DSS Prototype Analysis

Alvin Murphy

Table of contents

1 DSS Installation as a Docker Container

https://github.com/jupyter/docker-stacks
https://hub.docker.com/r/jupyter/r-notebook/tags/

(optional) docker pull jupyter/r-notebook:latest

We want to make the DDS Prototype ~/analysis/ directory linked to the Juypter container. Use the following to mount the analysis directory (i.e. current working directory) as a volume in the Juypter container. Note that the directory needed to be added via the Docker Desktop Dashboard on Mac.

docker run -it –rm -d -p 10000:8888 -v ${PWD}:/home/jovyan/work –name notebook jupyter/r-notebook:latest

To find the token from the container:
docker exec -it notebook jupyter server list

Navigate to the container UI and enter the token: http://localhost:10000

2 DSS System Context

Figure 1 depicts the context for the DSS. The DSS operator interacts with the DSS Prototype for decision assitance. The DSS relies on a aircraft database to gather real-time flight data to review in decision support algorithms.
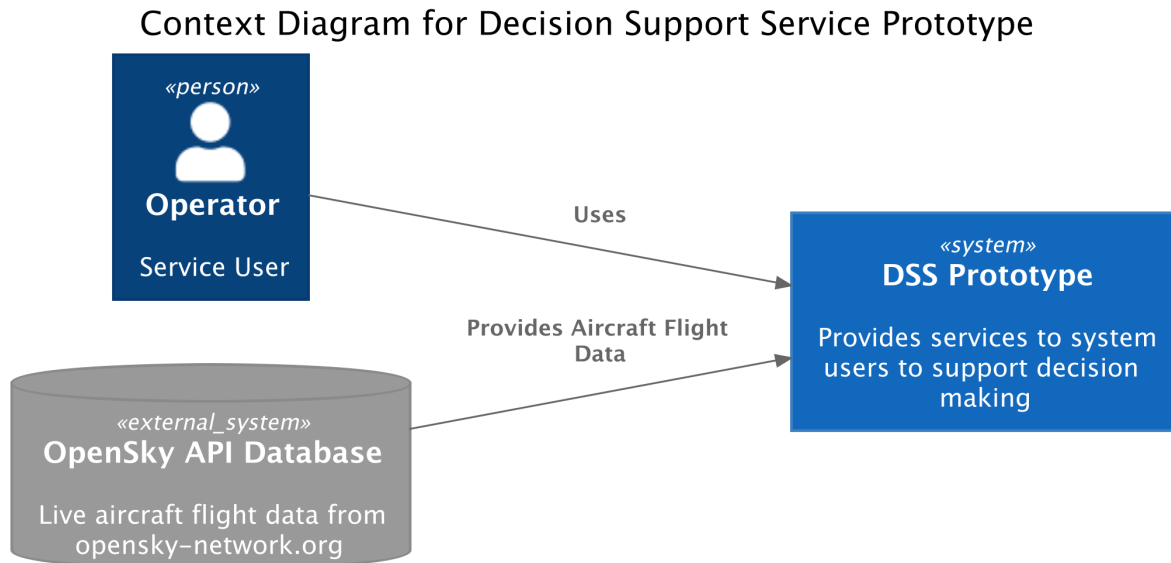
Context Diagram for Decision Support Service Prototype

Figure 1: DSS Context Diagram

2.1 DSS Container Architecture

Nine containers are instantiated as part of the DSS architecuture (see Figure 2). Six provide the DSS implementation while the additional 3 support collection and calculation of metrics. Each application container was designed around the 12-Factor Application "Single Responsibility Principle"; e.g. each app has one purpose to enable rapid insertion of new capabilities with low cohesion to other functionality. At this time, all responses are canned without underlying calculation to focus on meeting the 500 ms hypothesis pryor to burdening the application with calculation latency.

2.1.1 DSS Applications

- opensky-int: Provides the OpenSky API for flight data. The app provides data about aircraft within 60 NM of Richmond (RIC) or Dulles (IAD) airports.
- tm-server: Provides sensor track data (e.g. OpenSky) and system tracks to support DSS services. System tracks represents the system-wide common understand of a track objects state for decisions.
- wa-app: The Weapon Assessment Application determines which weapons are capable to successfully engage a target. The wa-app use the tm-server api to get track data.
- te-app: The Trail Engage Application determines the success rate of an engagement with a specific weapon target pairing. The predicted track kinematic data at engagement time is provided; therefore, the current track kinematics from the tm-server are not queried prior to providing a response.
- test-app: Provides and ability to generate automated test. the test-app uses the dss-ui to call dss-ui endpoint to replicate operator interactions with the DSS Prototype.
- dss-ui: Provides a simple graphical interface to launch DSS services.

2.1.2 DSS Tools

- telem-jaeger: The open source Jaeger containter collects "span" data from the DSS applications. Spans collection duration data for service call to over container; e.g. latency. This the fundamental data that is being analysed here.
- grafana: The open source Grafana container connects to the telem-jaeger container to create visualization dashboard. Also, Grafana faciliates the export of data as a .csv file for analysis.
- notebook: The Jupyter Notebook container support analysis of the data exported by Grafana. It is the core datafile use by this tool.
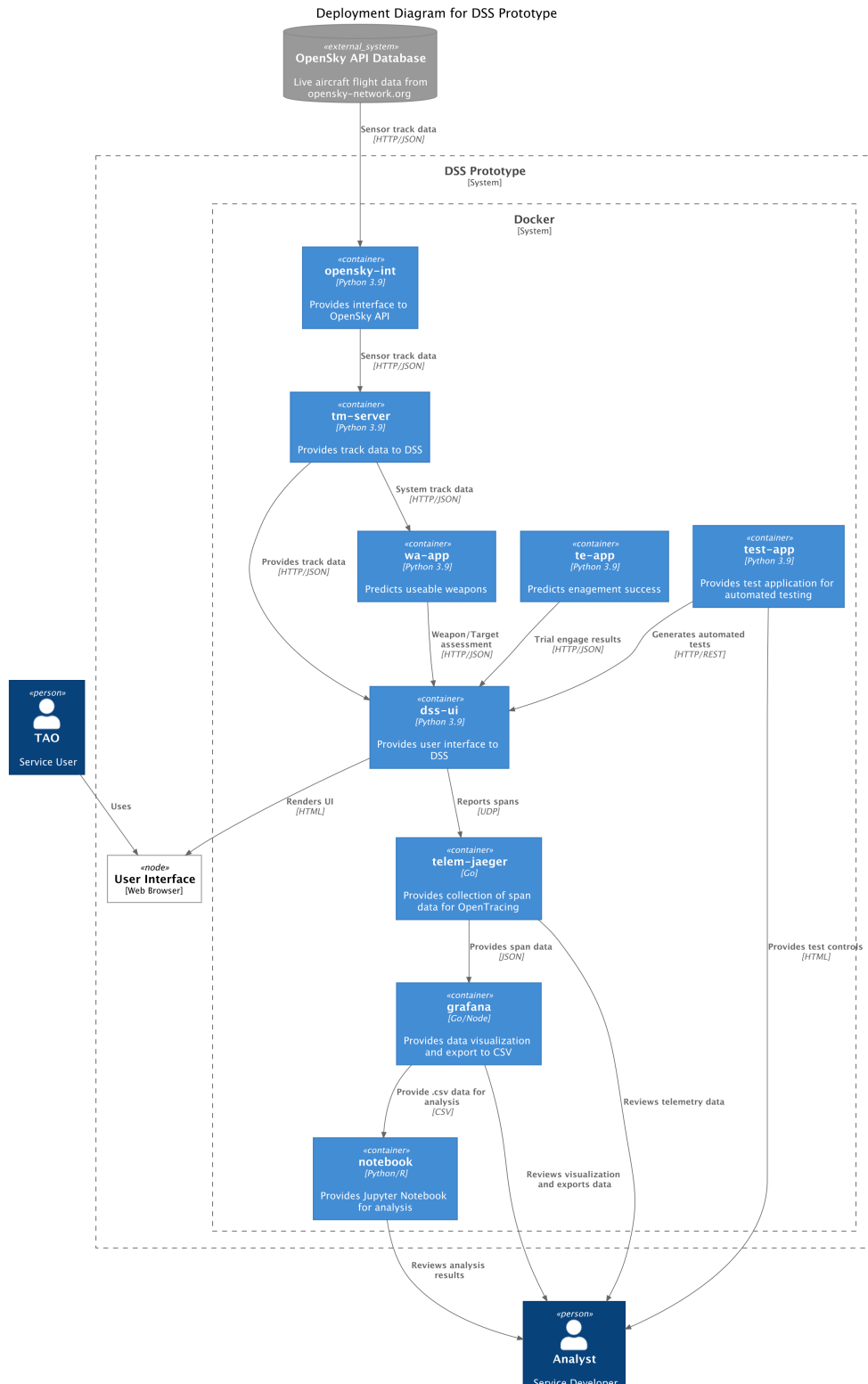
Deployment Diagram for DSS Prototype

«external_system»
OpenSky API Database

Live aircraft flight data from
opensky-network.org

Sensor track data
[HTTP/JSON]

DSS Prototype
[System]

Docker
[System]

«container»
opensky-int
[Python 3.9]

Provides interface to
OpenSky API

Sensor track data
[HTTP/JSON]

«container»
tm-server
[Python 3.9]

Provides track data to DSS

System track data
[HTTP/JSON]

Provides track data
[HTTP/JSON]

«container»
wa-app
[Python 3.9]

Predicts useable weapons

«container»
te-app
[Python 3.9]

Predicts enagement success

«container»
test-app
[Python 3.9]

Provides test application for
automated testing

Weapon/Target
assessment
[HTTP/JSON]

Trial engage results
[HTTP/JSON]

Generates automated
tests
[HTTP/REST]

«person»
TAO

Service User

«container»
dss-ui
[Python 3.9]

Provides user interface to
DSS

Uses

Renders UI
[HTML]

Reports spans
[UDP]

«node»
User Interface
[Web Browser]

Provides test controls
[HTML]

«container»
telem-jaeger
[Go]

Provides collection of span
data for OpenTracing

Provides span data
[JSON]

«container»
grafana
[Go/Node]

Provides data visualization
and export to CSV

Reviews telemetry data

Provide .csv data for
analysis
[CSV]

Reviews visualization
and exports data

«container»
notebook
[Python/R]

Provides Jupyter Notebook
for analysis

Reviews analysis
results

«person»
Analyst

Service Developer

Figure 2: DSS Deployment Diagram

5

2.2 Hypothesis

Hypotheses are "innocent until proven guilty." We'll assume that SpaceX and others have proven that DevSecOps tech can meet hard-real-time requirements but nothing available in the body of knowledge documents this.

