Massive Multiplayer Online Game Economies

Authors: Jordan Fields, Aaron Jacboson, Helen Ngo, Andrew Zhang, Junhao Zhang

Table of Contents

Abstract

MMO games like Runescape have robust in-game economies with assets that can be sold for fiat currency. In-game items like armor grant the holder benefits and possess rarity. Costs of in-game items and exchange rates fluctuate over time, like real world securities. Secondary markets exist for the exchange of RuneScape gold with US dollars. Our project seeks to examine through statistical means whether investments in in-game assets could potentially provide a financial return. Using the historical prices for in-game RuneScape items, we derive an optimal portfolio. This portfolio generated a return much higher than traditional equity markets during the time period examined. Following, we observed that in-game items can be strengthened or weakened through code changes, represented in the patch notes. We attempt to examine whether mentions of items in patch notes have an effect on price. Finally, we perform linear regression to determine whether increases in the total population of the player base have an effect on in-game currency (gold) value. Ultimately, we fail to discover evidence that these exogenous factors have an effect on price.

Introduction

At the beginning of the pandemic, as economic uncertainty became a central issue of the ongoing crisis, financial markets responded with significant losses. However, a few economies were left untouched. One such example are the black markets which facilitate real world trading (RWT) in popular MMO games such as Runescape 3, World of Warcraft, and Old School Runescape. RWT markets allow for players to exchange real world currency, such as a US Dollar, for in game currency or vice-versa. Could purchasing in game currency and investing in game markets serve as an effective investment vehicle? To answer this question, we will treat in-game items as if they were stocks. In addition to answering this question, we aim to understand some attributes of MMO economies to explore their similarities and differences to real world economies.

Data Set Description

Historical Pricing Data

Historical pricing data was kindly provided by WeirdGloop LLC. WeirdGloop is responsible for hosting and maintaining the Runescape 3 (RS3) wiki page, along with keeping and collecting historical pricing data for display on the wiki.

The Runescape3 historical pricing data set originally came as a collection of 6,006 files with each file corresponding to the pricing data of a unique item from the game. The data structure of each file was a Lua array with each element formatted as “timestamp:price” for the corresponding item. We opted to process the pricing files as plain text files using Python to take advantage of the Python ecosystem for data processing and analysis. Each of the 6,006 item files were read into a pandas DataFrame with timestamps in the columns and item names in the rows, and the corresponding prices filling the values.

While the quality of the data set was high, it did need a fair amount of cleaning and processing for it to be usable in our analysis. There were a number of repeated dates which had to be removed from the data set. This was most commonly a result of the price being measured twice on the same day by the data provider and seemed to be done on an inconsistent basis. We were able to implement a simple Python script to remove repeat dates. In addition, the data set was fairly sparse with a large number of null values. This was unavoidable, since items added into the game at a relatively late date would not have historical pricing data before their inception. This sparsity was handled on a question-by-question basis depending on the needs of the analysis method being employed.

Patch Note Data

In order to understand the effect of patches or public game fixes on items, we web scraped all RuneScape 3 patch notes between 2009-2020, indicating every item that was mentioned in a certain patch date. Collectively, 575 unique items were mentioned over 531 different patch notes. Then this data was then pivoted such that every individual item mentioned was a row, with item name and patch date as the column names; there were 1,242 rows in the finalized data set.

Player Count Data

We used the daily player count data obtained from misplaceditems.com by web scraping. This data contained the date and the daily number of active players from 2019 to 2020.

RS3 Gold Value Data

We also supplemented our analysis with the daily USD to RS3 Gold exchange rate from playerauctions.com (March 2019 - December 2020). This dataset contains the average value of RS3 gold in USD for each day, in dollars per million gold. This website had an interactive chart, and we scraped all the available data by using its API call.

Additional Data

We also used daily share prices of the SNP500 from Yahoo! Finance (2008-2020).

Questions

We aimed to understand if purchasing RS3 gold and items can serve as an effective investment vehicle by answering the following three questions:

1. What is the optimal portfolio within Runescape according to MPT?
    a. How does the optimal Runescape portfolio compare to the standard market cap index?
2. How does the mention of an item in the patch notes affect the percent change of that item's price?
3. Does the size of the playerbase have an effect on the price of gold (in USD)?

