Distributed Server Project

Matthew Angle, Josh Wood

Overview

The purpose of this project was to learn how to use CORBA to set up a distributed system across servers and measure the difference in transfer time from one server to another. The tools we used for this experiment were provided but included a Java implemented server and client. CORBA stands for "Common Object Request Broker Architecture", CORBA provides "vendor-independent architecture and infrastructure that computer applications use to work together over networks".[1] The Java program we were provided requires us to first setup a ORBD Daemon which the Java program will then use to establish a connection between the server and the client.

Experiment Conditions

Clipper is a server located at Shippensburg University. Clipper is an Ubuntu 18.04.4 server with Linux Kernel 4.15.0-88-generic, it has an Intel Xeon 2680 v3 and has 20 CPU cores and 24 GB's of ram, which have been assigned to the virtual machine running Clipper. The ORBD daemon was started on CLipper and our team used ports 1054 and 1055. Java 8 was used to build and run the server and the client.

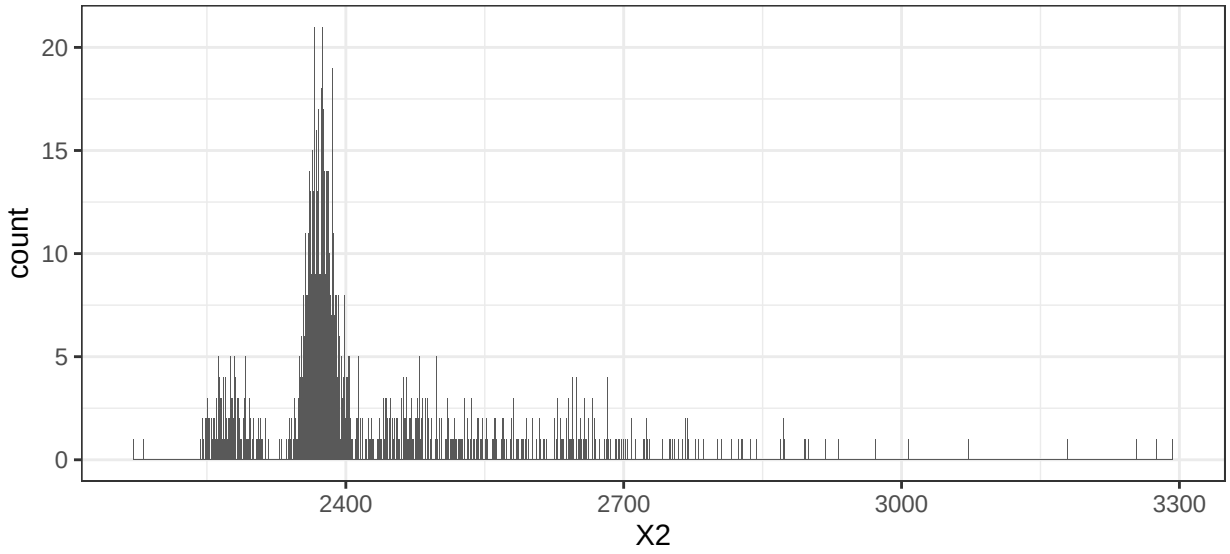
The German server is a remote virtual server hosted in Germany and uses an Intel Xeon e5-2660 v4 but is assigned one virtual core and 1 GB of ram. The server runs Ubuntu 18.04.5 with Linux Kernel 4.15.0-122-generic x86_64. The ORBD daemon was started on CLipper and our team used ports 1054 and 1055. Java 8 was used to build and run the server and the client.

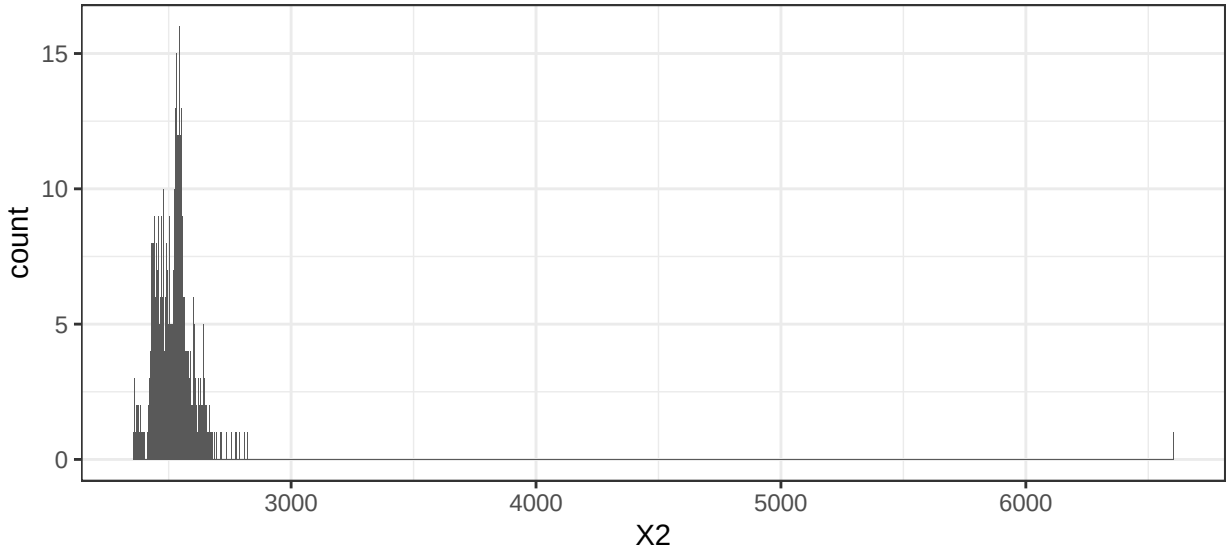
The Spanish server is also a remote virtual server hosted in Spain instead of Germany. It uses an Intel Xeon E5-2683 v3 but is also assigned one virtual core and 1 GB of ram. The server runs Ubuntu 18.04.5 with Linux Kernel 4.15.0-122-generic x86_64. The ORBD daemon was started on CLipperand our team used ports 1054 and 1055. Java 8 was used to build and run the server and the client.

First Experiment

We initially set up our experiment using a bash script that would start and stop the ORBD Daemon and the Java Server, then run the client and measure the time the client took to complete the file transfer. In our experiment we used a text document full of random characters that was 1 Mb in size, we stored that on our server in the appropriate folder in the Java Server build path and then transferred that file to our client. We ran this bash script 1000 times on the German server and the Spanish server and stored the results in a csv.

First Results





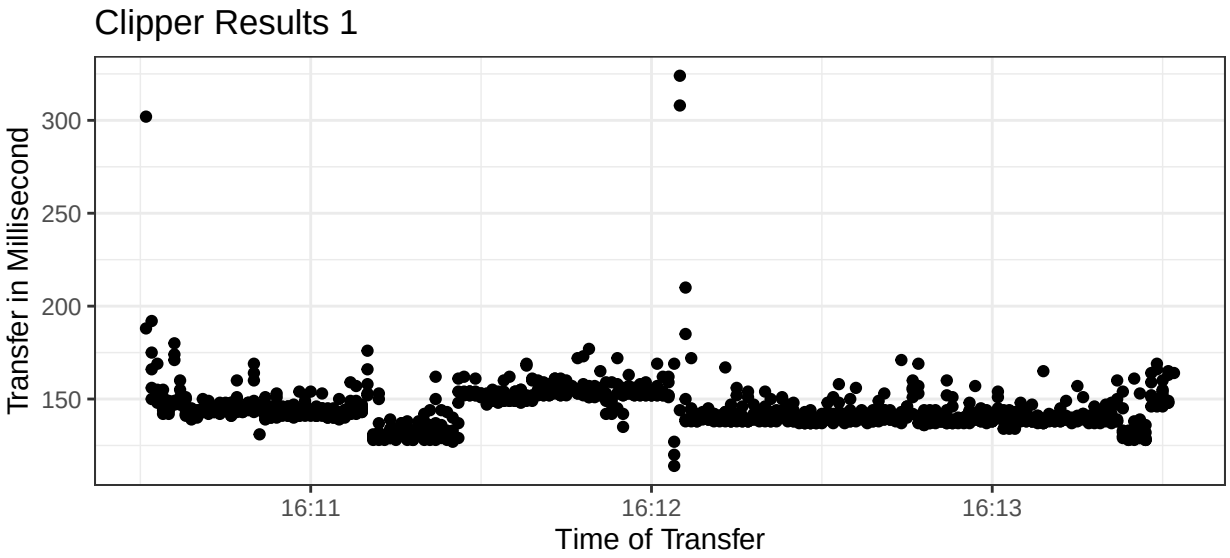
Second Experiment

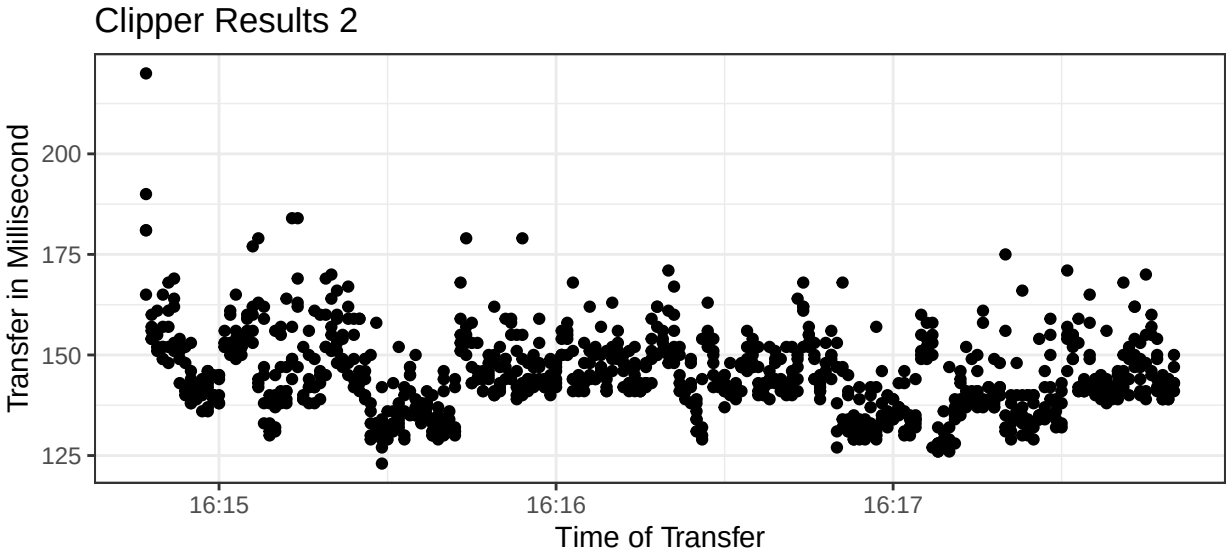
After some consideration we became concerned that the differences in compile time on the each server may have influenced our results and rendered them void.