Hypothesis: Modern DevSecOps architectures can be designed to meet hard-real-time latency (μ) requirements using modern computing environments and computing infrastructure.

$H_0 : \mu \leq 500ms$ with jitter within latency bounds
$H_a : \mu > 500ms$ with jitter exceeding latency bounds

Murphy, Alvin C. and Moreland Jr, James D. 'Integrating AI Microservices into Hard-Real-Time SoS to Ensure Trustworthiness of Digital Enterprise Using Mission Engineering'. 1 Jan. 2021 : 38 − 54.

3 Exploratory Data Analysis

```
  Trace.ID              Trace.name            Start.time            Duration
Length:100            Length:100            Length:100            Length:100
Class :character      Class :character      Class :character      Class :character
Mode  :character      Mode  :character      Mode  :character      Mode  :character
```

A data.frame: 6 × 2

	Trace.ID <chr>	Trace.name <chr>
1	9ee3577fb1b427bc4fc17fecc5154d7d	dss-prototype: /TE
2	f05ddc4dc13aff5c3098011b2a402401	dss-prototype: /tracks
3	2bd901fbbfc9ee8dfa7c9629d93a1567	dss-prototype: /IAD
4	69a48381a14e79da08aaa2353f7db4b2	dss-prototype: /RIC
5	e83037dcb9438c04dc12fba373b5502f	dss-prototype: /WA
6	7e381cd880adb670bb9627ca47020938	dss-prototype: /TE

A data.frame: 6 × 2

	Start.time <chr>	Duration <chr>
1	2022-05-02 10:25:01.366	36.0 ms
2	2022-05-02 10:25:00.309	43.3 ms
3	2022-05-02 10:24:58.818	464 ms
4	2022-05-02 10:24:57.307	494 ms
5	2022-05-02 10:24:56.128	139 ms

	Start.time <chr>	Duration <chr>
6	2022-05-02 10:24:55.081	30.3 ms

3.1 Convert Data into Useable Metrics

To make the data more usable and easier to understand well apply conversions from text to numeric and add additional columns with supporting information. A **useCase** column is added to identify specific request use cases; e.g. Get Dulles Airport Data. The data also indicates whether the request is managed internally or a connection to an external service is required to provided a response (i.e., https://opensky-network.org). A **numContainers** column is added to indicate the number of containers involved in providing a use case response (e.g. independent variable). An **extNetworkHops** column is added to include network hops for external request as an additional independent variable.

```
   Trace.ID            Trace.name             Start.time             Duration
 Length:100          Length:100          Min.   :1.651e+09    Min.    :0.01390
 Class :character    Class :character    1st Qu.:1.651e+09    1st Qu.:0.03275
 Mode  :character    Mode  :character    Median :1.651e+09    Median :0.07375
                                         Mean   :1.651e+09    Mean    :0.25404
                                         3rd Qu.:1.651e+09    3rd Qu.:0.48450
                                         Max.   :1.651e+09    Max.    :2.00000
   useCase            numContainers extNetworkHops
 Length:100          Min.   :2.0    Min.   : 0.0
 Class :character    1st Qu.:2.0    1st Qu.: 0.0
 Mode  :character    Median :3.0    Median : 0.0
                     Mean   :2.6    Mean   : 5.6
                     3rd Qu.:3.0    3rd Qu.:14.0
                     Max.   :3.0    Max.   :14.0
```

A data.frame: 6 × 3

	Trace.ID <chr>	Trace.name <chr>	Start.time <dbl>
1	9ee3	dss-prototype: /TE	1651487101
2	f05d	dss-prototype: /tracks	1651487100
3	2bd9	dss-prototype: /IAD	1651487098
4	69a4	dss-prototype: /RIC	1651487097
5	e830	dss-prototype: /WA	1651487096
6	7e38	dss-prototype: /TE	1651487095

A data.frame: 6 × 4

	Duration \<dbl>	useCase \<chr>	numContainers \<dbl>	extNetworkHops \<dbl>
1	0.0360	Trial Engage (Internal)	2	0
2	0.0433	Get Stored Local DSS Tracks (Internal)	2	0
3	0.4640	Get Dulles Airport Data (External)	3	14
4	0.4940	Get Richmond Airport Data (External)	3	14
5	0.1390	Assess Weapons (Internal)	3	0
6	0.0303	Trial Engage (Internal)	2	0

3.2 Exploratory Analysis Plots

Duration of Endpoint Responses from 'Trace'
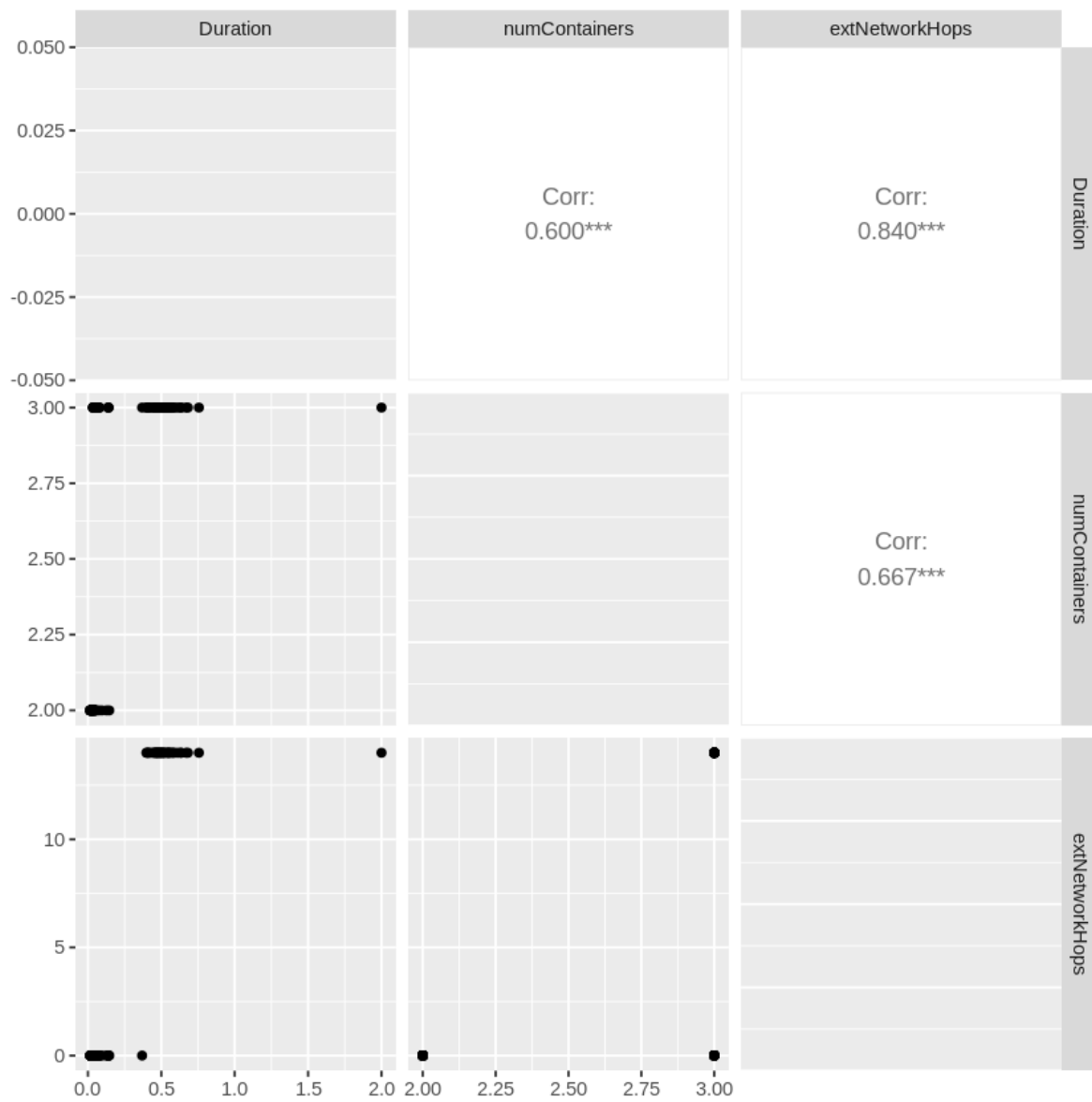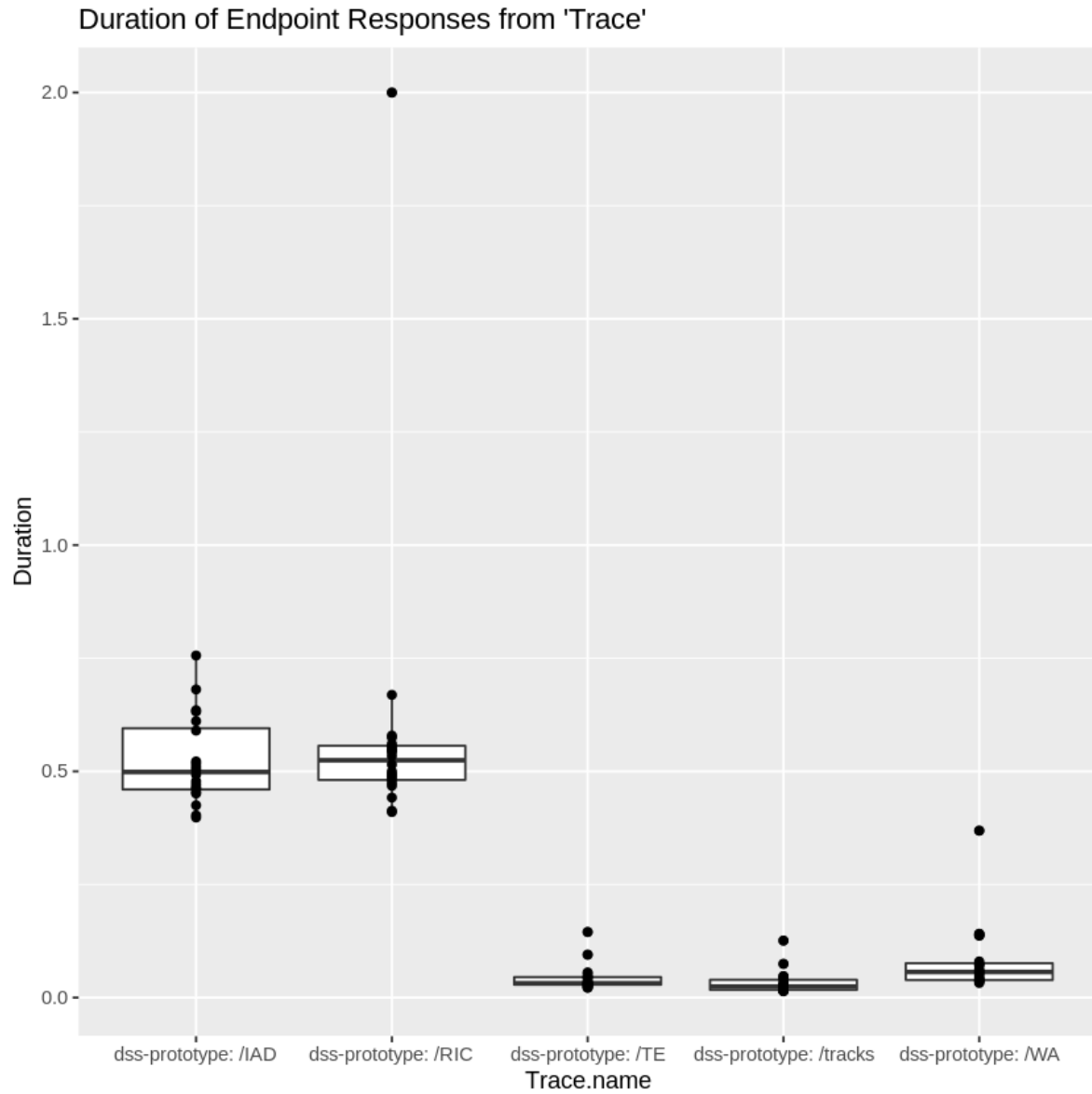
Histogram and Frequency of Durations