Statistical Methods

Markowitz Portfolio Theory

The primary assumption of Markowitz Portfolio theory as applied to RS3 is that we can model the investment return of any item as a random variable. This let us define an expected value and variance of the investment return with weekly pricing data as our samples. If we let $w_i \in [0, 1]$ be the proportion of initial capital invested in the i^{th} item S_i, then we can define the expected investment performance in terms of return as $\mathbf{E}(w_1 S_1 + w_2 S_2 + ... + w_m S_m)$ where m is the number of items considered for investment. The means are estimated in the usual way with $\mu_i = \frac{1}{n} \sum_{j=1}^{n} s_{ij}$ where s_{ij} is the return of item i during week j.

In the MAD formulation of Markowitz Portfolio Theory, we also assume that the risk of an item can be captured in its mean absolute deviation from the mean return of the item. The reason for this assumption is twofold: it serves as a proxy measure for variance (a proxy measure for risk) while also allowing for the portfolio optimization problem to remain a linear programming problem. The mean-absolute-deviation for item S_i is calculated by $d_i = \frac{1}{n} \sum_{j=1}^{n} |s_{ij} - \mu_i|$.

We have now established how we can model the expected return and risk of items, but how do we turn these models into a portfolio optimization model? The third assumption gives us the key. We assume that the investor is risk-averse. With the way we have modeled things in accordance with MPT, this means that for a fixed expected return the investor seeks the portfolio which minimizes his mean-absolute-deviation. With this final assumption we can identify that the optimal portfolio is one which maximizes expected return while minimizing the mean-absolute-deviation. Let $|s_{ij} - \mu_i| = D_{ij}$ be the absolute deviation from the mean of item i during week j and let $W = (w_1, ..., w_m)$. Then, we aim to solve

$$\max_{w} \sum_{i=1}^{m} w_i \mu_i - \frac{1}{n} \sum_{j=1}^{n} | \sum_{i=1}^{m} w_i D_{ij} |$$

While this objective function is not entirely linear due to the absolute value in the summation, it is easily turned into a linear function by the addition of new variables and constraints. For exact details on the form of the optimization problem we solved and how the data was transformed to satisfy the optimization problem, please see the appendix.

Game Patches and Item Prices

In question 2, we are trying to understand if there is an impact on the price of an item when it is involved in a patch note. Thus, the null hypothesis is that there is no difference in the normalized percent change in price of an item between an item a week before and a week after a patch mentioning the item.

Before completing our analysis, we transformed the historical pricing data so that every individual item mention was a row, with item name and patch date as the column names. This allowed us to append the historical pricing data to the item mentions by item name and date, including a week before and a week after the patch. Then we pivoted the dataset so that each row had the item, patch date, and the price of the item for the day of the patch as well as for the week before and week after the patch. Using this information we calculated the average price for the week before and the week after the patch and then divided that average price by the price of the item on the day of the patch to normalize the prices, which will be referenced as the normalized price.

For each instance that an item was mentioned in the patch notes, we normalized the time by the date of the patch, such that the time of the patch is t_0. Then we will use regression discontinuity (Fig. 1) to visualize the percent change in each item's price before and following the patch for a week.

