Daniel Gysi

Dr. Hallenbeck

CIS-241

October 27, 2020

Exploratory Analysis of Data Recovered from the Core Protect Database of the SalC1 Anarchy Server

Introduction

When running Minecraft servers, careful care and attention must be put toward the hardware and software resources involved. This is especially true for anarchy servers, where the requirement to keep all the players on a single world, which must run on a single thread on a single machine, requires intense optimization to meet demands for player count. By analyzing the player activity of an anarchy server over time, we can make better approximations for what players will be doing and when so that optimizations can take place as necessary over a server's lifetime. Additionally, much of this same information would be useful to the players on such a server as they conduct their activities.

The SalC1 anarchy server was a Minecraft server begun by Minecraft Youtuber SalC1 in late February of 2020 that lasted approximately two months. It consisted of a single world that ran on a modified instance of a Minecraft 1.12.2 server. When the server was shutdown, the administrators published all the data collected by the server and its various plugins. This included the world files themselves, a database of player kills, and a data base of player activity compiled by the Core Protect plugin. Core Protect is a plugin that aims to prevent griefing by logging and making reversable all tangible player activity including but not limited to block places, block breaks, player kills, entity kills, and player session logs. My dataset was derived

from that database, consisting of player session logs, sign data, block placement, and block destruction data for the first 1,000,000 blocks placed and destroyed by players (“SalC1”, “CoreProtect”).

After the data was published, the server’s community did some work to make much of it more accessible, like creating a website to view overhead maps of the server’s world and examine player inventories, publishing some statistics about player activity (who had the most kills, who was the first player to log on, etc), and compiling a list of notable locations. (“#salc1-Anarchy-Data-Mining”, “Salc1 Datamine”, “SalC1 anarchy server land marks”). These were all derived from the world files and the other database. There is record of some work having be done on deciphering the schemas of the Core Protect database, but the repository where it was hosted has since been lost. (“#salc1-Anarchy-Data-Mining”) As far as I am aware, there is no surviving work on the contents of the database.

The questions I’m interested in answering using the methods learned in class include: Can we identify any of the manually gathered landmarks on the server using clustering techniques on block placement data? How did player activity change over time and is there any correlation between player activity spikes and SalC1’s published videos? Can association rule mining reveal information about which blocks are likely to be placed together? Can association rule mining on sign data provide practical utility?

Data

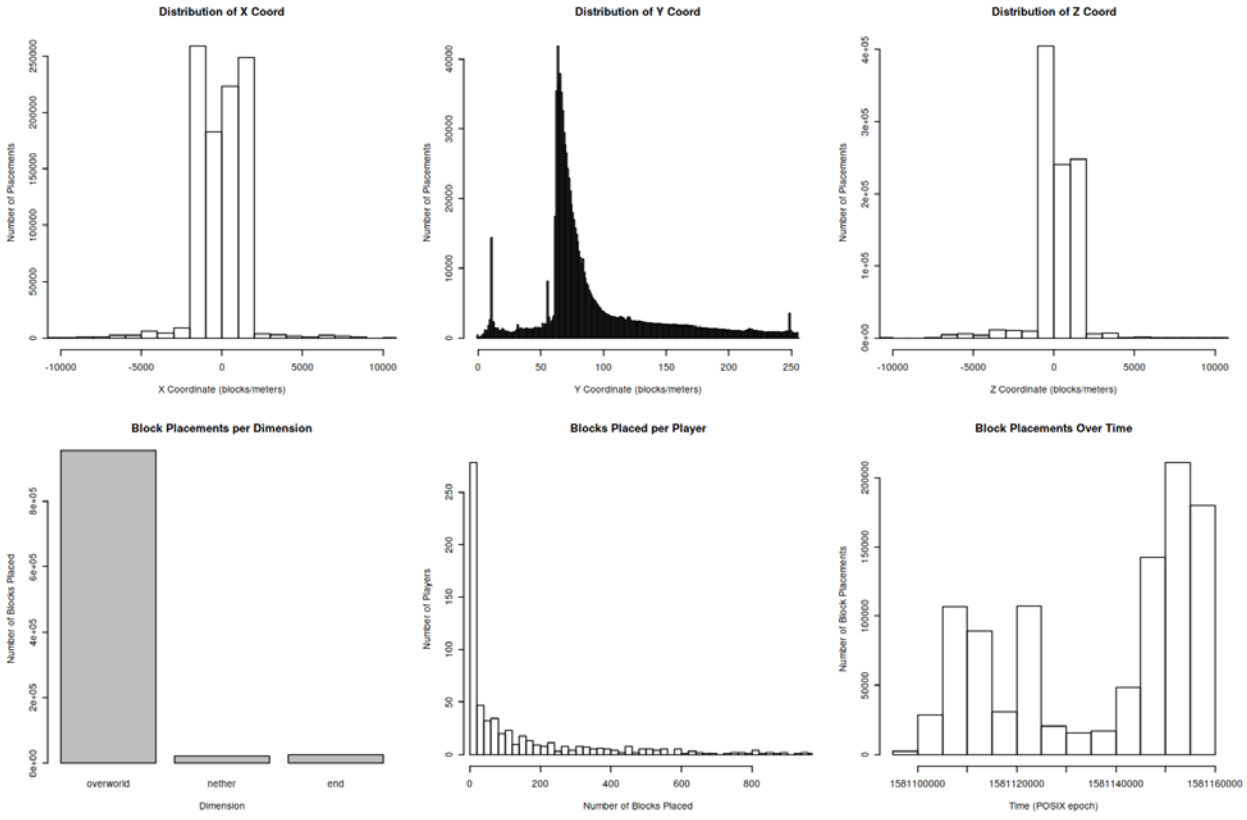
My data consists of four spreadsheets roughly parallel to the original database tables containing information I extracted from the database with SQL queries and lightly processed with Python. When entering them into R, I separated them into 7 data frames covering block

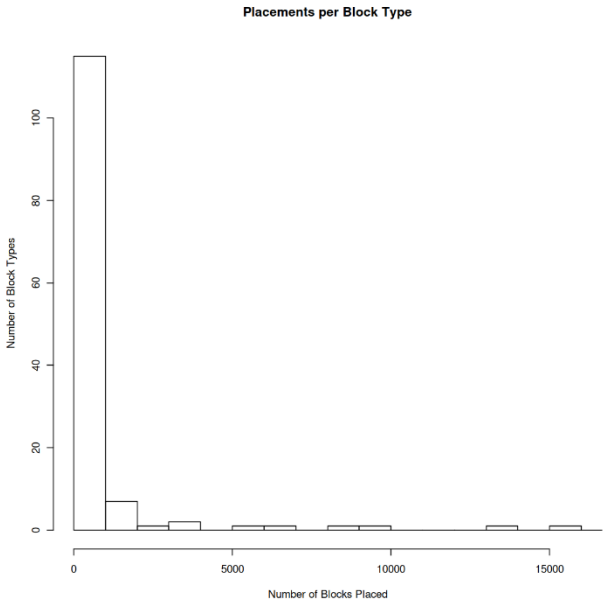
placements, block breaks, player session data (log-in and log-out actions), log-in actions, log-out actions, and sign data.