3.3 Q-Q Normality Test

Span Duration Q-Q Norm Plot
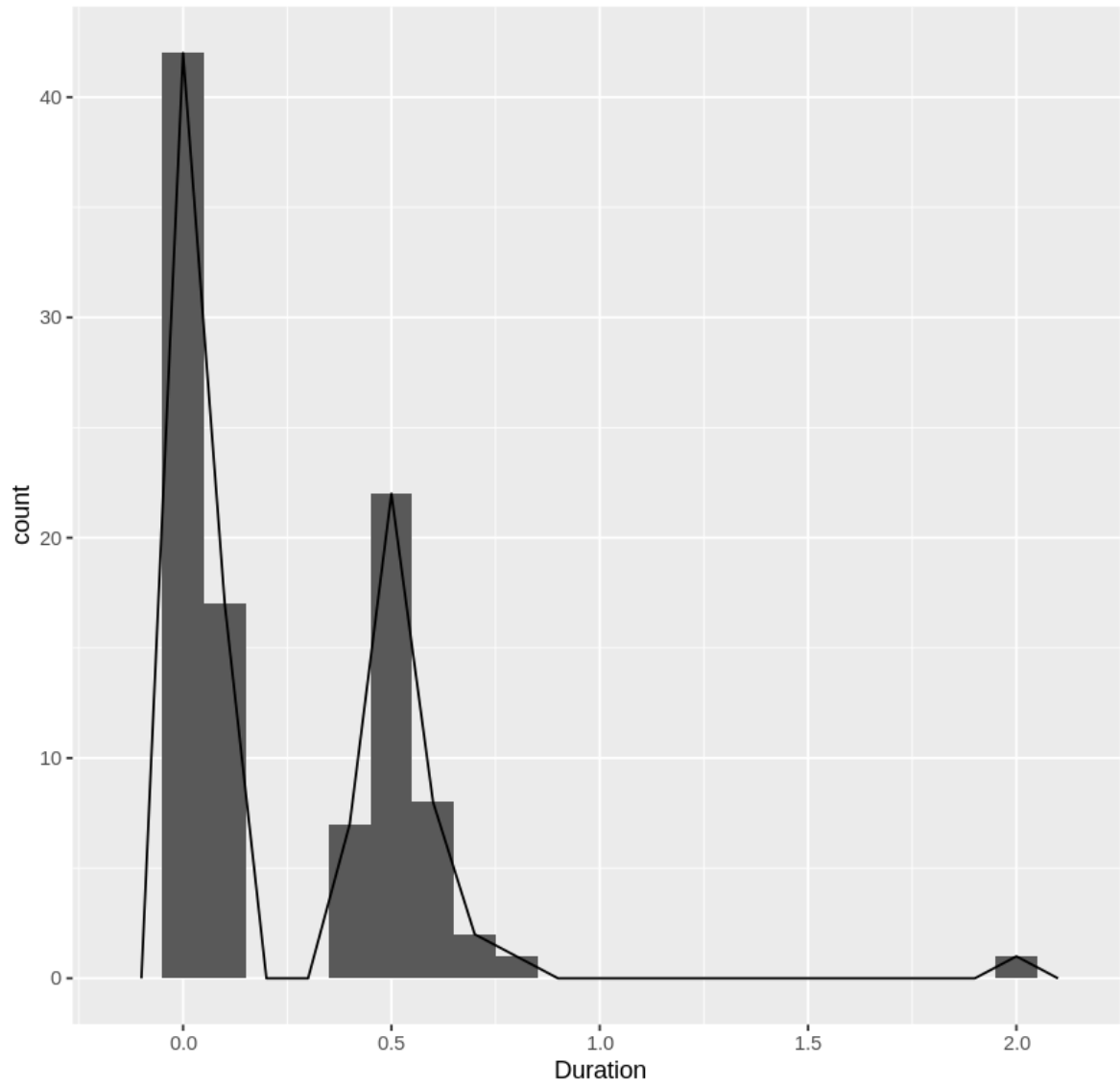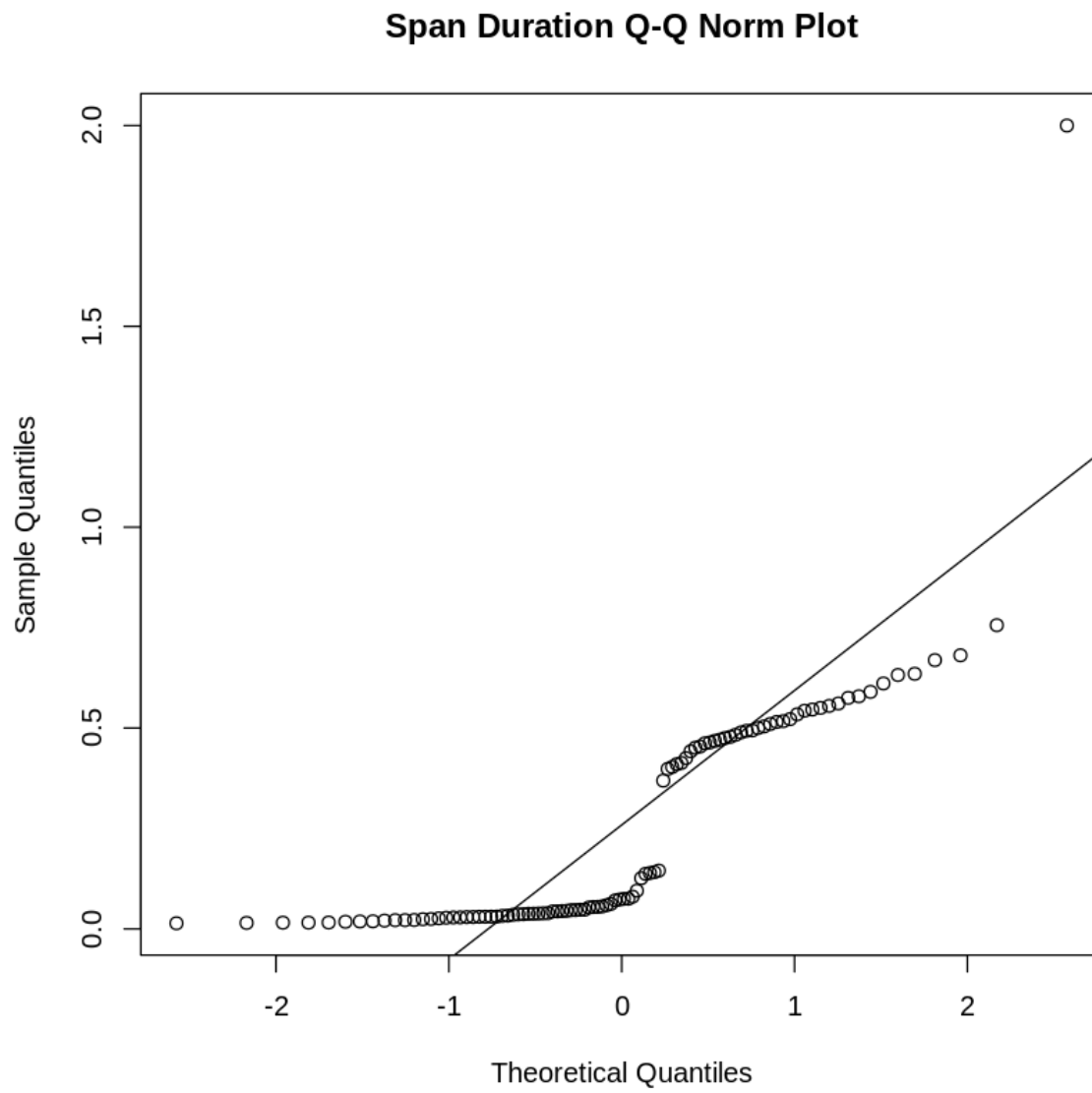
A transformation is needed to apply statistical analysis.

