Dissert-exp-dia-hora

Gabriel Robaina

2023-11-15

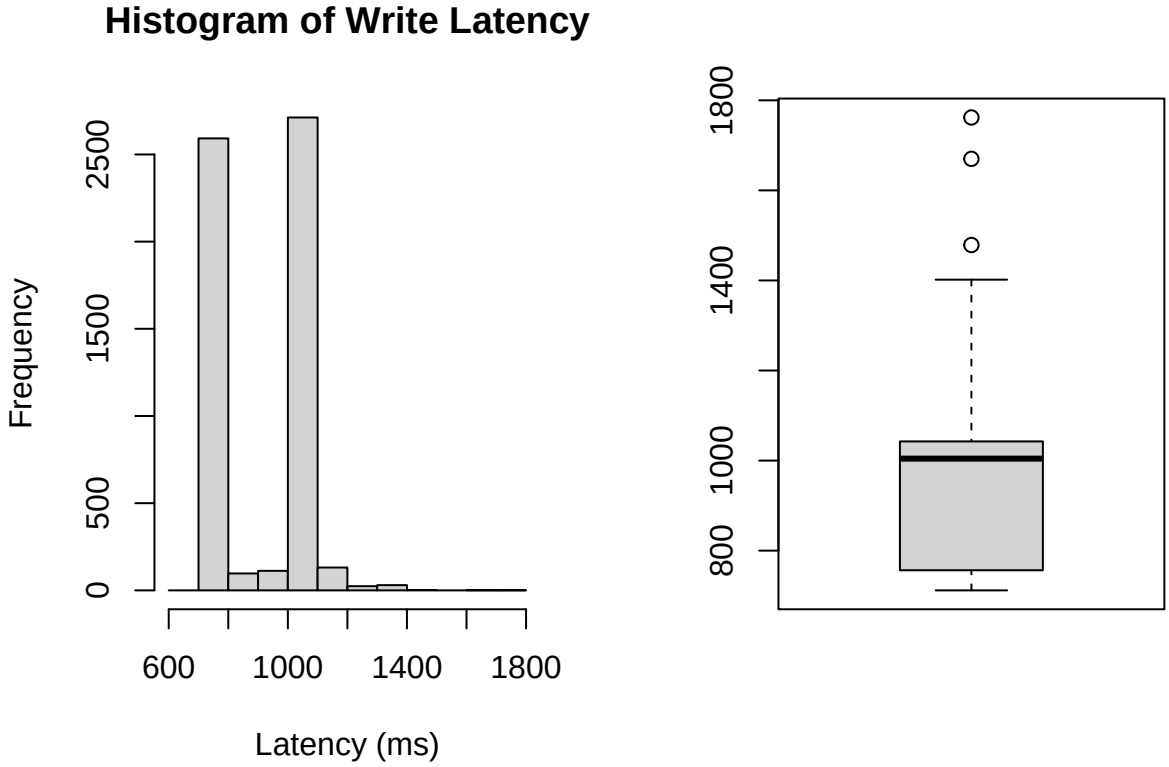
Day and time experiment

The goal of this experiment is to compare data from weekends and weekdays and see if there is any performance difference between them. Another goal is to compare business hours to off hours to see if there is any performance difference there. We will divide this into analysis of write and read operations.

All data in weekends are considered to be in off hours.

Write operations analysis

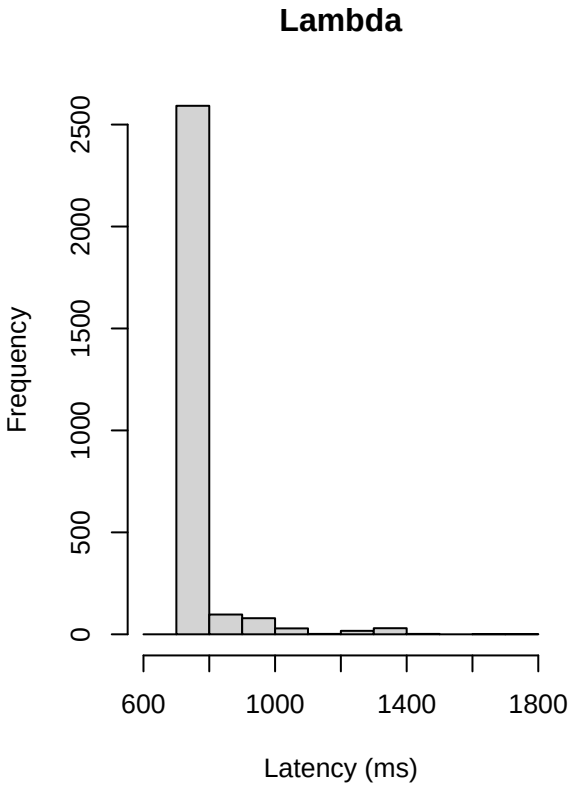
The first histogram was based on the latency of all data for write operations. We can see two modes: The one in the left is the interval (700,800] and is asymmetric. The right mode is symmetrical and is located in the interval (1000,1200]. For this reason, this is an asymmetric bimodal data distribution.

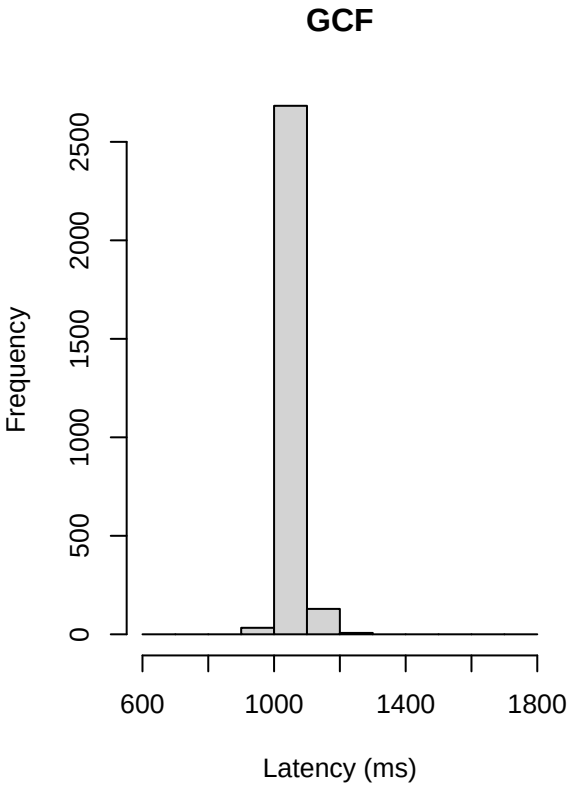


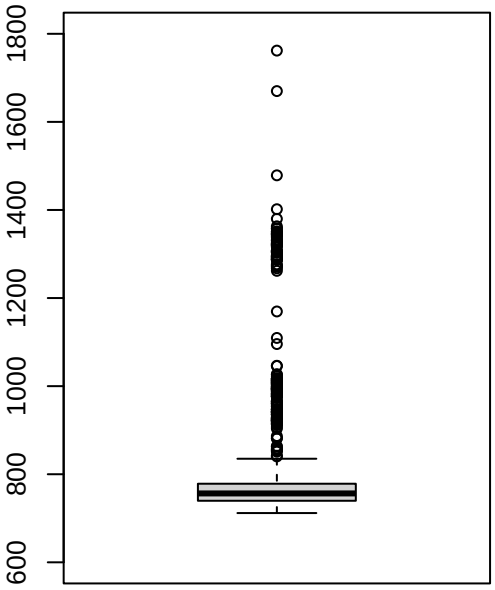
Let's look for the source of both modes. Maybe its from the provider.

