Assignment 4

Caolan McDonagh

2022-10-20

Assignment 4

A company is interested in investigating whether adding a new variant to a particular website will improve i) the visit through time and ii) the proportion of query items resolved compared to that observed in the current website. A sample of 300 customers was directed to either the current or new variant of the website and their visit through time whether or not their query was resolved was recorded.

Variables of Interest: Visit Through Time: Time (secs) Resolved: Resolved (Yes / No)

Between Subject Factor: Variant: Variant (A/B) where 'A' represents the current website and 'B' represents the website with a new variant.

Assumptions and notes

-The shorted the site through time, the better. E.g like a customer support site. -A resolved query is a good result. -I will refer to Site A as "old" and Site B as "new" sites. -Markdown was not printing results, e.g 1x1 DF, it returned NaN, but running the chunk/line returned the value. In these cases I've commented the value.

I initially started by reviewing the data set provided for this question. It consists of 3 variables and 300 objects. -Variant (A/B) -Time (number in seconds) -Resolution (Yes/No)

The above can all be adapted to number (variant and resolution can translate to 1's and 0's), so we are dealing with a quantitative set of data.

This dataset can be broken down into two populations, where the variant is our common characteristic. This will give us two subsets with a frequency of 150 samples each.

Question 1

To estimate the probability that a random customer will have a visit through time on the new site of less than 3 minutes (180 seconds).

Here I use the number of visits (per site) that were less than 180 seconds, divided by the count of all visits (150 per site).

```
library(tidyverse)
library(dplyr)
library(plyr)
options(rgl.useNULL = TRUE)
library(tolerance)
library(ggplot2)
library(ggstatsplot)
library(pROC)
library(ROCR)
library(ggpubr)
```

```r
ab_test = read.csv('ab_test.csv')

head(ab_test)
```

```
##   Variant     Time Resolved
## 1       B 226.3488       No
## 2       B 231.9260      Yes
## 3       B 195.0814      Yes
## 4       B 179.5926       No
## 5       B 227.2649      Yes
## 6       B 118.1920       No
```

```r
siteAvisit <- subset(ab_test,Variant=='A') #Old
siteBvisit <- subset(ab_test,Variant=='B') #New

#Simple calculation for probability: number of occurrences over total
#observations.
newSiteVisitTime <- (nrow(subset(siteBvisit, Time <180))/nrow(siteBvisit))

#Probability that a random customer will have a visit through time on the new
#site of less than 3 minutes:
newSiteVisitTime #0.3266667
```

```
## [1] 0.3266667
```

```r
#Old site
oldSiteVisitTime <- (nrow(subset(siteAvisit, Time <180))/nrow(siteAvisit))


#Site A:
print(oldSiteVisitTime) #0
```

```
## [1] 0
```

```r
#Site B:
print(newSiteVisitTime) #0.3266667
```

```
## [1] 0.3266667
```

From the above results we can see the new site does not experience any customers having a visit through time on the new site. Comparing this to the probability of 32.6% that a given customer may have a >3 minute through time on the old site, we can say the new site is better in terms of speed. This is indicator that the new site has benefits over the old.

Question 2

To estimate the probability that at least 70% of queries on the new site will be resolved in a day, where it is assumed 55 customers visit the site in a day.

Here I first got the probability of resolution between the old and new sites; -Count of resolved site visits, divided by the total visits to that site.

Then I generated a random sample of 55 customers using the sample_n() method.

These were then used in conjunction; -count of resolved queries (in the sample of 55), divided by the 55 total.

To get some more solid analytical data, the below was run 20 times. The results of newSiteQueryComplete and oldSiteQueryComplete were compiled into a data frame as seen below.

```r
ab_test = read.csv('ab_test.csv')

head(ab_test)
```

```
##   Variant     Time Resolved
## 1       B 226.3488       No
## 2       B 231.9260      Yes
## 3       B 195.0814      Yes
## 4       B 179.5926       No
## 5       B 227.2649      Yes
## 6       B 118.1920       No
```

```r
siteAvisit <- subset(ab_test,Variant=='A') #Old
siteBvisit <- subset(ab_test,Variant=='B') #New

#2)

#Probability of resolved:
resolvedA <- (nrow(subset(siteAvisit, Resolved =='Yes' ))/nrow(siteAvisit))
resolvedB <- (nrow(subset(siteBvisit, Resolved =='Yes' ))/nrow(siteBvisit))

print(resolvedA) #0.4933333
```

```
## [1] 0.4933333
```

```r
print(resolvedB) #0.6
```

```
## [1] 0.6
```

```r
#Random sample of 55:
newSiteSample <- sample_n(siteBvisit, 55)
oldSiteSample <- sample_n(siteAvisit, 55)

#New site completion probability.
newSiteQuryComplete <- (nrow(subset(newSiteSample, Resolved =='Yes' ))
                        /nrow(newSiteSample))
#Old site completion probability.
oldSiteQuryComplete <- (nrow(subset(oldSiteSample, Resolved =='Yes' ))
                        /nrow(oldSiteSample))


print(newSiteQuryComplete)
```

```
## [1] 0.6545455
```

```r
print(oldSiteQuryComplete)
```

```
## [1] 0.5272727
```

```r
#Forgive long DF creation, did this to circumnavigate submitting an
#additional file.

Site<- c("New"  ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
```