4 Clean the Data

4.1 Search for Outliers

2

Histogram of cleanMetrics$Duratio
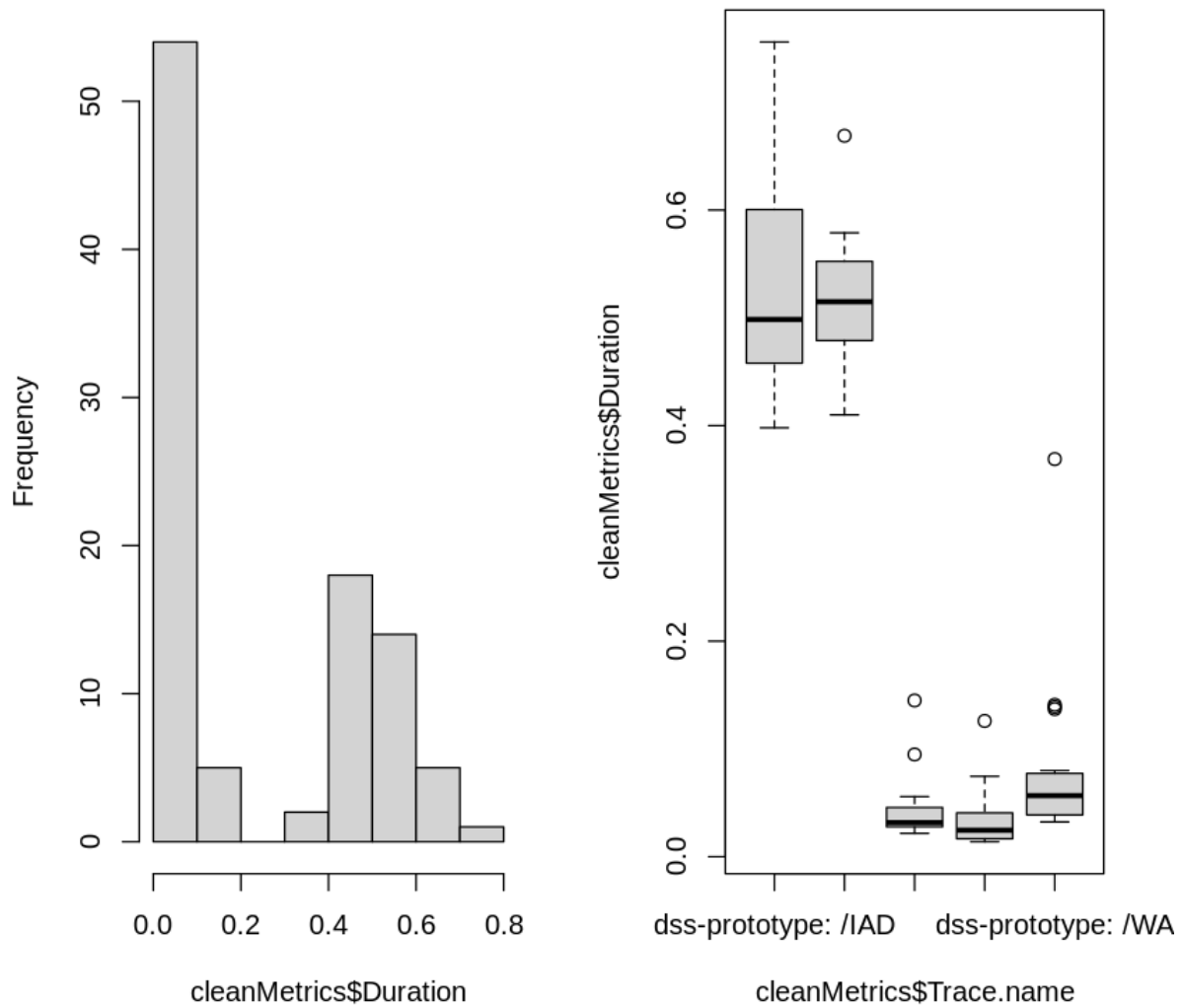
Duration of Endpoint Responses from 'useCase'
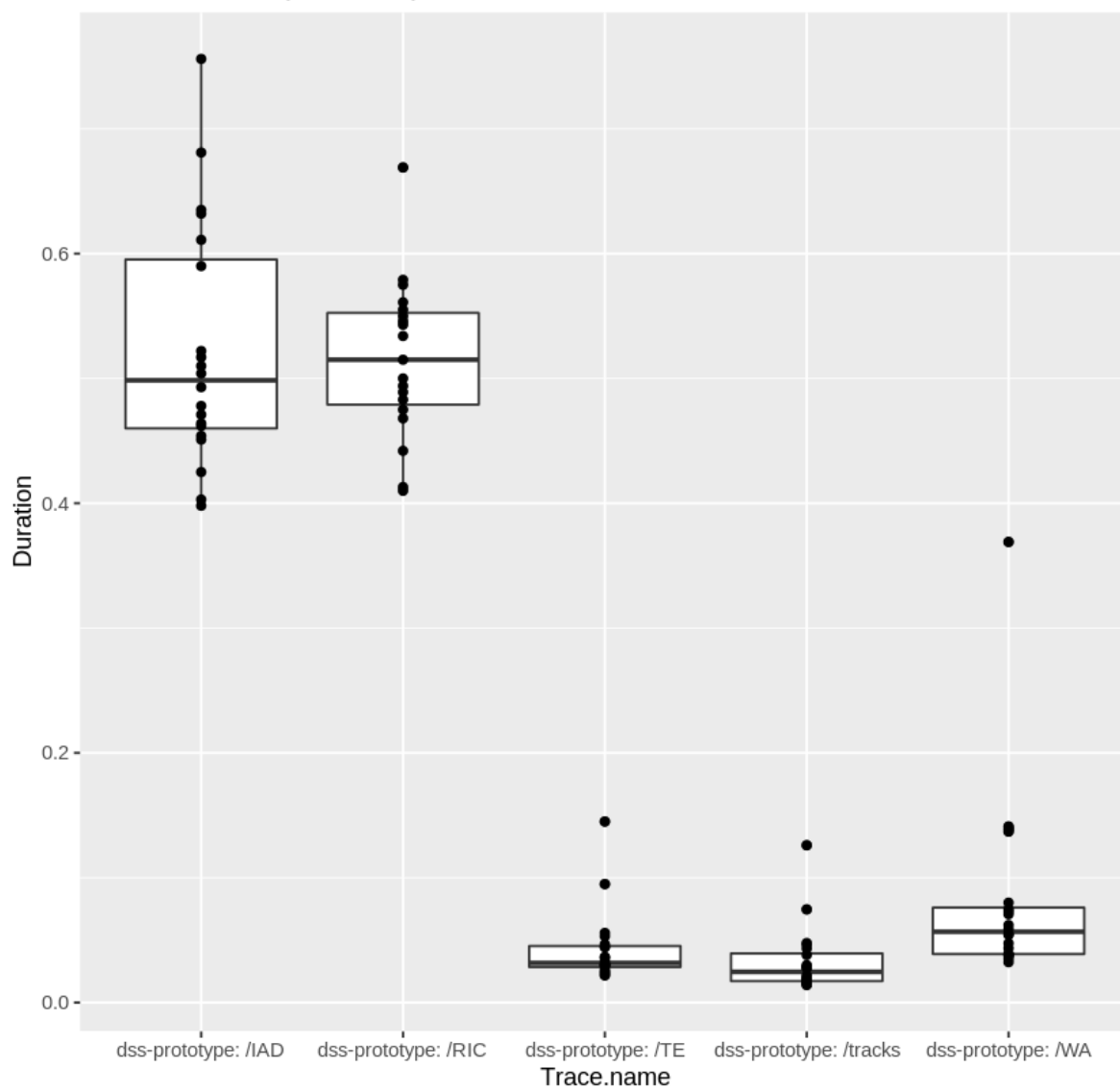
4.2 Transformation of Clean Metrics