The block placement data frame consists of the type of block placed as a category, the name of the player who placed it as a category, the date and time of the event represented as a string, the dimension the block was placed in as a category between “overworld”, “nether”, and “end”, the x coordinate where the block was placed as an integer representing the number of blocks or meters from the origin in the East direction, the y coordinate where the block was placed as an integer representing the number of blocks or meters from the origin in the upward direction, and the z coordinate where the block was placed as an integer representing the number of blocks or meters from the origin in the North direction. This data frame has exactly 1,000,000 rows representing block placements with no missing data. We see a wide range of X and Z coordinates centered on 0. This makes sense as players spawn at approximately 0,0 and it takes significant effort to move beyond a few thousand blocks of that point. We see that the Y coordinate is roughly normal centered at 64 but ranges between -1 and 255 with a spike at 11. This makes sense as y=64 is ground level and the y coordinate of a block cannot be lower than -1 or higher than 255. The additional spike at y=11 is consistent with players mining for diamonds as that is widely known to be the optimal y level to find them. We would expect roughly this same behavior for coordinate data in all other data frames from this database. We see that block placements over time is roughly consistent with player activity over time from the sessions data frame. We see that blocks based per player shows an exponential curve which speaks to the discrepancy in player activity on an individual level. If this were plotted as a histogram of number of players per number of blocks placed, we would expect to see a normal curve. We see that most block placements are in the overworld. This makes sense as players spawn in the

overworld and most gameplay takes place here. We would expect this dimensional pattern for all of the data in the dataset. Finally, we see a sharp exponential curve in the number of blocks per block type. This is due to liquids and the high prevalence of easily acquired blocks. Liquids like water and to a lesser extent lava can self-replicate and all the resulting blocks will be attributed to the player that placed the original blocks. Additionally, some blocks, like cobblestone and wood planks, are significantly easier to acquire in bulk than other blocks so we would expect and we see that these blocks are placed more often.

Coord	**Min.**	**1st Quart.**	**Median**	**Mean**	**3rd Quart.**	**Max.**
X	-62073	-1317	2	544.3	1496	47797
Y	-1	66	75	92	105	255
Z	-434114	-269	1	-262.6	1366	66707

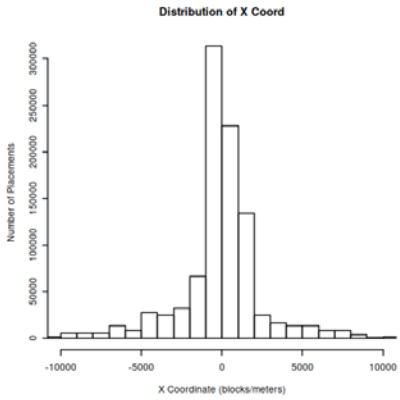


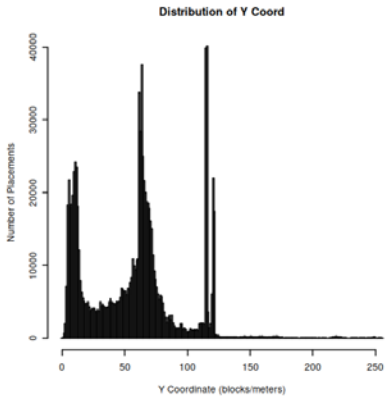


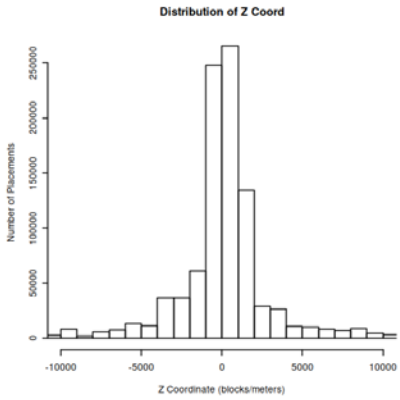
The block break data frame consists of the type of block broken as a category, the name of the player who broke it as a category, the date and time of the event represented as a string, the dimension the block was broken in as a category between “overworld”, “nether”, and “end”, the x coordinate where the block was broken as an integer representing the number of blocks or meters from the origin in the East direction, the y coordinate where the block was broken as an integer representing the number of blocks or meters from the origin in the upward direction, and the z coordinate where the block was broken as an integer representing the number of blocks or meters from the origin in the North direction. This data frame has exactly 1,000,000 rows representing block breaks with no missing data. The distribution of our data is mostly consistent with the block placement data so we won’t repeat the same rationale. Though we note the higher peaks at lower y-levels for block breaking due to mining. We don’t have a clear explanation for the peak at y=128.

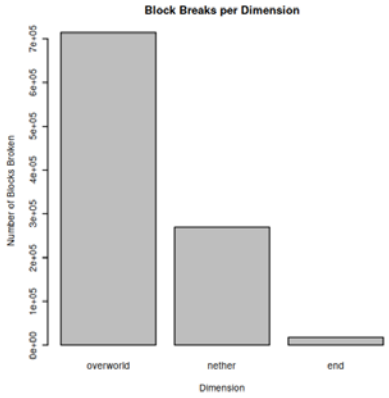
Coord	**Min.**	**1st Quart.**	**Median**	**Mean**	**3rd Quart.**	**Max.**

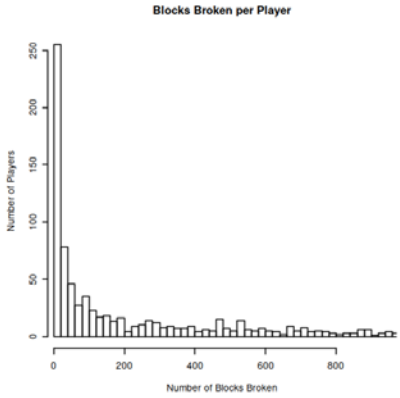
X	-85231	-633	0	228.4	1023	61846
Y	1	22	62	58.61	75	255
Z	-196731	-727	33	-458.3	1098	84869