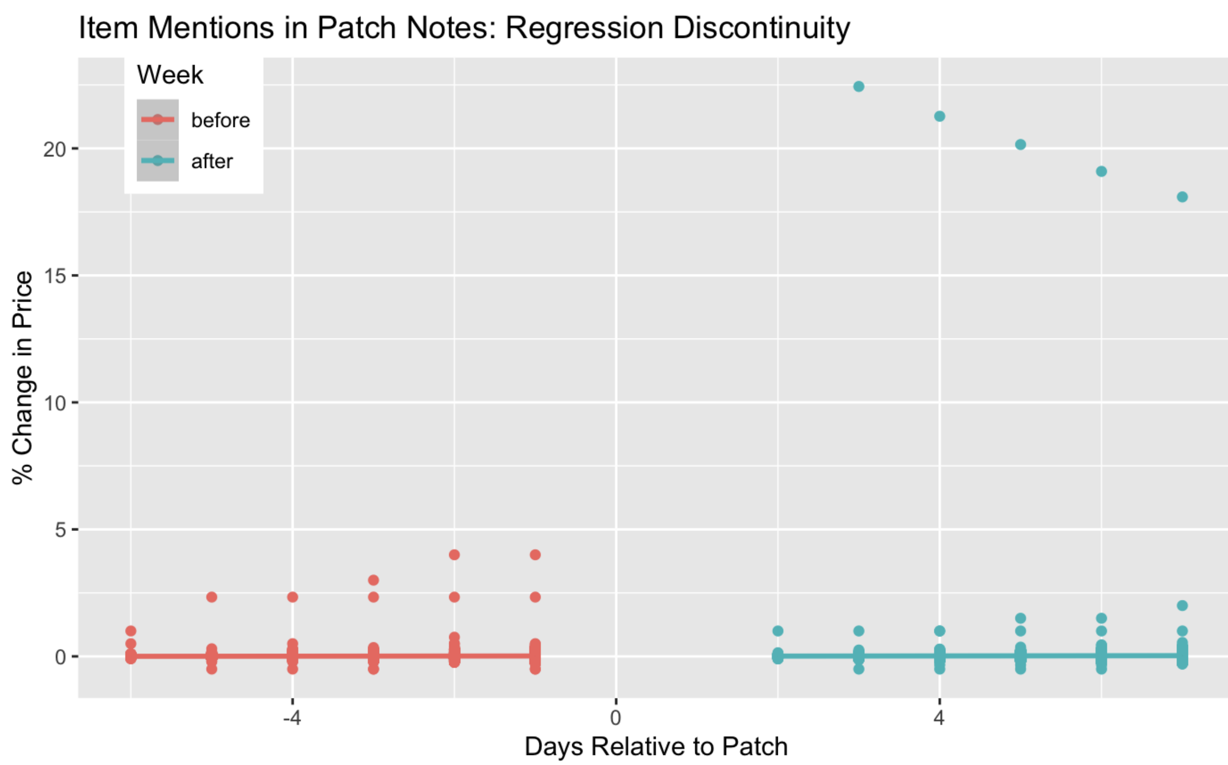

Figure 1: Regression discontinuity of patch item prices using the patch date as the break point

While the simple regression models for percent change in each item's price before and following the patch include residuals, the density of the values makes it impossible to see. By visual inspection, there seems to be little difference in price percent change relative to the item being mentioned in a patch, even as we zoom into the percent change between [0,2], as seen below.