4.2.1 Sqrt, Log, and Cube Transformations
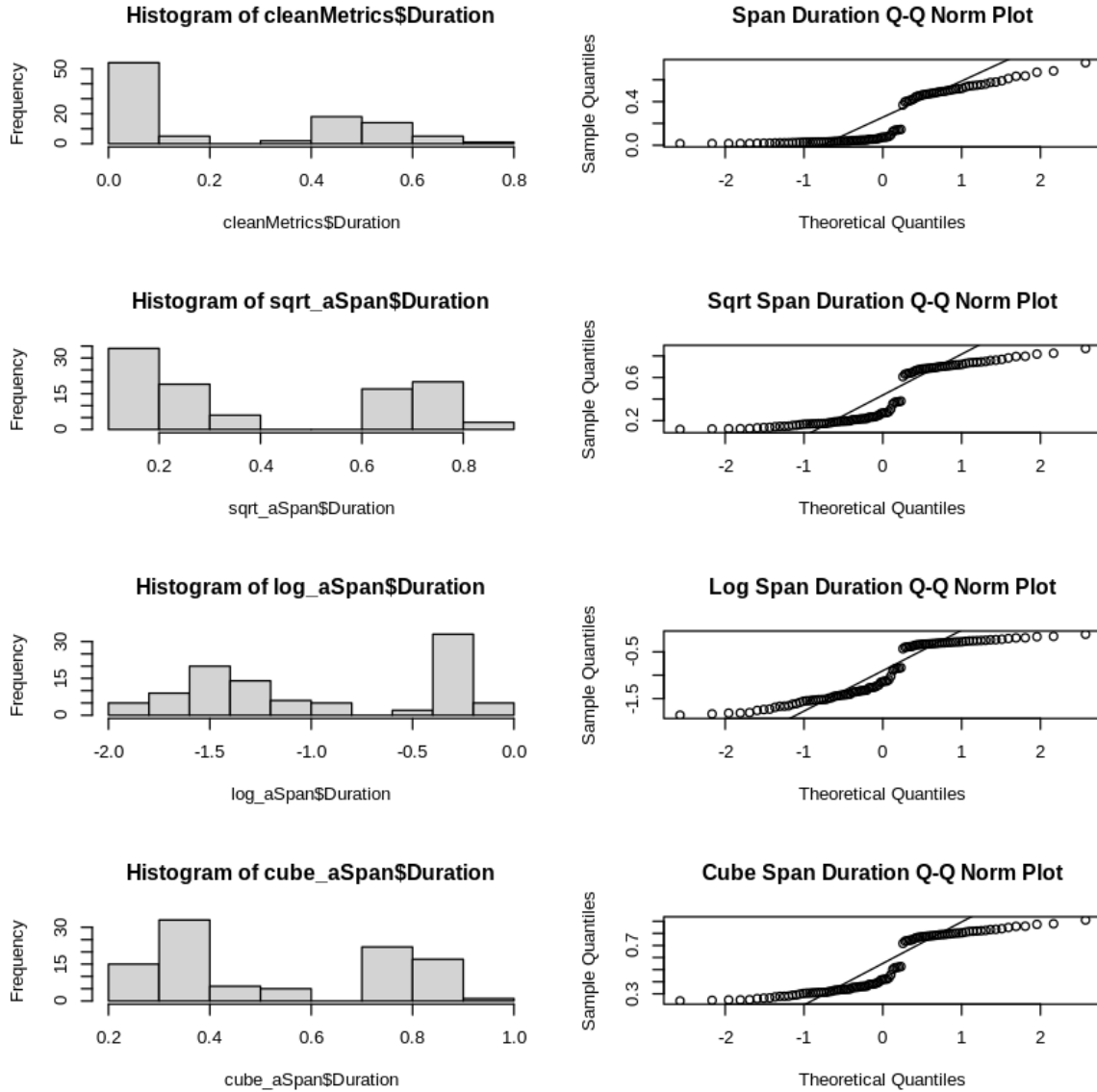
None of these transformation yield distributions that would be considered normal. Most likely due to access to external and internal services with differing latency. Let try another transformation.

4.2.2 Box-Cox Transformation

Box and Cox (1964) developed a family of transformations designed to reduce nonnormality of the errors in a linear model. Applying this transform often reduces non-linearity as well, and heteroscedascity.

The idea is to transform the response variable Y to a replacement response variable $Y_i^{(\lambda)}$, leaving the right-hand side of the regression model unchanged, so that the regression residuals become normally-distributed. Note that the regression coefficients will also change, because the response variable has changed; therefore, the regression coefficients must be interpreted with respect to the transformed variable. Also, any predictions made with the model have to be back-transformed, to be interpreted in the original units.

The standard (simple) Box-Cox transform is:

$$Y_i^{(\lambda)} = \begin{cases} \frac{Y_i^\lambda - 1}{\lambda}, & (\lambda \neq 0) \\ log(Y_i), & (\lambda = 0) \end{cases}$$

Box, G. E. P., & Cox, D. R. (1964). An Analysis of Transformations. Journal of the Royal Statistical Society, Series B (Metholodogical), 26(2), 211-252.

http://www.css.cornell.edu/faculty/dgr2/_static/files/R_html/Transformations.html

Histogram of cleanMetrics$Duratio   **Histogram of bcData$Duration**
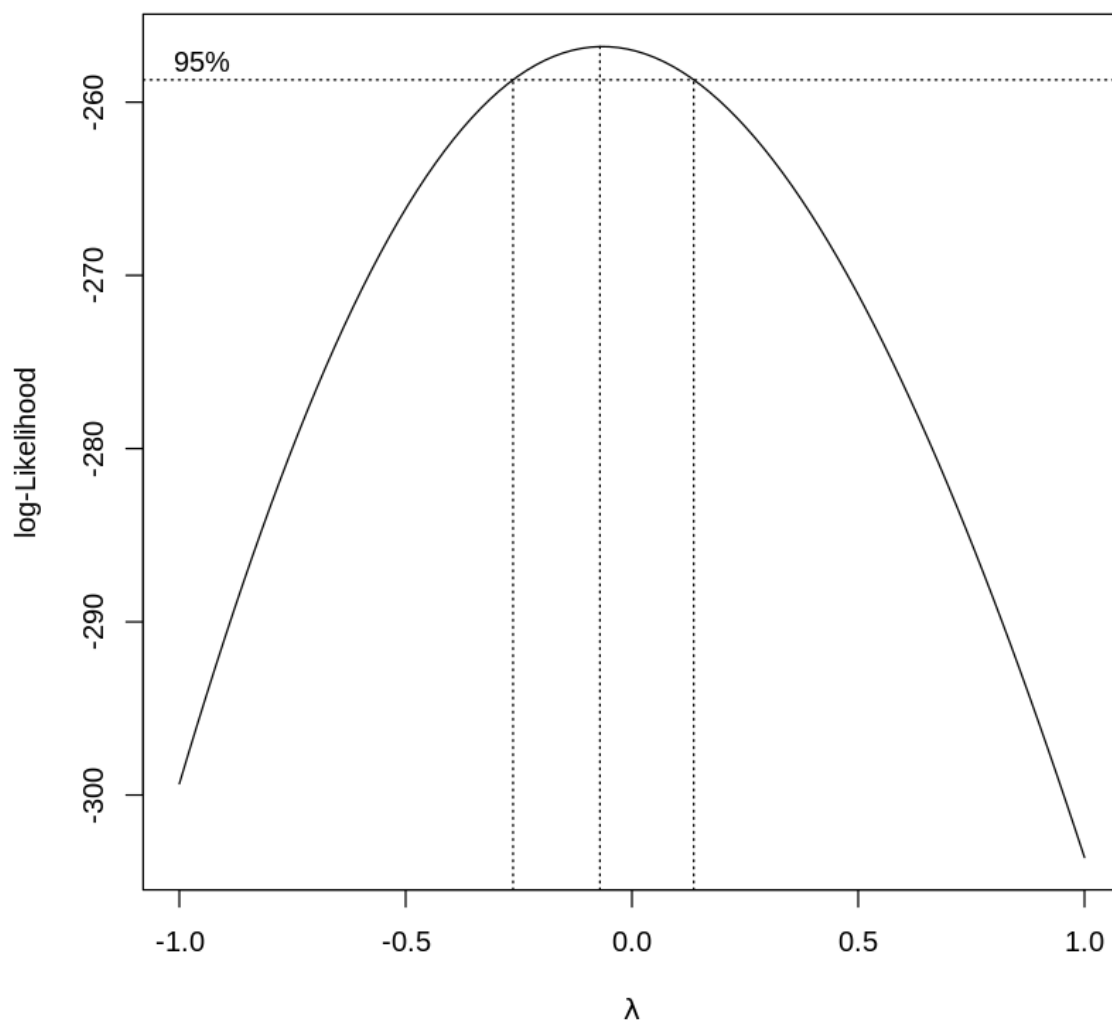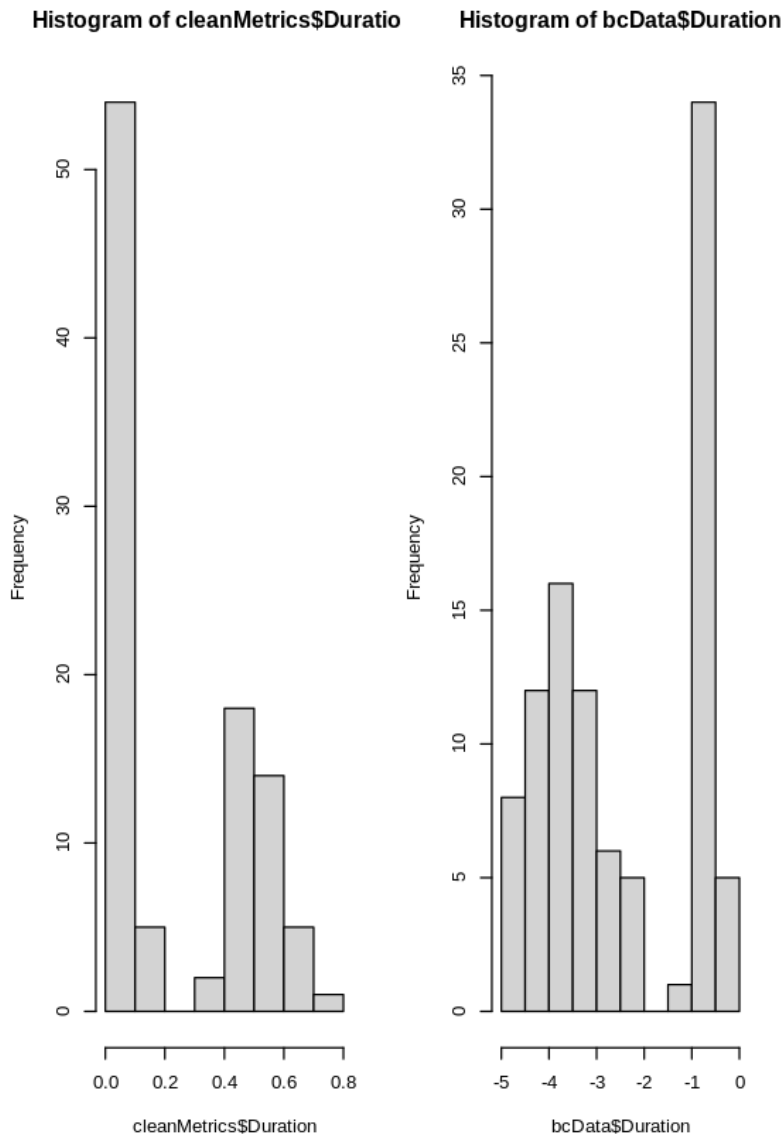
4.3 Normality Testing of the Trasformation