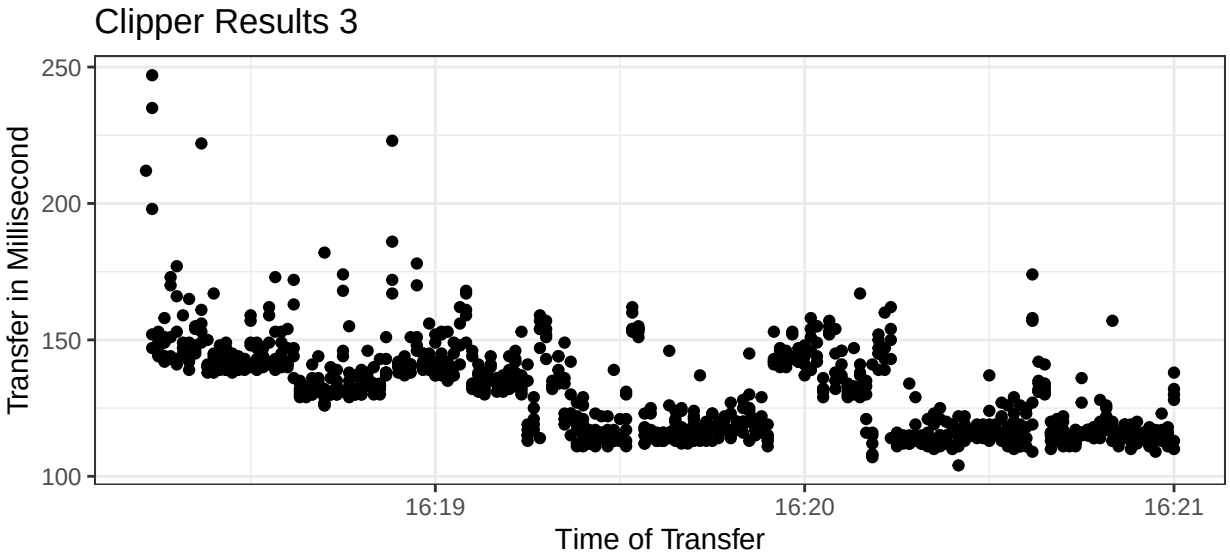
We set up a second experiment where we transferred in the same manner a 2 MB text file, but instead of measuring total time to compile and transfer we only measured the time that it took to transfer using a Java method to measure the start and stop time of the transfer file method. We nested this method in a loop and ran it 1000 times on Clipper, the German and the Spanish server. We ran these tests 3 times each for each server and stored the results in corresponding csv files. The Clipper test was ran as a baseline to see best case scenarios for the transfer. The file size chose was intended to be large enough to be significant but not too large to be overwhelming for the remote servers. We ran the tests 3 times on each server to ensure we could have a large enough sample size to make significant analysis.

Clipper Server Results

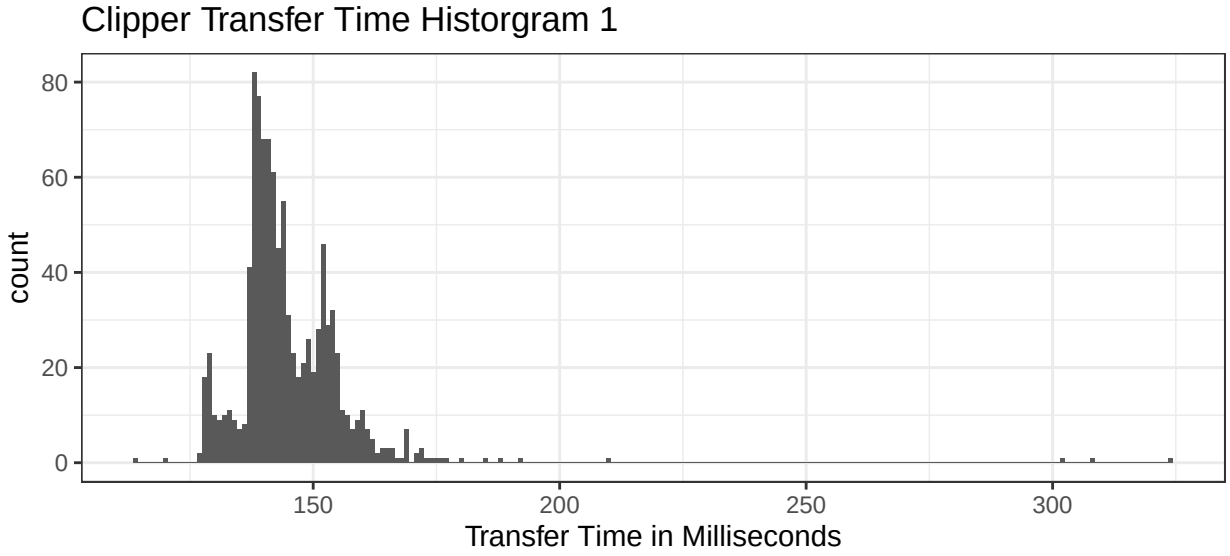
We started analysis by looking at a point plot of the data over time.