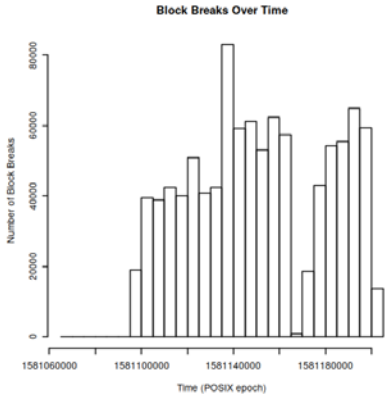


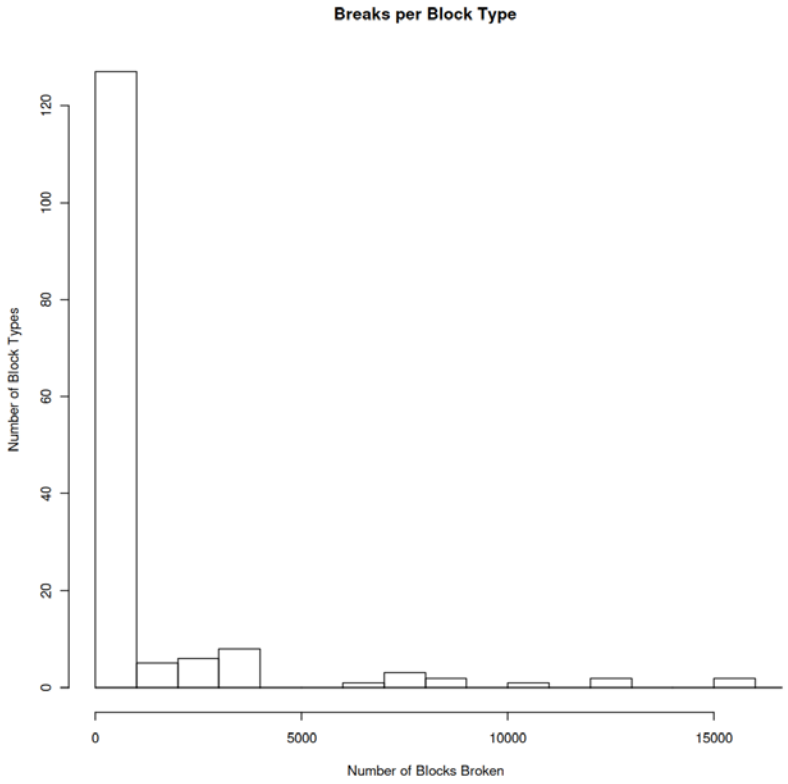








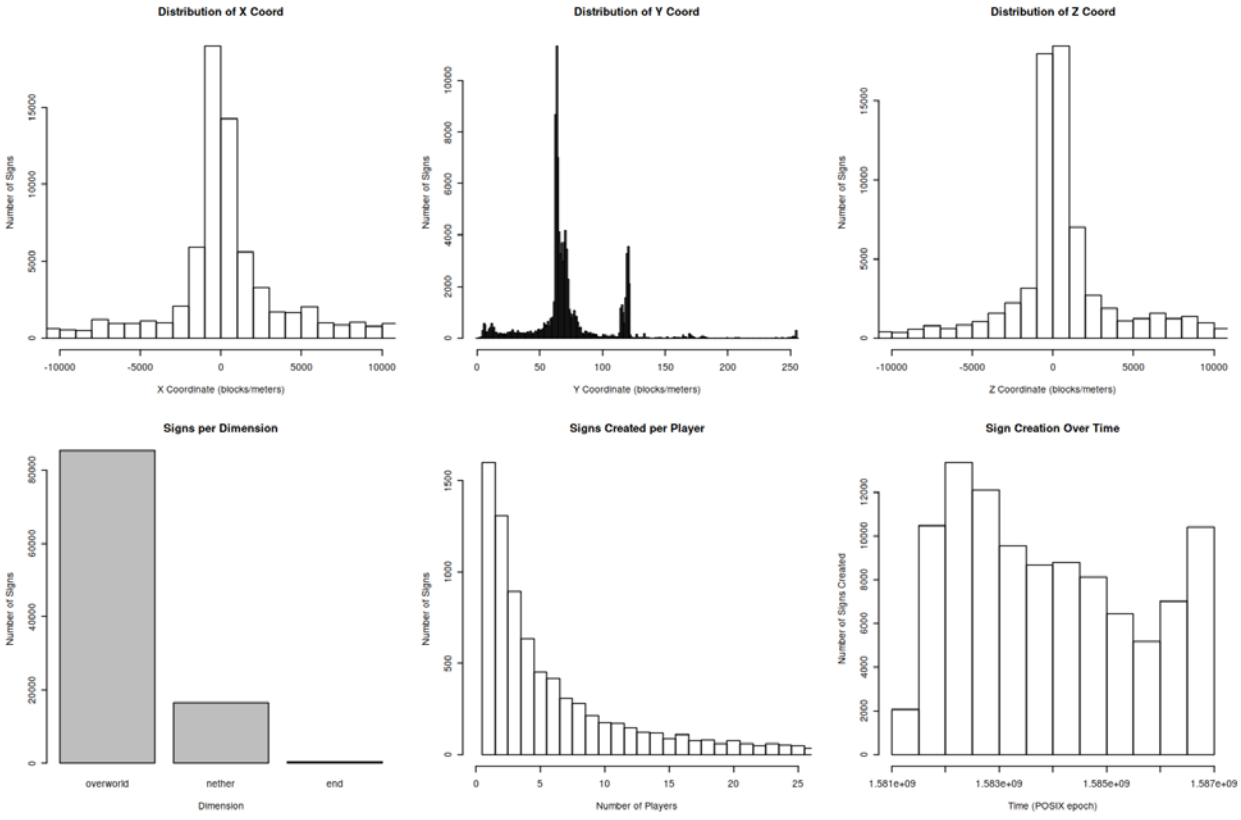


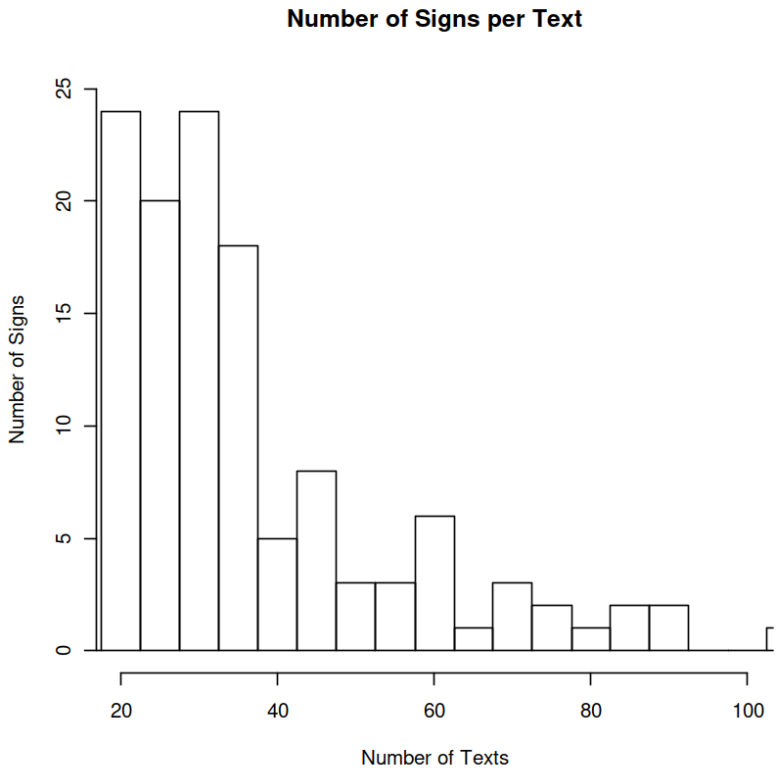


The sign data frame consists of the text on the sign as a string, the name of the player who created the sign as a category, the date and time the sign was created as a string, the dimension where the sign existed as a category between “overworld”, “nether”, and “end”, the x coordinate where the sign was placed as an integer representing the number of blocks or meters from the origin in the East direction, the y coordinate where the sign was placed as an integer representing the number of blocks or meters from the origin in the upward direction, and the z coordinate where the sign was placed as an integer representing the number of blocks or meters from the origin in the North direction. This data frame had 130,000 rows representing signs before cleaning, and 100,000 after blank signs were removed. We note that the distribution of the data is consistent with the general block placement data, which we would expect as sign placements are a subset of blocks. We do note the additional data of the signs per text histogram which points to a roughly normal distribution of the number of texts per number of signs, though

out of frame on the graph are sharp peaks due to spam that occurred when a small handful of players placed hundreds of signs with the same texts.

Coord	Min.	1st Quart.	Median	Mean	3rd Quart.	Max.
X	-13001781	-2151	0	-2667	4196	29999983
Y	1	63	67	73.76	77	255
Z	-15015226	-1427	116	81284	5190	29999983