4.3.1 Shapiro-Wilk

The null-hypothesis of this test is that the population is normally distributed. Thus, if the p value is less than the chosen alpha level, then the null hypothesis is rejected and there is evidence that the data tested are not normally distributed. On the other hand, if the p value is greater than the chosen alpha level, then the null hypothesis (that the data came from a

normally distributed population) can not be rejected (e.g., for an alpha level of .05, a data set with a p value of less than .05 rejects the null hypothesis that the data are from a normally distributed population).

https://en.wikipedia.org/wiki/Shapiro–Wilk_test

```
    Shapiro-Wilk normality test

data:  bcData$Duration
W = 0.85873, p-value = 2.852e-08
```

With p-value of 2.852e-08 < 0.05 we reject the null hypothesis that the data are from a normally distributed population. But we'll also do a Q-Q Norm plot to visually see the results.

"if the p value is greater than the chosen alpha level, then the null hypothesis (that the data came from a normally distributed population) can not be rejected"

4.3.2 Q-Q Norm

Q-Q Plot of Duration

Q-Q Plot of Duration
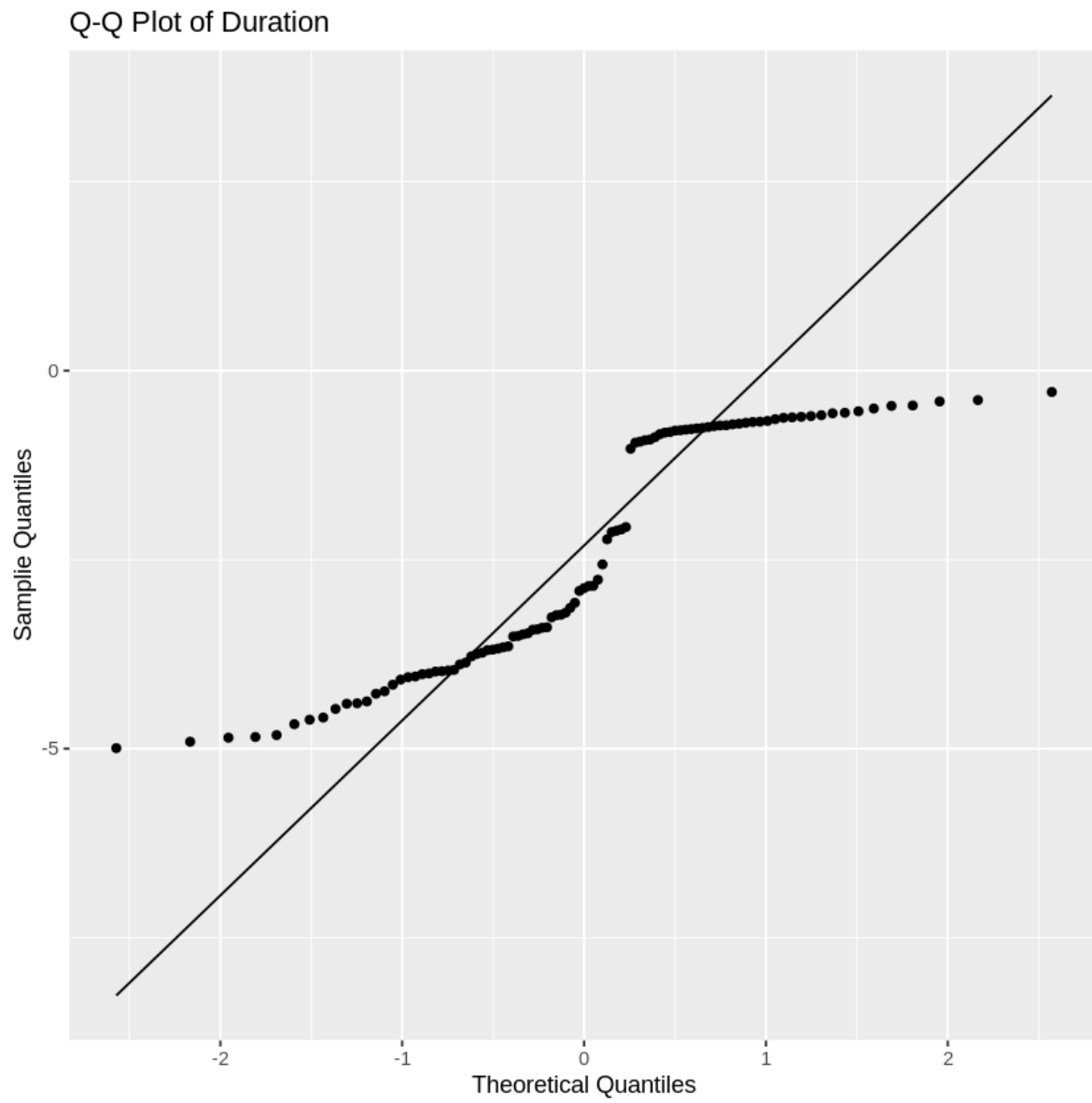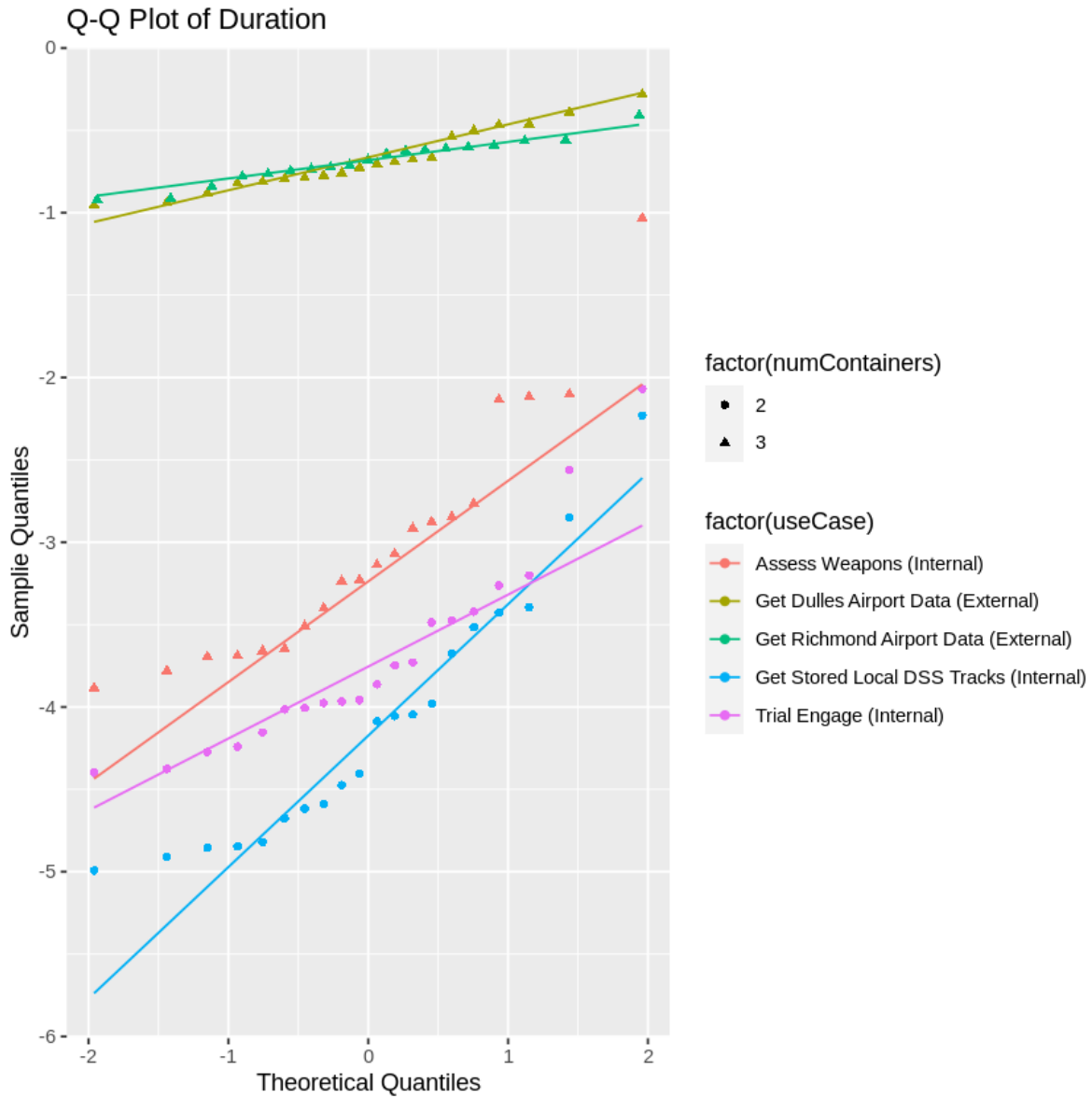
Our assumption here is that the separation of **Sample Quantiles** is from the difference between internal and external span durations (e.g. latency). Let's see what happens when we split the samples.
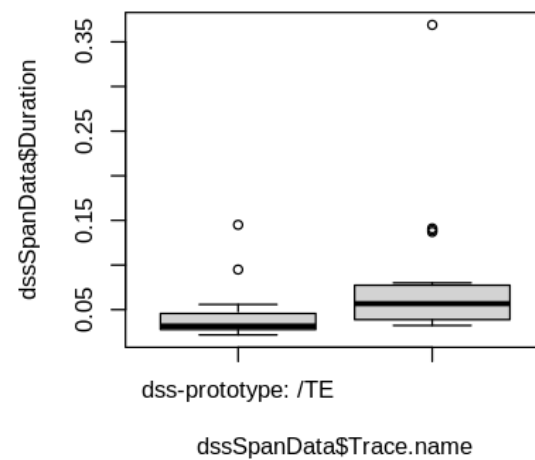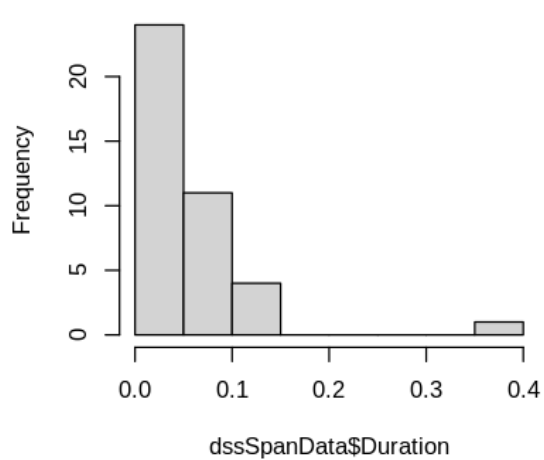
5 Separating "Original" Internal from External Data