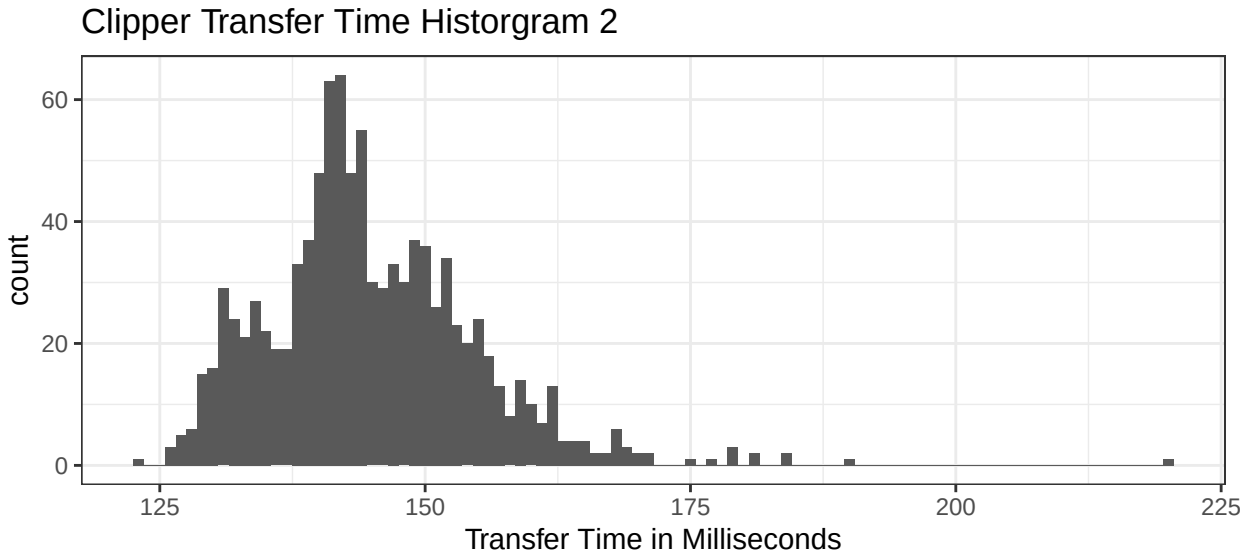


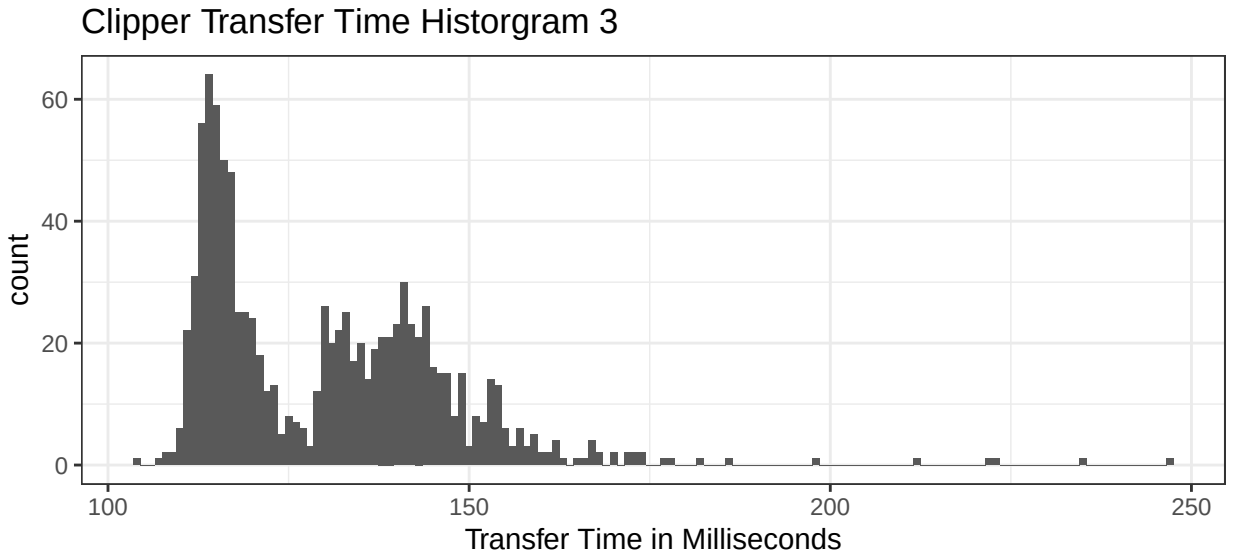




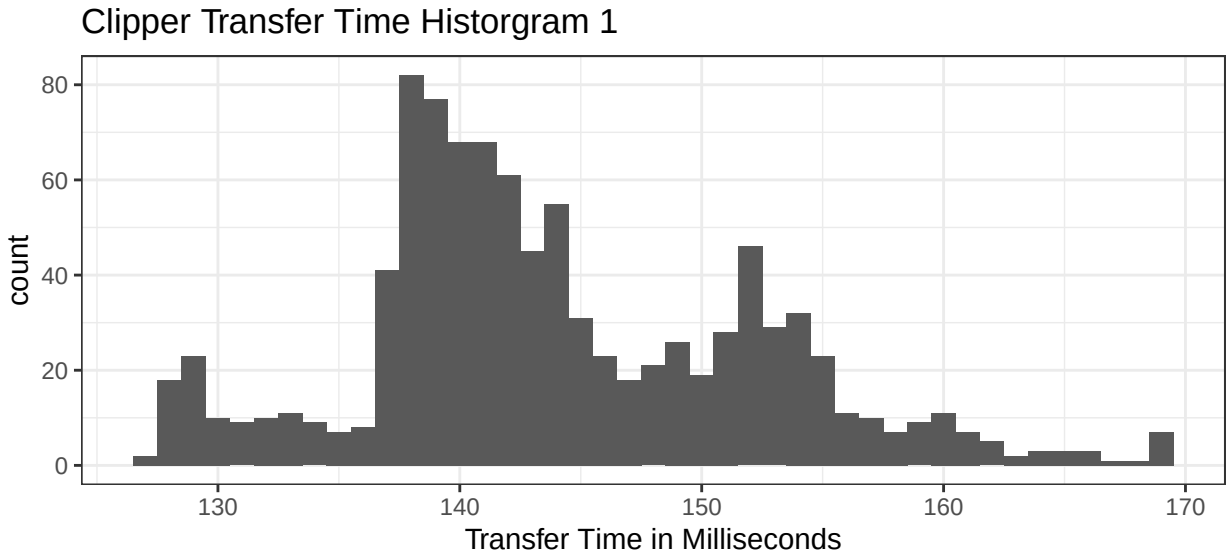
The time series data seems to show a mostly flat line across the first two tests with a few outliers except for the third plot which seems to stepped into two separate lines. To get a better look at these and to see if the data was normally distributed we made histograms using the data.

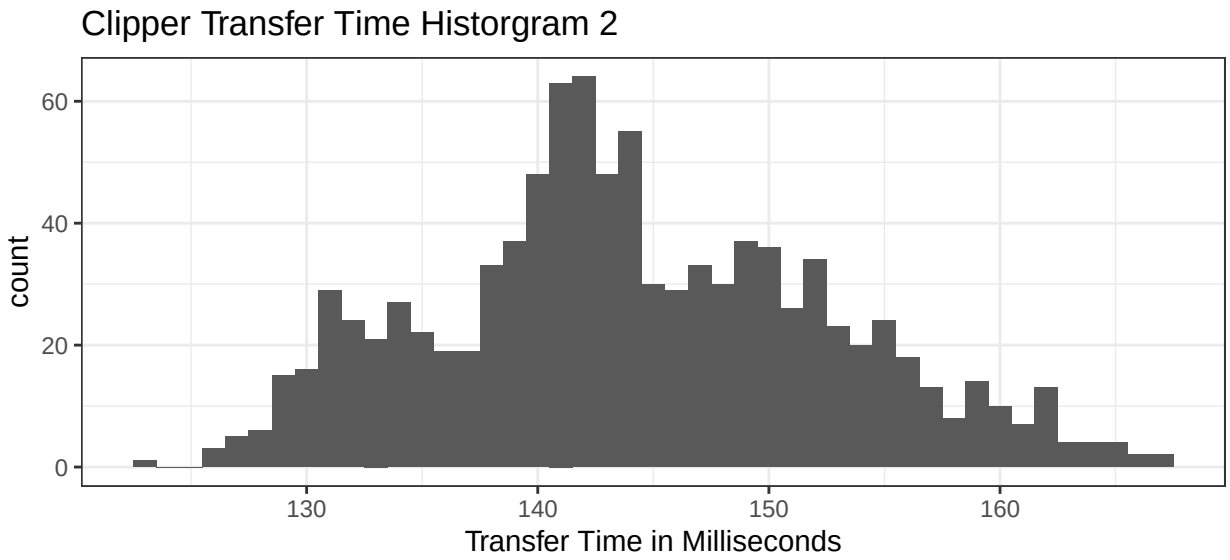


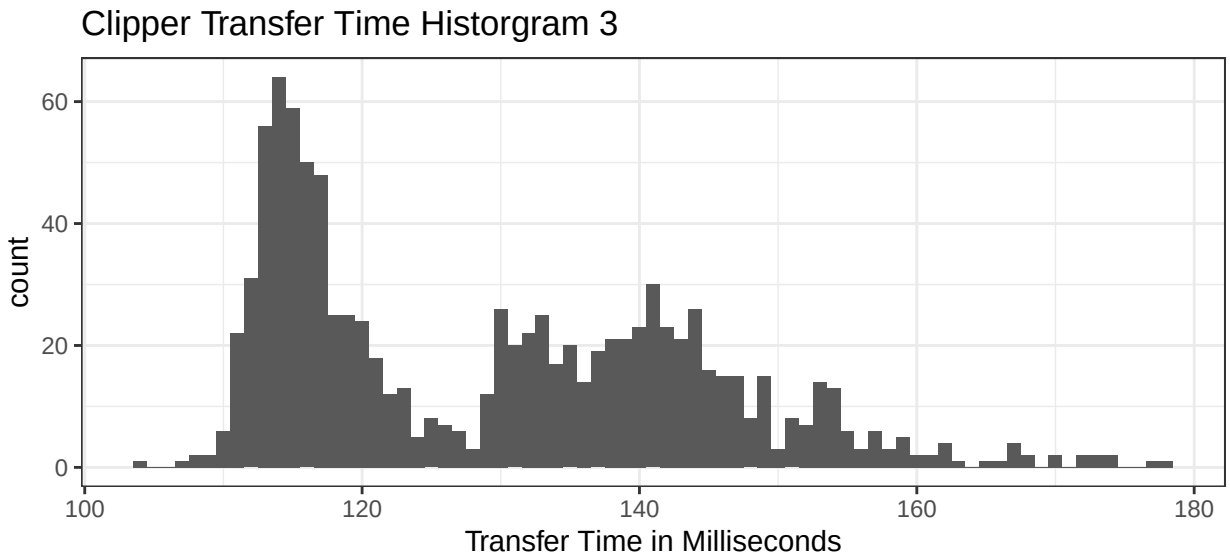




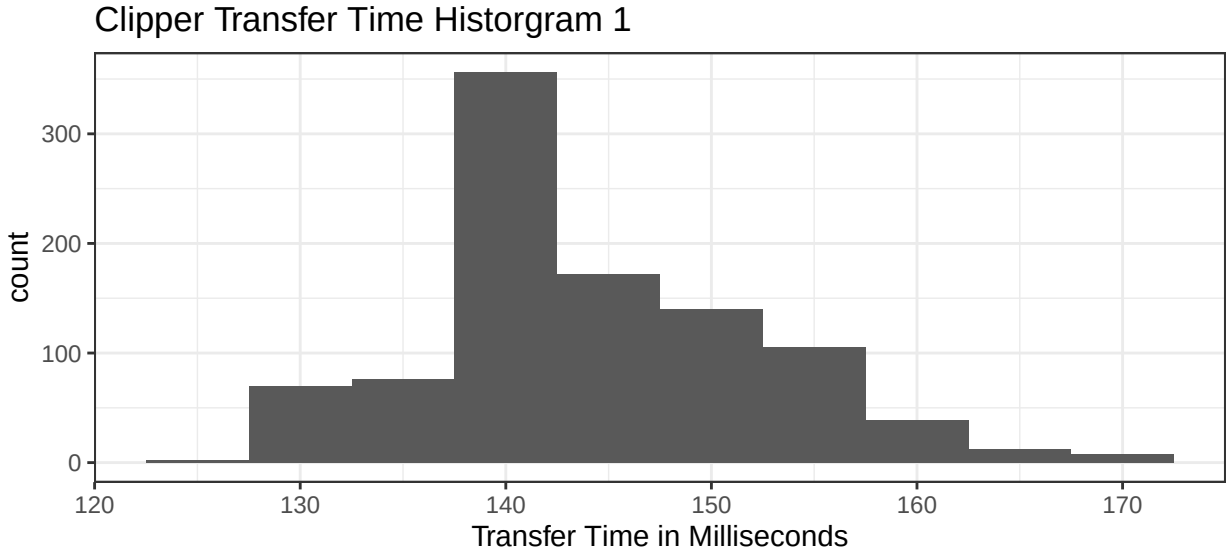
It became evident that we did have a few outliers and likely a bimodal distribution in the third set of data. To account for the outliers we used the boxplot function contained in the ggstatsplot package to remove outliers and clean up the data. The boxplot function can be used to identify outliers "which lie beyond the extremes of the whiskers". [2] After removing the outliers, the following were our resulting histograms.

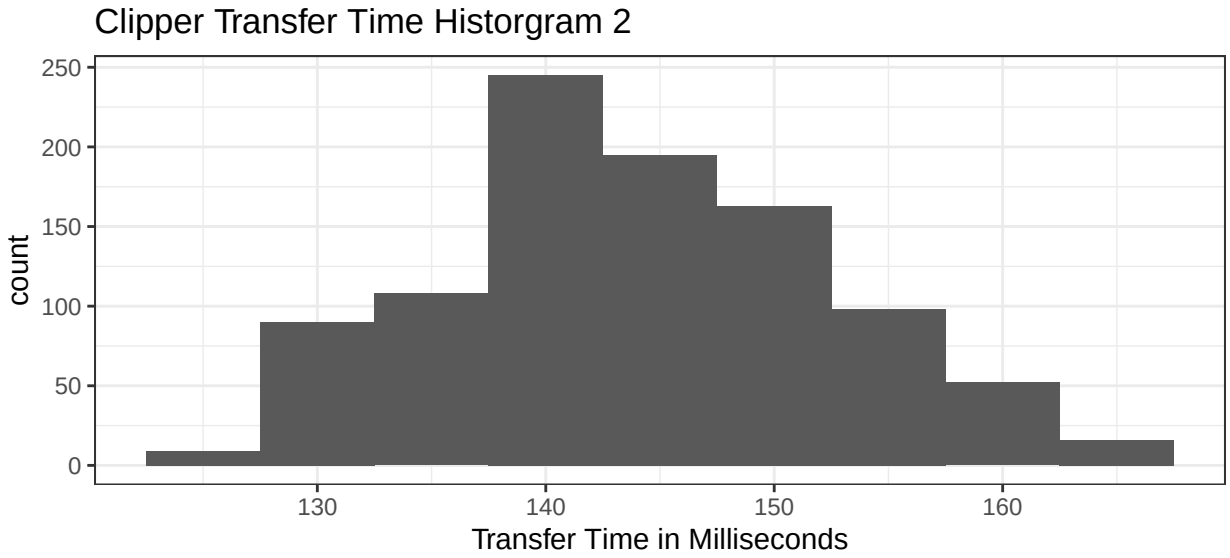

Clipper Transfer Time Historgram 1
count
80
60
40
20
0
130
140
150
160
170
Transfer Time in Milliseconds

