Checkout Process Analysis
Manas Desai
11/14/23

Introduction

First of all, thank you for giving me the opportunity to analyze this data! In this documentation, I delve into the checkout process data to see if I can find any insights to help with future decision-making. While the data itself is fabricated, I will treat this as a real world dataset to simplify my explanations. This is a short investigation and small dataset but I will do my best to identify anomalies, explore conversion rates, examine correlations between different steps, and provide a general overview of the story I believe the data tells. To conduct my analysis, I primarily used python as it has extensive libraries for analysis (e.g., pandas, NumPy, Matplotlib, Seaborn). Python packages provide visualization capabilities which R Studio lacks and complex statistics tools which Excel lacks. Ideally I would have used Tableau for my graphs but my free subscription ended. All of my code is documented and the github repository for this project is given in the email accompanying this document.

Data Overview

The dataset showcases various aspects of the checkout process as described in the instructions. The 'Data' workbook has no null values and is organized by descending date values housed in the first column. Throughout this paper I will refer to the individual steps of checkout as 'variables.'

General Analysis

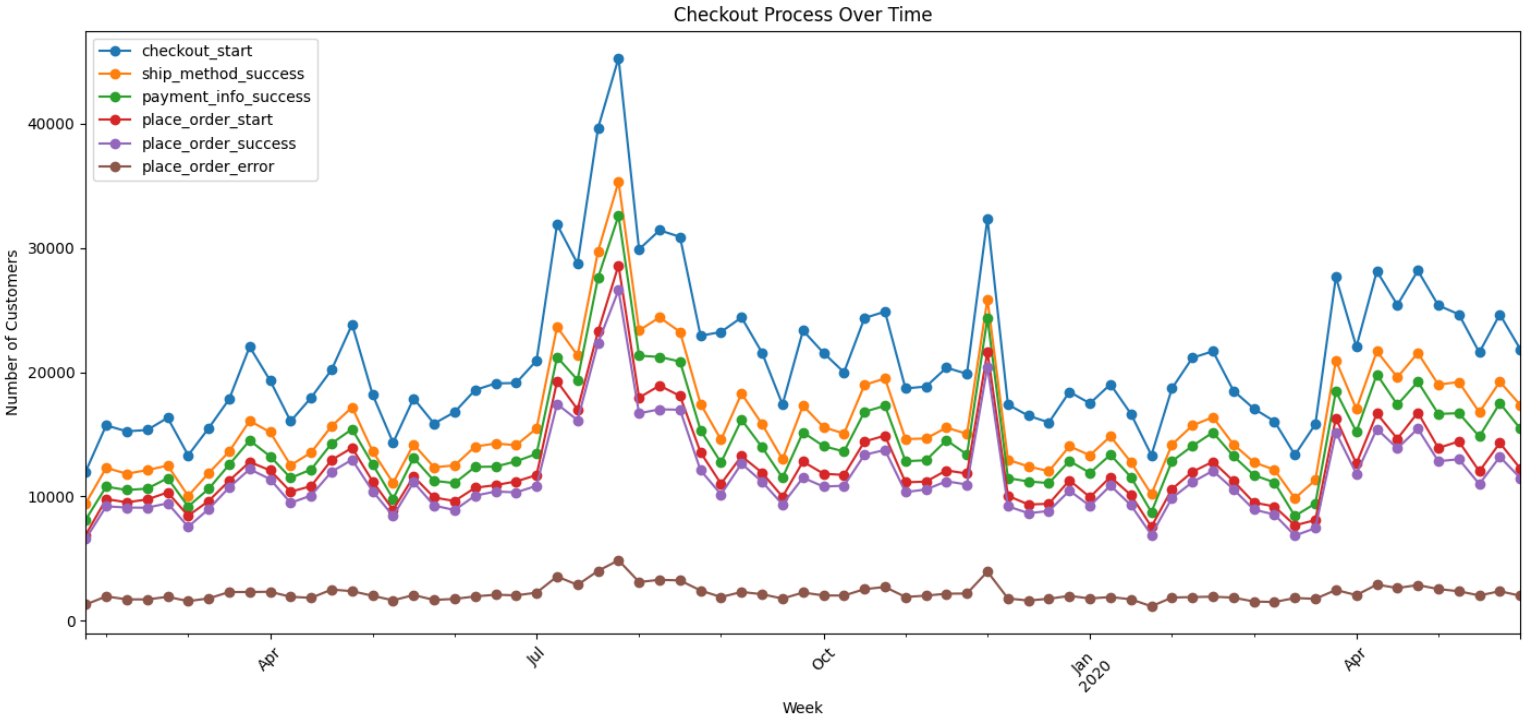

Figure 1

Let's take a surface level look at this data before we do any analysis. Figure 1 is a visualization I have created to show how many customers (y-axis) partake in each step (color coded) of the process in the weeks (x-axis) the data was collected. The graph simply puts the numbers from the 'Data' workbook in an easier to digest format. We can already