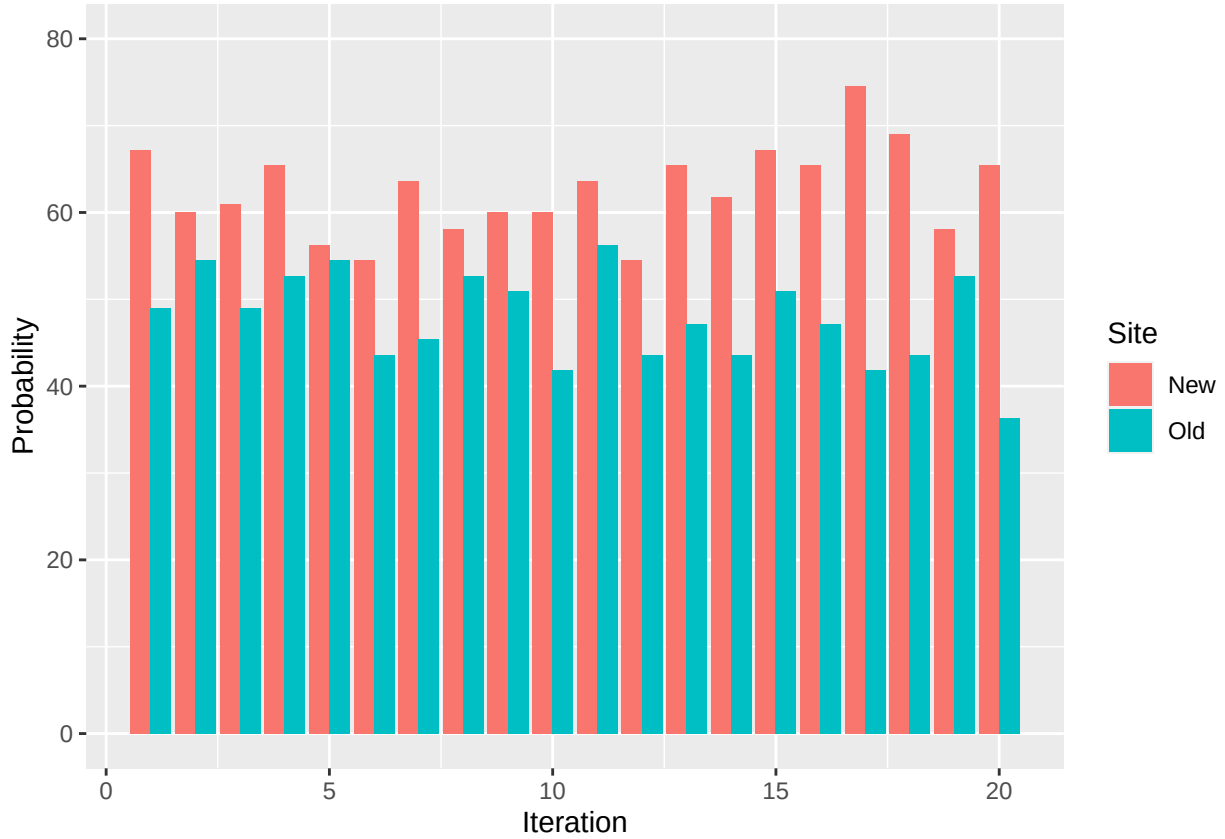
```
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"New"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   ,
"Old"   )
Iteration <- c(1    ,
2   ,
3   ,
4   ,
5   ,
6   ,
7   ,
8   ,
9   ,
10  ,
11  ,
12  ,
13  ,
14  ,
15  ,
16  ,
17  ,
18  ,
```

```
19  ,
20  ,
1   ,
2   ,
3   ,
4   ,
5   ,
6   ,
7   ,
8   ,
9   ,
10  ,
11  ,
12  ,
13  ,
14  ,
15  ,
16  ,
17  ,
18  ,
19  ,
20  )
Probability <- c(67.2   ,
60  ,
61  ,
65.4    ,
56.3    ,
54.5    ,
63.6    ,
58.1    ,
60  ,
60  ,
63.6    ,
54.5    ,
65.4    ,
61.8    ,
67.2    ,
65.4    ,
74.5    ,
69  ,
58.1    ,
65.4    ,
49  ,
54.5    ,
49  ,
52.7    ,
54.5    ,
43.6    ,
45.4    ,
52.7    ,
50.9    ,
41.8    ,
56.3    ,
```

```
43.6    ,
47.2    ,
43.6    ,
50.9    ,
47.2    ,
41.8    ,
43.6    ,
52.7    ,
36.3    )

testDF <- data.frame(Iteration,Site,Probability)

head(testDF)
```

```
##   Iteration Site Probability
## 1         1  New        67.2
## 2         2  New        60.0
## 3         3  New        61.0
## 4         4  New        65.4
## 5         5  New        56.3
## 6         6  New        54.5
```

```
#None of the random tests were >75, so kept y max to 80.
ggplot(testDF, aes(fill=Site, y=Probability, x=Iteration)) +
    geom_bar(position="dodge", stat="identity") + ylim(0,80)
```



From the above sample test data, in all of the 20 iterations, the new site had both a consistently higher

probability than the old, and never had a lower probability of success. From this we can extrapolate the newer site is statistically, more likely to resolve customer queries when compared to the older site, in a given day. This was an experiment more so, next I will use Bernoulli distribution to get a clearer answer.

This is another indicator towards the superiority of the newer site, over the old site.

```
library(Rlab)
```

```
## Rlab 4.0 attached.
```

```
##
## Attaching package: 'Rlab'
```

```
## The following objects are masked from 'package:plyr':
##
##     count, ozone
```

```
## The following object is masked from 'package:dplyr':
##
##     count
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```
## The following objects are masked from 'package:stats':
##
##     dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,
##     qweibull, rexp, rgamma, rweibull
```

```
## The following object is masked from 'package:datasets':
##
##     precip
```

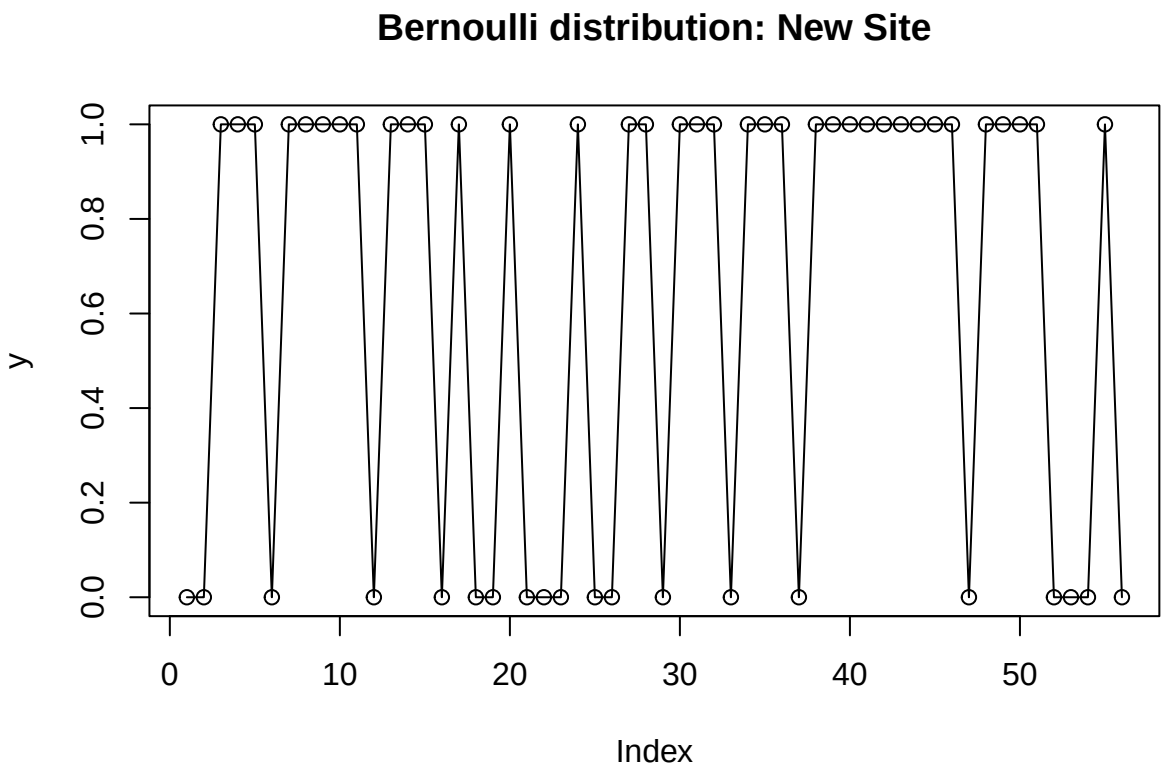
```
resolvedA #0.4933333
```

```
## [1] 0.4933333
```