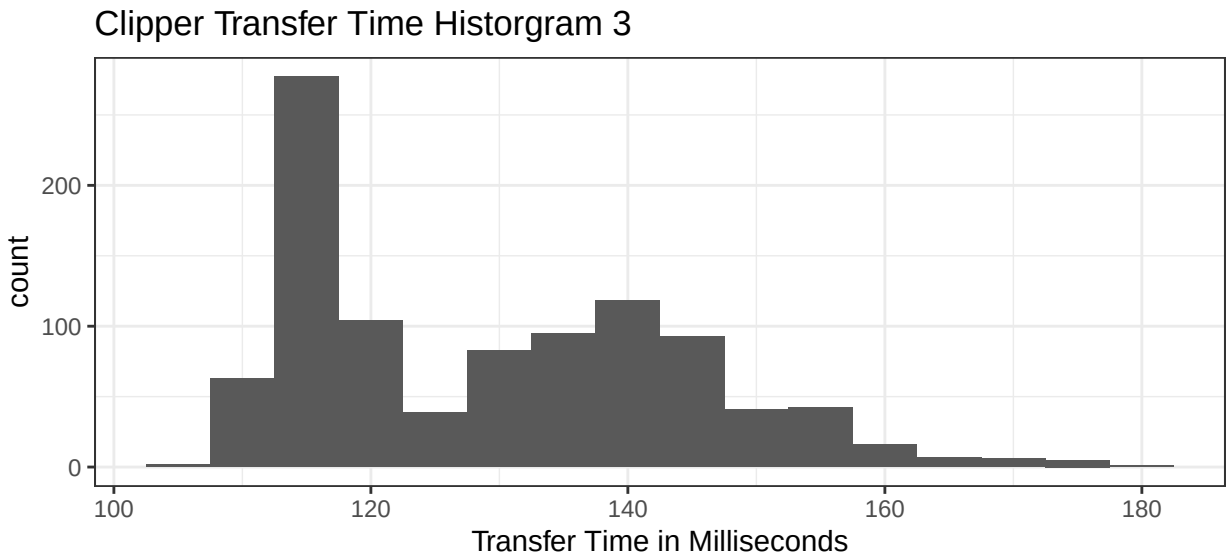

Clipper Transfer Time Historgram 2
count
60
40
20
0
130
140
150
160
Transfer Time in Milliseconds



As can be seen in the resulting histograms the second test on Clipper seems to approach a normal distribution. The first and third test both seem to be bimodal at first. Open further examination we found that if we changed the bin width for the histograms we can see that with an adjusted bin width of 5, the first test on Clipper also seems to be normally distributed. The third test is certainly bimodal.







After we removed the outliers and confirmed that the first two data sets were normally distributed we needed ensure that the first and second test on clipper accurately represent the same underlying distribution. We chose a t-test to compare the first and the second dataset, because a t-test can be used to determine if there is a significant difference between the means of two groups. First we'll assume a null hypothesis which is that Test 1 = Test 2 , Test 2 = Test 3, and Test 1 = Test 3, that is the data from three separate tests are = and there is no statistical difference between them. Our Alternate Hypothesis will be that Test 1 != Test 2 , Test 2 != Test 3, and Test 1 != Test 3, or that is there is a statistically significant difference between the three sets of data.

```
##
##  Welch Two Sample t-test
##
## data:  javaResults01$X2 and javaResults02$X2
## t = -0.24975, df = 1950, p-value = 0.8028
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8343446  0.6458451
```

```
## sample estimates:
## mean of x mean of y
##  143.9969  144.0912

##
##  Welch Two Sample t-test
##
## data:  javaResults02$X2 and javaResults03$X2
## t = 26.784, df = 1579, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  13.57308 15.71817
## sample estimates:
## mean of x mean of y
##  144.0912  129.4456

##
##  Welch Two Sample t-test
##
## data:  javaResults01$X2 and javaResults03$X2
## t = 26.888, df = 1539.8, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  13.48985 15.61289
## sample estimates:
## mean of x mean of y
##  143.9969  129.4456
```

As can be seen in the results the mean of the first test was 144.973 and the mean of the second test was 144.859. The p-value was .8258 indicating the underlying distributions are similar and we can not reject a null hypothesis in this instance. Our second t-test has an extremely low p-value (2.2e-16) and shows a sample mean of 144.859 for Test 1 and 130.115 for Test 2. This low p-value indicates that we should reject the null hypothesis. There is a statistically significant difference between our results in Test 2 and our results in Test 3. Likewise when comparing Test 1 and Test 3 we get similar results with a p-value of 2.2e-16. With all that in mind we can conclude that test's 1 and 2 belong to the same underlying distribution and will be ok to use when comparing the performance of other servers. As test 3 was bimodal and did not fit with Test 1 and Test 2 we will not use Test 3 as a baseline or include it in further analysis.

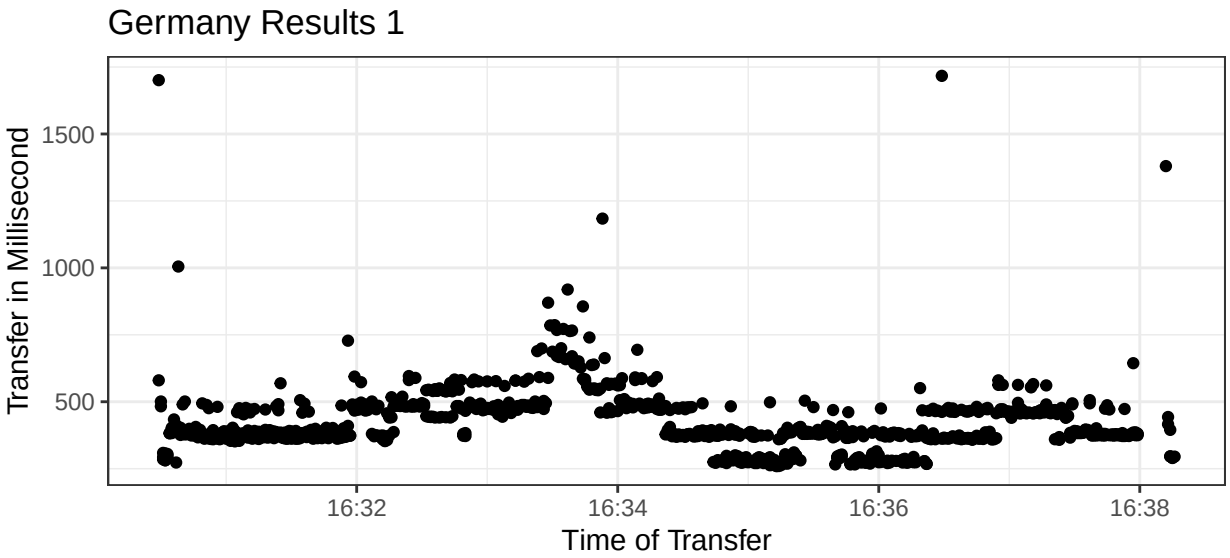
```
##   0%  25%  50%  75% 100%
##  123  139  143  150  169

## Max = 169

## Min = 123

## Mean = 144.04

## SD = 8.34
```

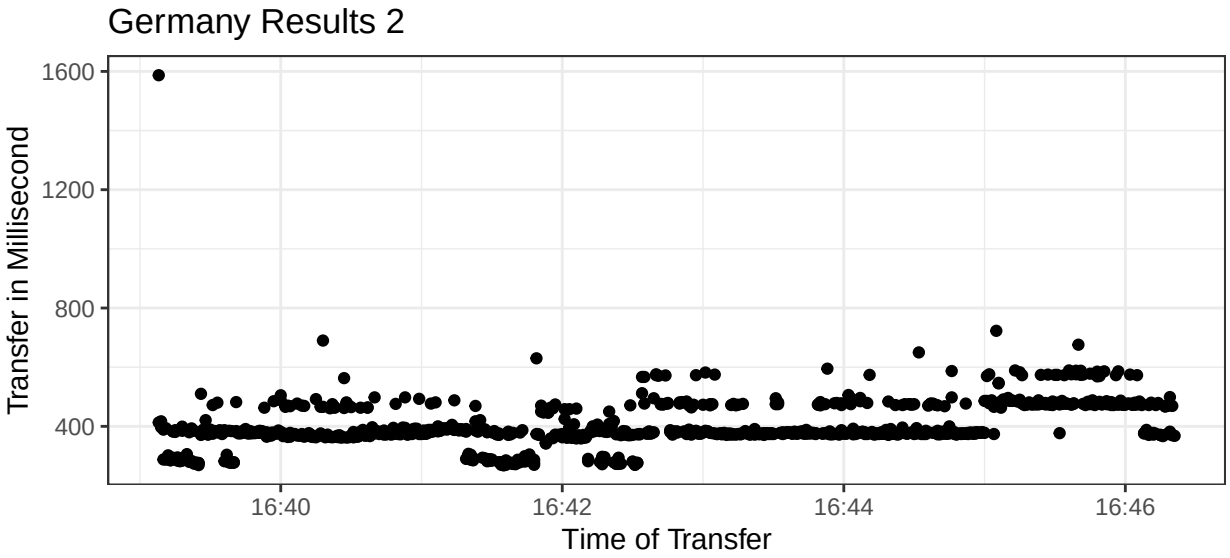
So our final results for best case scenario transfer time is a mean time of 144.92, with a max time of 324 milliseconds and a min time 144 milliseconds. Our standard deviation is 11.58 and 50% of all our values fall between 139 and 151 milliseconds.