put together multiple inferences from this visualization before we utilize any statistics:
1. There will always be fewer customers as the steps progress. This might seem obvious but the graph tells me that the data collection system has no immediately glaring flaws. For example, if there were more people reaching the 'place_order_success' part of the checkout than 'checkout_start,' then the rest of the analysis wouldn't be valid since there is a flaw in the data collection process - it's impossible to place an order successfully if the order was never initiated.
2. There seems to be a positive correlation between all of the variables. The more people that start the process, the more people finish it. This is another one that might seem obvious but also gives us some key information. It is important to check for negative correlations between the steps to see if any individual step is actively driving away customers. For example, if there was a decrease in customers completing an order successfully every time there was an increase in customers reaching the payment page, we could infer that the payment page needed to be fixed in some manner. These correlations are calculated later in the analysis as a failsafe to make sure our other calculations dependent on the assumption of correlation are valid.
3. We can see some spikes in the graph. An easy way to see if these spikes have any significance is by checking the z-score (to see if those spikes are statistically varied from the mean). The spikes might also dictate seasonality (more products sell at specific times of the year). Seasonality would be difficult to test for since the whole dataset takes up only the span of a year and a half.
Okay so we have some surface level assumptions, lets see if they hold any validity and if we can find any weeks in the dataset that are varied in any way.

