price Quantity CustomerNo Country Out[4]: 536345 C536548 12/1/2018 22168 Organiser Wood Antique White 18.96 -2 12472.0 Germany 536346 C536548 12/1/2018 21218 Red Spotty Biscuit Tin 14.09 12472.0 Germany 536347 C536548 12/1/2018 20957 Porcelain Hanging Bell Small 11.74 -1 12472.0 Germany 536348 C536548 12/1/2018 22580 Advent Calendar Gingham Sack 16.35 12472.0 Germany 536349 C536548 12/1/2018 22767 Triple Photo Frame Cornice 20.45 -2 12472.0 Germany data.shape (536350, 8)data.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 536350 entries, 0 to 536349 Data columns (total 8 columns): Column Non-Null Count Dtype -----TransactionNo 536350 non-null object Date 536350 non-null object ProductNo 536350 non-null object 536350 non-null object ProductName 536350 non-null float64 Price 5 536350 non-null int64 Quantity 6 CustomerNo 536295 non-null float64 7 536350 non-null object Country dtypes: float64(2), int64(1), object(5) memory usage: 32.7+ MB Additional info • TransactionNo (categorical): A six-digit unique number that defines each transaction. The letter "C" in the code indicates a cancellation. • Date(numeric): The date when each transaction was generated. • ProductNo (categorical): A five or six-digit unique character used to identify a specific product. • Product (categorical): Product/Item name. • Price (numeric): The price of each product per unit in pound sterling. • Quantity (numeric): The quantity of each product per transaction. Negative values related to cancelled transactions. • CustomerNo (categorical): A five-digit unique number that defines each customer. • Country (categorical): Name of the country where the customer resides. Turn the Date column to DateTime type data['Date']=pd.to_datetime(data['Date']) In [7]: data.head() In [8]: **TransactionNo** Date ProductNo Out[8]: ProductName Price Quantity CustomerNo Country 581482 2019-12-09 22485 Set Of 2 Wooden Market Crates 21.47 12 17490.0 United Kingdom 581475 2019-12-09 22596 Christmas Star Wish List Chalkboard 10.65 36 13069.0 United Kingdom 581475 2019-12-09 13069.0 United Kingdom 2 23235 Storage Tin Vintage Leaf 11.53 12 581475 2019-12-09 13069.0 United Kingdom 23272 Tree T-Light Holder Willie Winkie 10.65 12 13069.0 United Kingdom 581475 2019-12-09 23239 Set Of 4 Knick Knack Tins Poppies 11.94 6 data.info() In [9]: <class 'pandas.core.frame.DataFrame'> RangeIndex: 536350 entries, 0 to 536349 Data columns (total 8 columns): Column Non-Null Count Dtype -----O TransactionNo 536350 non-null object 536350 non-null datetime64[ns] 1 Date 2 ProductNo 536350 non-null object 536350 non-null object ProductName 536350 non-null float64 Price Quantity 536350 non-null int64 CustomerNo 536295 non-null float64 Country 536350 non-null object dtypes: datetime64[ns](1), float64(2), int64(1), object(4) memory usage: 32.7+ MB In [10]: data.describe() Out[10]: Price Quantity CustomerNo **count** 536350.000000 536350.000000 536295.000000 12.662182 9.919347 15227.893178 mean 216.662300 std 8.490450 1716.582932 -80995.000000 min 5.130000 12004.000000 1.000000 13807.000000 25% 10.990000 **50**% 11.940000 3.000000 15152.000000 10.000000 **75**% 14.090000 16729.000000 660.620000 80995.000000 18287.000000 max data.isnull().sum() In [11]: TransactionNo Out[11]: Date 0 ProductNo 0 ProductName 0 Price 0 0 Quantity CustomerNo 55 Country dtype: int64 data[data['CustomerNo'].isnull()] In [12]: ProductName Price Quantity CustomerNo Out[12]: **TransactionNo** Date ProductNo Country 6511 C581406 2019-12-08 46000M Polyester Filler Pad 45x45cm 6.19 -240 NaN United Kingdom 6512 C581406 2019-12-08 46000S 6.19 -300 Polyester Filler Pad 40x40cm NaN United Kingdom 90098 C575153 2019-11-08 22947 Wooden Advent Calendar Red -1 NaN United Kingdom 102671 