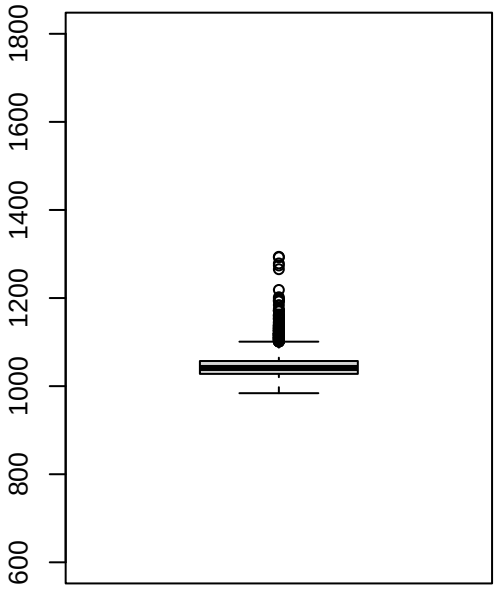
Write operations analysis per provider

When we plot the latency distribution per provider, we can find the source of the two modes. The left mode belongs to Lambda, while the other one belongs to GCF. The distributions for the providers are asymmetric. At first glance, it seems GCF is slower than Lambda for write operations. Lets divide the analysis into two providers so we can check the relevance of day of the week and time of day.


Lambda
Frequency
0
500
1000
1500
2000
2500
600
1000
1400
1800
Latency (ms)


GCF
Frequency
0
500
1000
1500
2000
2500
600
1000
1400
1800
Latency (ms)

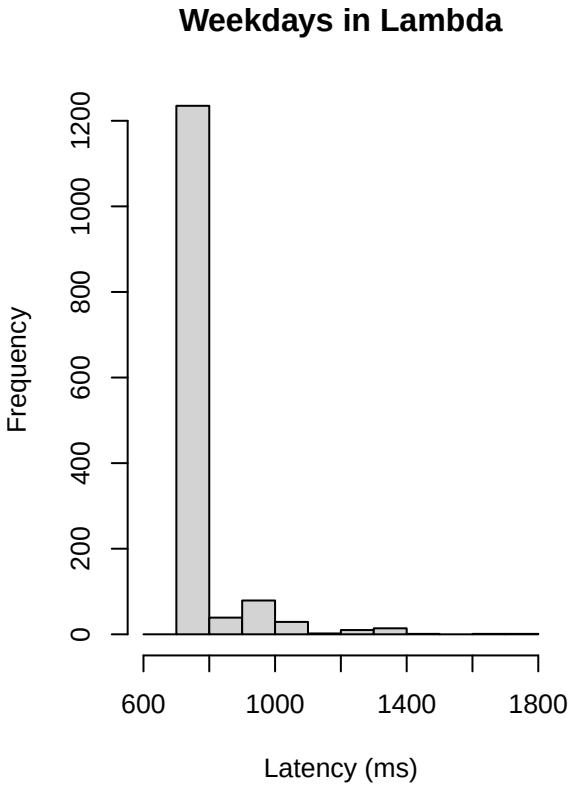

600
800
1000
1200
1400
1600
1800

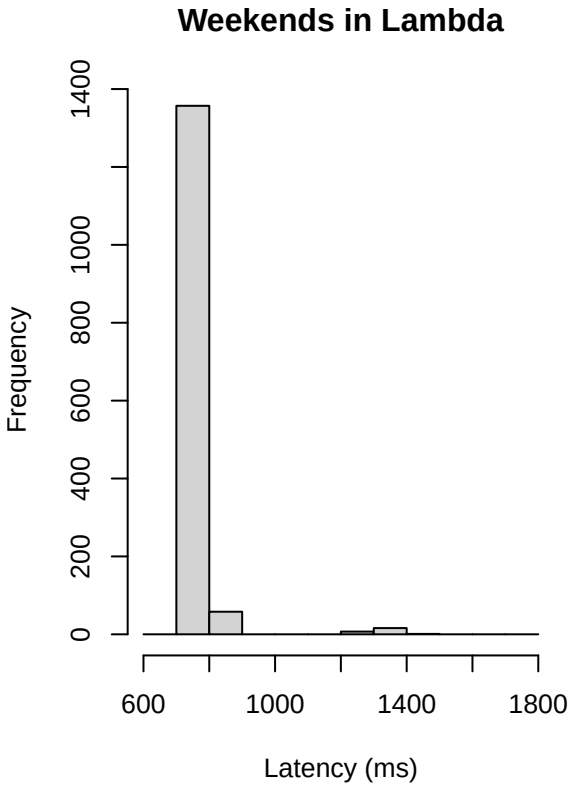

600
800
1000
1200
1400
1600
1800

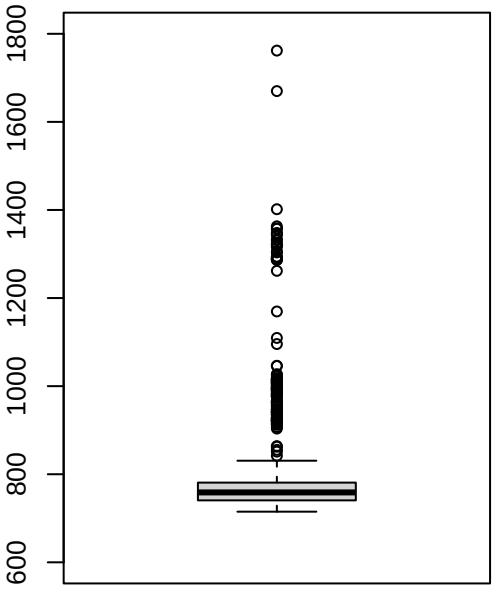
Lambda analysis

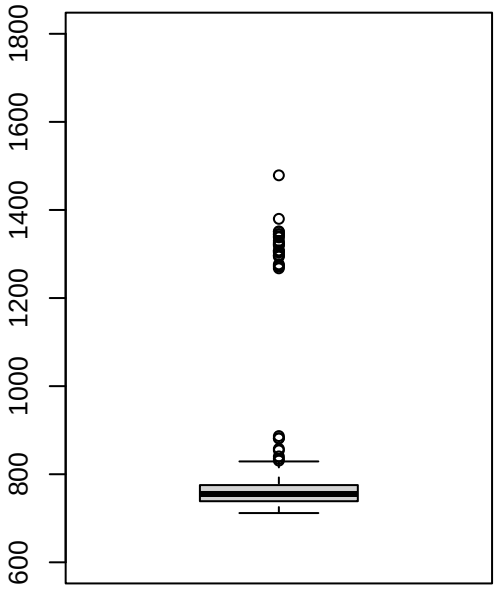
Weekdays and weekends in Lambda

First, lets plot histograms for both weekdays and weekends and see if they have a similar distribution. No visible difference other than weekends aparently having more stable performance with less outliers.