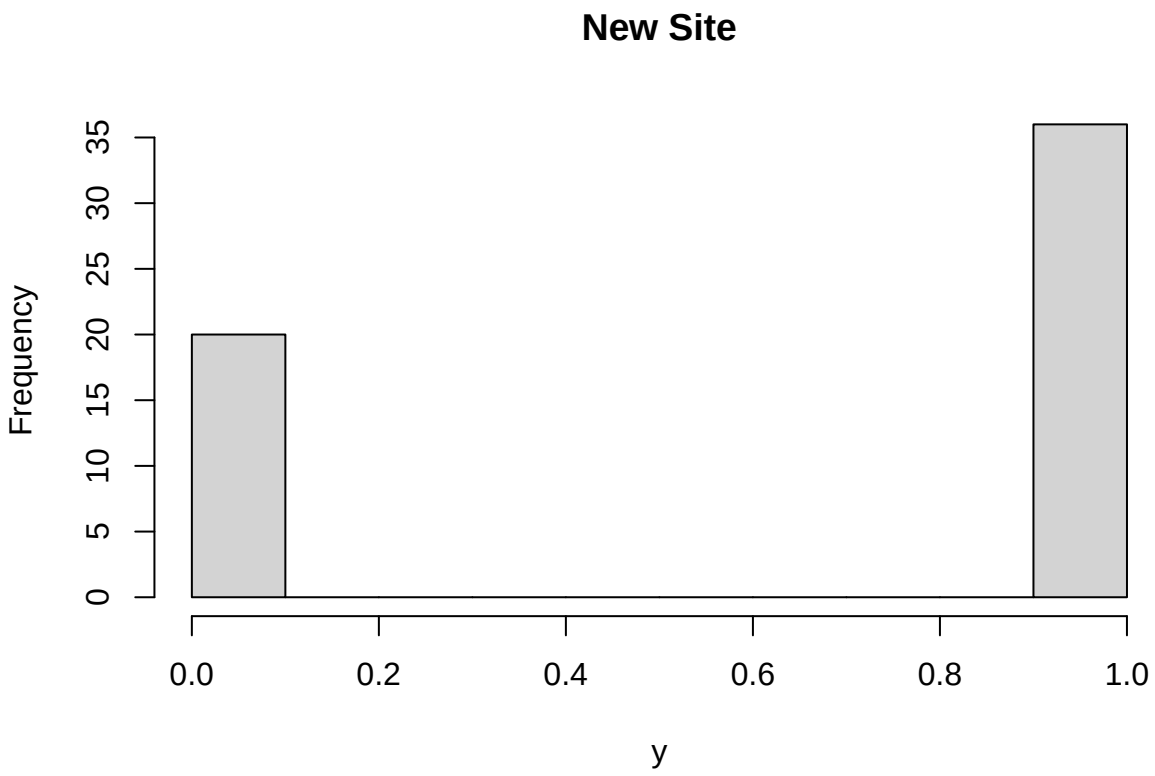
```
resolvedB #0.6
```

```
## [1] 0.6
```

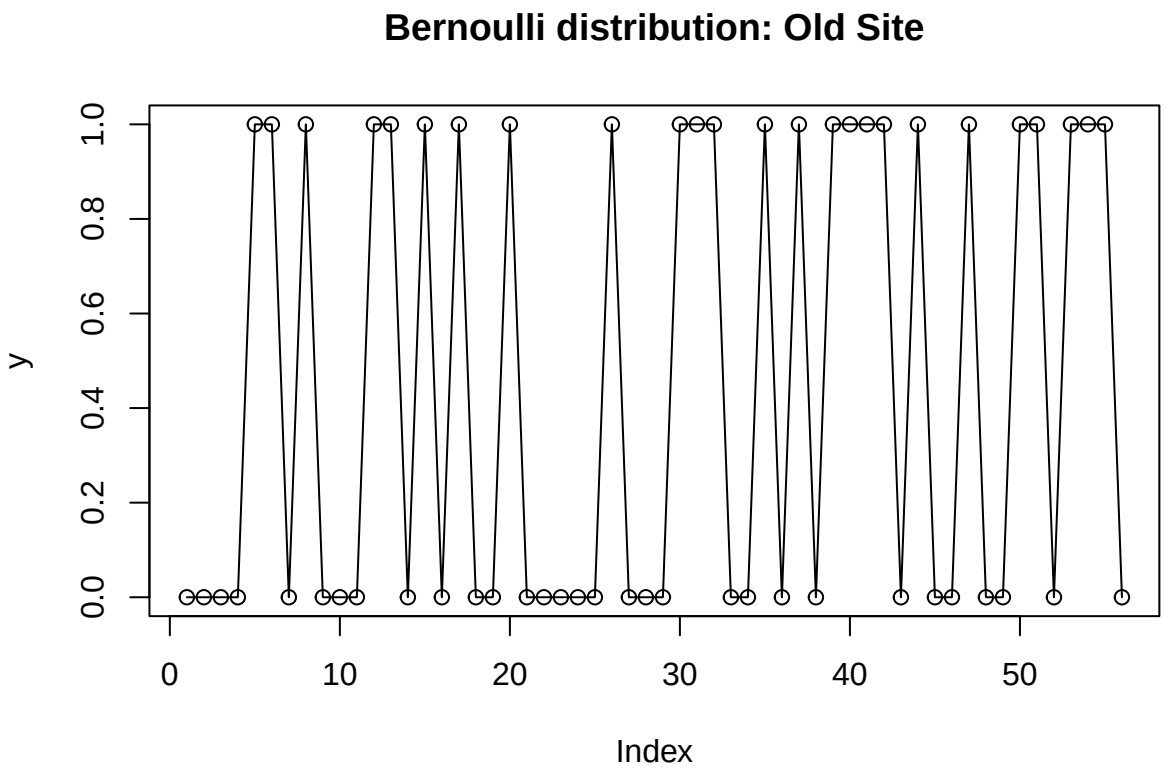
```
x <- seq(0, 55, by = 1)

y <- rbern(x, prob = 0.6)
plot(y, type = "o",main="Bernoulli distribution: New Site")
```