Anomalies

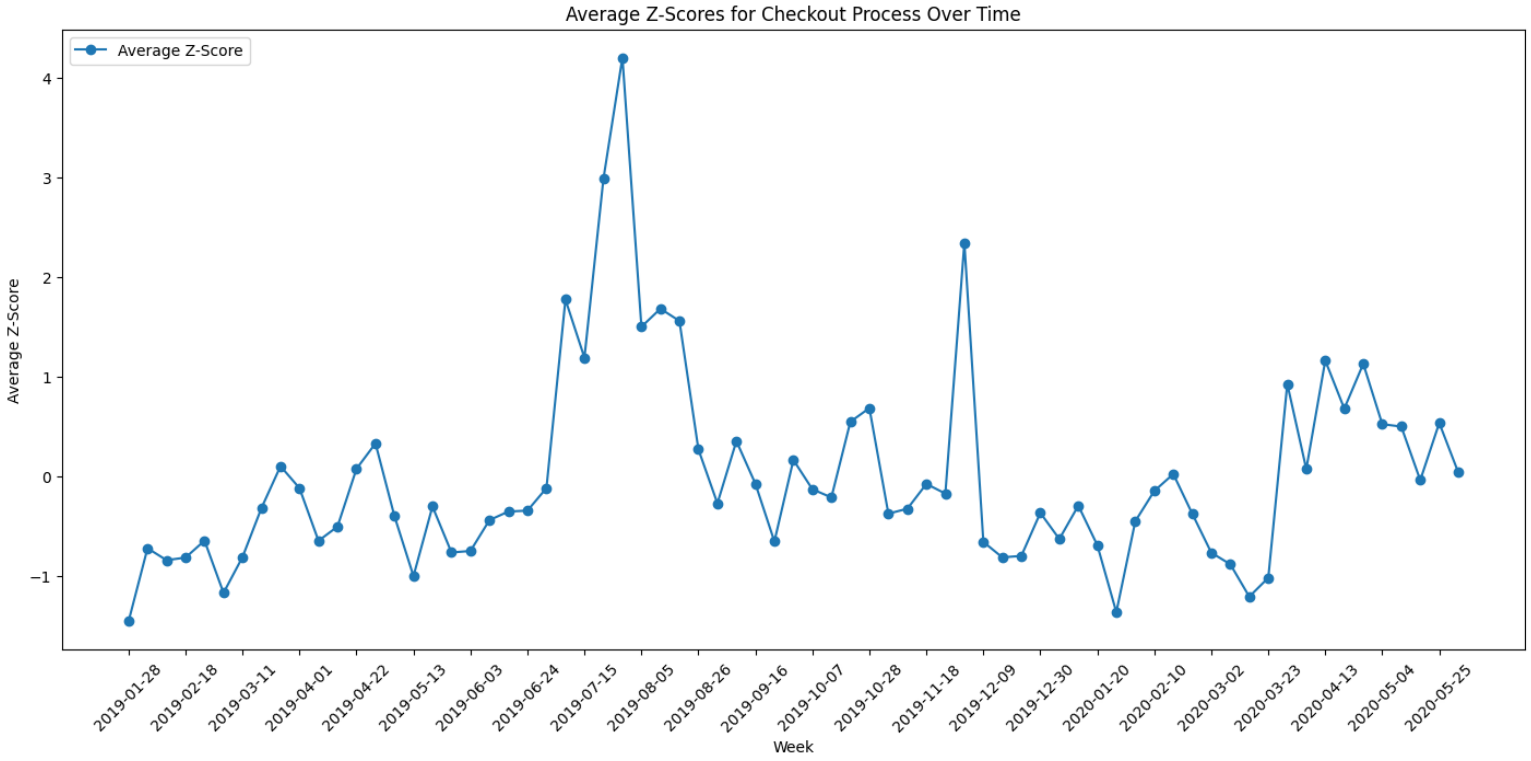

Figure 2

Let's break down the visualization in Figure 2. The z-score shows us how many standard deviations a specific value is from the mean. Initially I had made a visualization which showed the z-score for each variable over time; however

since the variables are heavily correlated, the difference between the z-scores between variables was close to none. This updated visualization shows the average z-score between the six different variables over time. I simplified the graph for readability but the highest z-score is on 07/29/2019, this is the large spike that surpasses the z-score of 4 (which is quite extreme). I could normalize the data to reign in the z-scores but this graph is a good visual demonstration of which weeks are significantly different from the average. Let's take a look at the two dates with the highest z-scores and theorize why this might have taken place.

07/29/2019 : This column deviated most from the mean, and taking a look at the individual variable values it makes sense why. Every single variable from this date was the highest value in its respective column. Three weeks before and three weeks after 07/29/2019 also had a spike in checkouts. While it is not possible for me to figure out the exact cause without knowledge of the product and company, I can guess that this spike could be because of a marketing campaign or the product blowing up on social media. It is also possible that a popular influencer or celebrity spoke about the product which caused a spike in sales.

12/02/2019 : A similar spike happened toward the end of the year but this one was short-lived. The week was not preceded or followed by high checkout rates; so a strategy different from the initial spike was used. The drastic return to average sales before and after makes me think this was a weeklong or shorter event where the company held a cheaper price for the product. Since this is towards the end of the year it is possible there was a christmas, new year, or general holiday sale.

There are two other dates worth mentioning: **1/28/2019** and **1/27/2020**. This is a week in which the company consistently suffers year over year. If I had more data, I would have tested for seasonality to see if this is a recurring dropoff in sales. I cannot say with statistical significance that these weeks are relevant since they fall less than two standard deviations from the mean; however, I do think they are worth looking at.