Weekdays in Lambda
Frequency
0
200
400
600
800
1000
1200
600
1000
1400
1800
Latency (ms)

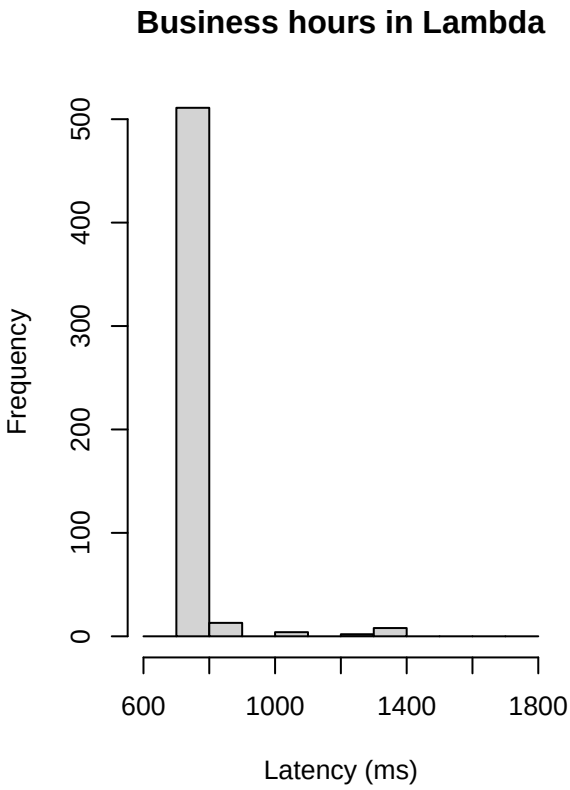

Weekends in Lambda
Frequency
0
200
400
600
800
1000
1400
600
1000
1400
1800
Latency (ms)

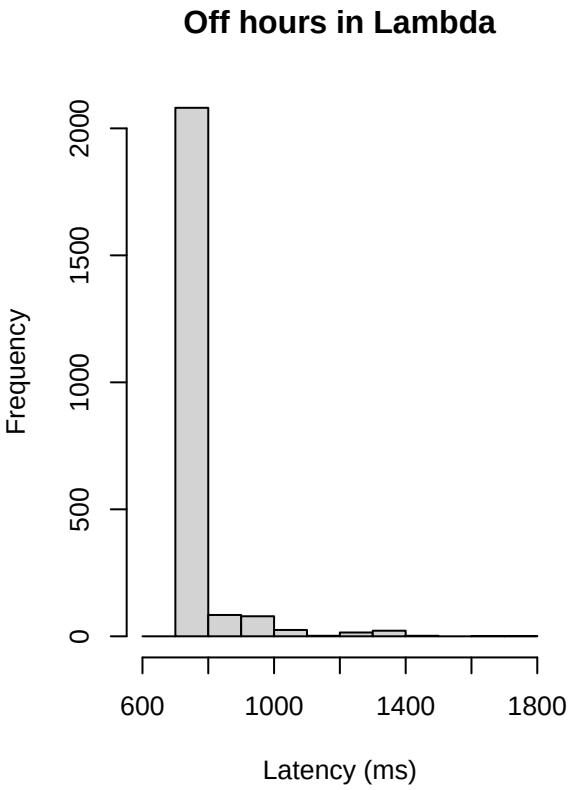

600
800
1000
1200
1400
1600
1800

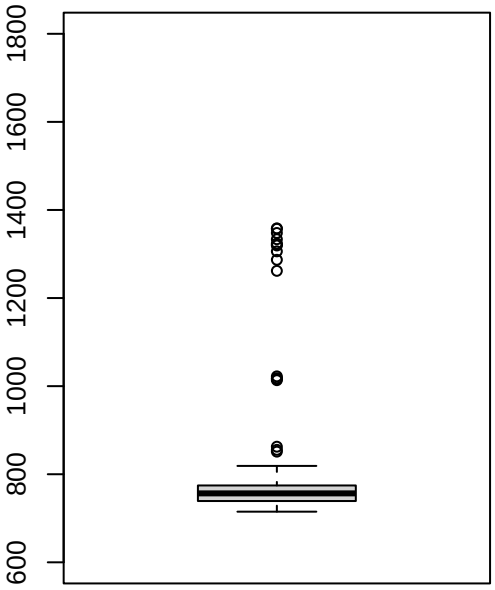

600
800
1000
1200
1400
1600
1800

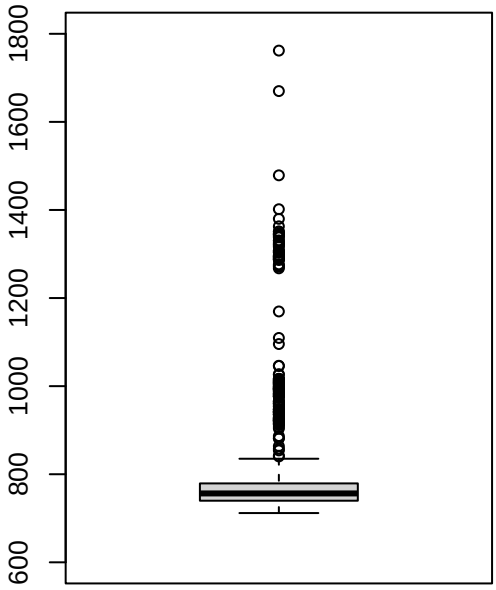
Business and off hours in Lambda

Second, lets plot histograms for both business hours and off hours and see if they have a similar distribution in Lambda. Also no noticeable performance difference here besides business hours apparently having more stable performance.


Business hours in Lambda
Frequency
0
100
200
300
400
500
600
1000
1400
1800
Latency (ms)