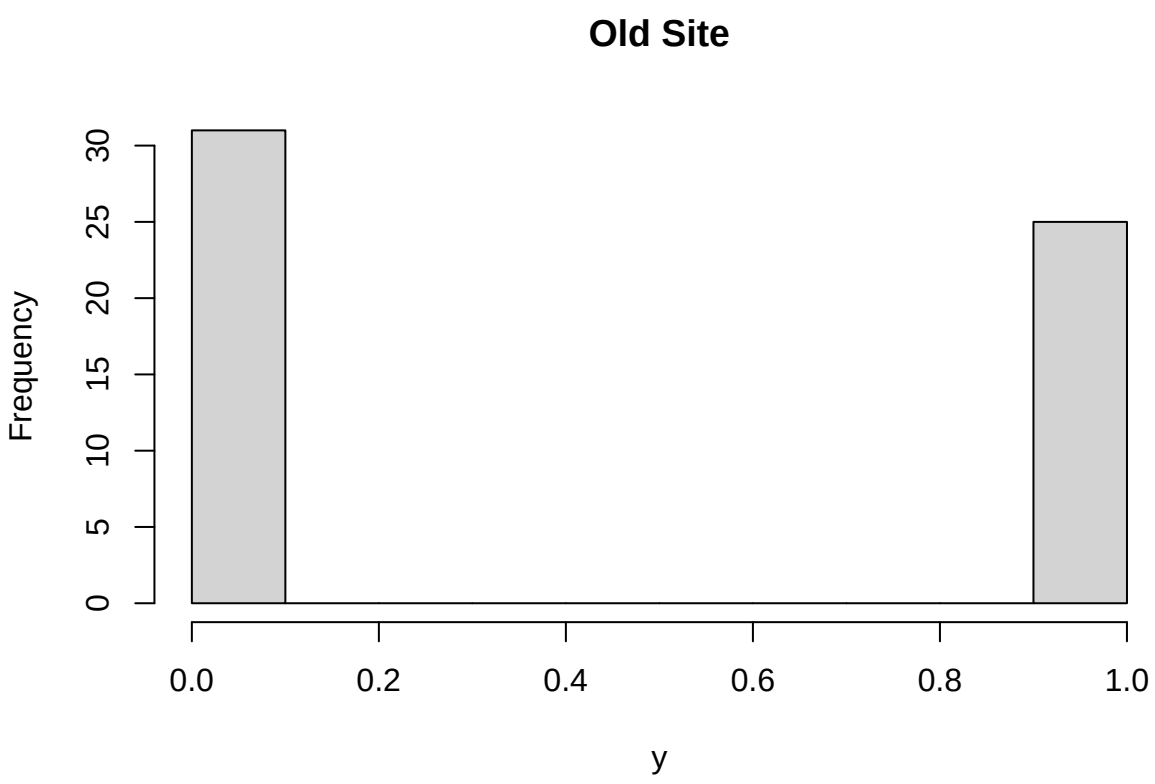

```
hist(y,breaks = 10,main = "New Site")
```




```
y <- rbern(x, prob = 0.4933333)
plot(y, type = "o",main="Bernoulli distribution: Old Site")
```




```
hist(y,breaks = 10,main = "Old Site")
```



Above shows a series of Bernoulli trials, using the probability of a resolution from the data sets of both old and new sites.

When plugging this data into a histogram, it is easy to see it is more favored to the new site, as the trials show more "Yes" resolutions in the new site, compared to the old.

Now if we want to put a solid probability to the above question, we can use binomial distribution:

```
#70% of 55 = 38.5

#New
x <- pbinom(38.5,55,.6)
print(x)
```

```
## [1] 0.9369371
```

```
#Old
x <- pbinom(38.5,55,.4933333)
print(x)
```

```
## [1] 0.9990379
```

The above function returns the pbinom(), the cumulative probability of an event. It returns the probability that x<70%. To get the answer to the question (x>70%) we simply take this away from 1:

New site: 1 - 0.9369371 = 0.0630629 = ~6.3%

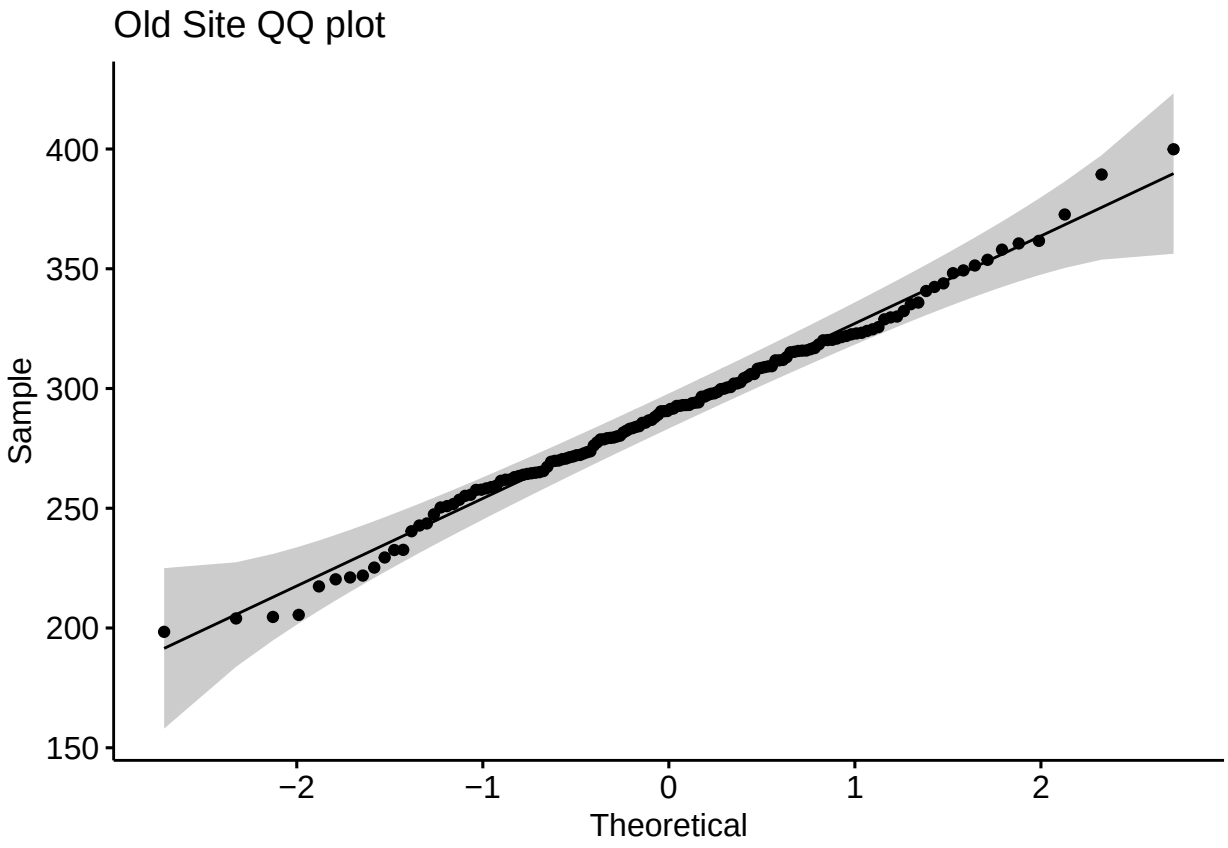
Old site: 1 - 0.9990379 = 0.0009621 = ~0.009%

This is essentially a 6.2% better chance that the new site will see 70% of queries on a day (55 queries on a given day) are resolved. These are much more favorable odds to see.