C574288 2019-11-03 22178 Victorian Glass Hanging T-Light 25.37 -1 NaN United Kingdom 117263 C573180 2019-10-28 23048 Set Of 10 Lanterns Fairy Light Star 14.50 -1 NaN United Kingdom 163160 C569495 2019-10-04 21843 -1 Red Retrospot Cake Stand 21.47 NaN United Kingdom 190598 C567518 2019-09-20 22846 Bread Bin Diner Style Red 27.62 -1 NaN United Kingdom 192284 C567518 2019-09-20 21871 Save The Planet Mug 11.94 -12 NaN United Kingdom -160 242406 C563015 2019-08-11 46000M Polyester Filler Pad 45x45cm 10.25 NaN United Kingdom 242407 C563015 2019-08-11 46000S -220 Polyester Filler Pad 40x40cm 10.25 NaN United Kingdom -2 246334 C562617 2019-08-08 23243 Set Of Tea Coffee Sugar Tins Pantry 15.32 United Kingdom Airline Bag Vintage Tokyo 78 14.48 254250 C561966 2019-08-01 22371 -1 NaN United Kingdom 254254 C562030 2019-08-01 84595E Large Tortilla Design Red Bowl 13.62 -1 NaN United Kingdom 260633 C561239 2019-07-26 22427 -1 Enamel Flower Jug Cream 16.35 NaN United Kingdom 260634 C561239 2019-07-26 22426 Enamel Wash Bowl Cream 14.09 -1 NaN United Kingdom 260635 C561239 2019-07-26 84755 -4 Colour Glass T-Light Holder Hanging 10.92 NaN United Kingdom -2 276396 C559949 2019-07-14 22699 Roses Regency Teacup And Saucer 13.27 NaN United Kingdom 285494 C559296 2019-07-07 23001 Travel Card Wallet Dotcomgiftshop 10.25 -200 NaN United Kingdom 285495 C559296 2019-07-07 22616 Pack Of 12 London Tissues 10.25 -204 NaN United Kingdom 285496 C559296 2019-07-07 23298 Spotty Bunting 10.25 -170 NaN United Kingdom 297259 C558364 2019-06-28 22960 Jam Making Set With Jars 18.75 -1 NaN United Kingdom 298352 558245 2019-06-27 8 22734 Set Of 6 Ribbons Vintage Christmas 10.25 NaN United Kingdom 303165 C557664 2019-06-21 22847 Bread Bin Diner Style Ivory 25.57 -1 NaN United Kingdom 304635 C557664 2019-06-21 23183 Mother's Kitchen Spoon Rest 14.09 -1 NaN United Kingdom 311943 C556939 2019-06-15 46000M Polyester Filler Pad 45x45cm 10.25 -80 NaN United Kingdom C556939 2019-06-15 Polyester Filler Pad 40x40cm 10.25 NaN United Kingdom 311944 46000S -160 NaN United Kingdom 333735 C554716 2019-05-26 22662 Lunch Bag Dolly Girl Design 5.97 -1 333736 C554716 2019-05-26 23206 Lunch Bag Apple Design 5.97 -1 NaN United Kingdom 342908 C553862 2019-05-19 84945 Multi Colour Silver T-Light Holder 21.28 -1 NaN United Kingdom 342953 C553854 2019-05-19 22053 -1 Empire Design Rosette 11.53 NaN United Kingdom 342954 C553854 2019-05-19 21703 Bag 125g Swirly Marbles 10.68 -1 NaN United Kingdom 342955 C553854 2019-05-19 22891 Tea For One Polkadot 14.61 -1 NaN United Kingdom 361482 C552230 2019-05-06 20735 Black Mini Tape Measure 10.25 -1 NaN United Kingdom 361483 C552230 2019-05-06 22617 -1 Baking Set Spaceboy Design 10.25 NaN United Kingdom 361484 C552230 2019-05-06 37327 Asstd Multicolour Circles Mug -1 NaN United Kingdom 361485 C552230 2019-05-06 84375 Set Of 20 Kids Cookie Cutters 10.25 -1 NaN United Kingdom 361486 C552230 2019-05-06 84969 Box Of 6 Assorted Colour Teaspoons 10.25 -1 NaN United Kingdom 395771 C548605 2019-04-01 22453 -1 Measuring Tape Babushka Blue 13.27 United Kingdom 412419 C547091 2019-03-20 22766 Photo Frame Cornice 13.27 -12 NaN United Kingdom NaN United Kingdom 412420 C547091 2019-03-20 21316 Small Chunky Glass Roman Bowl 13.27 -1 412421 C547091 2019-03-20 22800 Antique Tall Swirlglass Trinket Pot 14.09 -4 NaN United Kingdom 413780 C547008 2019-03-18 22567 20 Dolly Pegs Retrospot 11.53 -1 NaN United Kingdom 416704 C546758 2019-03-16 22212 Four Hook White Lovebirds 12.40 -1 NaN United Kingdom 429733 C545506 2019-03-03 22907 Pack Of 20 Napkins Pantry Design 11.12 -12 NaN United Kingdom C543384 2019-02-07 454036 48173C Doormat Black Flock 18.40 -1 NaN United Kingdom C543185 2019-02-04 455511 22333 -14 Retrospot Party