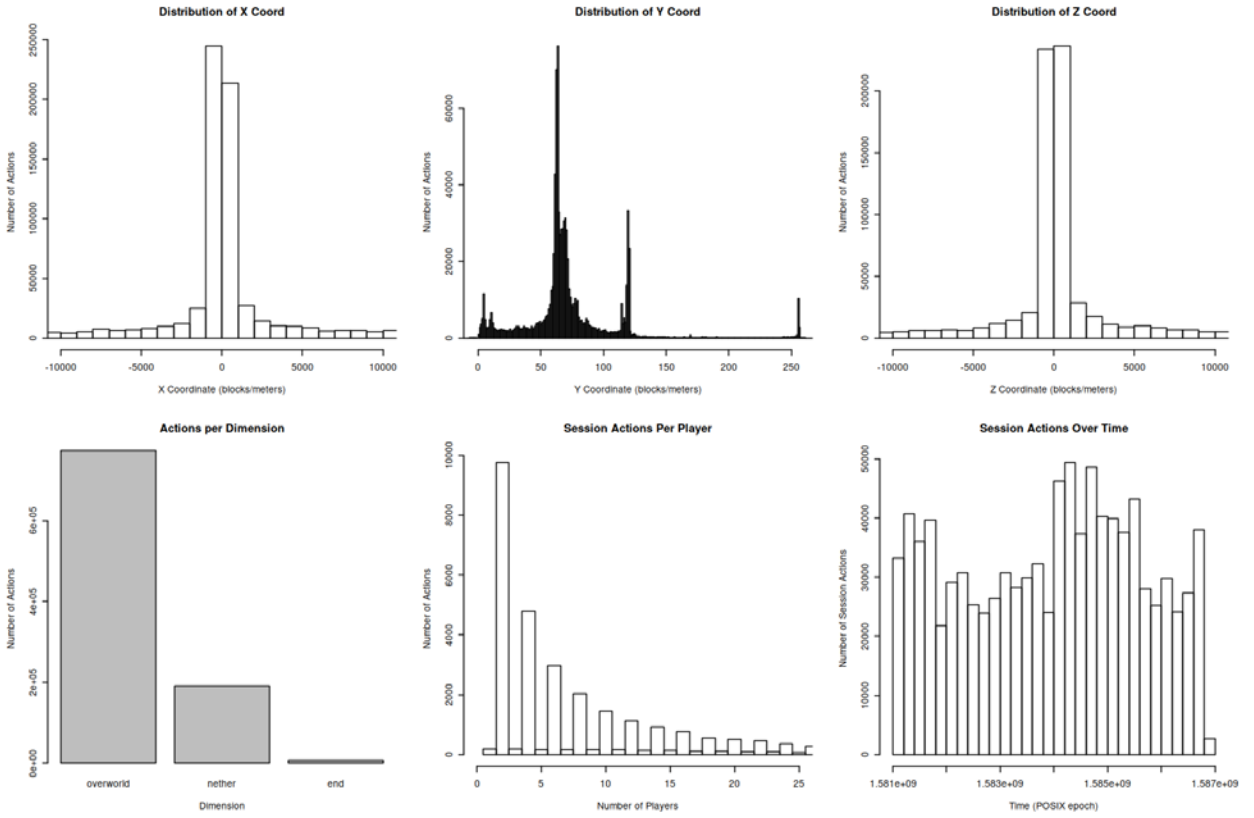




The session data frame consists of the action being recorded as a category (“in” or “out), the name of the player who did the action, the date and time the action occurred as a string, the dimension where the action occurred as a category between “overworld”, “nether”, and “end”, the x coordinate where the action occurred as an integer representing the number of blocks or meters from the origin in the East direction, the y coordinate where the action occurred as an integer representing the number of blocks or meters from the origin in the upward direction, and the z coordinate where the action occurred as an integer representing the number of blocks or meters from the origin in the North direction. This data frame has 970,000 rows representing player log in/out actions with no empty items. We note that the distribution of the data is roughly consistent with the analogous distributions for the block placement and break data frames. Though we do note that the y-value for logging in and out is not constrained to the grid that blocks are so there is a small number of actions that occurred below and above the world due to

flying and exploits. We also note that the session actions per player has an alternating pattern because session actions necessarily come in pairs (one log in for one log out) except in the rare case of a server crash.

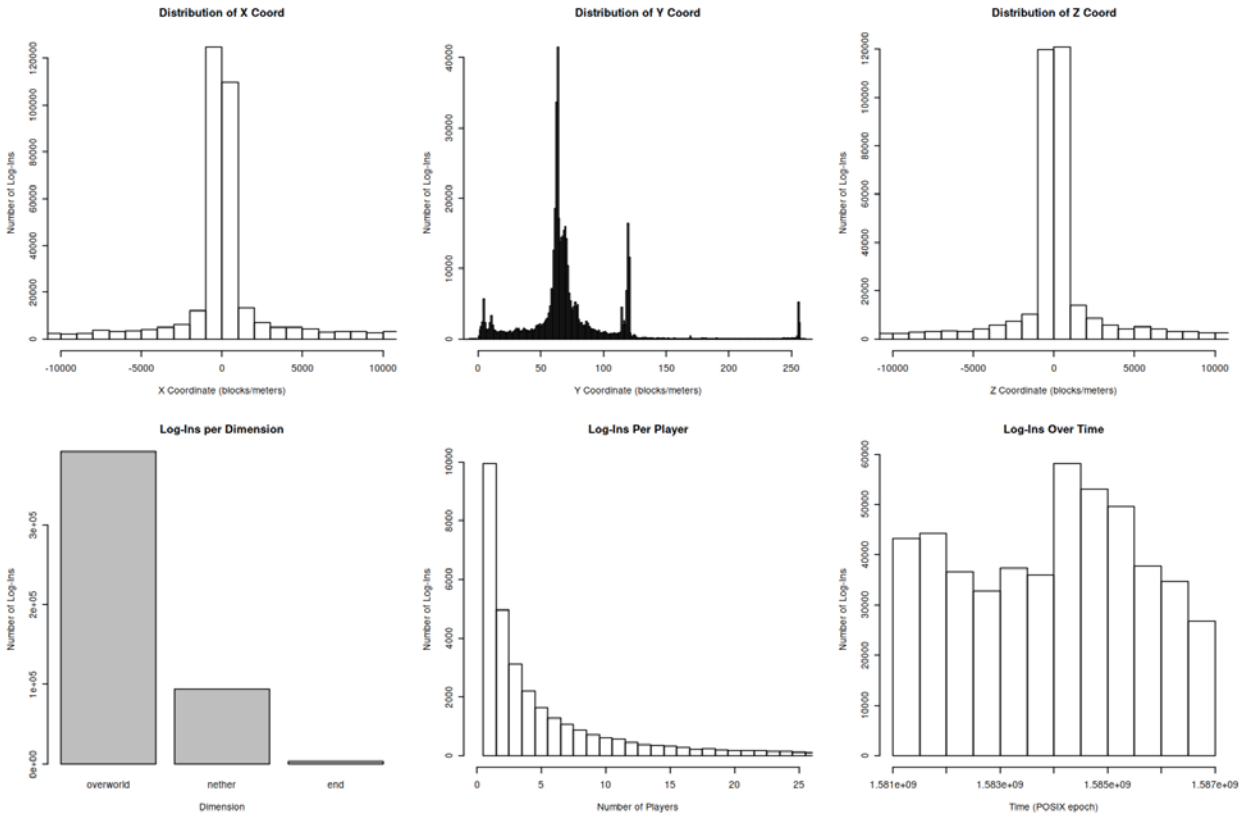
Coord	Min.	1st Quart.	Median	Mean	3rd Quart.	Max.
X	-13001795	-753	4	8117	2673	30000000
Y	-2669	61	66	74.23	79	14610
Z	-15015097	-481	17	-5068	2320	9999965



The log-in data frame consists of the name of the player who logged in, the date and time the player logged in as a string, the dimension where the player logged in as a category between “overworld”, “nether”, and “end”, the x coordinate where the player logged in as an integer representing the number of blocks or meters from the origin in the East direction, the y

coordinate where the player logged in as an integer representing the number of blocks or meters from the origin in the upward direction, and the z coordinate where the player logged in as an integer representing the number of blocks or meters from the origin in the North direction. This data frame has 489,643 rows representing player log-in actions. The distributions of this data is consistent with the session data frame in general, as this is a subset.