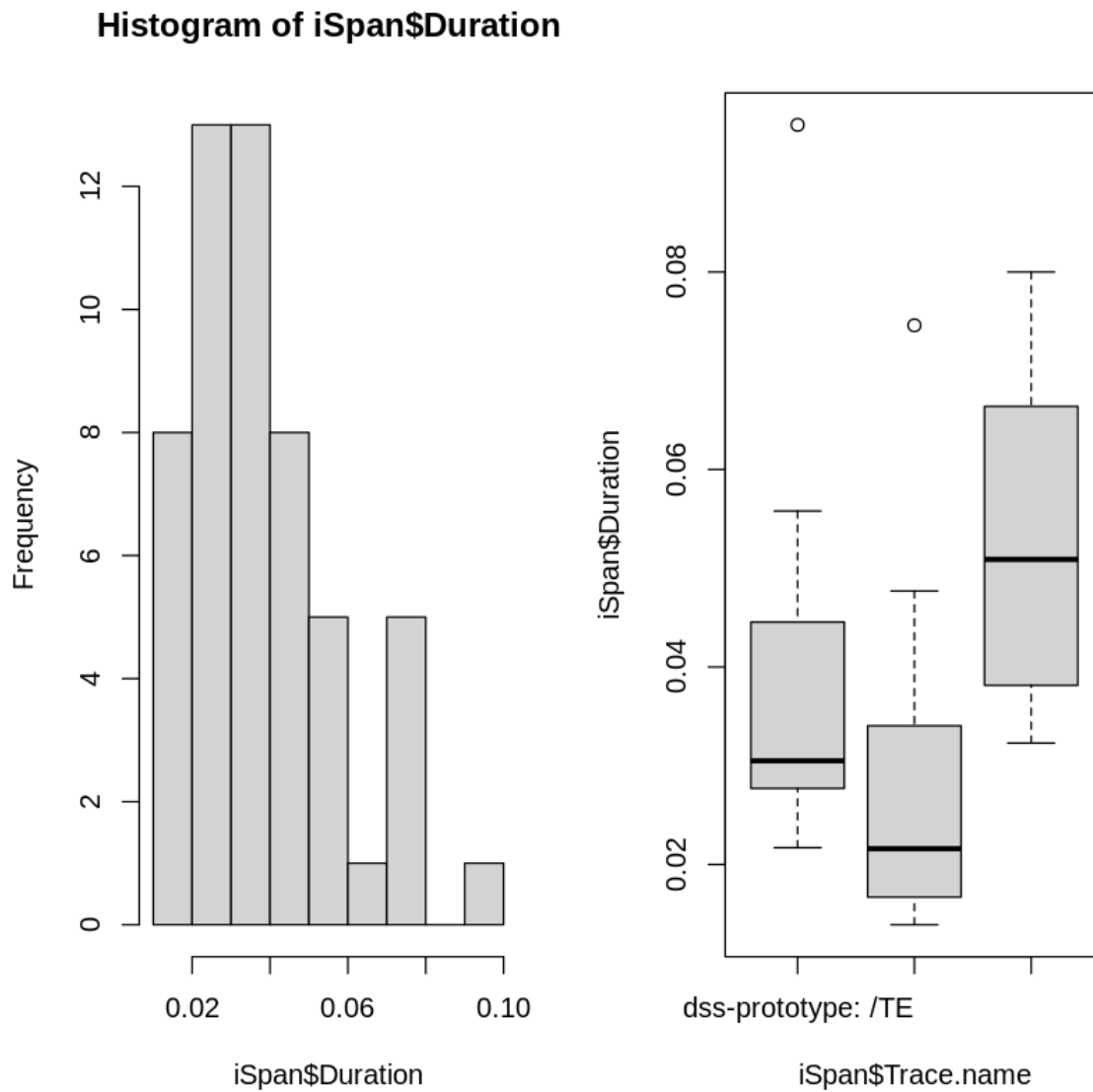## 5.1 Internal Data

Histogram of internalSpanData$Duration
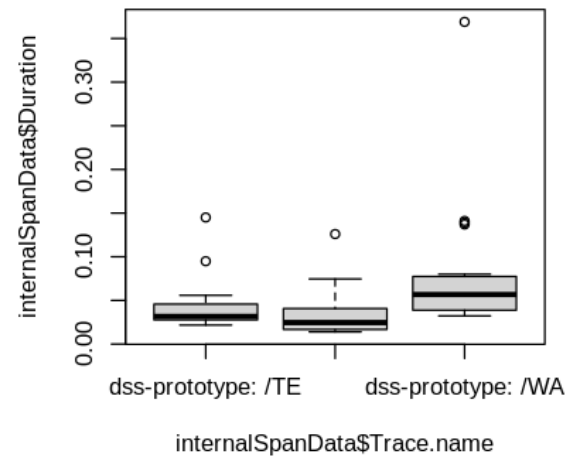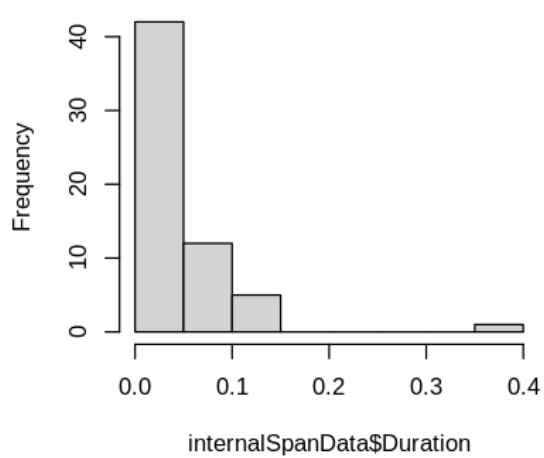
Histogram of dssSpanData$Duration

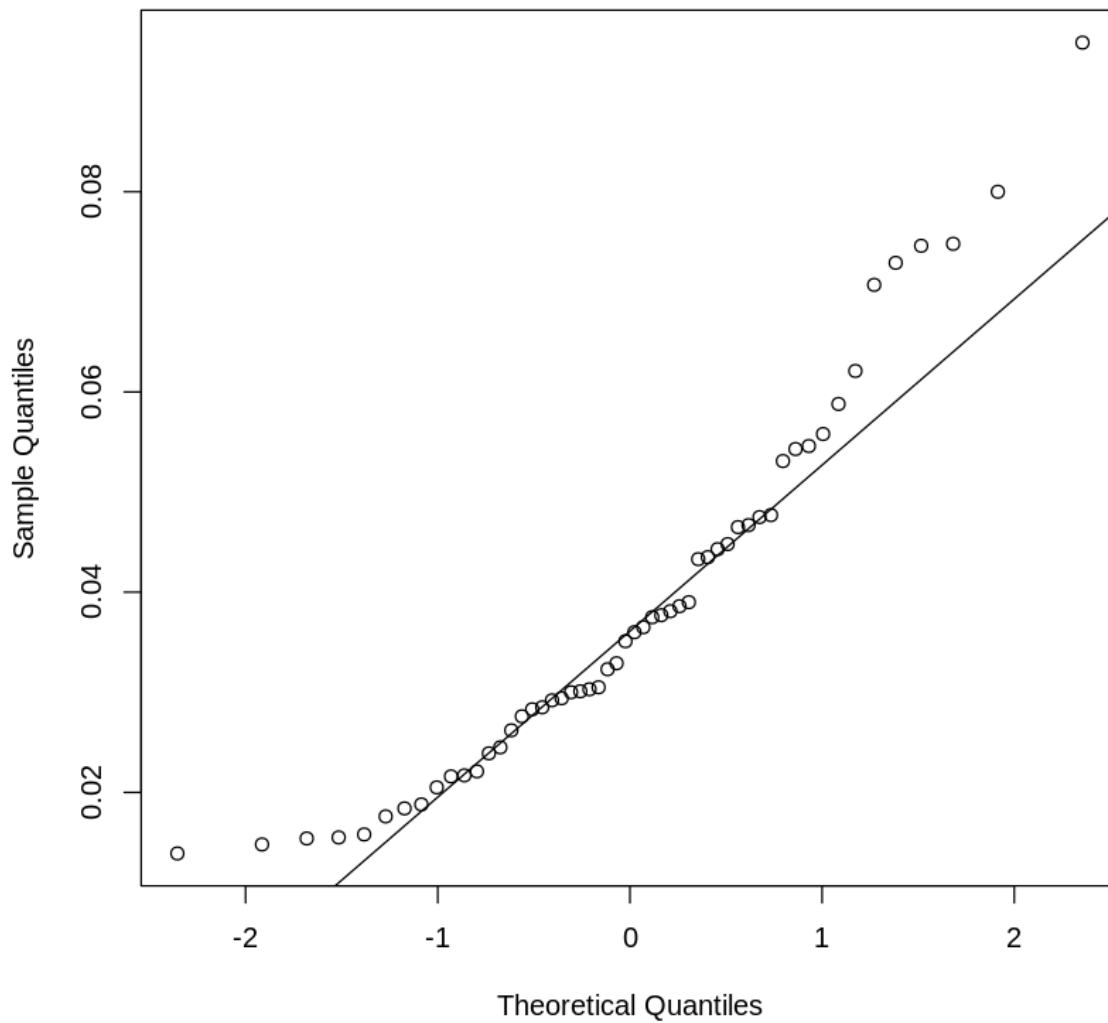This result looks much better. However, we'll remove internal span outliers.