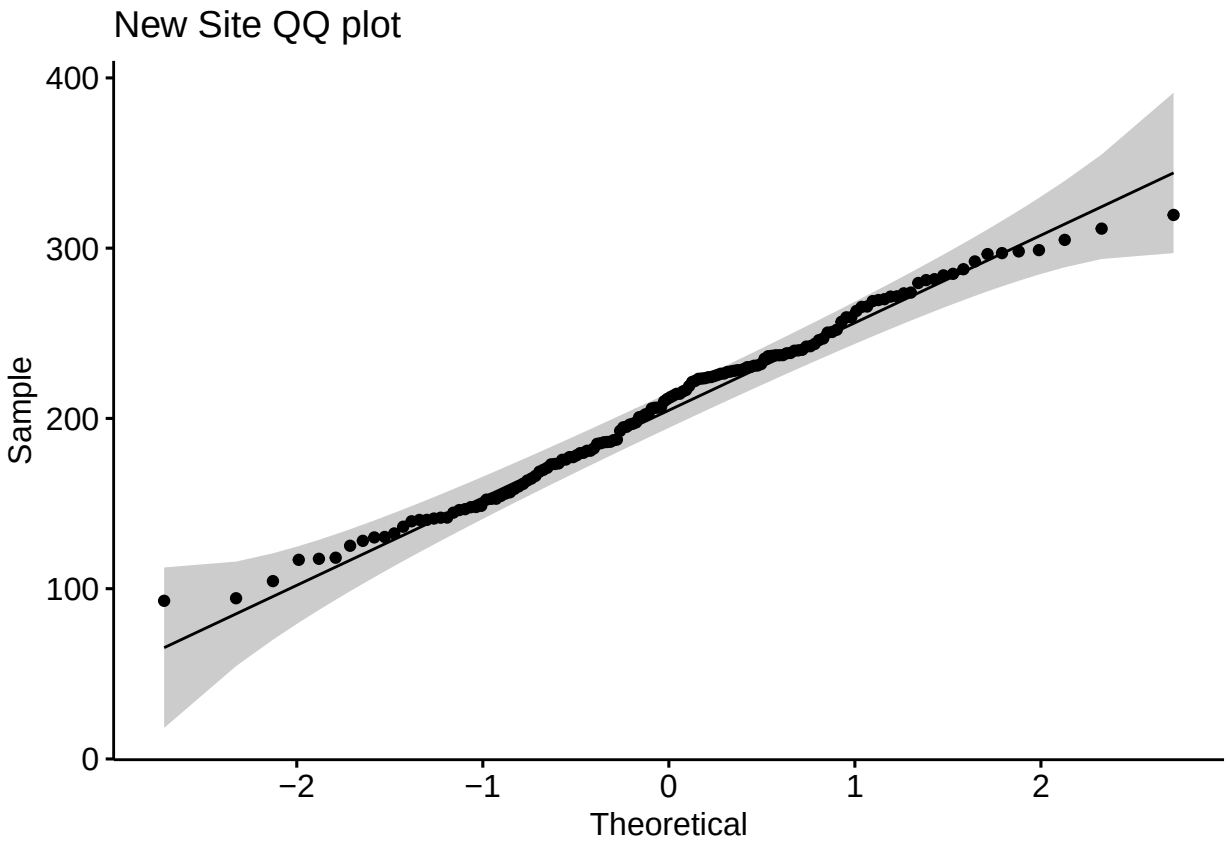
Further report

Below I have complied useful plots and graphs for statistical analysis. The first of which is the mean visit times.

```
#Old Site
ggqqplot(siteAvisit$Time,main="Old Site QQ plot")
```




```
#New Site
ggqqplot(siteBvisit$Time,main="New Site QQ plot")
```



Above are two QQ (Quantile Quantile) plots, showing our old and new sites. We can see both data sets look to have normally distributed data. This reinforces our analysis above and further down in this report.

```
shapiro.test(siteAvisit$Time)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  siteAvisit$Time
## W = 0.99228, p-value = 0.5955
```

```
shapiro.test(siteBvisit$Time)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  siteBvisit$Time
## W = 0.98765, p-value = 0.2054
```

Above is the results of the Shapiro-Wilks test, this is another test of normality to bolster my argument that this data is normal.

If Shapiro-Wilks p-value shows greater than 0.05, data has no significant departure from normality. We can see both sites produced 0.59 and 0.20, adhering to our QQ plot results.

```
meanTimeA = mean(siteAvisit$Time)
meanTimeB = mean(siteBvisit$Time)

print(meanTimeA) #Old Mean
```

```
## [1] 290.1473
```

```
print(meanTimeB) #New Mean
```

```
## [1] 207.1608
```

The old mean: 290 The new mean: 207

This is an 83 second gap, on average, between the two sites. This a relatively large jump in average speeds between the two, another indicator towards the new sites advantages.

Below are compiled quanitles for through speed ranging from 0% to 100% in 25% intervals.

```
quantileTimeA = quantile(siteAvisit$Time)
quantileTimeB = quantile(siteBvisit$Time)

print(quantileTimeA)
```

```
##       0%      25%      50%      75%     100%
## 198.3982 265.9794 291.0173 315.2672 399.8961
```

```
print(quantileTimeB)
```