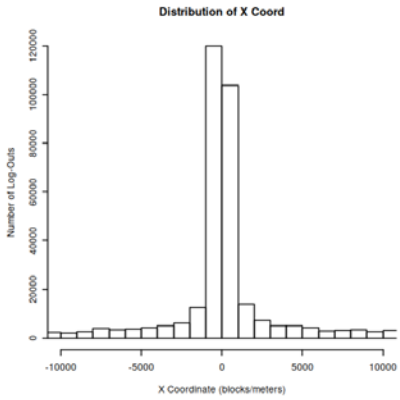
Coord	**Min.**	**1st Quart.**	**Median**	**Mean**	**3rd Quart.**	**Max.**
X	-13001795	-679	5	8061	2500	29999999
Y	-2669	61	66	74.23	79	299
Z	-15015097	-435	17	-5076	2284	9999965

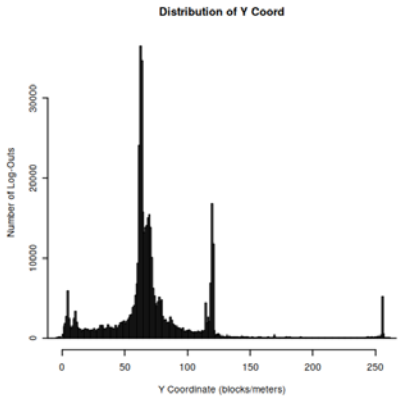


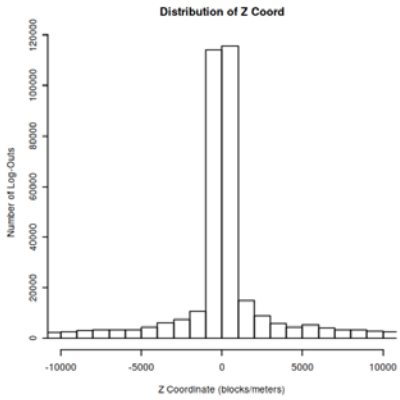
The log-out data frame consists of the name of the player who logged out, the date and time the player logged out as a string, the dimension where the player logged out as a category

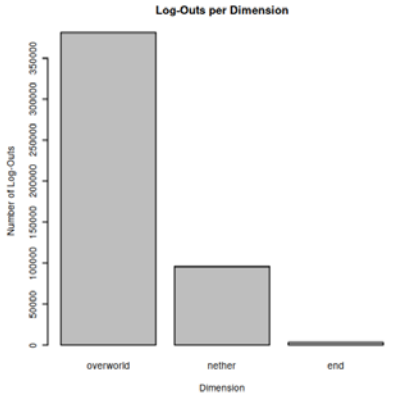
between “overworld”, “nether”, and “end”, the x coordinate where the player logged out as an integer representing the number of blocks or meters from the origin in the East direction, the y coordinate where the player logged out as an integer representing the number of blocks or meters from the origin in the upward direction, and the z coordinate where the player logged out as an integer representing the number of blocks or meters from the origin in the North direction. This data frame has 480,051 rows representing player log-out actions. The distributions of are consistent with the session data as a whole as this is a subset.

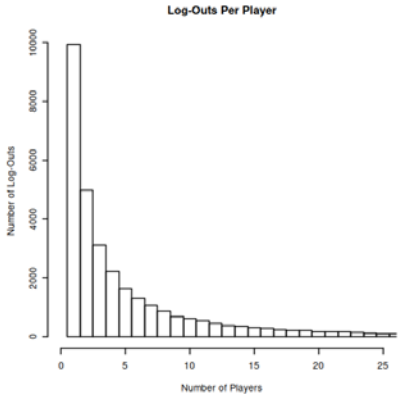
Coord	**Min.**	**1st Quart.**	**Median**	**Mean**	**3rd Quart.**	**Max.**
X	-13001795	-816	4	8174	2870	30000000
Y	-2665	61	66	74.23	79	14610
Z	-15015097	-527	18	-5061	2442	9999965

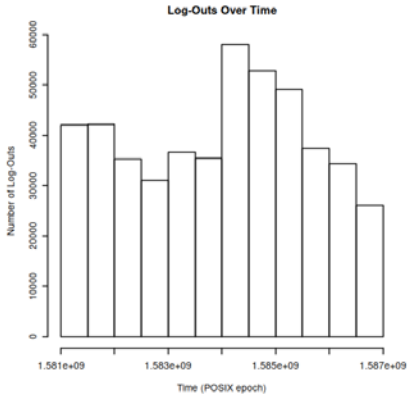

Distribution of X Coord
Number of Log-Outs
0
20000
40000
60000
80000
100000
120000
-10000
-5000
0
5000
10000
X Coordinate (blocks/meters)


Distribution of Y Coord
Number of Log-Outs
0
10000
20000
30000
0
50
100
150
200
250
Y Coordinate (blocks/meters)


Distribution of Z Coord
Number of Log-Outs
0
20000
40000
60000
80000
100000
120000
-10000
-5000
0
5000
10000
Z Coordinate (blocks/meters)


Log-Outs per Dimension
Number of Log-Outs
0
50000
100000
150000
200000
250000
300000
350000
overworld
nether
end
Dimension


Log-Outs Per Player
Number of Log-Outs
0
2000
4000
6000
8000
10000
0
5
10
15
20
25
Number of Players

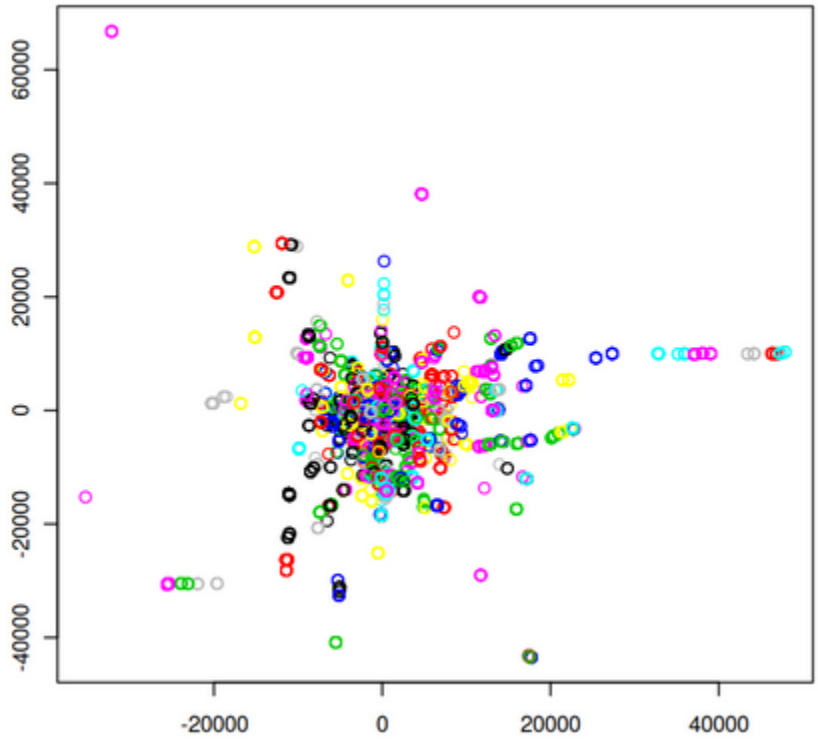

Log-Outs Over Time
Number of Log-Outs
0
10000
20000
30000
40000
50000
60000
1.581e+09
1.583e+09
1.585e+09
1.587e+09
Time (POSIX epoch)

Analysis

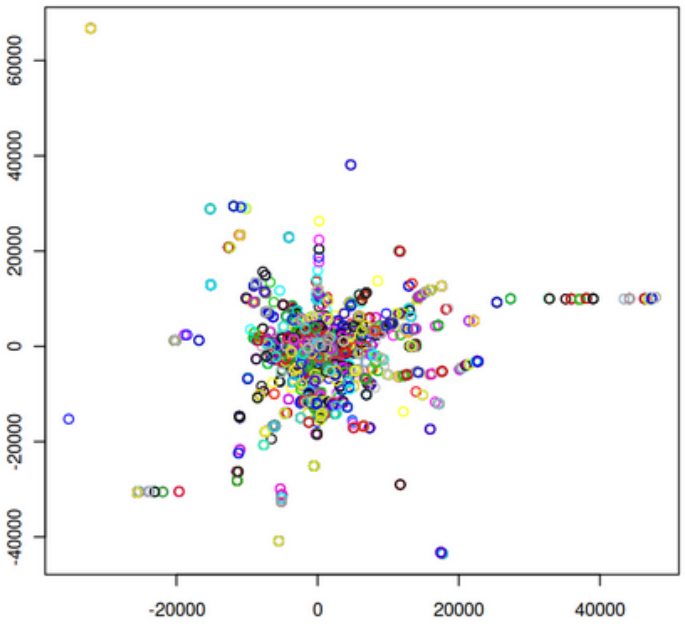
The first question we examine is whether clustering can be used on coordinate data to identify locations of player builds. This question proved difficult to answer due to the size of our data. Firstly, we had to omit much of the data on block placements due to the quantity of it. This means that if we were to be successful, we would only be able to identify builds that were created within the first 1,000,000 block placements.