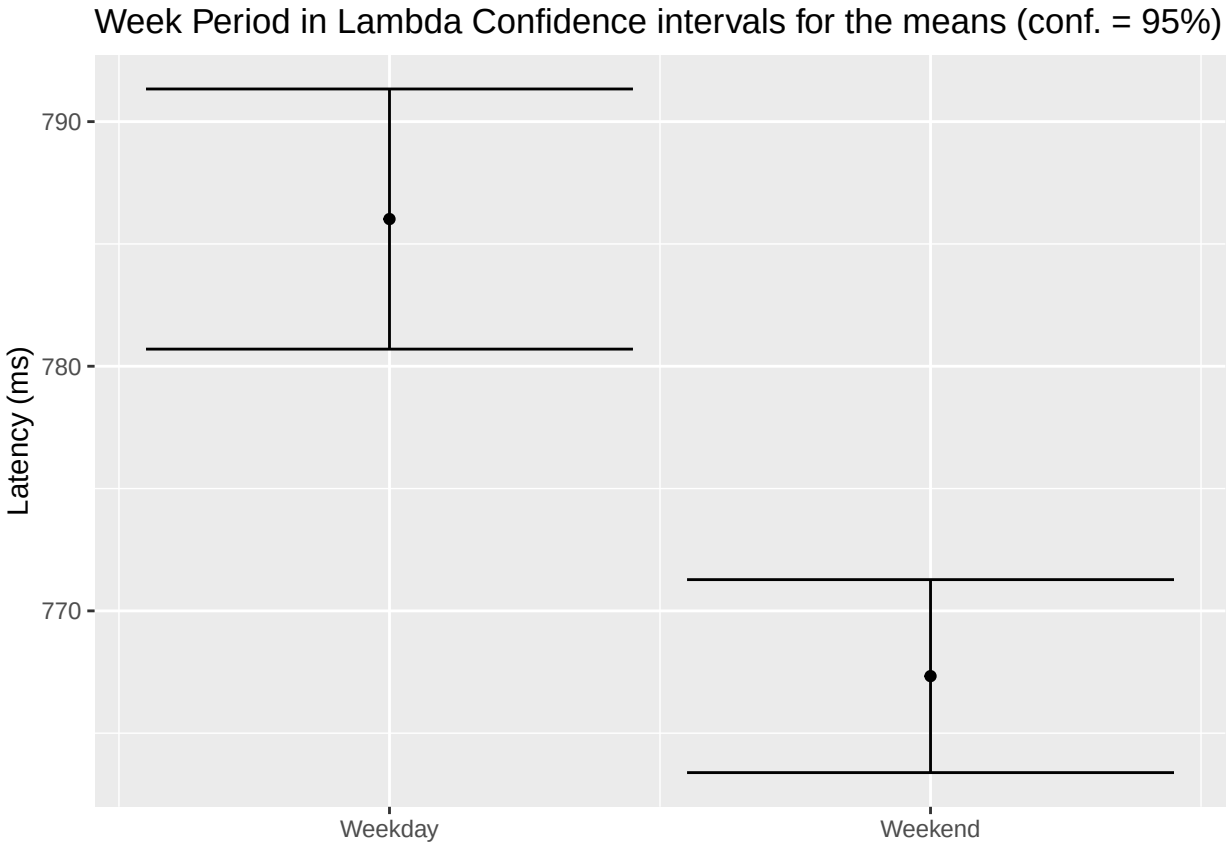

Off hours in Lambda
Frequency
0
500
1000
1500
2000
600
1000
1400
1800
Latency (ms)


600
800
1000
1200
1400
1600
1800

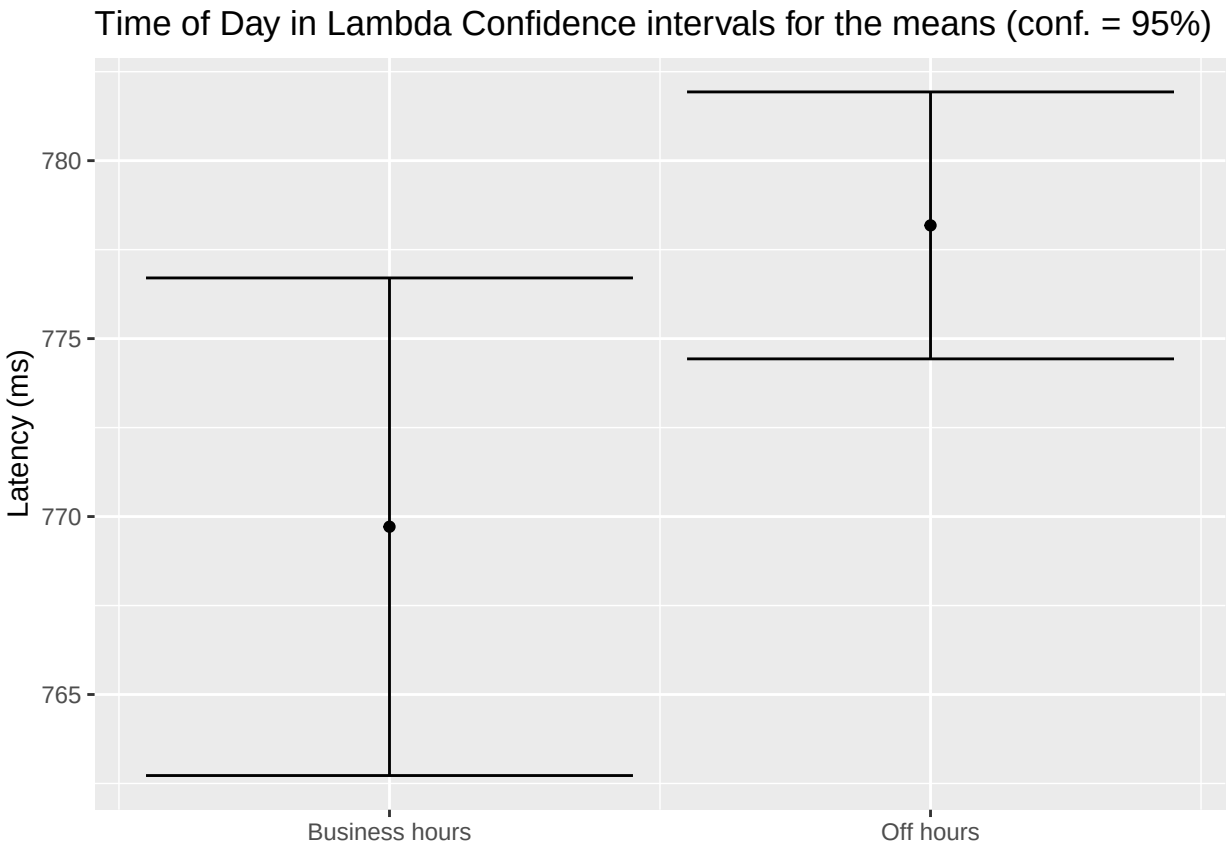

600
800
1000
1200
1400
1600
1800

Confidence intervals in Lambda

As we can see from the confidence intervals for the means on week periods, we can confirm there is statistically significant difference between latencies measured during weekdays and weekends, since the confidence intervals have no overlap.



Even though the confidence intervals for business and off hours have some overlap, we cannot confirm there are no statistically significant difference between them since this overlap does not include either mean values of any interval. This means we need further testing to see if the performance between business hours and off hours is different, even though we already know it is for weekends and weekdays.



ANOVA in Lambda

By looking at the ANOVA results, both time of day and week period factors are statistically significant for a confidence level of 95%.

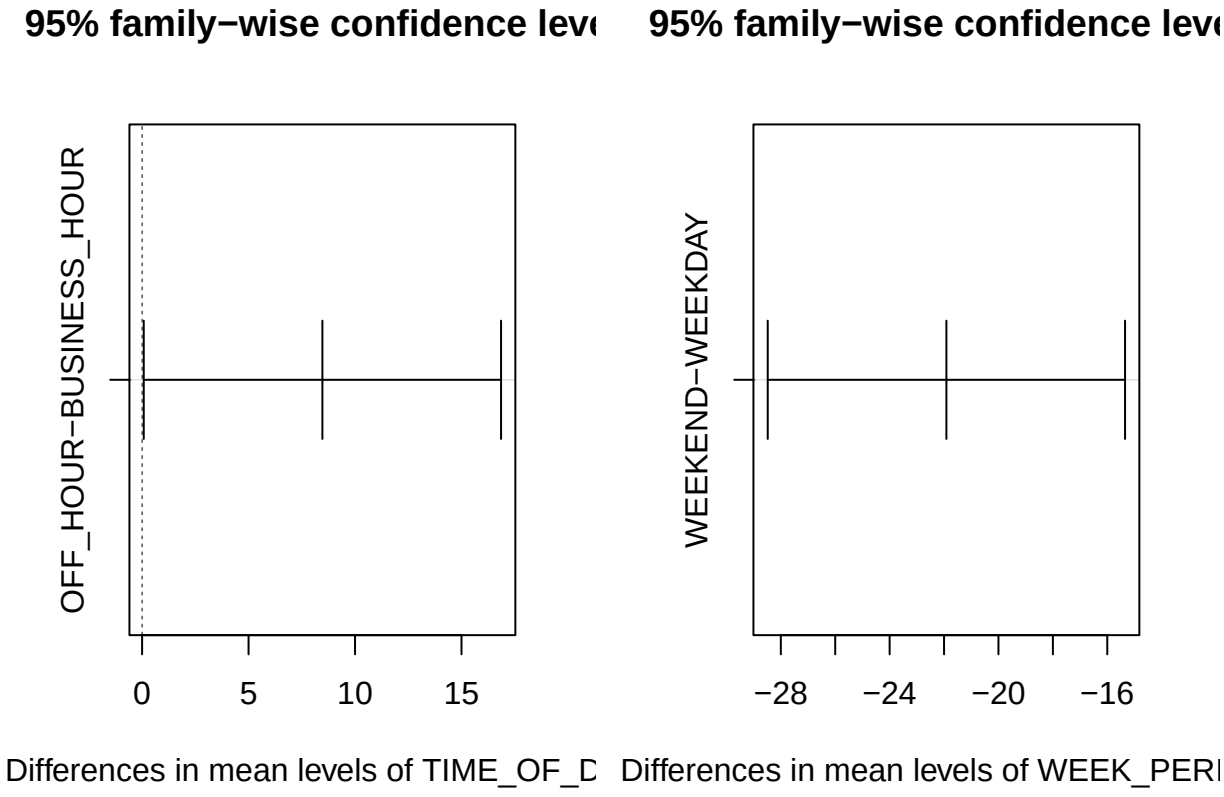
```
##                Df   Sum Sq Mean Sq F value   Pr(>F)
## TIME_OF_DAY     1    31291   31291   3.915   0.0479 *
## WEEK_PERIOD     1   448561  448561  56.127 9.02e-14 ***
## Residuals    2847 22753115    7992
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The allocation of variation shows most of the variation is due to random error. Even though time of day is a relevant factor it participates very little in the total variation.