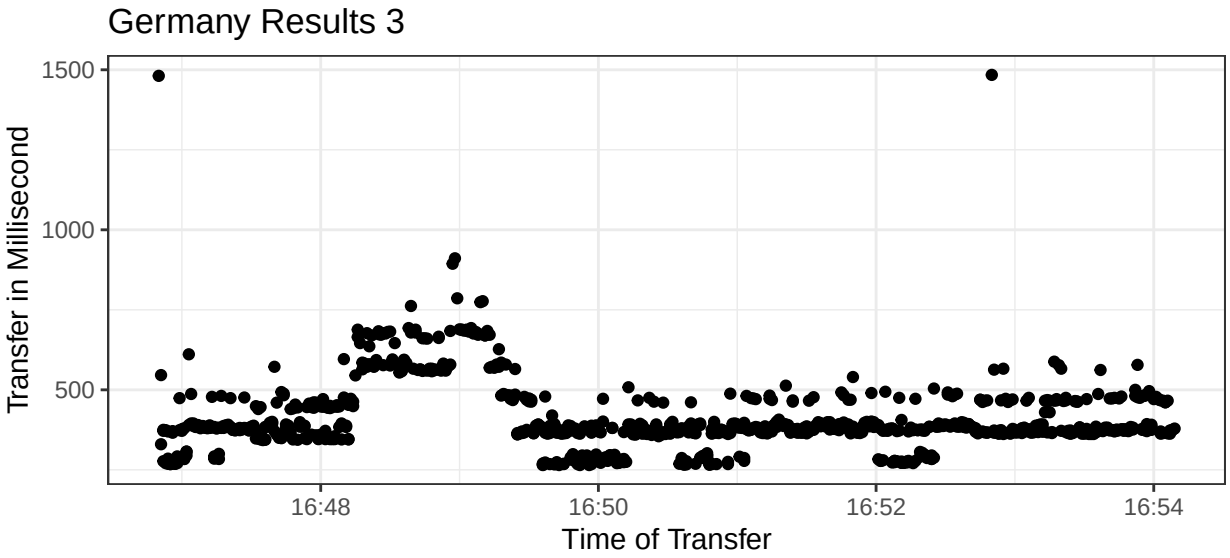
As can be seen the combined dataset approaches a normal distribution.

German Server Results

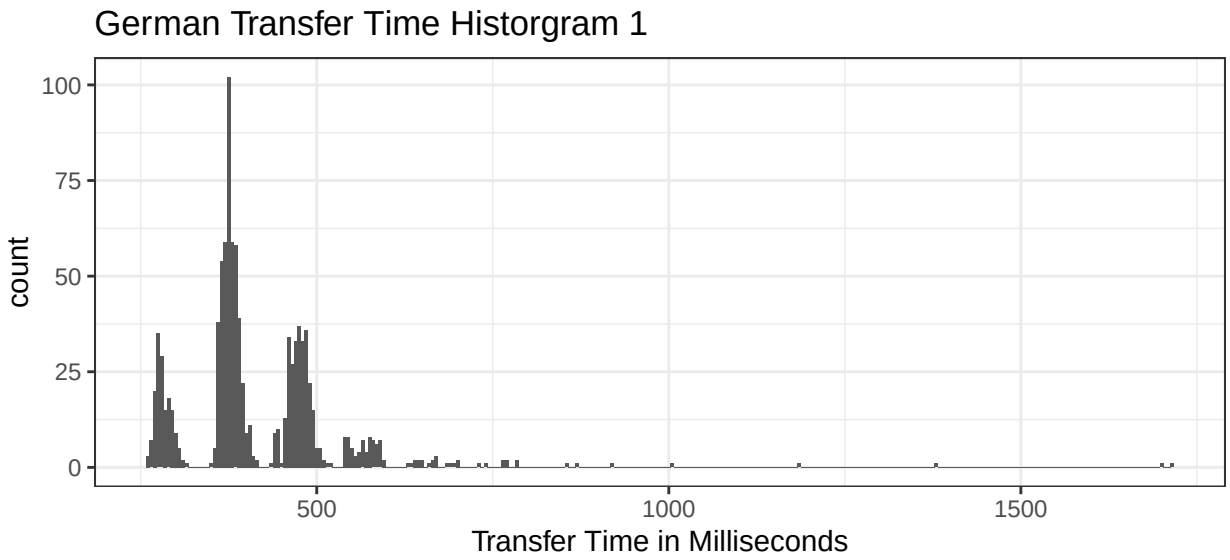
Having established a baseline we ran the same tests on the German server

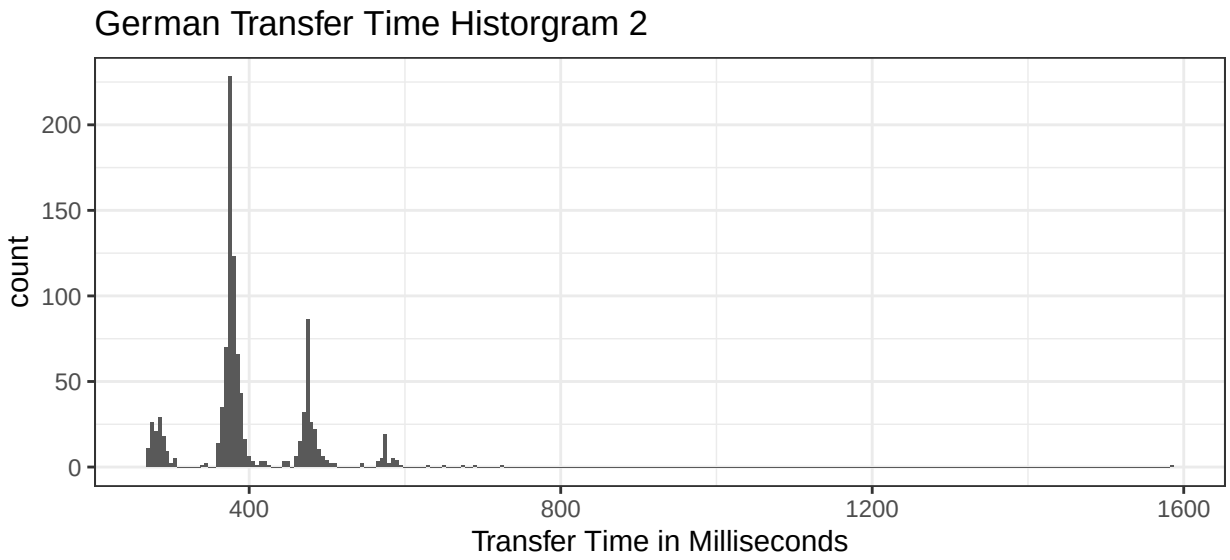


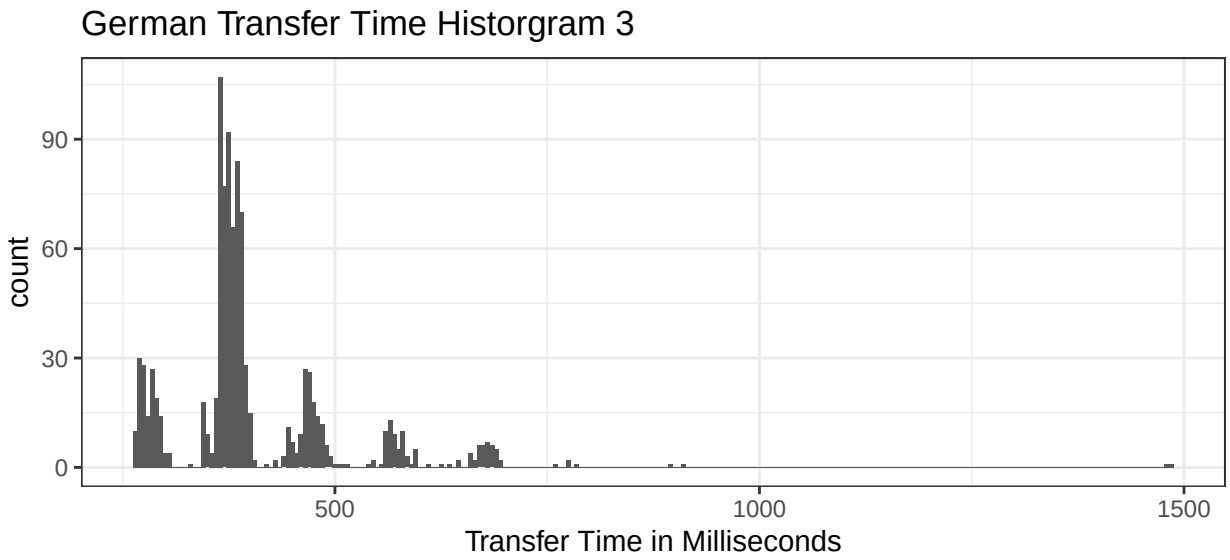




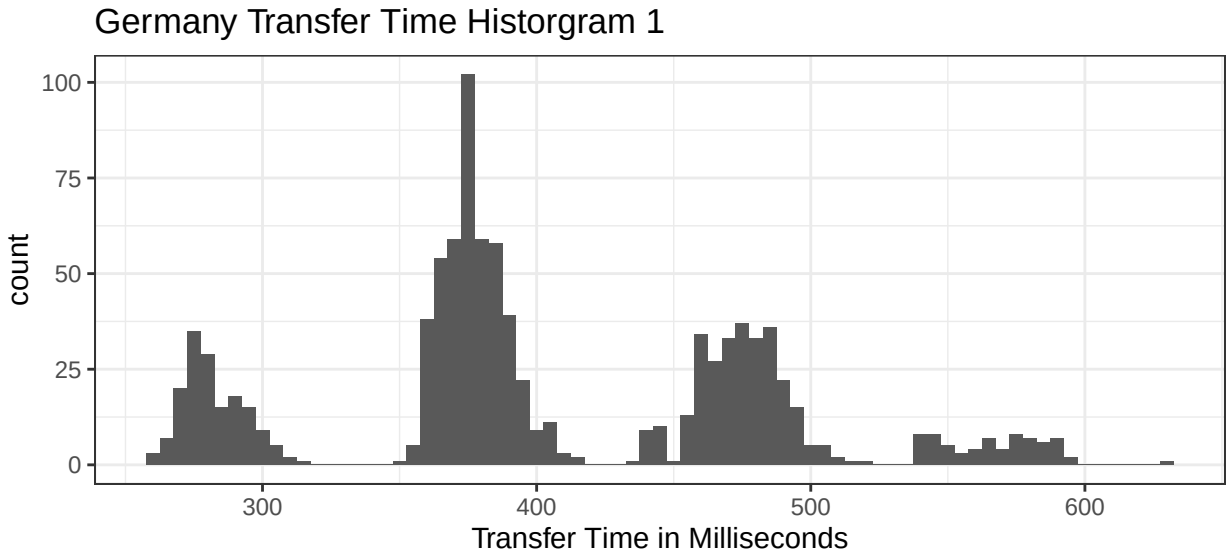
As can be seen the time series data looks similar to our previous test. There seems to be a spike in activity that happens around a minute in for all three tests but all three seem to follow a similar pattern.