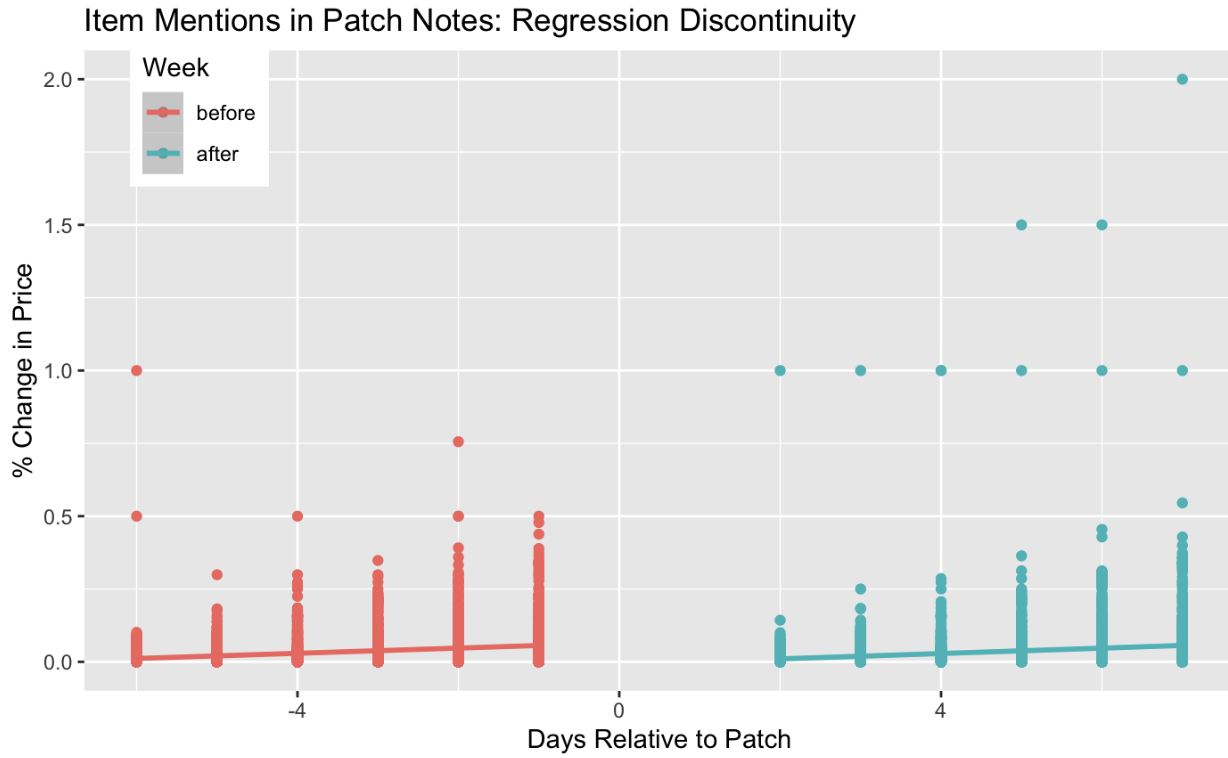

Figure 2: Regression discontinuity of patch item prices focusing on the interval 0-2% change in price.

Since we are comparing the prices before and after a patch, we use a paired t-test with the normalized prices. The paired sample t-test only requires that the observation or difference of each pair is normally distributed and that each item mentioned is independent from each other (Xu).

As part of the examination of the distribution of the observations, we examined outliers. From the summary statistics of the difference of pairs, we could see that there were outliers and NA's in the data set. Coincidentally, there were 144 outliers and, by the 1.5 x Interquartile Rule from the box plot, 144 outliers or 13% of the data (separate from the null values), as seen in Table 1 and Fig. 3, respectively.

Min	1st Qu.	Median	Mean	3rd Qu.	Max	NA's
-15.38625	-0.03414	0.00000	0.14337	0.03165	107.9571	144

Table 1: Summary statistics of difference in normalized prices prior to examining outliers and missing values.

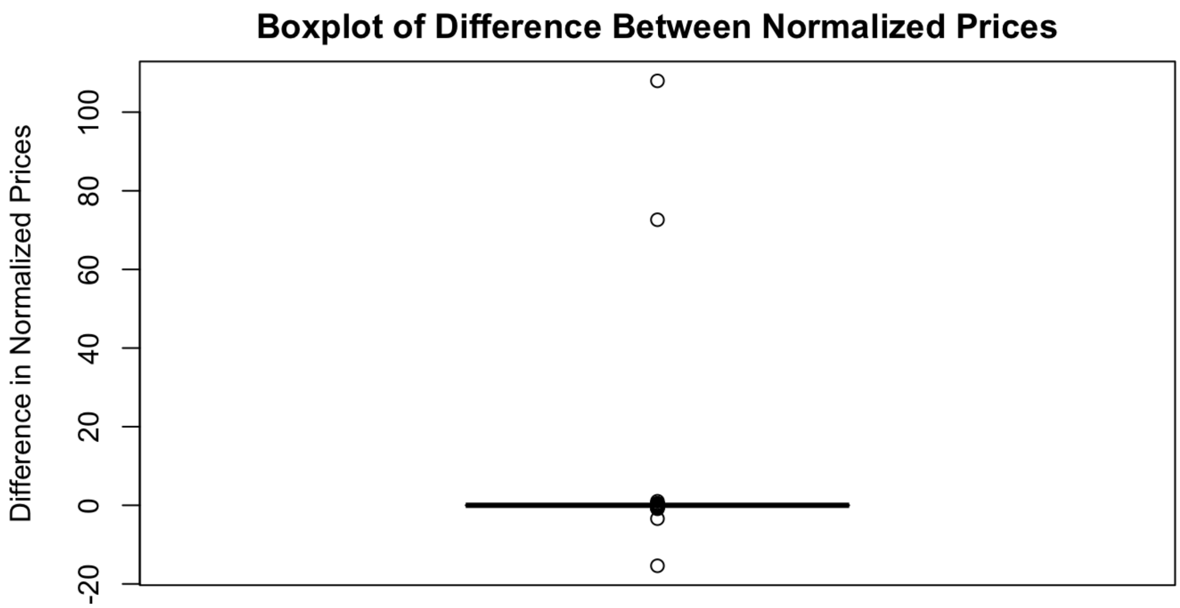

Figure 4: Boxplot of difference between the normalized price between before and after the patch.

To avoid bias by removing all 144 observations, we chose to identify outliers as observations whose absolute difference between normalized prices are greater than 1, which can be interpreted as items whose prices changed by more than 100% of the item at patch. By this definition, there are 5 observations that qualify as outliers, which can be seen in Table 2. Once we removed the null values and the 5 outliers, there remained 1,093 observations, which have the following summary statistics.