```
##              Sum Sq
## TIME_OF_DAY    0.13
## WEEK_PERIOD    1.93
## Residuals     97.93
```

```
## [1] 3.844727
```

From the Tukey test we can see the confidence intervals for the difference between business hours and off hours does not include 0, but it is very close to it. We can also see that the difference between weekend and weekday is negative, meaning that weekdays are less performant.



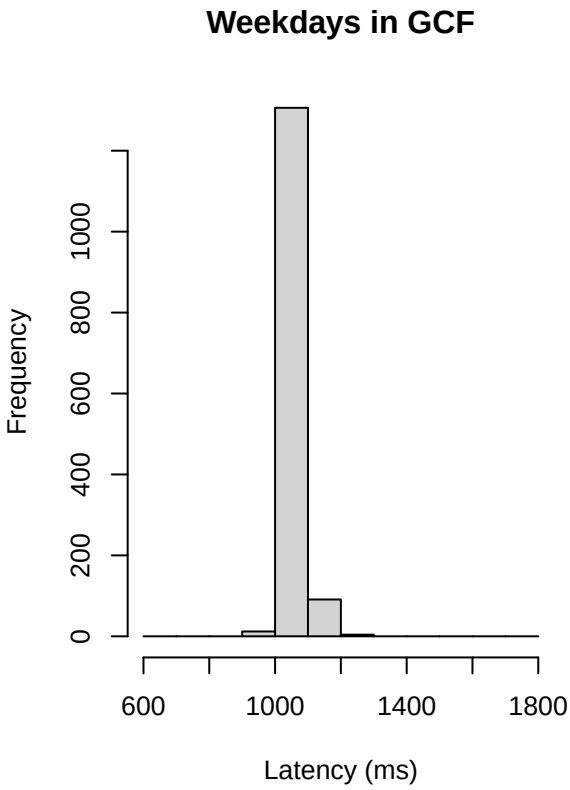
GCF analysis

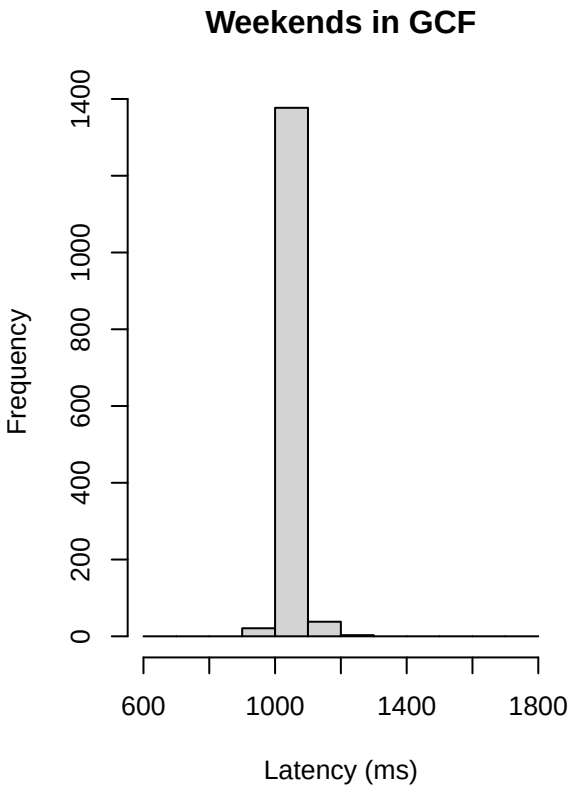
Weekdays and weekends in GCF

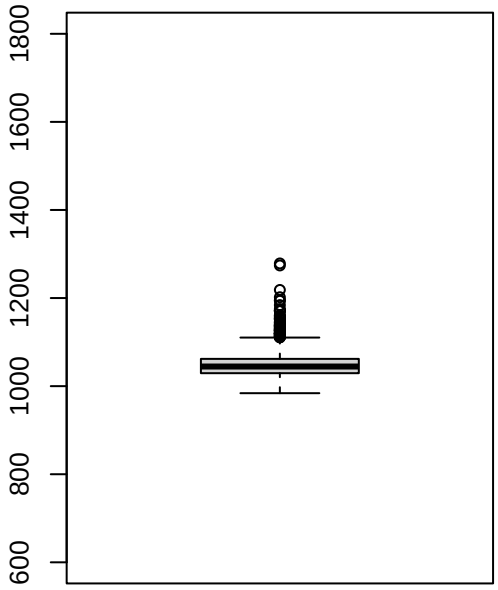
```
data_write_weekday_gcf <- data_write_gcf[data_write_gcf$WEEK_PERIOD == 'WEEKDAY',]
data_write_weekend_gcf <- data_write_gcf[data_write_gcf$WEEK_PERIOD == 'WEEKEND',]
```

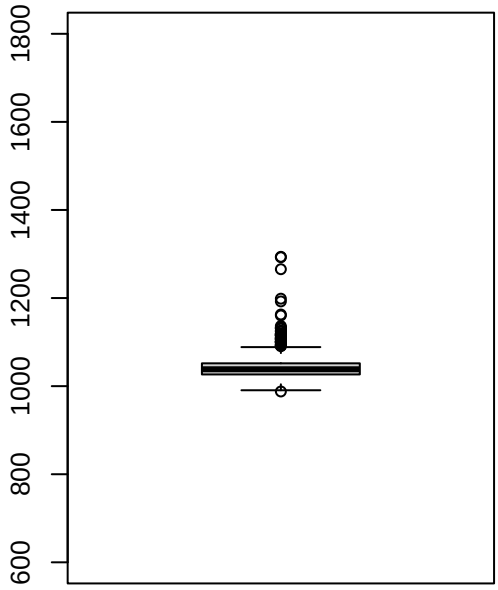
First, lets plot histograms for both weekdays and weekends and see if they have a similar distribution.

No visible difference other than weekends aparently having more stable performance with less outliers. This difference is very small though.


Weekdays in GCF
Frequency
0
200
400