Conversion Rate Analysis

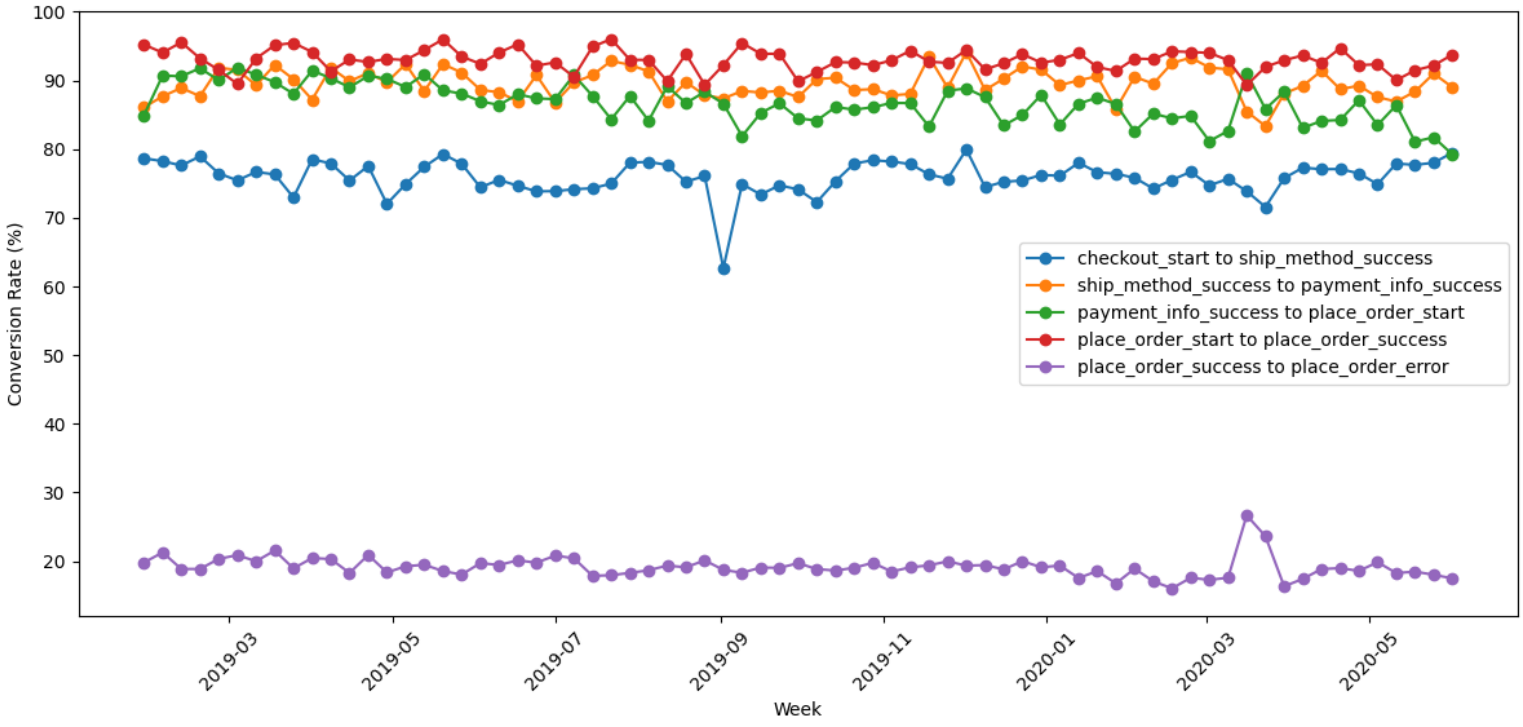

Figure 3

Analyzing conversion rates at each step of the checkout process provides valuable insights into user engagement. By understanding patterns in user engagement and disengagement, we can pinpoint areas of improvement and optimize the checkout journey for enhanced user satisfaction. In Figure 3, I have shown my conversion rate calculations through a visual. The individual points on the graph show what percentage of people continued from one step to the next, allowing us to study which areas of the checkout process are weak points. First let's take a look at the individual variables and theorize why certain parts of the checkout process seem to drive people away.

```
Mean Conversion Rates:
conversion_rate_checkout_start_to_ship_method_success          75.962190
conversion_rate_ship_method_success_to_payment_info_success    89.568277
conversion_rate_payment_info_success_to_place_order_start      86.731972
conversion_rate_place_order_start_to_place_order_success       92.989626
```