Item Name	Patch Date	Difference
Armadylean yellow	4/6/20	-15.386254
Flies	2/24/20	107.957143
Luminite injector	1/7/19	-3.400000
Off-hand dragon throwing axe	10/8/13	72.635220
Shadow glaive	4/4/16	1.032065

Table 2: Items identified as outliers and removed from the data prior to conducting paired t-test.

Min	1st Qu.	Median	Mean	3rd Qu.	Max	NA's
-0.885714	-0.034059	0.00000	-0.004955	0.031207	0.429476	0

Table 3: Summary statistics of difference in normalized prices after removing outliers and null values.

From the histogram of the observations, we can see that its distribution is roughly normal centering at 0 with a longer tail on the left-hand side.

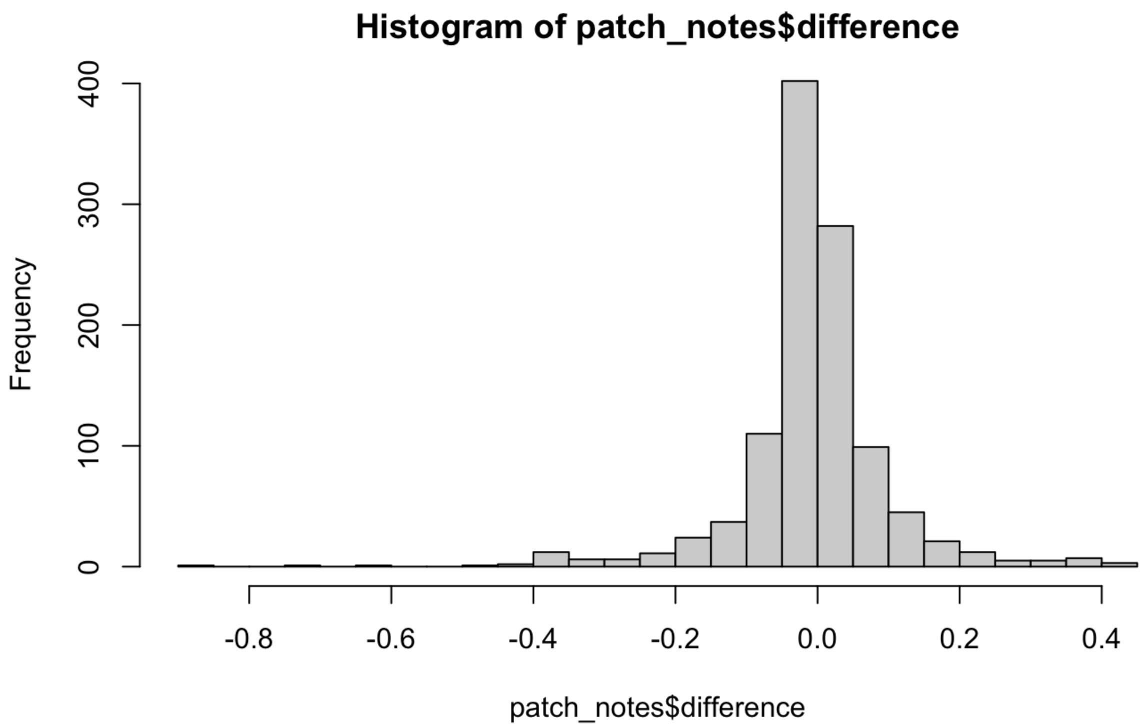

Figure 4: Histogram of difference between normalized price before and after a patch.

Since items may be patched repeatedly, a patch may include multiple items, and some items can be produced by other items, e.g. a necklace with gold, there is slight dependence among item mentions. Due to the timespan of all patches, over more than a decade, and that we normalized the time according to patch frequency (1 week), for this analysis we assumed that the observations are independent of one another. With the data meeting all the assumptions for the paired t-test, we tested the null hypothesis using the normalized prices for before and after the patch.

Player Base Size and Price of Gold

Due to lack of the availability of the active player count data set and RS3 gold value data set, we focused on the player counts and gold values from March 2019 till end of 2020. We computed the weekly averages of both data sets to normalize any day-of-week effects, since there tends to be more players on the weekends than weekdays, and this trend should not have an effect on gold values. In addition, since the data was collected for consecutive weeks, it was not really independent, so we measure the percent changes from the previous week in both datasets. While there might still be dependence between data, this was the best we could do to reduce dependencies. Another thing we did was to offset gold value data by one week to reflect possible delays of the market's reaction to player count changes.