600
800
1000
600
1000
1400
1800
Latency (ms)

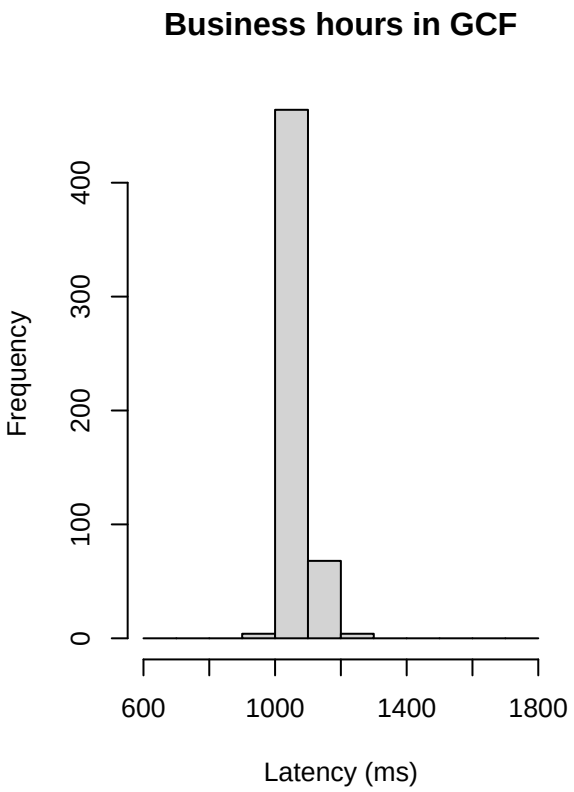

Weekends in GCF
Frequency
0
200
400
600
800
1000
1400
600
1000
1400
1800
Latency (ms)

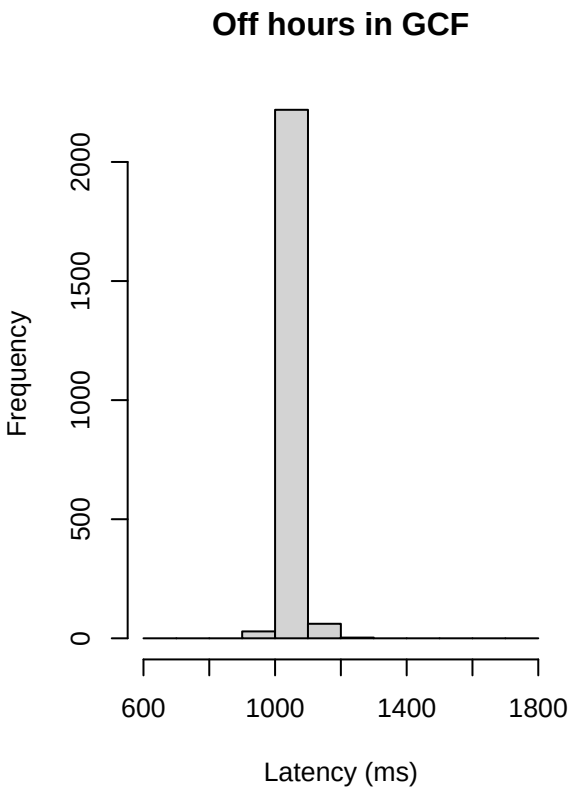

600
800
1000
1200
1400
1600
1800

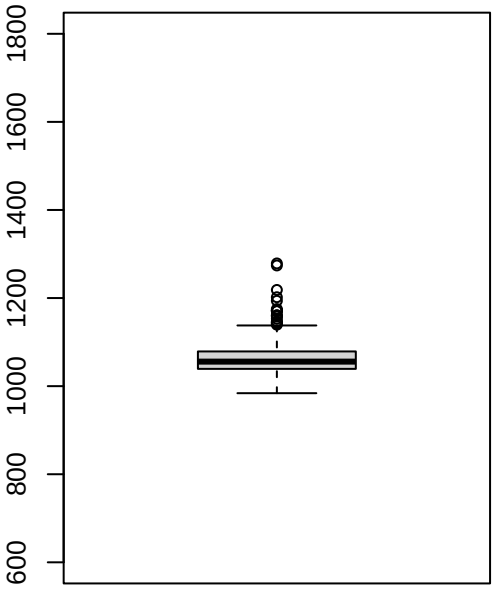

600
800
1000
1200
1400
1600
1800

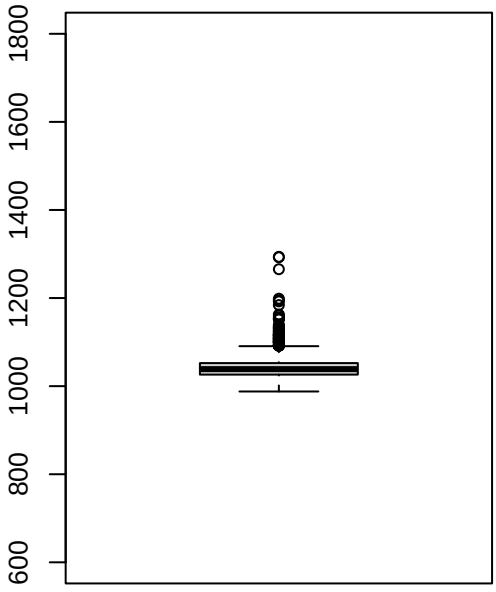
Business and off hours in GCF

Second, lets plot histograms for both business hours and off hours and see if they have a similar distribution in GCF. Also no noticeable performance difference here besides off hours apparently having more stable performance.


Business hours in GCF
Frequency
0
100
200
300
400
600
1000
1400
1800
Latency (ms)

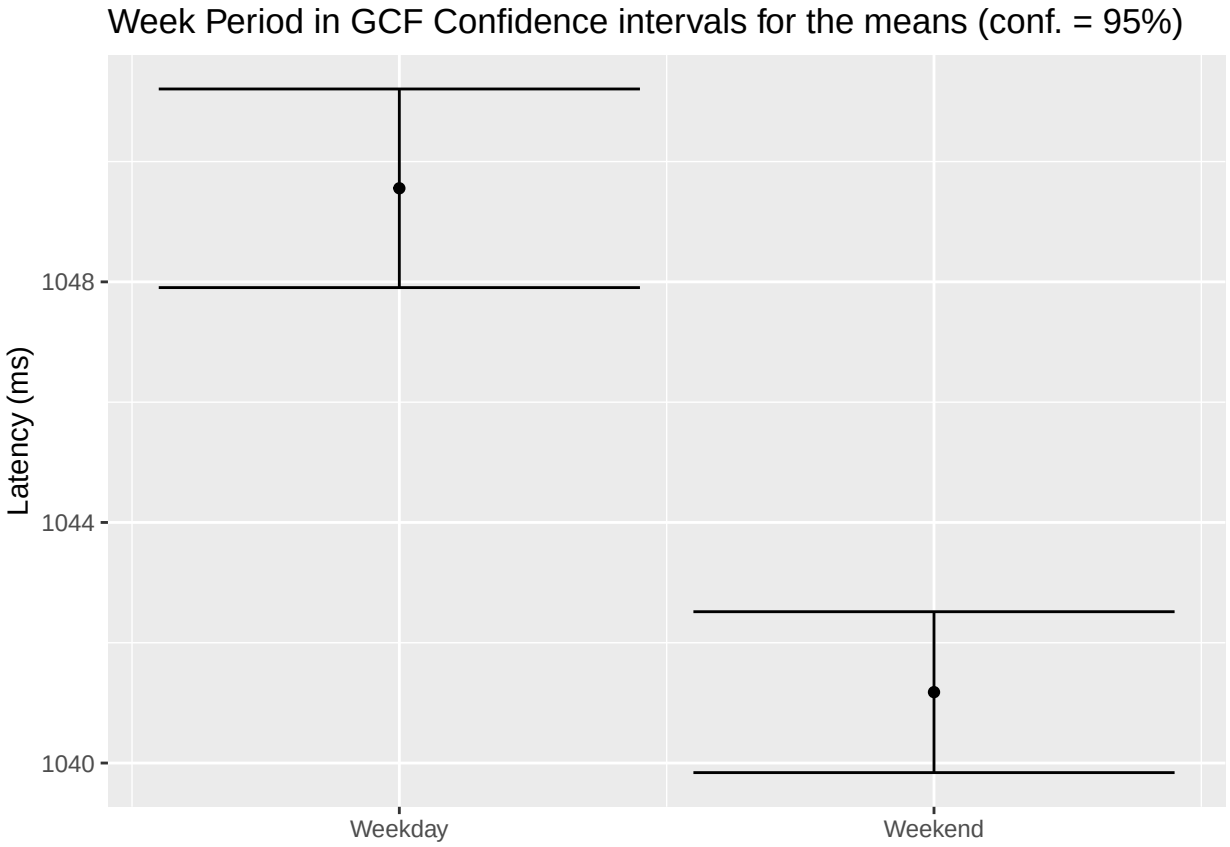

Off hours in GCF
Frequency
0
500
1000
1500
2000
600
1000
1400
1800
Latency (ms)


600
800
1000
1200
1400
1600
1800

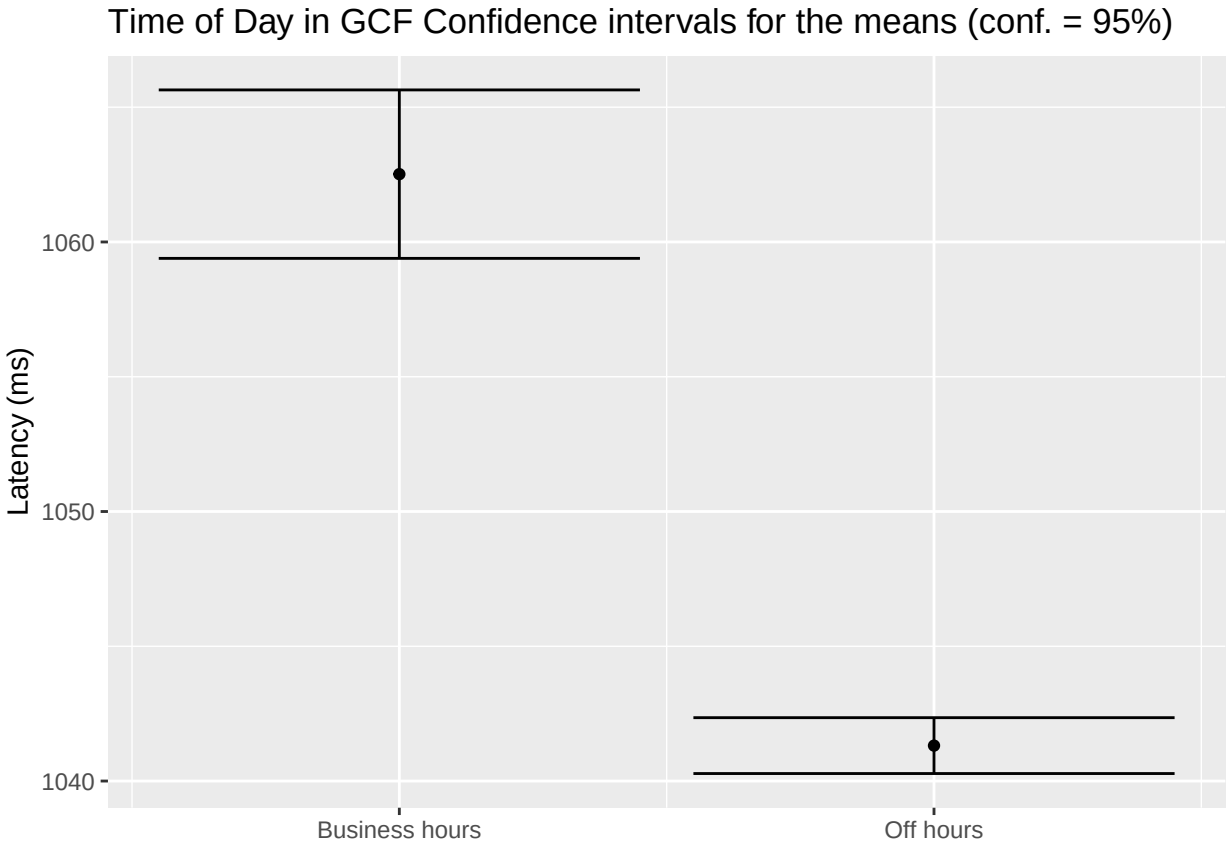

600
800
1000
1200
1400
1600
1800

Confidence intervals in GCF