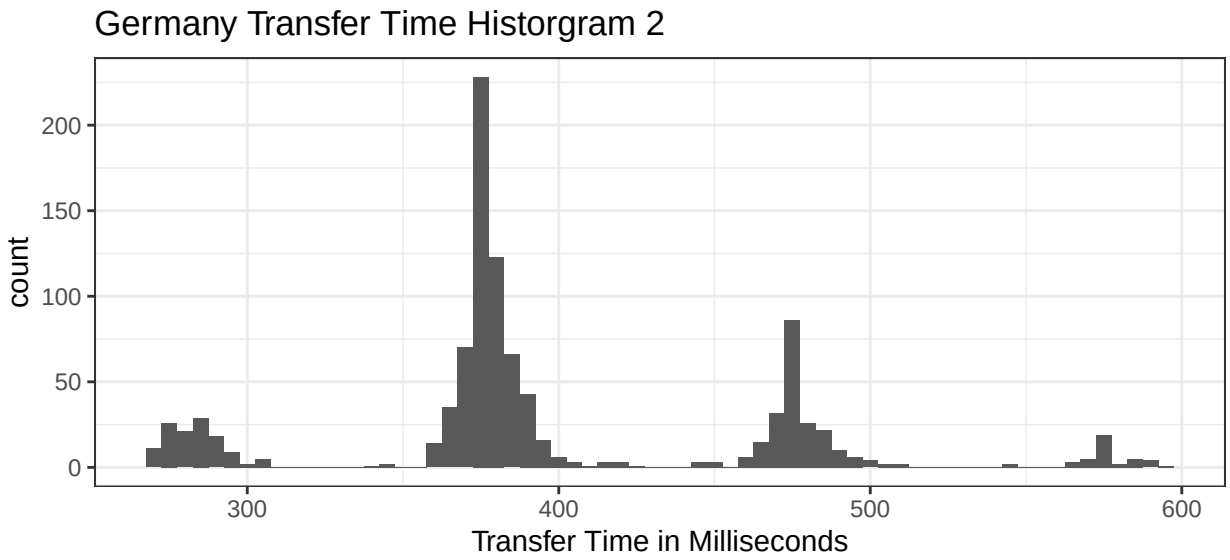

German Transfer Time Historgram 1
count
100
75
50
25
0
500
1000
1500
Transfer Time in Milliseconds

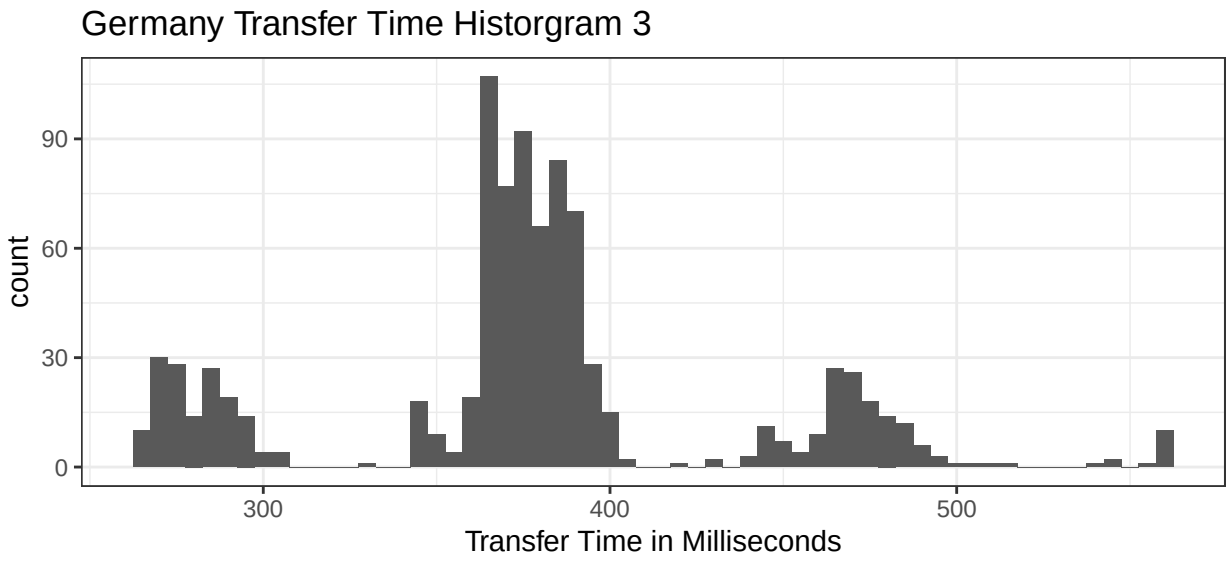

German Transfer Time Historgram 2
count
200
150
100
50
0
400
800
1200
1600
Transfer Time in Milliseconds



When making our histograms there is an very apparent difference from our first results on Clipper. All three tests run on the German server seem to group around similar transfer time intervals.







After removing outliers as we did before and grouping these into larger bins (width of 5) the grouping is even more apparent. There seems to be groupings around 275 milliseconds, 375 milliseconds, and 475 milliseconds on tests 1 and 2. While this does not seem to be exactly normally distributed, it is a consistent distribution. Test 3 seems to be normally distributed after removing the outliers. Running the t-tests as before seems to indicate all these tests belong to the same underlying distribution.

```
##
##  Welch Two Sample t-test
##
## data:  javaResultsGermany01$X2 and javaResultsGermany02$X2
## t = 2.2735, df = 1909.7, p-value = 0.02311
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   1.037737 14.072502
## sample estimates:
## mean of x mean of y
##  403.5994  396.0443
```

```
##
##  Welch Two Sample t-test
##
## data:  javaResultsGermany02$X2 and javaResultsGermany03$X2
## t = 6.1258, df = 1893.5, p-value = 1.094e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  12.18674 23.66481
## sample estimates:
## mean of x mean of y
##  396.0443  378.1185
```

```
##
##  Welch Two Sample t-test
##
## data:  javaResultsGermany01$X2 and javaResultsGermany03$X2
## t = 7.9047, df = 1804.8, p-value = 4.636e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  19.15865 31.80313
## sample estimates:
## mean of x mean of y
##  403.5994  378.1185
```

As we can from our two sample t-test's the means for Test 1 = 403.6, Test 2 = 396.04, and Test 3 = 378.12 and all three tests fall below a significant level of .05. So that means in all three cases we can assume test 1, 2, and 3 are all part of the same underlying distribution.

```
##   0%  25%  50%  75% 100%
##  123  141  149  360  562
```

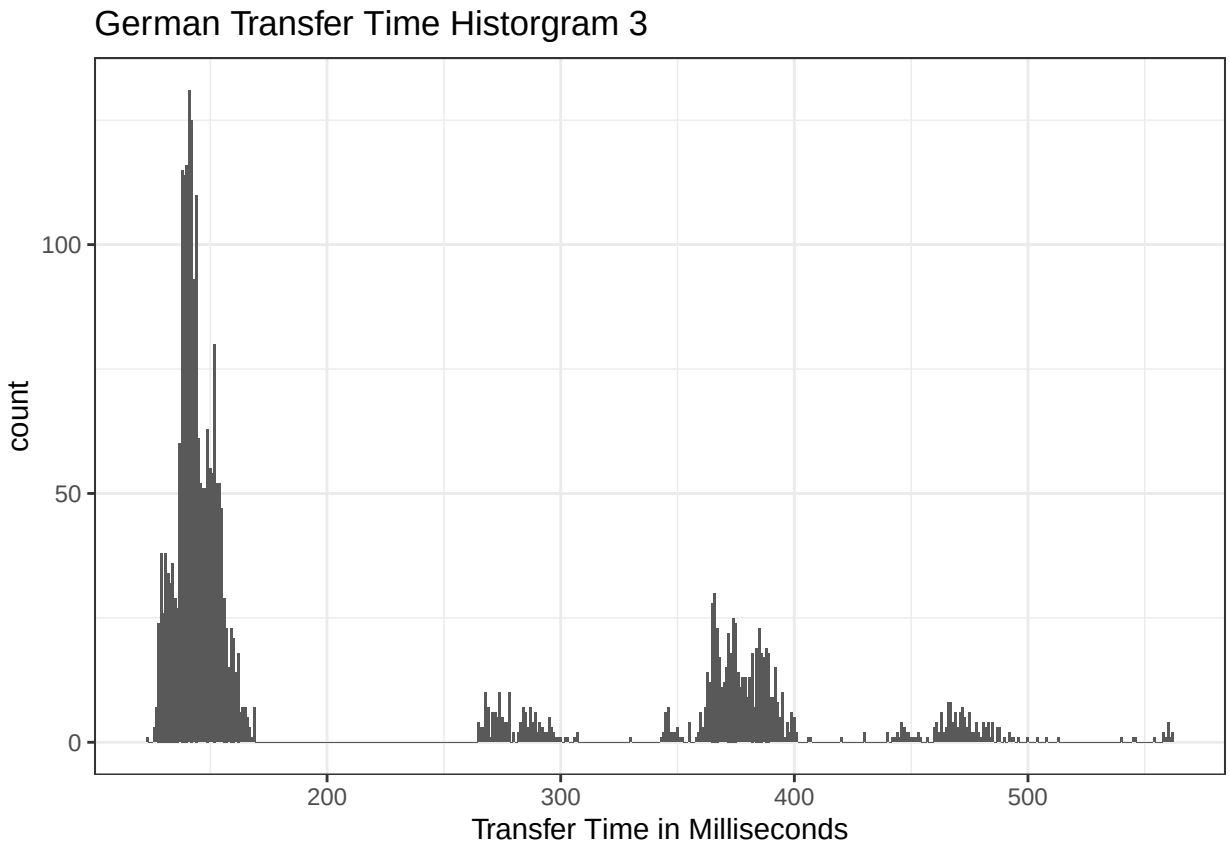
```
## Max = 169
```

```
## Min = 123
```