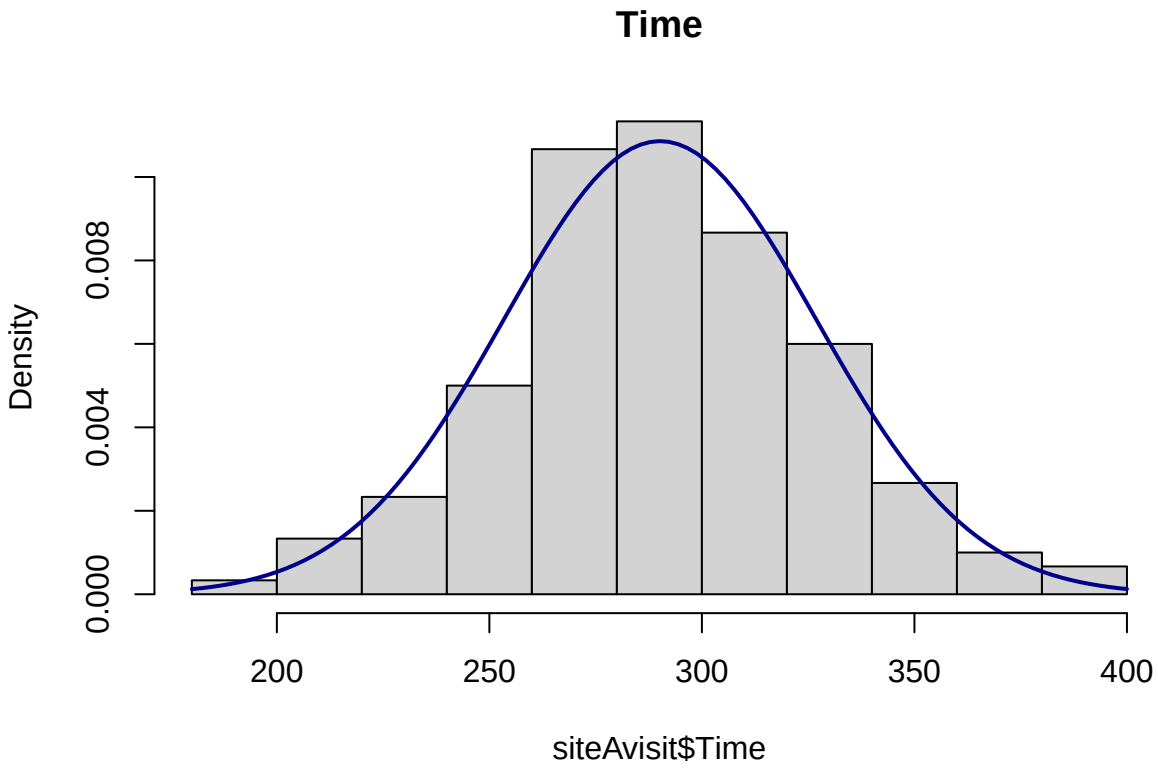
```
##        0%       25%       50%       75%      100%
##  92.88389 170.10585 211.97506 239.43593 319.51047
```

Each quantile of the new site is marginally faster than the old, again reinforcing the hypothesis that the new site is a better representation for this company.

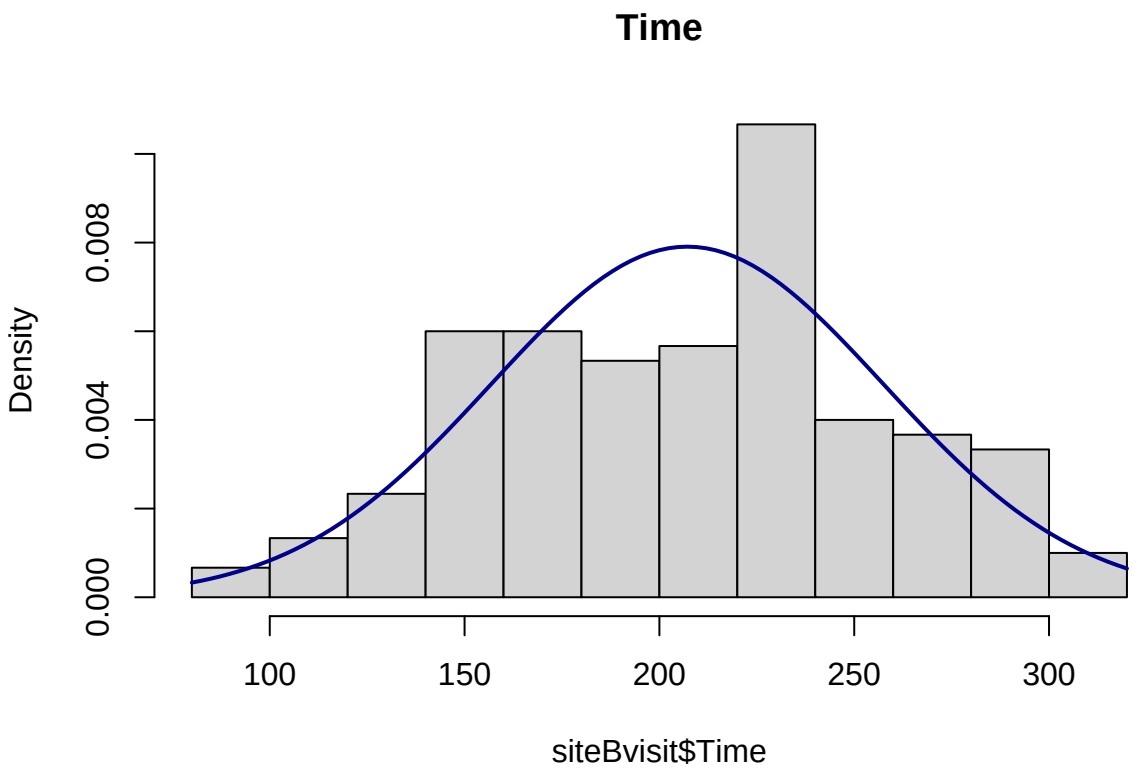
Differences between quantiles, new site leading in all by x seconds: 0%: 106 second lead 25%: 95 second lead 50%: 80 second lead 75%: 76 second lead 100%: 80 second lead

This all compiles to an average of 87.6 seconds faster across the range of the dataset, for the new website.

```
m<-mean(siteAvisit$Time);std<-sqrt(var(siteAvisit$Time))
hist(siteAvisit$Time,prob=T,main="Time")
curve(dnorm(x, mean=m, sd=std), col="darkblue", lwd=2, add=TRUE)
```