Bag + Sticker Set 11.94 NaN United Kingdom C543185 2019-02-04 455512 22332 Skulls Party Bag + Sticker Set 11.94 -14 NaN United Kingdom 459432 C542860 2019-02-01 22580 Advent Calendar Gingham Sack 16.35 -1 NaN United Kingdom 462607 C542540 2019-01-28 21658 Glass Beurre Dish 18.75 -1 NaN United Kingdom C540854 2019-01-12 483197 22461 Savoy Art Deco Clock 23.32 -1 NaN United Kingdom Bingo Set 14.09 486650 C540559 2019-01-10 21888 -4 NaN United Kingdom 493036 C540155 2019-01-05 72802B Ocean Scent Candle In Jewelled Box 14.16 -54 NaN United Kingdom 495203 C539948 2018-12-23 21888 Bingo Set 14.09 -4 NaN United Kingdom C539756 2018-12-21 497059 22720 Set Of 3 Cake Tins Pantry Design 15.32 -1 United Kingdom 507547 C538686 2018-12-14 22467 Gumball Coat Rack 12.86 -1 NaN United Kingdom All the null values in CustomerNo have the same country and negative Quantity (except one, whose TransactionNo does not start with the letter 'C' and does not have a negative Quantity). Let's drop all null and duplicated values data.dropna(inplace=True) data.isnull().sum() 0 TransactionNo Out[13]: 0 Date 0 ProductNo 0 ProductName Price 0 Quantity 0 CustomerNo 0 Country dtype: int64 data.duplicated().sum() In [14]: 5200 Out[14]: data.drop_duplicates(inplace=True) data.duplicated().sum() Out[15]: Answering the Questions 1. How was the sales trend over the months? data['Total']=data['Quantity']*data['Price'] data['Month_Year']=pd.to_datetime(data['Date']).dt.to_period('M') data.head() **Total Month Year** TransactionNo Date ProductNo ProductName Price Quantity CustomerNo Country Out[16]: Set Of 2 Wooden Market Crates 21.47 17490.0 United Kingdom 257.64 581482 2019-12-09 22485 12 2019-12 581475 2019-12-09 22596 Christmas Star Wish List Chalkboard 10.65 36 13069.0 United Kingdom 383.40 2019-12 581475 2019-12-09 23235 Storage Tin Vintage Leaf 11.53 12 13069.0 United Kingdom 138.36 2019-12 23272 581475 2019-12-09 Tree T-Light Holder Willie Winkie 10.65 12 2019-12 13069.0 United Kingdom 127.80 581475 2019-12-09 23239 Set Of 4 Knick Knack Tins Poppies 11.94 13069.0 United Kingdom 71.64 2019-12 total_month=data.groupby(['Month_Year'], as_index=False)['Total'].sum() In [17]: total_month.columns=['Date', 'Total'] total month.head() **Total** Out[17]: **Date 0** 2018-12 4217281.53 **1** 2019-01 3638948.12 **2** 2019-02 3292324.69 **3** 2019-03 4340035.60 **4** 2019-04 3406002.17 total_month.iplot(x='Date',y='Total',title='Sales Trend Over The Months', In [18]: yTitle='Total Money', xTitle='Month', mode='lines+markers', size=6) Sales Trend Over The Months 8M 7M 6M Total Money 5M 4M 3M 2M Mar 2019 Sep 2019 Nov 2019 Jan 2019 May 2019 Jul 2019 Month Export to plot.ly In [19]: data_no_return=data[data['Quantity']>0] number_of_sales=data_no_return.groupby(by='Month_Year', as_index=False)['Total'].count() number_of_sales Month_Year Total Out[19]: 2018-12 40899 2019-01 33860 1 2 2019-02 26759 3 2019-03 35318 4 2019-04 28707 2019-05 35756 2019-06 35533 7 2019-07 38138 2019-08 34092 9 2019-09 48635 10 2019-10 58269 11 2019-11 81819 12 2019-12 24816 In [20]: number_of_sales.columns=['Date', 'Sum of Sales'] number_of_sales.iplot(x='Date',y='Sum of Sales',title='Number of Sales Each Month', yTitle='Number of Sales', xTitle='Months', mode='lines+markers', size=6) Number of Sales Each Month 80k 70k Sales 60k Number of 50k 40k 30k Jan 2019 Mar 2019 May 2019 Jul 2019 Sep 2019 Nov 2019 Months Export to plot.ly > data.sort_values('Date', ascending=False).head(5) **TransactionNo** Date ProductNo Out[21]: ProductName Price Quantity CustomerNo Country Total Month_Year 17490.0 United Kingdom 581482 2019-12-09 22485 Set Of 2 Wooden Market Crates 21.47 12 257.64 2019-12 1076 581498 2019-12-09 21930 2019-12 Jumbo Storage Bag Skulls 6.19 14498.0 