Figure 4

The snippet from Figure 4 shows the average conversion rates between the steps in the checkout process. The conversion rate from ship_method_success onwards stays above 85% which is quite good. On the other hand, it seems about a quarter of the people leave at checkout_start. I believe a good way to negate this is to provide offers on the checkout page or find another way to excite the customer on the product so we don't lose them at checkout. Some websites provide coupon codes on the checkout page so the customer has more incentive to start the checkout process. There could be a deeper analysis done on this if we had more data on the checkout page itself.

```
Minimum Conversion Rates:
conversion_rate_checkout_start_to_ship_method_success          62.666208
conversion_rate_ship_method_success_to_payment_info_success    83.349527
conversion_rate_payment_info_success_to_place_order_start      79.232462
conversion_rate_place_order_start_to_place_order_success       89.319515
conversion_rate_place_order_success_to_place_order_error       16.021019
dtype: float64

Maximum Conversion Rates:
conversion_rate_checkout_start_to_ship_method_success          79.942561
conversion_rate_ship_method_success_to_payment_info_success    94.120601
conversion_rate_payment_info_success_to_place_order_start      91.874252
conversion_rate_place_order_start_to_place_order_success       96.029892
conversion_rate_place_order_success_to_place_order_error       26.664915
```

Figure 5

Lets refer back to Figure 3, Figure 4 and Figure 5 to analyze some of the weeks where the graph spikes.

9/02/2019 : For this week there is a significant drop-off from checkout_start to ship_method_success. The average conversion rate for this step is already quite low at 75.96%, but this week it fell to an all time low of 62.66%. This is also the lowest conversion rate in the entire dataset. My theory is that the checkout page had either a bug or stated a

price that was higher than the rest of the website which drove away customers. Another interesting point is that the correlations between the different conversion rates are usually positive except for this week which would further support the argument of a webpage bug.

3/16/2020 & 3/23/2020 : These two weeks had error rates higher than the usual as you can see by the spike in the graph. When comparing 'place_order_error' with 'place_order_success' for these two weeks, the percentages are at 26.66% error rate and 23.61% error rate respectively. These error rates must have been caused by a bug on the place order button since the other steps on these dates are not significantly different from the mean. The bug was contained within these two weeks as the error rate fell back to normal levels in the following weeks.

Quick note before moving on, the conversion rate to the place order error screen should be low. If most people ended up on an error screen, most people are not successfully checking out. I found the data for the error page a bit confusing when I was doing the math. For example, in the first week there were 6907 people who started to place an order and 6577 people got a successful order through. This means there were about 330 people who either should have gotten an error message or just chose to leave the site; however, there were 1301 people who ended up on the error screen. Does this mean some people can get a success message *and* an error message? Or are these people retrying after an error message and getting a success message? If they are retrying, do they go back to the 'place_order_start' or a different step? Since I do not have knowledge on this, I figured I would mention my concerns as it was a point of difficulty when running some of the statistical analysis. If you are interested in the deeper math for these calculations, you can find them in the descriptions surrounding the python code on github.