Due to the varying sizes of builds, hierarchical clustering was deemed the best approach here. However, the algorithm requiring a cartesian product of our points makes the problem intractable to do on the raw data. Perhaps with a setup that uses more memory, hierarchical clustering could be used to greater effect. In our case, we attempted to continue the project by first performing k-means clustering with k equal to the number of unique players that placed blocks within the first 1,000,000 placements and then perform hierarchical clustering on the centers of the k-means clusters. The value for k was slightly arbitrary as it wasn't feasible to rerun the algorithm for multiple values of k. We then ran the hierarchical clustering algorithm on these clusters and chose 300 as the number of hierarchical clusters. Due to many of the factors above, including additionally the vast amount of noise in the spawn region and the limited amount of data we had, the resulting clusters did not easily align with our list of server landmarks when checking manually. Our approach may still be valid, but to do it optimally on a full set of data would require more computational resources and/or time than is feasible for a problem of this limited utility.

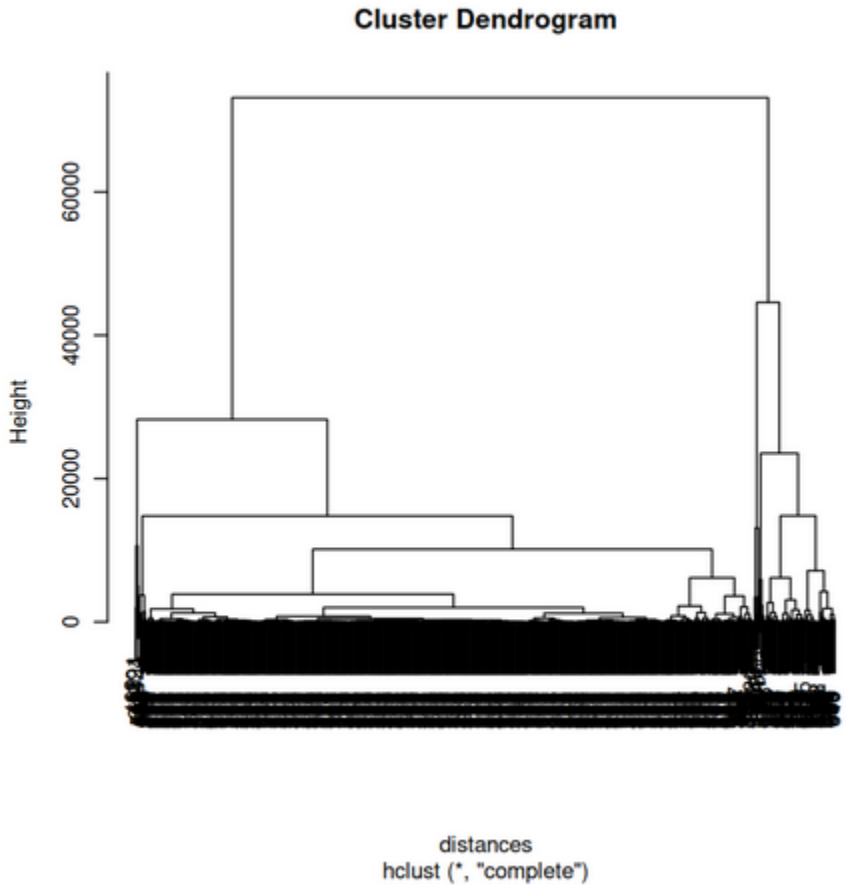
K-Means Clusters:



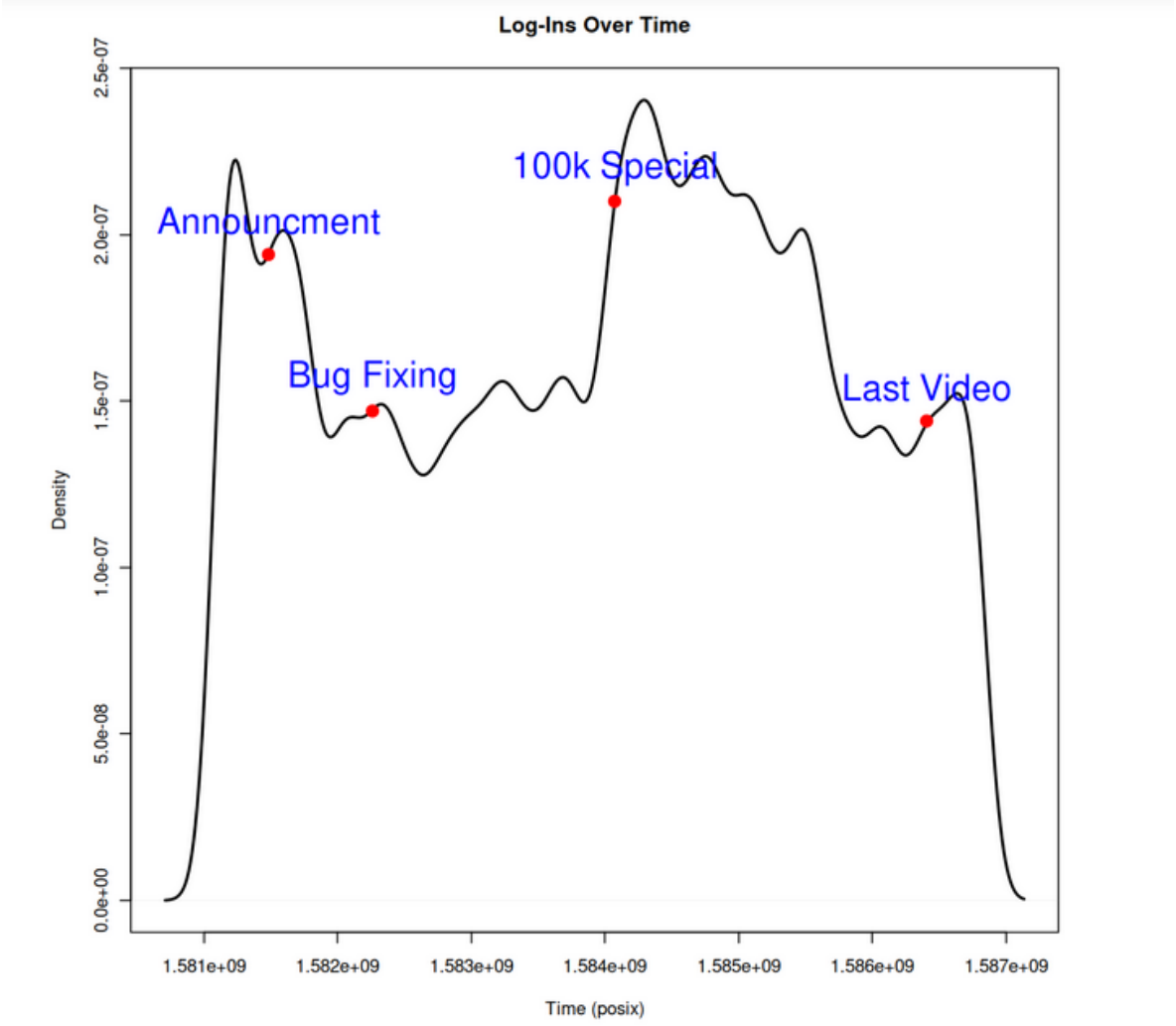
Hierarchical Clusters:



Hierarchical Cluster Dendrogram:



Our second question is whether we can see a clear correlation between player activity and the videos posted by the YouTube channel SalC1 that owned the server. This question was more straightforward to answer by analyzing and comparing distributions. When we plot the log-in data as a density curve, we can see some clear peaks in the data. The biggest hurdle is handling the dates and comparing the spikes to posted videos. When some server-related videos are overlaid on top of the density curve, we can see some patterns.