As we can see from the confidence intervals for the means on week periods, we can confirm there is statistically significant difference between latencies measured during weekdays and weekends, since the confidence intervals have no overlap.



Similarly, the confidence intervals for business and off hours have no overlap, meaning we can confirm there are statistically significant differences between latencies measured during business hours and off hours in GCF



ANOVA in GCF

By looking at the ANOVA results, both time of day is statistically significant for a confidence level of 95%. On the other hand, the period of the week is not statistically significant, even though latencies measured during the week are different from those in weekends.

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## TIME_OF_DAY     1  196774  196774  251.55 <2e-16 ***
## WEEK_PERIOD     1      71      71    0.09  0.764
## Residuals    2849 2228657     782
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

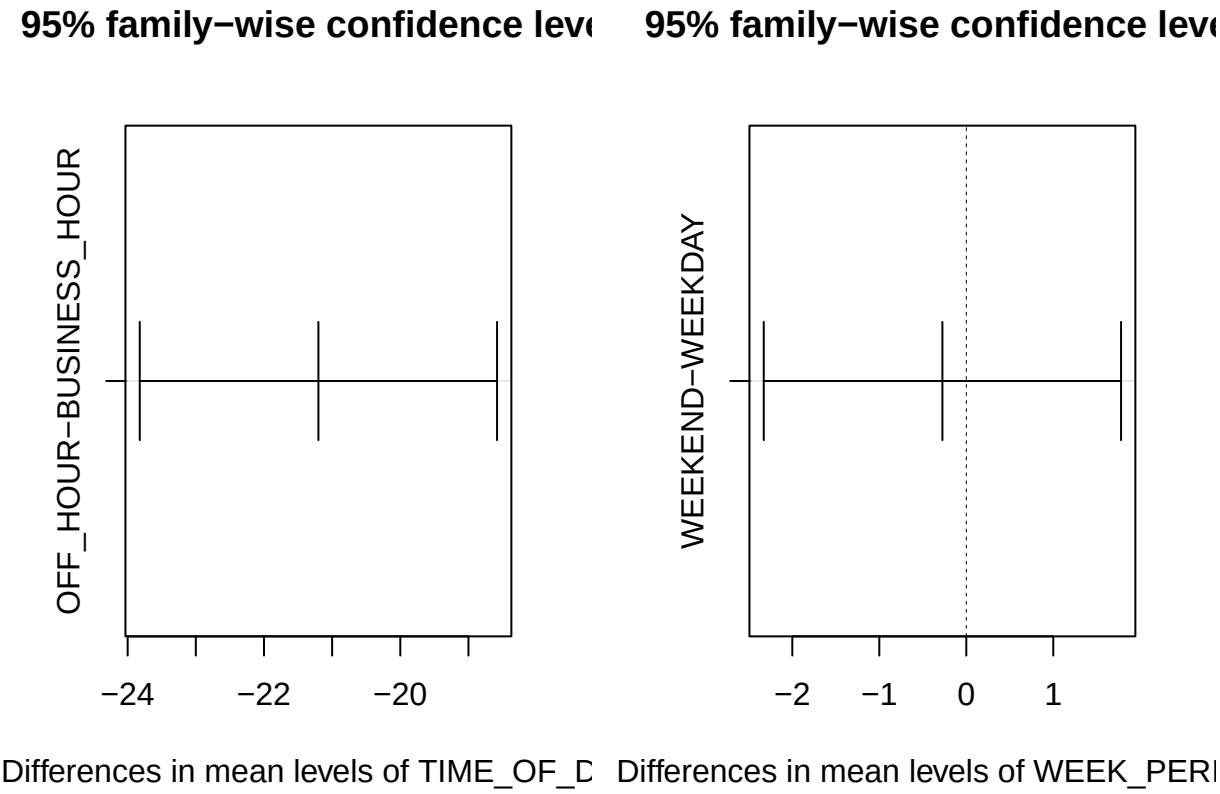
The allocation of variation shows most of the variation is due to random error with small participation from the time of day.