We then fit a linear regression model with % change of player account as the predictor variable and % change of gold value as the response variable. We checked the necessary assumptions for a linear regression model. We also perform a hypothesis test of whether the coefficient of the predictor’s variable is equal to zero.

Results

Markowitz Portfolio Theory

The optimal portfolio as produced by the model contains 224 items and has an annualized expected return of ~73.5%. See the appendix for the full list of items in the portfolio along with their associated investment proportions. We also built a naive 1/n portfolio which invests an equal proportion of our initial capital into all items in RS3. This will serve as a baseline of comparison for the performance of the MAD Markowitz portfolio. The results are summarized in the plot below. Note that the SNP500 is hiding behind the naive portfolio since the differences between the SNP500 and the naive portfolio are not visible at this scale.

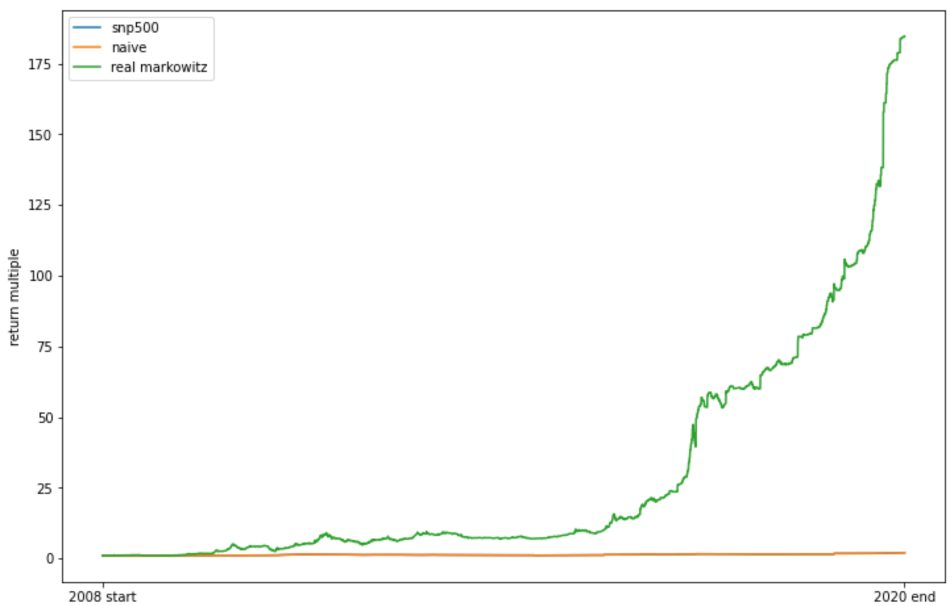

Figure 5: Return multiple of three portfolios: SNP500, naive RunScape 3 items, and MPT RunScape 3 items.

The results are striking. While the naive portfolio had a return over the 12 year period of about 2 times the initial investment, and the SNP500 had a return of about 2.4 times the initial investment, the optimal portfolio in RS3 had a whopping return of 183 times the initial investment.

Game Patches and Item Prices

In the paired t-test comparing normalized item prices before and after a patch, we tested the null hypothesis that the normalized price of an item is the same before and after a patch. The results of that test were t-statistics was -1.5194 and the p-value = 0.129, leading us to conclude that we could not reject the null hypothesis, complete results from R can is shown in Table 4. With that in mind, we can conclude that only knowing that an item is affected by a patch results in no change to the price of an item.

```
        Paired t-test

data:  patch_notes$before_avg_normalized and patch_notes$after_avg_normalized
t = -1.5471, df = 1094, p-value = 0.1221
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.016124786  0.001907064
sample estimates:
mean of the differences
           -0.007108861
```

Table 4: Game patches and item prices Paired T-test results.

Player Base Size and Price of Gold

The linear regression model may be described as (gold price percent increase) = 0.0948 × (player base size percent increase)-0.0031. Thus, for every percent increase in player base size, gold price, on average, increases by 0.0948 percent. For the hypothesis test mentioned in the "statistical methods" section, the t-statistics is 0.757 and the p-value is 0.451, meaning that our linear regression model is not significant. Increases in player count does not have a significant effect on the price of gold (in USD). A scatter plot of the data along with the regression line, as well as the hypothesis test output is shown in Fig. 6 and Table 5, respectively.