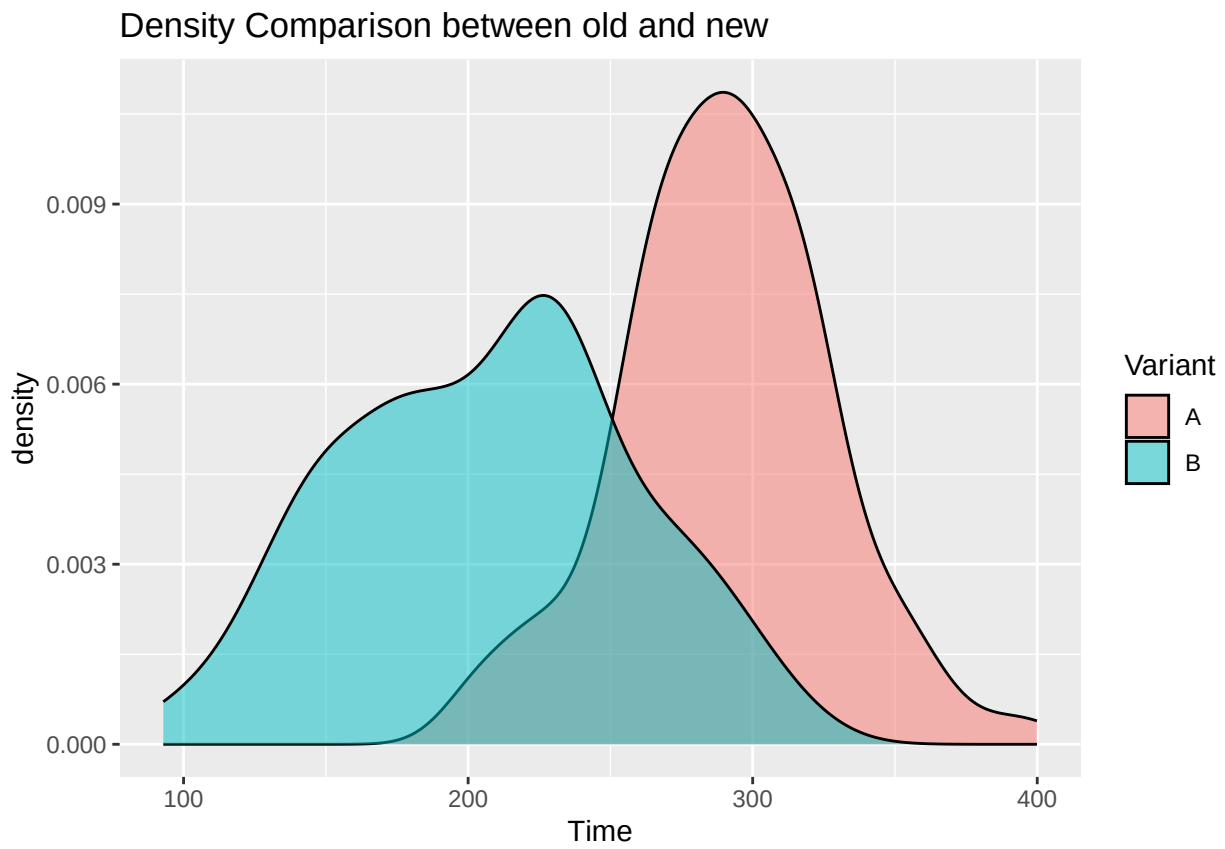

```
m<-mean(siteBvisit$Time);std<-sqrt(var(siteBvisit$Time))
hist(siteBvisit$Time,prob=T,main="Time")
curve(dnorm(x, mean=m, sd=std), col="darkblue", lwd=2, add=TRUE)
```



Above is a set of histograms, each represent the old and new sites, showing the times and the probability density function (density). It is easy to see new site has a shallower peak compared to the old, this is better seen in the below graph.

There is a central tendency within the A site visit time between 275 - 325. Relatively slow compared the central tendency of B site, which is more widely spread between 150-250. This spread is another observation to take note of, B is more dispersed compared to A, this isn't bad though as you can see this larger range, is still lower than the centralized results of A.

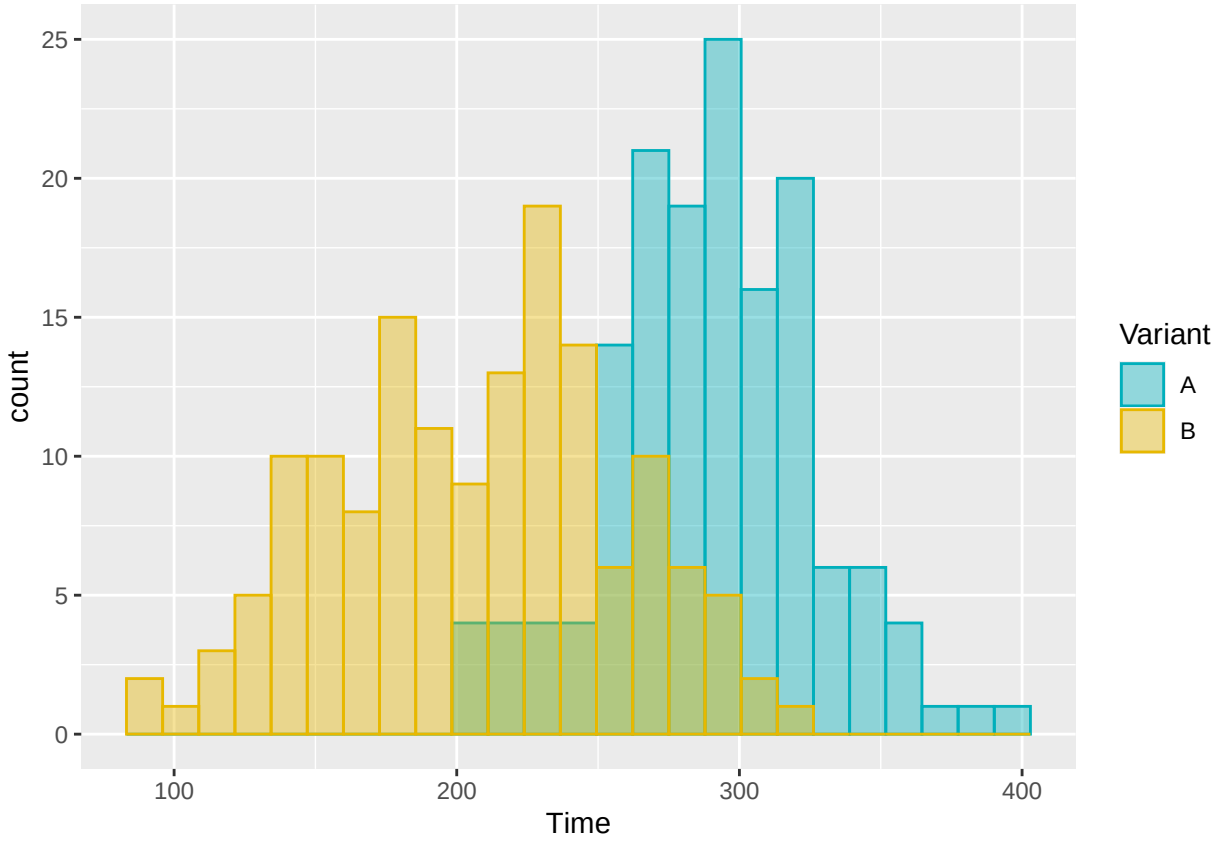
```
ggplot(ab_test, aes(x = Time, fill = Variant)) + geom_density(alpha = 0.5) +
  ggtitle("Density Comparison between old and new ")
```



Now we can properly compare the two sites density plots. The Old site is more consistent, but it is more consistently slower to get through. A large portion of the new sites traffic resides in the sub 250 second mark, where only a fraction of the old sites traffic sits. This is a very good look for the new site, as it is much faster for the majority of their traffic.