```
##              Sum Sq
## TIME_OF_DAY    8.11
## WEEK_PERIOD    0.00
## Residuals     91.88
```

```
## [1] 3.844725
```

From the Tukey test we can see the confidence intervals for the difference between business hours and off hours is negative, meaning that off hours have more performance. Conversely, the confidence interval for

the difference between weekends and weekdays includes 0, confirming that week period is not significant to overall latency.

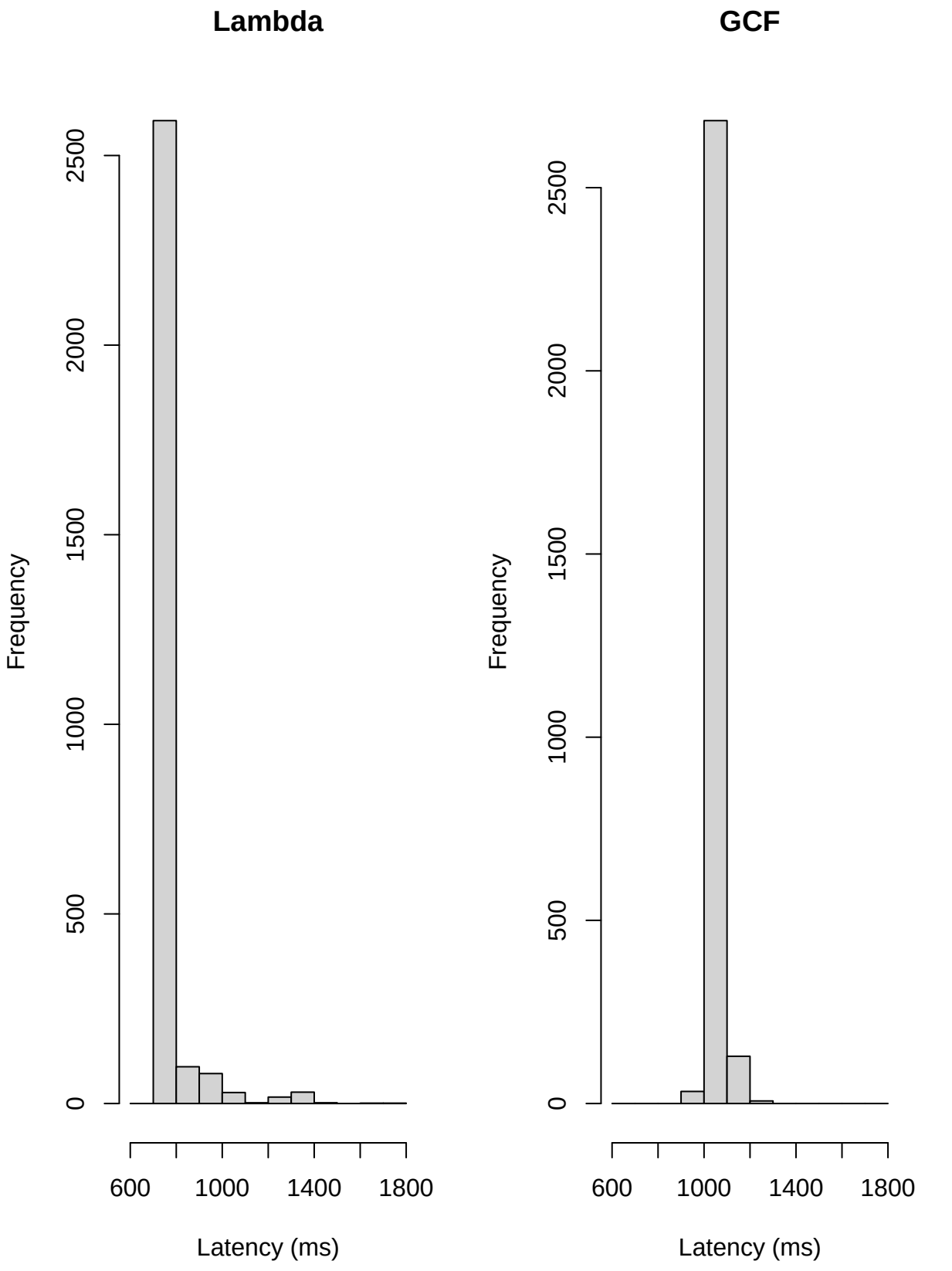


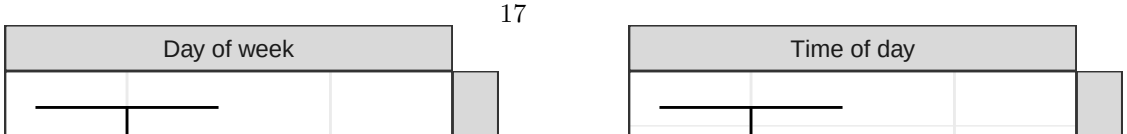
Key takeaways

For Lambda, 1. Latency for weekdays and weekends is different. 2. Latency for business hours and off hours is different, but not a lot. Small advantage to business hours for performance. 3. Both time of day and week periods are significant for overall performance. 4. Performance over the weekends is better.

For GCF, 1. Latency for weekdays and weekends is different. 2. Latency for business hours and off hours is different. 3. Time of day is significant for overall latency. Day of the week is not. 4. Performance during off hours is better.

Figure playground





```r
medians = data.frame(name = c("Time of day", "Day of week","Time of day", "Day of week"), provider = c(
                     diff_median = c(median(data_write_business_lambda$LATENCY_SECONDS) * 100 / median(
medians
```

```
##          name provider diff_median
## 1 Time of day   Lambda    100.0130
## 2 Day of week   Lambda    100.4725
## 3 Time of day      GCF    101.6469
## 4 Day of week      GCF    100.6029
```