Correlations

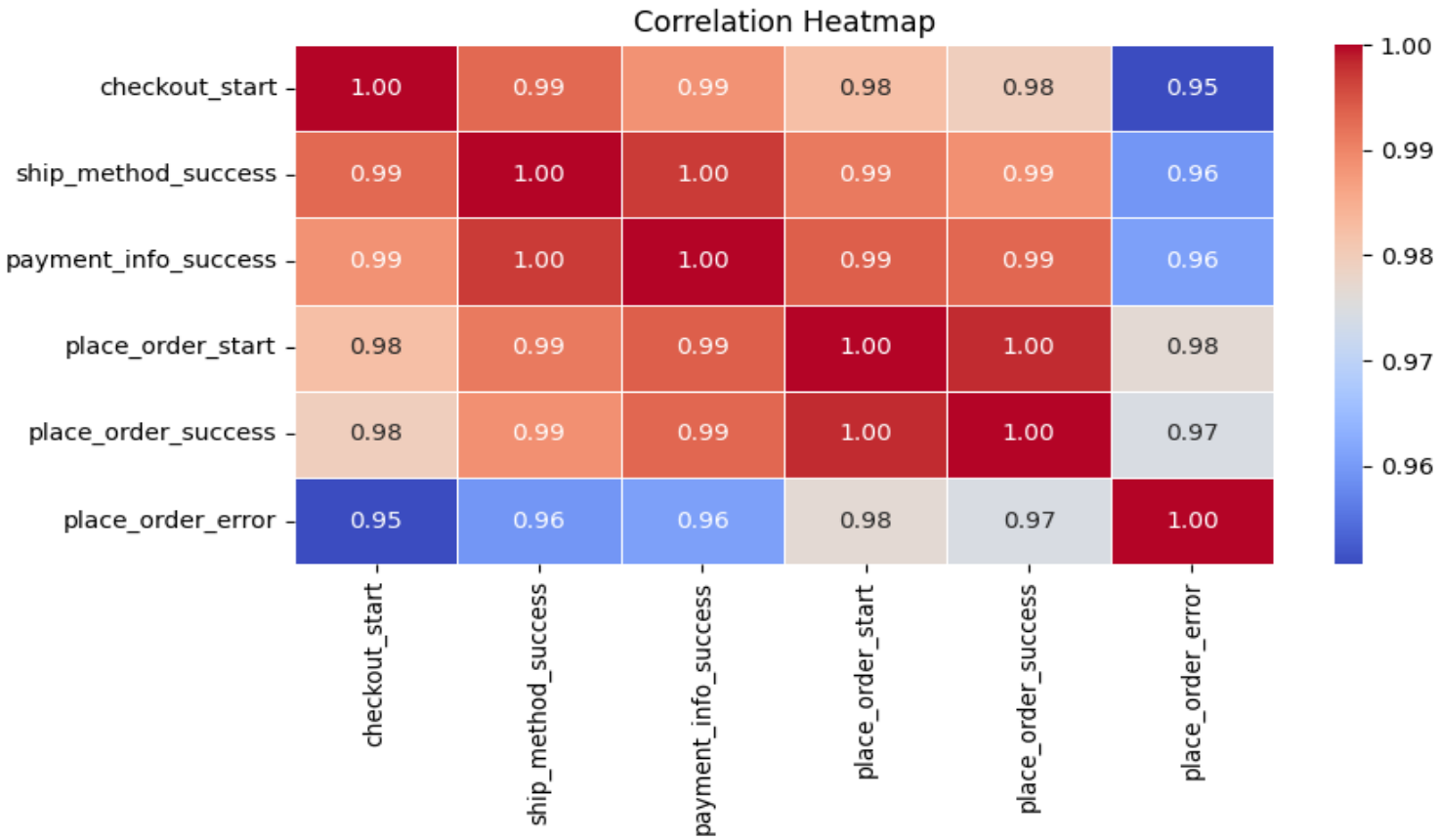

Figure 6

This section is mostly to rest my doubts from a data quality and analysis perspective. The rest of our analysis has shown us that there is a positive correlation between each of the steps and whilst there are spikes, most of the time the variables are correlated. In order to make extra sure, I ran a correlational analysis and provided a heatmap (Figure 6) to show how correlated each of the variables are to one another. For example, ‘checkout_start’ is most correlated with the very next step, ‘ship_method_success,’ and least correlated with ‘place_order_error.’ This makes sense as this error step is furthest from ‘checkout_start.’ While this final analysis might tell us what we already know, it is important to double check. If I saw here that there was a negative correlation which I missed in my other calculations I would need to revisit my analysis.

Final Notes

I hope this analysis provided some insights on some notable weeks and ways we can optimize the checkout process. From uncovering some outliers to scrutinizing conversion rate, a lot of statistics packages and python code was utilized so I would really love for you to check out the github project link. I really hope the descriptions in the code are valuable as well. Recommendations for improving documentation and analysis would be greatly appreciated!

Github Link to Project:

https://github.com/MDclassiccoder/CheckoutAnalysis