United Kingdom 12.38 1086 581498 2019-12-09 22041 Record Frame 7" Single Size 6.19 2 14498.0 United Kingdom 12.38 2019-12 1085 2019-12 581498 2019-12-09 21989 Pack Of 20 Skull Paper Napkins 6.19 14498.0 United Kingdom 6.19 1084 581498 2019-12-09 Pack Of 6 Skull Paper Cups 2019-12 21987 6.19 14498.0 United Kingdom 12.38 **Answer** We can see that from December 2018 to August 2019 there is no big change in total income or number of sales. But things are getting different from that point, maybe due to Christmas, we can see a rising trend on our charts. According to them, December 2019 was the worst month in sales and income, however, we can observe, that it is due to a lack of data. Our dataset is only up to 9th December, whereas all other months have full data. There is only info of 9 days in December 2019, but we already have more than half the number of sales than in December 2018 and almost half of income. So we can guess that December 2019 would have been richer in sales and income than a year ago. 2. What are the most frequently purchased products? popular_products=data.groupby(by='ProductName', as_index=False)['Quantity'].sum() popular_products=popular_products.sort_values('Quantity', ascending=False).head(10) popular_products.iplot(kind='bar', x='ProductName', y='Quantity', title='10 Most Frequently Purchased Products') 10 Most Frequently Purchased Products 50k 30k 20k 10k Pack Of 60 Pink Paisley Cake Cases World War 2 Gliders Assta Designs Assorted Colour Bird Ornament Pack Of 72 Retrospot Cake Cases Cream Hanging Heart T-Light Holder Pack Of 12 London Tissues Jumbo Bag Red Retrospor Mini Paint Set Vintage Popcorn Holder **Export to plot.ly** 3. How many products does the customer purchase in each transaction? Let's see the top 10 quantity amount top_quantity = data.sort_values('Quantity', ascending=False)[['Quantity', 'TransactionNo']].head(10).reset_index(drop = True) x=top_quantity['TransactionNo'] y=top_quantity['Quantity'] plt.bar(x,y,color='orange',ec="black") plt.title("Top 10 Quantity", fontsize=16) # plt.ylabel("Salary", fontsize=12) plt.xlabel("Transaction Number", fontsize=12) plt.xticks(fontsize=10, rotation=45) plt.grid(True, linestyle='dashed') plt.show() Top 10 Quantity 80000 70000 60000 50000 40000 30000 20000 10000 Transaction Number I will calculate the mean of positive quantities, that will be my answer to question round(data_no_return['Quantity'].mean()) Out[25]: **Answer** The average number of products bought per transaction is 11. 4. What are the most profitable segment customers? Let me check which countries are the most and least profitable. most_profitable_countries = data.groupby(by='Country', as_index = False)['Total'].sum().sort_values('Total', ascending = False).head(10) In [26]: least_profitable_countries = data.groupby(by='Country', as_index = False)['Total'].sum().sort_values('Total').head(10) In [27]: most_profitable_countries.iplot(kind='bar', x ='Country', y='Total', title='Most Profitable Countries', yTitle='Income', colors='Green') least_profitable_countries.iplot(kind='bar', x ='Country', y='Total', title='Least Profitable Countries', yTitle='Income', colors='Red') Most Profitable Countries 50M 40M 30M 20M **United Kingdom EIRE** Australia Switzerland Netherlands Germany France Sweden Japan Belgium Export to plot.ly > **Least Profitable Countries** 12k 10k 8k Income 6k 4k Saudi Arabia Bahrain RSA European Community Czech Republic Brazil Lithuania **United Arab Emirates Export to plot.ly** The most popular products in UK popular_UK=data_no_return[data_no_return['Country']=='United Kingdom'] popular_UK1=popular_UK.groupby(by='ProductName', as_index=False)['Total'].count().head().sort_values('Total', ascending=False) popular_UK1.iplot(kind='pie', labels='ProductName', values='Total', hole =.4, title='Most Popular Products in UK') O Most Popular Products in UK 10 Colour Spaceboy Pen 12 Coloured Party Balloons 12 Egg House Painted Wood 12 Daisy Pegs In Wood Box 25.3% 12 Hanging Eggs Hand Painted 45.4% 14.8% 13% **Export to plot.ly**: In [29]: popular_UK_2=popular_UK.groupby(by='ProductName', as_index = False)['Total'].sum().head().sort_values('Total', ascending = False) popular_UK_2.iplot(kind='pie', labels='ProductName', values='Total', hole =.4, title='Top High-Income Products in the UK') O Top High-Income Products in the UK 10 Colour Spaceboy Pen 12 Coloured Party Balloons 12 Daisy Pegs In Wood Box 12 Egg House Painted Wood 25.4% 12 Hanging Eggs Hand Painted 4.69% 65.4% 4.08% **Export to plot.ly**: Let's see now which products are providing the highest income worlwide. most_profitable_products = data.groupby(by='ProductName', as_index=False)['Total'].sum().sort_values('Total', ascending=False).head(10) most_profitable_products.iplot(kind='bar', x='ProductName', title='Top 10 High Income Products Worldwide', yTitle='Income', colors='#FF8552') Top 10 High Income Products Worldwide 600k 500k 400k Income 300k 200k 100k Cream Hanging Heart T-Light Holder Assorted Colour Bird Ornament Pack Of 72 Retrospot Cake Cases World War 2 Gliders Assta Designs Paper Craft Little Birdie Jumbo Bag Red Retrospot Regency Cakestand 3 Tier Mini Paint Set Vintage Rabbit Night Light **Export to plot.ly Answer** Most profitable segment customers are customers from countries included in 'Most Profitable Countries' (especially UK). 5. Based on your findings, what strategy could you recommend to the business to gain more profit? We need a little more research data_negative_quantity = data[data['Quantity'] < 0]</pre> data_negative_quantity['Quantity']=abs(data_negative_quantity['Quantity']) data_negative_quantity_1=data_negative_quantity.groupby(by='ProductName', as_index=False)['Quantity'].sum().sort_values('Quantity', ascending=False).head(10) data_negative_quantity_1 Out[31]: ProductName Quantity 1144 Paper Craft Little Birdie 990 74494 Medium Ceramic Top Storage Jar 1409 Rotating Silver Angels T-Light Hldr 9367 566 Fairy Cake Flannel Assorted Colour 3150 414 Cream Hanging Heart T-Light Holder 685 Gin And Tonic Diet Metal Sign 2030 801 Herb Marker Basil 1527 592 Feltcraft Doll Molly 1447 1694 Tea Time Party Bunting 1424 1148 Paper Pocket Traveling Fan 1385 data_negative_quantity_1.iplot(kind='bar', x='ProductName', y='Quantity', title='Top Canceled Products', colors='#FF5A5F') **Top Canceled Products** 80k 60k 40k 20k Cream Hanging Heart T-Light Holder Medium Ceramic Top Storage Jar Rotating Silver Angels T-Light Hlor Fairy Cake Flannel Assorted Colour Gin And Tonic Diet Metal Sign Paper Craft Little Birdie Tea Time Party Bunting Paper Pocket Traveling Fan Feltoraft Doll Molly **Export to plot.ly**: data_negative_quantity_2=data_negative_quantity.groupby(by='Country', as_index=False)['Quantity'].sum().sort_values('Quantity', ascending=False).head(10) data_negative_quantity_2.iplot(kind='bar', x='Country', y='Quantity', title='Countries With Most Canceled Transactions', colors='#A6808C') **Countries With Most Canceled Transactions** 250k 200k 150k 100k 50k France USA Switzerland Germany Japan Australia **Export to plot.ly** Answer 1. The company needs to improve shipping or advertising more to other countries than the UK. 1. I would suggest checking on the first two products on 'Top Canceled Products'. The company lost 150k orders because of these two products' order cancelation. 1. Advertising best-selling products in other countries than the UK. 1. Increasing the ads budget a couple months before Christmas may be a good idea.