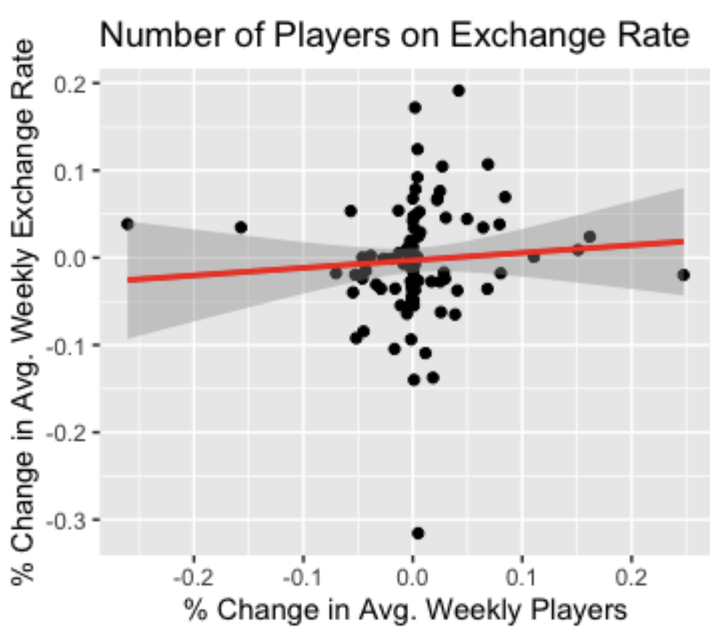

Figure 6: Player Base Size and Price of Gold Linear Regression Model

```
Call:
lm(formula = newCurrency ~ newCounts)

Residuals:
      Min        1Q    Median        3Q       Max
-0.313057 -0.033075 -0.000293  0.031903  0.192442

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.003112   0.007149  -0.435    0.664
newCounts    0.094780   0.125205   0.757    0.451

Residual standard error: 0.06753 on 88 degrees of freedom
Multiple R-squared:  0.00647,   Adjusted R-squared:  -0.00482
F-statistic: 0.5731 on 1 and 88 DF,  p-value: 0.4511
```

Table 6: Player Base Size and Price of Gold Hypothesis Test Results

Discussion

Markowitz Portfolio Theory

The results of the analysis are extremely promising when it comes to showing that RS3 investment might be a feasible investment vehicle. However, a hidden assumption in this analysis is the assumption of perfect liquidity. The RS3 market trades slowly as the game only has 10-20 thousand players on at one given time meaning that even approaching perfect liquidity might not be feasible. In addition, a significant investment in the RS3 marketplace could cause significant price disruptions. This is a potential that we were not able to capture in our analysis.

Game Patches and Item Prices

Based on our data, we fail to reject the null hypothesis that there is no effect on an item's price when the item is mentioned in a set of patch notes. This may seem intuitive but the key factor is that our data did not capture whether a game patch made an item better, worse, easier to get, or harder to get. Lacking this distinction results in seeing patches as the same thing regardless of the effect on the item. Future research should categorize each patch as whether it makes a specific item better, worse, or the same and then repeat our analysis separately for each category. With that limitation, we can conclude that an item being referenced in a patch does not have a significant effect on the price of that item overall and that more specific knowledge about the patch is necessary to conclude anything more meaningful.

Player Base Size and Price of Gold

As discussed previously, we fail to reject the null hypothesis, and our model does not indicate a strong correlation between player base size and price of gold. One limitation we had is the data. The size of our data is quite small because we only have around 2 years of data with one data point per day, and then we have to average data over given weeks. In addition, due to dependence, we need to calculate percent changes of week to week data, and since the weekly player count is pretty steady most of the time, the percent changes are pretty small, so we do not have a wide range of data to perform analysis. If we have known some techniques to better remove the weekly trends of the player base and to better remove dependence, we can perform better analysis.

References

“OSRS Gold: Buy OSRS Gold: OSRS GP.” PlayerAuctions, www.playerauctions.com/osrs-gold/.

“Patch Notes.” RuneScape Wiki, runescape.wiki/w/Patch_Notes.

Runescape Population Avg by Qtr Hour, www.misplaceditems.com/rs_tools/graph/.

“Yahoo Finance - Stock Market Live, Quotes, Business & Finance News.” Yahoo! Finance, Yahoo!, finance.yahoo.com/.