We would expect the video uploads to be in the valleys before the spikes, but we don't necessarily see that. We would especially expect the server announcement video to be posted before the first spike of player activity, but we see that it only correlates with a smaller spike. The other videos seem to fall on upward slopes that lead into spikes when we would expect them to be in the valleys. The explanation for this unexpected behavior may be that videos had less influence on server activity than expected and the server was announced by some other means to start the initial spike, or there was some discrepancy in how the times were handled, possibly a misrepresentation of the correct time zone.

Our third question is whether association rule mining can be used to determine to uncover information about players' block pallets. To find information on this, we create association rules on block placement players and block types. When we do this without any other restrictions beyond a small support to ensure the rule is present, we see a lot of rules regarding liquids. This is due to the self-duplicating property of liquid blocks so their placement has a high representation in our data set. To uncover more interesting information, we restrict the rules so they can't include any form of liquid. At this point, we come up with some more interesting rules. For one, we see the commonality of cobblestone represented in multiple rules that state that certain players are most likely to place cobblestone. We also uniquely see a relationship with endstone to and from the player "Barendome." When we plot this information, we find that this player placed a long trail of endstone in the end dimension. Since endstone is a relatively rare block and this player didn't place much else, this rule has a high confidence. Ultimately then, this reveals that our method does work, but it may be biased toward uncovering rules with blocks that aren't placed often by the average player.

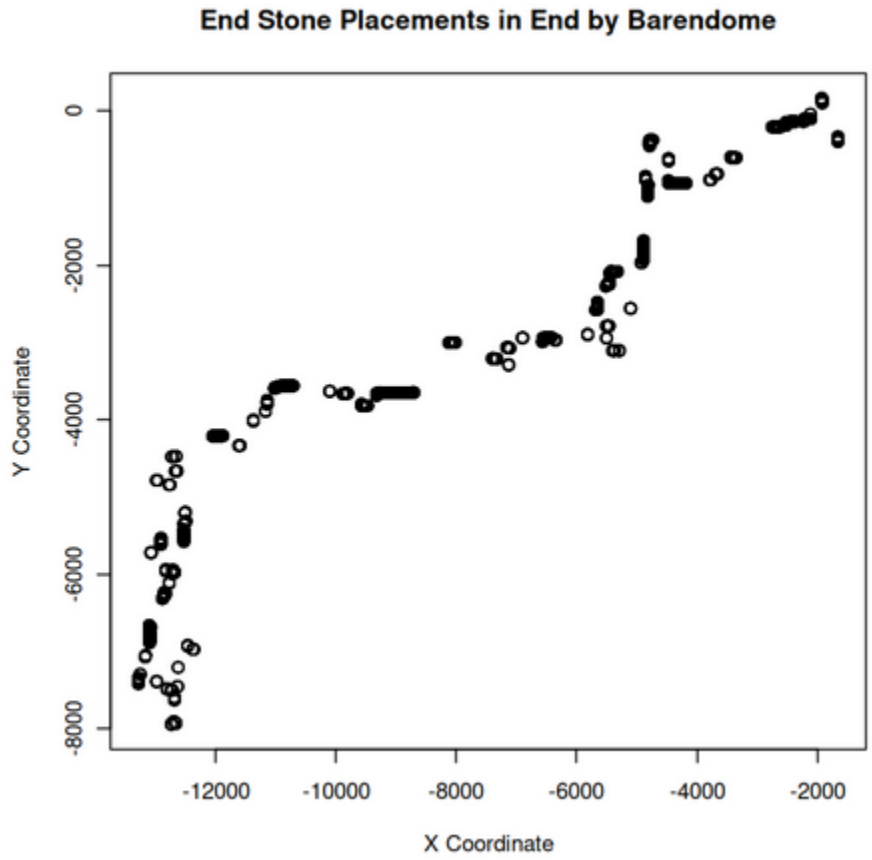
Association Rules Including Liquids:

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{place_data.block=lava}	=>	{place_data.user=Check9light}	0.052494	0.5931928	0.088494	2.408005	52494
[2]	{place_data.user=Check9light}	=>	{place_data.block=lava}	0.052494	0.2130940	0.246342	2.408005	52494
[3]	{place_data.user=lavablade02}	=>	{place_data.block=stationary_water}	0.051933	0.5590927	0.092888	1.224920	51933
[4]	{place_data.block=stationary_water}	=>	{place_data.user=lavablade02}	0.051933	0.1137804	0.456432	1.224920	51933
[5]	{place_data.user=artofcroissant}	=>	{place_data.block=stationary_water}	0.073616	0.5956999	0.123579	1.305123	73616
[6]	{place_data.block=stationary_water}	=>	{place_data.user=artofcroissant}	0.073616	0.1612858	0.456432	1.305123	73616
[7]	{place_data.user=GermanEngineer12}	=>	{place_data.block=stationary_lava}	0.085845	0.3667618	0.234062	1.445076	85845
[8]	{place_data.block=stationary_lava}	=>	{place_data.user=GermanEngineer12}	0.085845	0.3382374	0.253801	1.445076	85845
[9]	{place_data.user=GermanEngineer12}	=>	{place_data.block=stationary_water}	0.126361	0.5398612	0.234062	1.182786	126361
[10]	{place_data.block=stationary_water}	=>	{place_data.user=GermanEngineer12}	0.126361	0.2768452	0.456432	1.182786	126361
[11]	{place_data.user=Check9light}	=>	{place_data.block=stationary_lava}	0.069575	0.2824326	0.246342	1.112811	69575
[12]	{place_data.block=stationary_lava}	=>	{place_data.user=Check9light}	0.069575	0.2741321	0.253801	1.112811	69575
[13]	{place_data.user=Check9light}	=>	{place_data.block=stationary_water}	0.115777	0.4699848	0.246342	1.029693	115777
[14]	{place_data.block=stationary_water}	=>	{place_data.user=Check9light}	0.115777	0.2536566	0.456432	1.029693	115777

Association Rules Discluding Liquids:

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{place_data.user=BarenDome}	=>	{place_data.block=ender_stone}	0.005860	0.9401572	0.006233	95.071011	5860
[2]	{place_data.block=ender_stone}	=>	{place_data.user=BarenDome}	0.005860	0.5925776	0.009889	95.071011	5860
[3]	{place_data.user=EpicCrafter03}	=>	{place_data.block=ender_stone}	0.001137	0.5748231	0.001978	58.127521	1137
[4]	{place_data.user=Tsuro}	=>	{place_data.block=netherrack}	0.001129	0.6056867	0.001864	45.867981	1129
[5]	{place_data.user=freddy46}	=>	{place_data.block=leaves}	0.001986	0.6598007	0.003010	42.384574	1986
[6]	{place_data.user=KirbyClan}	=>	{place_data.block=leaves}	0.001204	0.5727878	0.002102	36.795004	1204
[7]	{place_data.user=I_Like_Trains_27}	=>	{place_data.block=leaves}	0.001072	0.5705162	0.001879	36.649080	1072
[8]	{place_data.user=WishMKR}	=>	{place_data.block=leaves}	0.001885	0.3945991	0.004777	25.348437	1885
[9]	{place_data.user=MtnL}	=>	{place_data.block=cobblestone}	0.001058	0.8390167	0.001261	14.265837	1058
[10]	{place_data.user=mmuuli}	=>	{place_data.block=cobblestone}	0.001275	0.8220503	0.001551	13.977357	1275
[11]	{place_data.user=Matii}	=>	{place_data.block=cobblestone}	0.001022	0.8189103	0.001248	13.923967	1022
[12]	{place_data.user=SpiritMayo}	=>	{place_data.block=cobblestone}	0.001274	0.4851485	0.002626	8.249001	1274
[13]	{place_data.user=Viktisen}	=>	{place_data.block=cobblestone}	0.001619	0.4441701	0.003645	7.552243	1619