Noting the shapes here, the A variant has a relatively normal/bell curve in the upper half of our time range, where as B has a rather positively skewed tail, where the majority sits in the lower half of the time range. A good observation for the new site, as the company would want more visits in that lower range of time.

```
ggplot(ab_test, aes(x = Time)) +
  geom_histogram(aes(color = Variant, fill = Variant),
                 position = "identity", bins = 25, alpha = 0.4) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  scale_fill_manual(values = c("#00AFBB", "#E7B800"))
```



Above is a grouped histogram showing the times by variant, you can see the similarity with the previous density plot, with the normal curve of A and positively skewed tail of B.

```
welch.test <- t.test(siteAvisit$Time, siteBvisit$Time)
welch.test
```

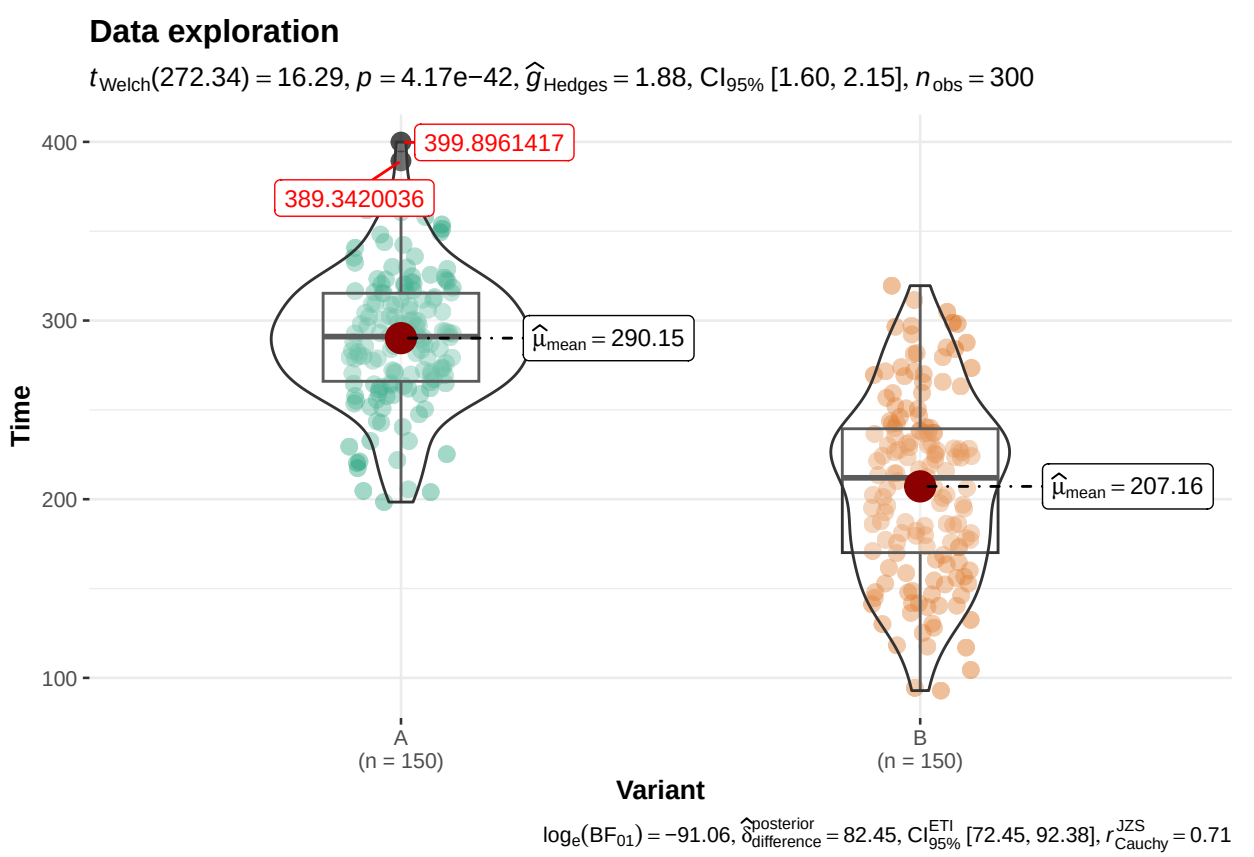
```
##
##  Welch Two Sample t-test
##
## data:  siteAvisit$Time and siteBvisit$Time
## t = 16.286, df = 272.34, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  72.95486 93.01806
## sample estimates:
## mean of x mean of y
##  290.1473  207.1608
```

Above I've calculated the confidence interval via Welsch's t-interval: 72.95486 & 93.01806

p-value $<$ 2.2e-16, so we can reject the null hypothesis and accept there is a difference between the two true averages of the old and new site.

```
statsplot <- ab_test %>%
ggbetweenstats(x = Variant, y = Time, type = "Parametric",
               outlier.tagging = TRUE,
               outlier.label.args = list(color = "red", size = 3)) +
  ggtitle("Data exploration")
```

```
statsplot
```



Above is a very interesting package for R: ggbetweenstats

It comes preloaded with several useful visuals for your statistical data.

Here I've used the initial dataframe, splitting between groups A/B (Old/New). It provides an easy to read set of graphics, above we can see: -Box plot -density plot -Raw data (dot plot) -Centrality measure -Outliers -Mean

It also adds hypothesis testing, I've configured the ggbetweenstats to use parametric tests. Because there is only 2 groups, it makes use of Welch's T-test

It also shows the Effect size estimation, which for parametric 2 groups, it uses Hedges' g for standardized difference estimation (with bias correction).

Confidence interval can also be seen alongside this.

With the understanding that a lower time is better, it is easy to extract from this visualization that group B (new site) is a superior group.

```
#Old Site
normtol.int(siteAvisit$Time,alpha = 0.05, P = 0.95, side = 2)
```

```
##   alpha    P    x.bar 2-sided.lower 2-sided.upper
## 1  0.05 0.95 290.1473       210.219      370.0756
```

```
#New Site
normtol.int(siteBvisit$Time,alpha = 0.05, P = 0.95, side = 2)
```

```
##   alpha    P    x.bar 2-sided.lower 2-sided.upper
```

```
## 1  0.05 0.95 207.1608      97.39847      316.9232
```

Above is our 95/95 Tolerance interval for the visit times through old and new sites.

We can be 95% confident that 95% of the population will be between: Old Site: 210 and 370 seconds New Site: 97 and 316 seconds

The new site is obviously the better choice here as the lower is 113 seconds faster, and the upper 54 seconds faster.

For 95% of predicted users to be within this range, it is a much more ideal to make use of the new site.

Our data set is rather limited in terms of information available to us, it is simply a yes/no and time. We can't really measure any meaningful relationships from this information, so the correlation and causation isn't available to us, which I think would be a worthwhile investigation for the company to see what is causing the improvements between the two sites, and hone then.

Overall, it is easy to see the optimal choice between the two sites. The new site (Group B) has consistently been both quicker for the customer through time and more successful in resolving customer queries. There is no arguable reason to continue using the old site, based on the results of my statistical analysis. The sum of all of the above investigation shows the importance and value in statistical investigation and evaluation of data.

I believe the company should move to the new site, and reap the benefits it provides.