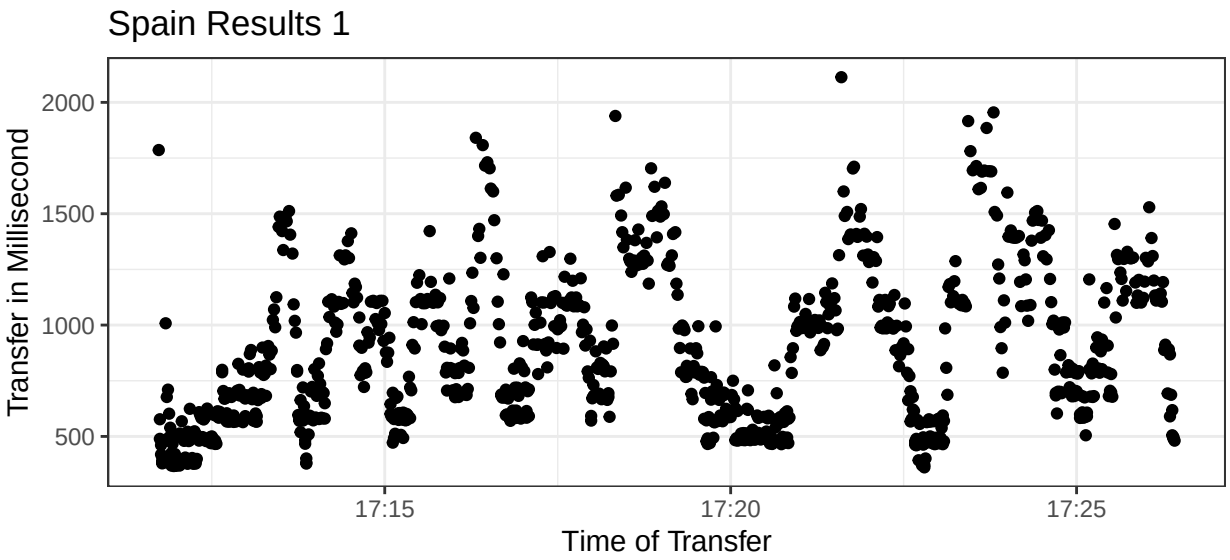
```
## Mean = 144.04
```

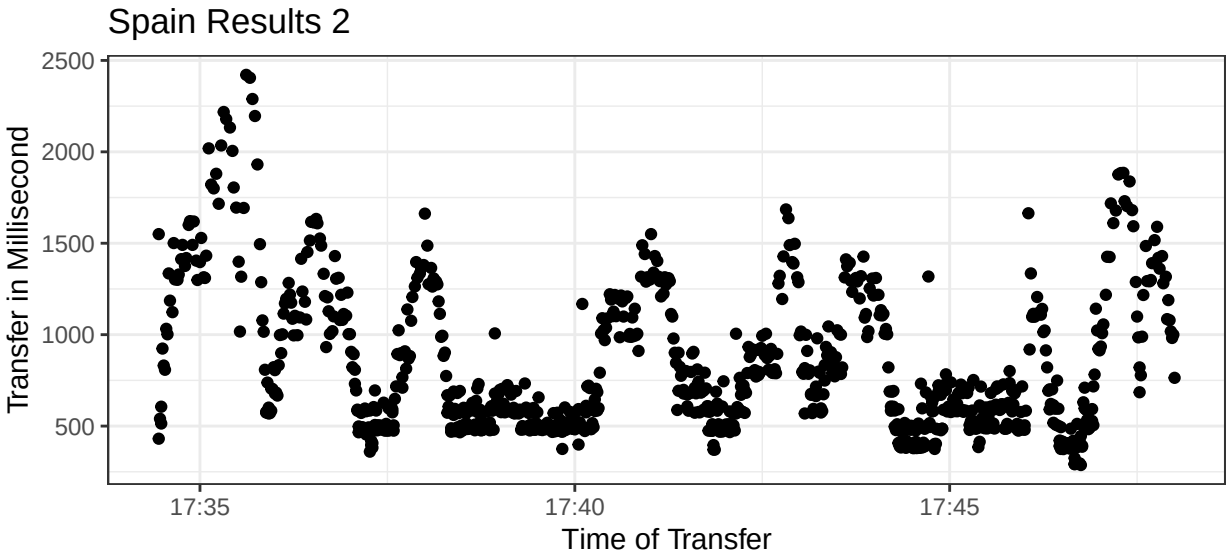
```
## SD = 8.34
```

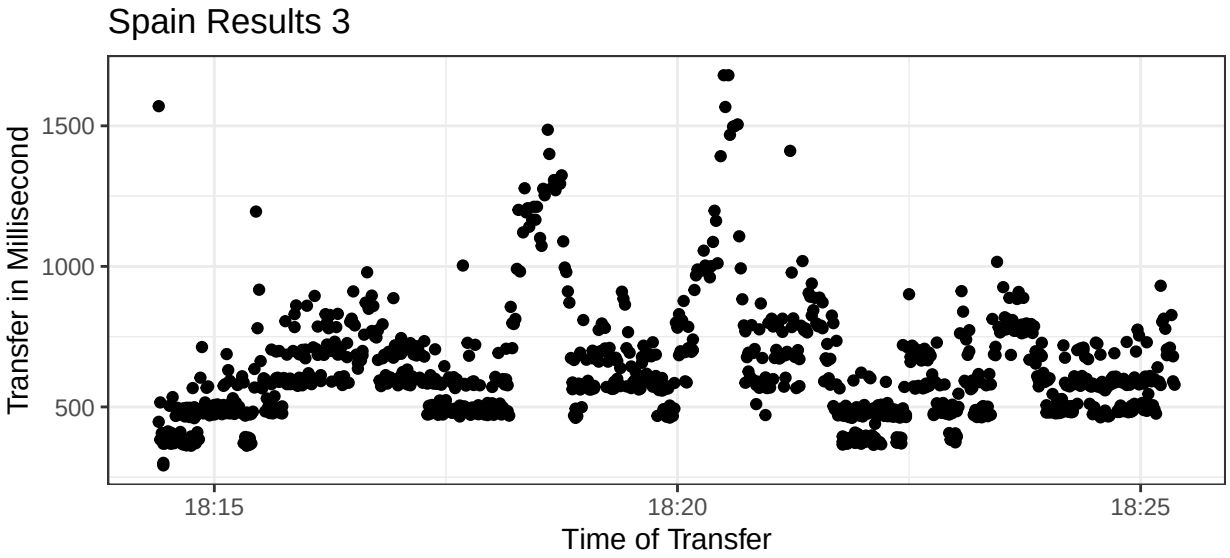
Once combined we can see 50% of all transfer time values fall between 369 and 400.5 milliseconds and our mean transfer time is 394.17 milliseconds. Our fastest time was 260 milliseconds and max was 596.

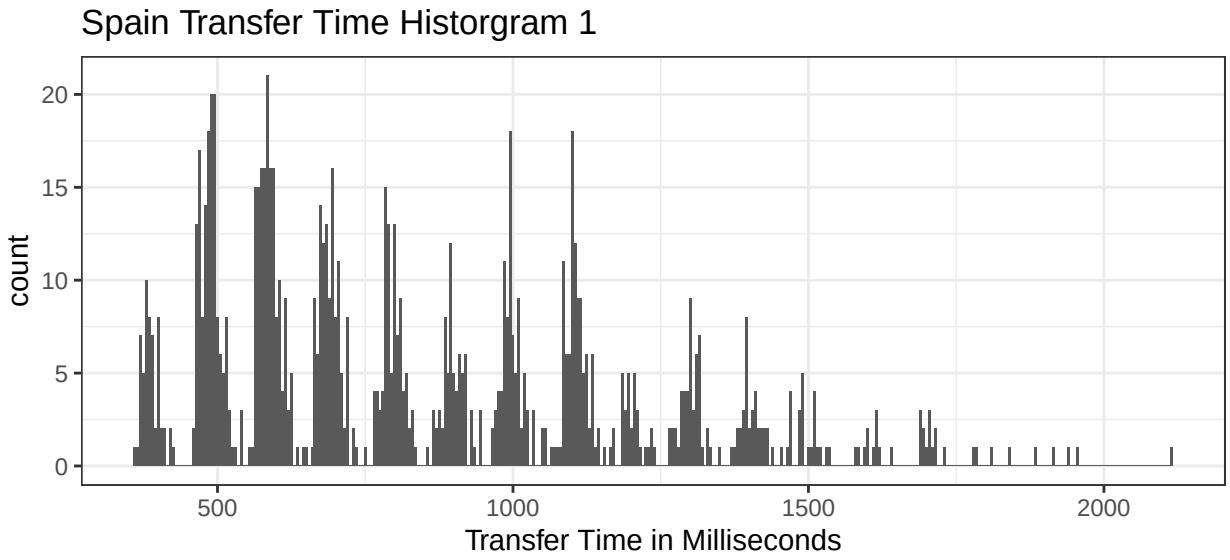

German Transfer Time Historgram 3
count
100
50
0
200
300
400
500
Transfer Time in Milliseconds