Cesarone, F., Scozzari, A. & Tardella, F. Linear vs. quadratic portfolio selection models with hard real-world constraints. Comput Manag Sci 12, 345–370 (2015). https://doi.org/10.1007/s10287-014-0210-1

Bower, Beth and Wentz, Pamela (2005) "Portfolio Optimization: MAD vs. Markowitz," Rose-Hulman Undergraduate Mathematics Journal: Vol. 6 : Iss. 2 , Article 3

Xu, Manfei et al. “The Differences and Similarities Between Two-Sample T-Test and Paired T-Test.” Shanghai archives of psychiatry vol. 29,3 (2017): 184-188. doi:10.11919/j.issn.1002-0829.217070

Appendix

MAD Markowitz Formulation

As discussed in section 1, to derive the optimal portfolio we aim to solve the (close to) linear optimization problem.

$$\max_{W} \sum_{i=1}^{m} w_i \mu_i - \frac{1}{n} \sum_{j=1}^{n} \left| \sum_{i=1}^{m} w_i D_{ij} \right|$$

We can transform this optimization into an equivalent linear one which can be solved using the simplex method.

$$\begin{aligned}
\max_{W} \quad & \sum_{i=1}^{m} w_i \mu_i - \frac{1}{n} \sum_{j=1}^{n} y_j \\
\text{s.t.} \quad & -y_j \leq \sum_{i=1}^{m} w_i D_{ij} \leq y_j \\
& \sum_{i=1}^{m} w_i = 1 \\
& w_i \geq 0 \quad \text{for all i} \\
& y_j \geq 0 \quad \text{for all weeks j}
\end{aligned}$$

Transforming for Scipy Simplex

We will use Scipy's SIMPLEX implementation. In order to do this, we need to reformulate our LP to match Scipy's preferred format. This format has the following form.

$$\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & A_{ub} x \leq b_{ub} \\
& A_{eq} x = b_{eq}
\end{aligned}$$

where x is a vector of decision variables; c, b_{ub}, b_{eq}, are vectors; and A_{ub} and A_{eq} are matrices.

Transforming the Objective Function

Our objective function is of the form

$$\max_{W} \quad \sum_{i=1}^{m} w_i\mu_i - \frac{1}{n}\sum_{j=1}^{n} y_j = w_1\mu_1 + ... + w_m\mu_m - 1/ny_1 - ... - 1/ny_n$$

Let the vector

$$c^T = (\mu_1, ..., \mu_m, -1/n, ..., -1/n)$$

which contains n of the $-1/n$ term. Then our decision variable vector is of the form

$$x = (w_1, ..., w_m, y_1, ..., y_n)^T$$

To convert our maximization problem into a minimization problem, we will solve min $-c^T x$.

Transforming the Constraints

Let

$$D = \begin{pmatrix} D_{11} & \cdots & D_{m1} \\ \vdots & & \vdots \\ D_{1n} & \cdots & D_{mn} \end{pmatrix}$$

and also let $I_{a,b}$ be the identity matrix in $\mathbb{R}^{a,b}$. Then, define $A_{ub} \in \mathbb{R}^{(3n+m),(m+n)}$ a block matrix by

$$A_{ub} = \left(\begin{array}{c|c} -D & -I_{n,n} \\ \hline D & -I_{n,n} \\ \hline \multicolumn{2}{c}{-I_{(m+n),(m+n)}} \end{array}\right)$$

Recall that $x = (w_1, ..., 2_m, y_1, ..., y_n)^t$. Then, $A_{ub}x \leq 0$ satisfies the constraints involving inequalities. The last step is to enforce that $\sum_i w_i = 1$. Scipy requires that this be expressed in the form $A_{eq}x = b_{eq}$. This is accomplished by simply letting $A_{eq} = I_{(m+n),(m+n)}$ and $b_{eq} = (1, ..., 1)^T$.

Putting it all together

We will use scipy on the linear program

$$\begin{aligned} \text{min} \quad & -(\mu_1, ..., \mu_m, -1/n, ..., -1/n)(w_1, ..., w_m, y_1, ..., y_n)^T \\ \text{s.t.} \quad & \left(\begin{array}{c|c} -D & -I_{n,n} \\ \hline D & -I_{n,n} \\ \hline \multicolumn{2}{c}{I_{(m+n),(m+n)}} \end{array}\right)(w_1, ..., w_m, y_1, ..., y_n)^T \leq 0 \\ & I_{(m+n),(m+n)}(w_1, ..., w_m, y_1, ..., y_n)^T = 1 \end{aligned}$$