Plot of Barendome’s endstone placements:

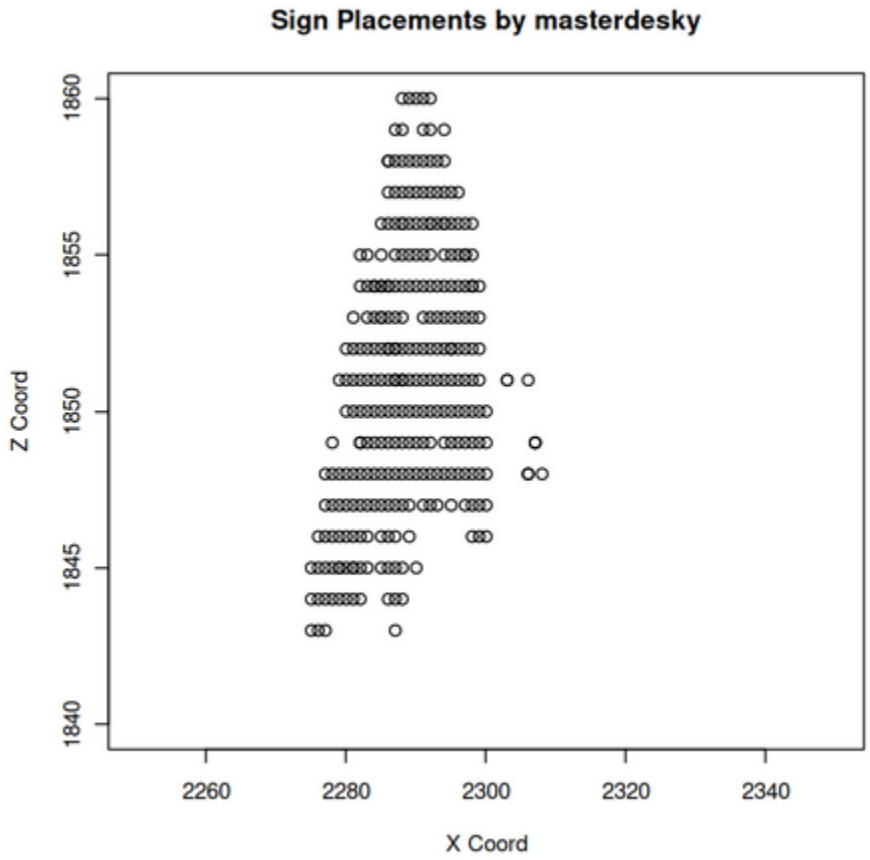


Our final question is related to text on signs and whether association rules can convey meaningful information of any use. To answer this question, we create association rules on the sign text and user in the sign data frame. Recall that in our distributions, we saw a number of players who placed extremely high quantities of signs and a number of texts found on signs in high quantities. When we run these rules, we find that these outliers in each group are often related and the rules can help us identify sign spam. We can, for example, use this method to associate certain texts with certain players and then filter our sign data to find where high quantities of signs were placed, which is useful to stop server exploits related to signs and world storage. We show a proof of concept for this regarding sign placements by user “masterdesky” with the sign text “Valley of signs.”

Association Rules on Sign Text and Users:

```
     lhs                                                                              rhs
support confidence    coverage      lift count
[1]  {sign_data.text=sal is cute    }                                              => {sign_data.user=CactusDupe
r}                                                        0.009936598  0.9768786 0.010171784  91.62457  1014
[2]  {sign_data.user=CactusDuper}                                                  => {sign_data.text=sal is cute
}                                                     0.009936598  0.9319853 0.010661754  91.62457  1014
[3]  {sign_data.text=tip:if there isnt a snow to spam on just move him slightly till } => {sign_data.user=SmashyMC}
0.003145609  0.9968944 0.003155409 135.64011   321
[4]  {sign_data.user=SmashyMC}                                                     => {sign_data.text=tip:if ther
e isnt a snow to spam on just move him slightly till } 0.003145609  0.4280000 0.007349555 135.64011   321
[5]  {sign_data.text=123456478957398 893475893758937 734563875637856 874356783563875}  => {sign_data.user=BrickBinde
r}                                                        0.002743834  1.0000000 0.002743834 313.02761   280
[6]  {sign_data.user=BrickBinder}                                                  => {sign_data.text=12345647895
7398 893475893758937 734563875637856 874356783563875}  0.002743834  0.8588957 0.003194606 313.02761   280
[7]  {sign_data.text= Valley of signs  }                                           => {sign_data.user=masterdesk
y}                                                        0.002596843  1.0000000 0.002596843 317.90343   265
[8]  {sign_data.user=masterdesky}                                                  => {sign_data.text= Valley of
signs  }                                                 0.002596843  0.8255452 0.003145609 317.90343   265
[9]  {sign_data.text=Join kroz.dog and check out the forum https:/ /niggasin.space}    => {sign_data.user=Portal_Tra
p}                                                        0.001352318  1.0000000 0.001352318 611.05988   138
[10] {sign_data.user=Portal_Trap}                                                  => {sign_data.text=Join kroz.d
og and check out the forum https:/ /niggasin.space}    0.001352318  0.8263473 0.001636501 611.05988   138
[11] {sign_data.text=centri3    }                                                  => {sign_data.user=Centri3}
0.001028938  1.0000000 0.001028938 426.97490   105
[12] {sign_data.user=Centri3}                                                      => {sign_data.text=centri3    }
0.001028938  0.4393305 0.002342058 426.97490   105
```

masterdesky’s sign placements:



Conclusions

Unfortunately, the work on this data set and even on similar data sets seems to be limited to only extracting data rather than working with it. This means that there is no peer to which we can compare our results.

We find that clustering techniques to identify builds may be possible but is infeasible at this level. We also find that the association between server activity and the SalC1 videos posted about the server are not necessarily related, at least not intuitively. Additionally, we find that association rules may be able to reveal information about blocks placed by certain players that may subsequently be used to identify areas or activities of interest. Finally, we find that

association rules may be used to identify players posting spam signs which can then be used to curb sign-related exploits where they exist.

Works Cited

"#salc1-Anarchy-Data-Mining." Discord. www.salc1.com/discord.

CoreProtect. Retrieved December 06, 2020, from
https://www.spigotmc.org/resources/coreprotect.8631/

Nerdsinspace. "Nerdsinspace/Leaky-Leaky." *GitHub*, github.com/nerdsinspace/leaky-leaky.

SalC1 anarchy server land marks. Retrieved December 06, 2020, from
https://pastebin.com/6g2c7LJF

"Salc1 Datamine." *SALC DATAMINE*, web.archive.org/web/20200725220031/salc.n00bs.info/.

SalC1. *The Future of the Salc1 Anarchy Server*. 9 Apr. 2020,
www.youtube.com/watch?v=ErqC6PXVm9k&t=2s. Accessed 3 Oct. 2020.