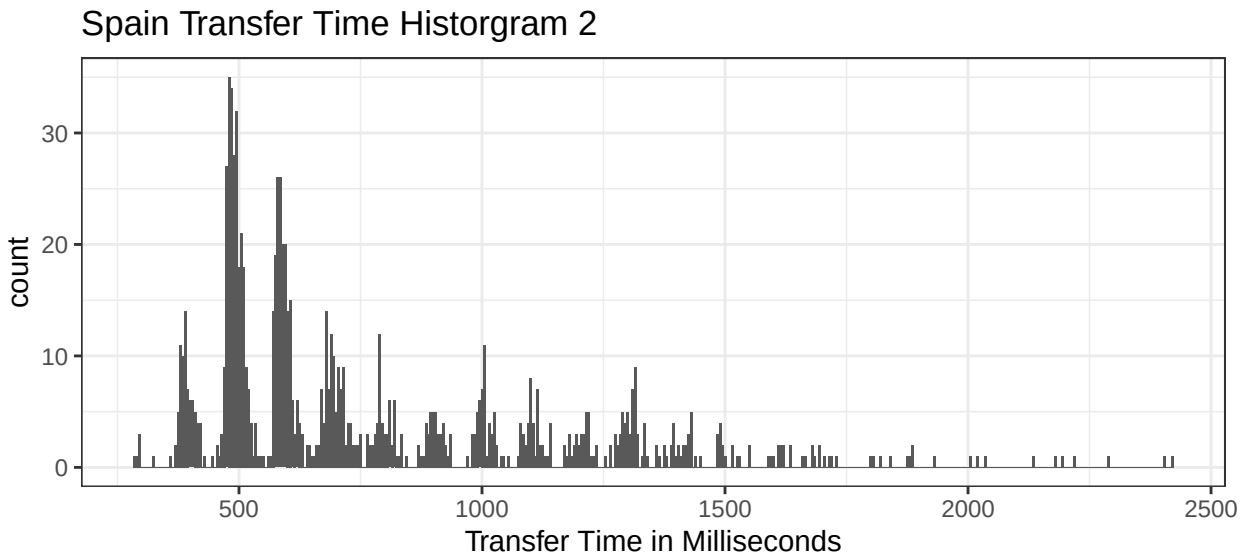
Spanish Server Results

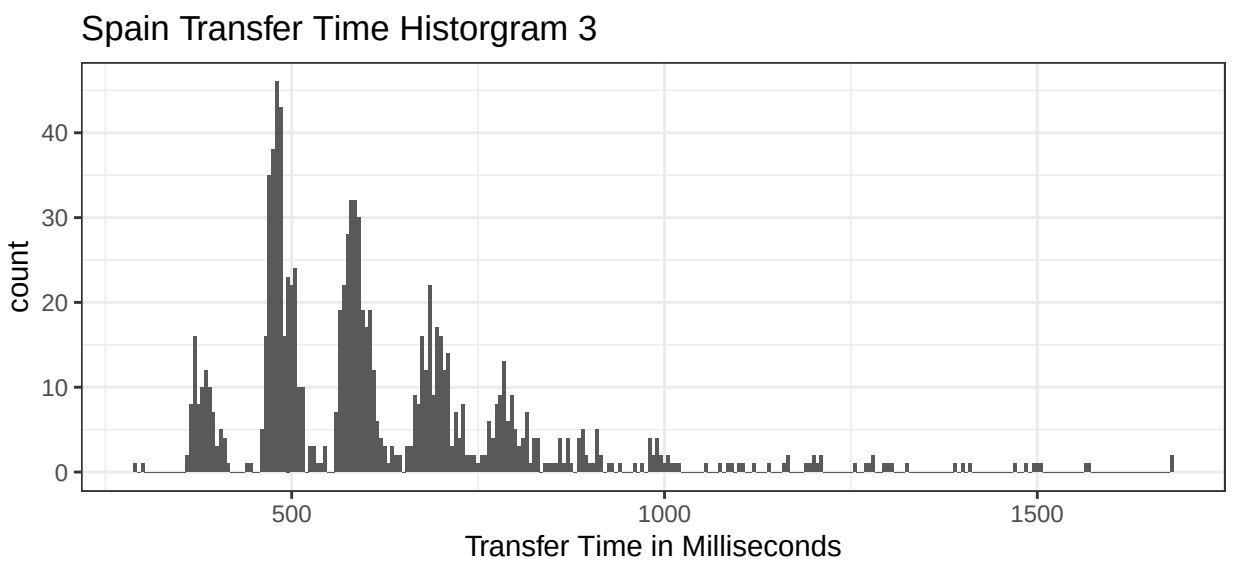


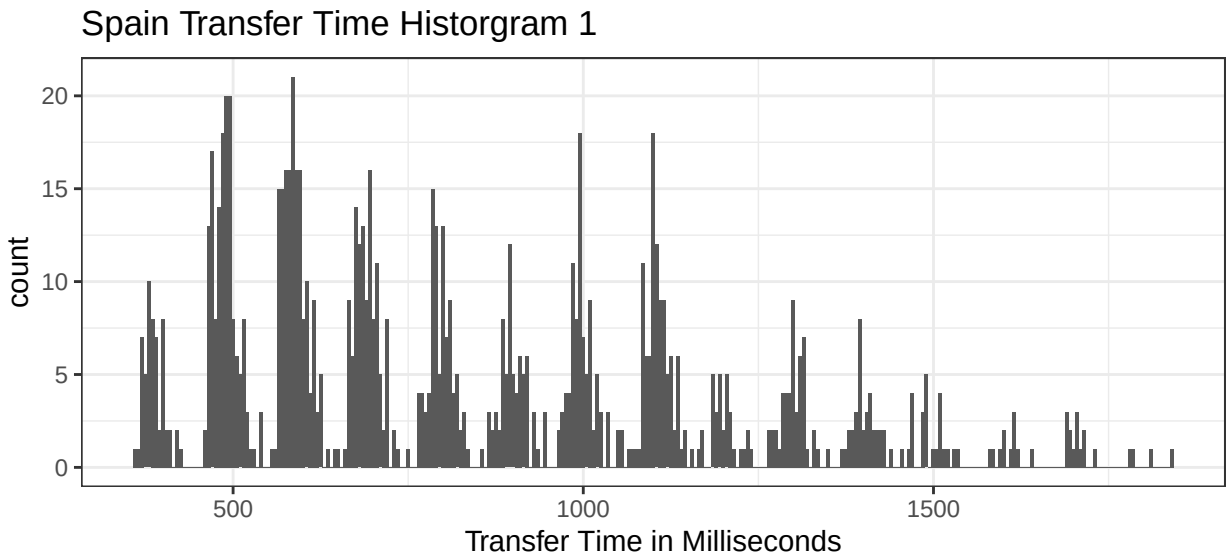


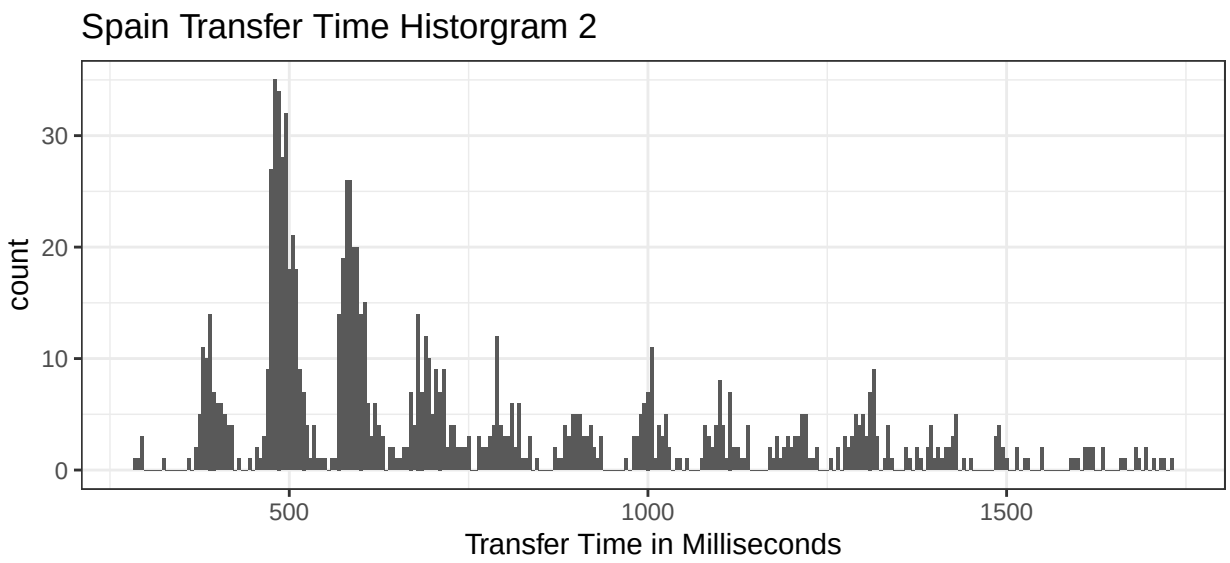

Spain Results 3
Transfer in Millisecond
1500
1000
500
18:15
18:20
18:25
Time of Transfer

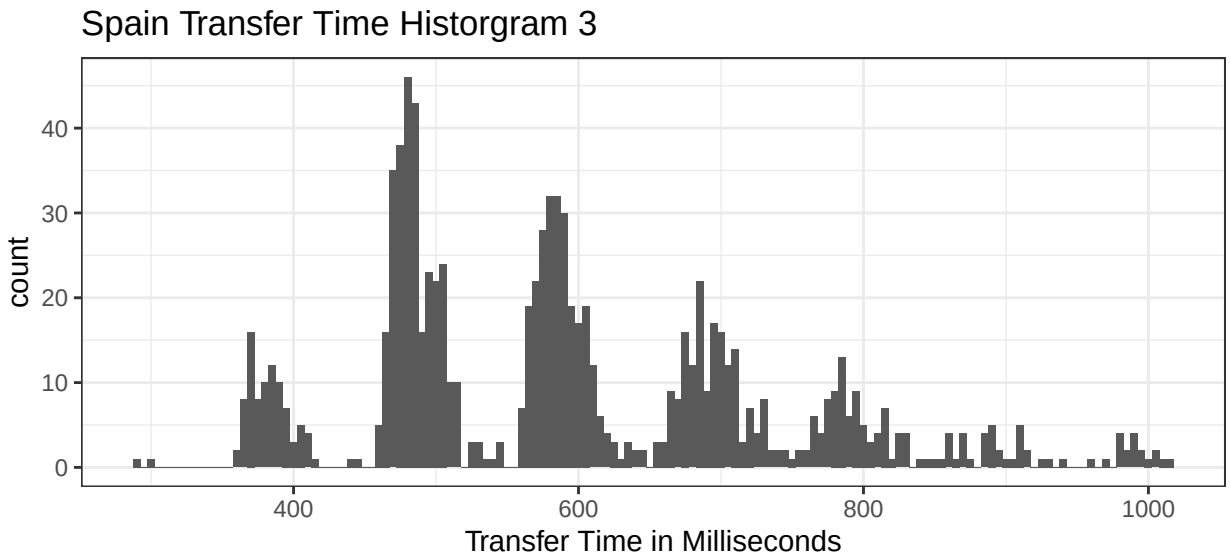

Spain Transfer Time Historgram 1
count
20
15
10
5
0
500
1000
1500
2000
Transfer Time in Milliseconds


Spain Transfer Time Historgram 2
count
0
10
20
30
500
1000
1500
2000
2500
Transfer Time in Milliseconds


Spain Transfer Time Historgram 3
count
0
10
20
30
40
500
1000
1500
Transfer Time in Milliseconds


Spain Transfer Time Historgram 1
count
20
15
10
5
0
500
1000
1500
Transfer Time in Milliseconds


Spain Transfer Time Historgram 2
count
30
20
10
0
500
1000
1500
Transfer Time in Milliseconds



Hypothesis Testing

Conclusion

[1] "CORBA FAQ." [Online]. Available: https://corba.org/faq.htm.

[2] "graphics." [Online]. Available: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/boxplot.