1. 0.126
2. 0.145
3. 0.139

4. 0.369
5. 0.137
6. 0.141

Histogram of iSpan$Duration

5.1.1 Q-Q Norm Plot of "Clean" Internal Span Data

We'll look a the Q-Q Norm Plot and Shapiro-Wilk Test

Internal Span Duration Q-Q Norm Plot

5.1.2 Autocorrelation

Autocorrelation plots are a commonly-used tool for checking randomness in a data set. This randomness is ascertained by computing autocorrelations for data values at varying time lags. If random, such autocorrelations should be near zero for any and all time-lag separations. If non-random, then one or more of the autocorrelations will be significantly non-zero.

ACF of Internal Span Duration
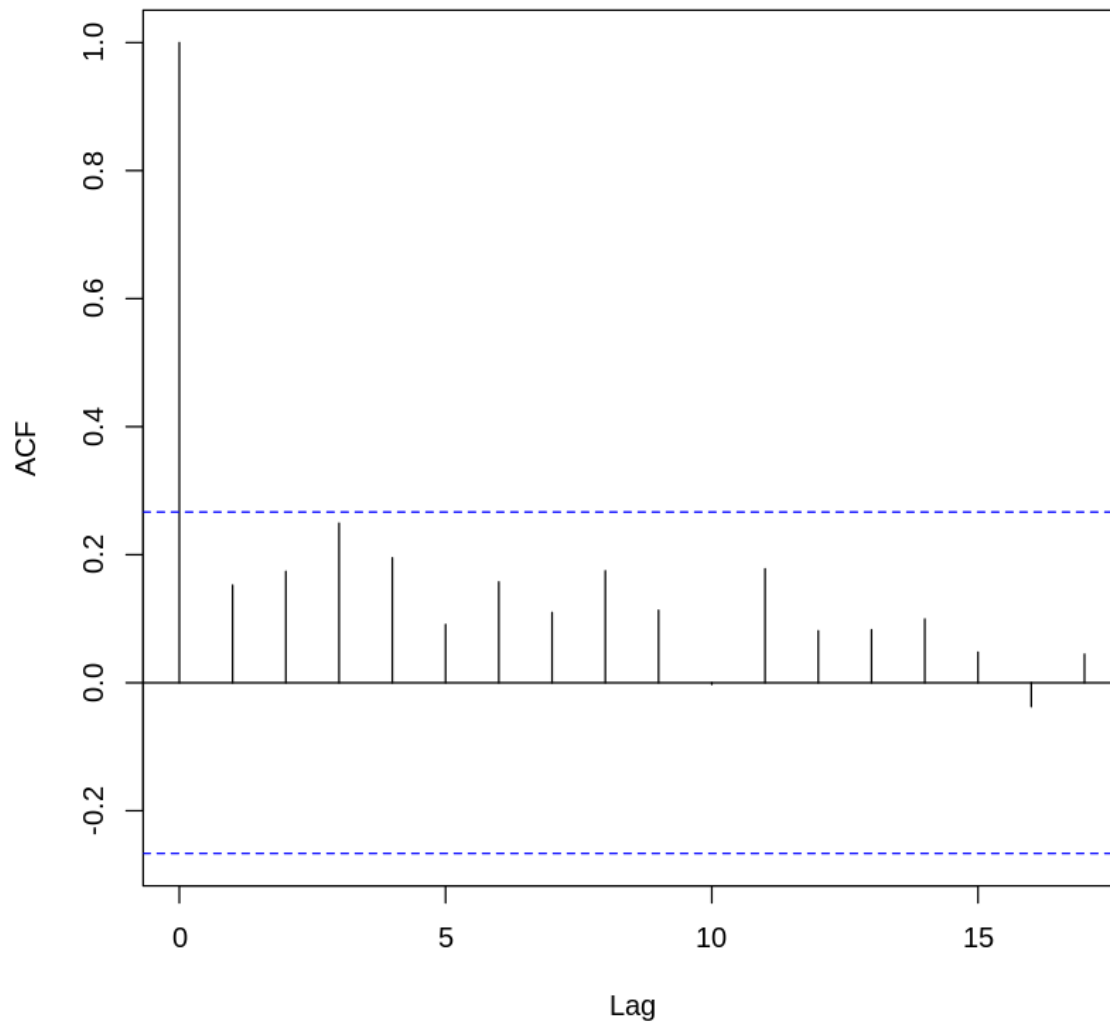
5.1.3 Shapiro-Wilk Normality Test

```
    Shapiro-Wilk normality test

data:  iSpan$Duration
W = 0.92499, p-value = 0.002321
```
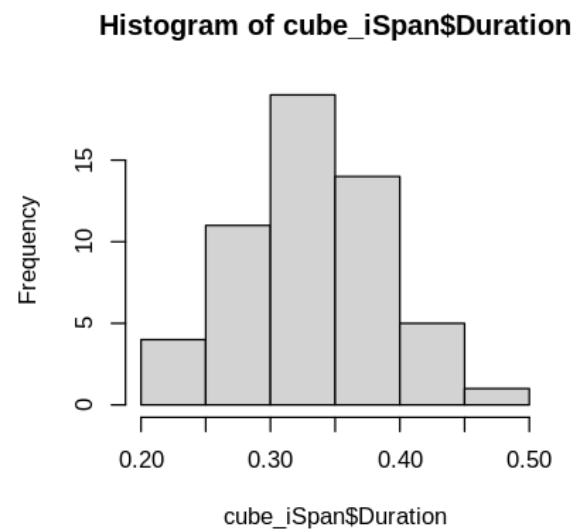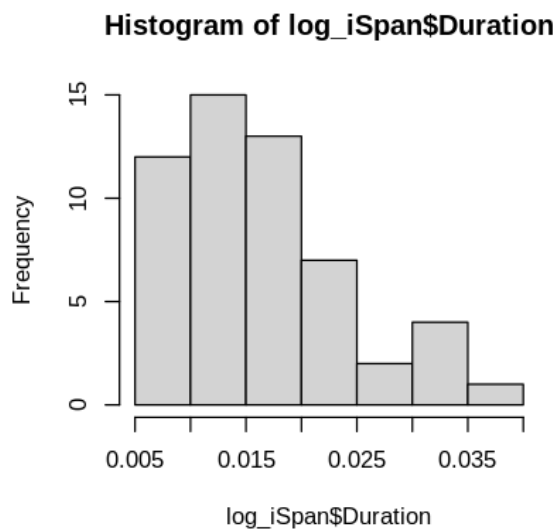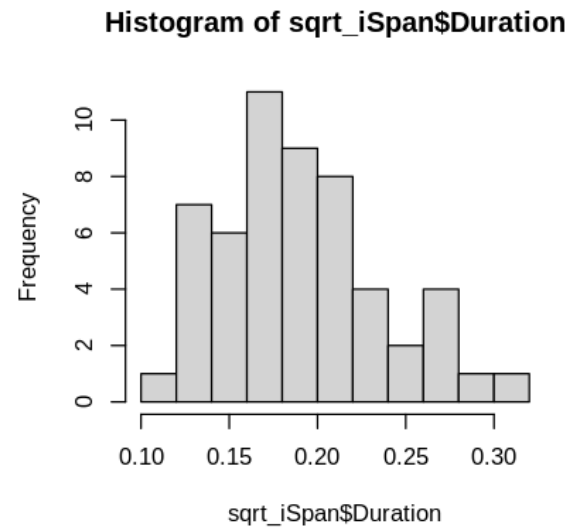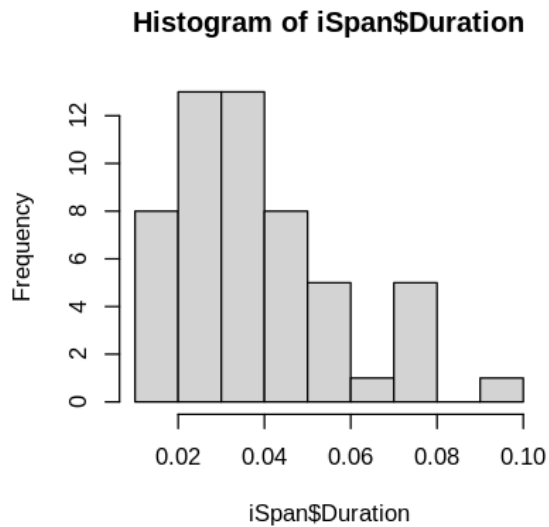
With p-value of $0.002321 < 0.05$ we reject the null hypothesis that the data are from a normally
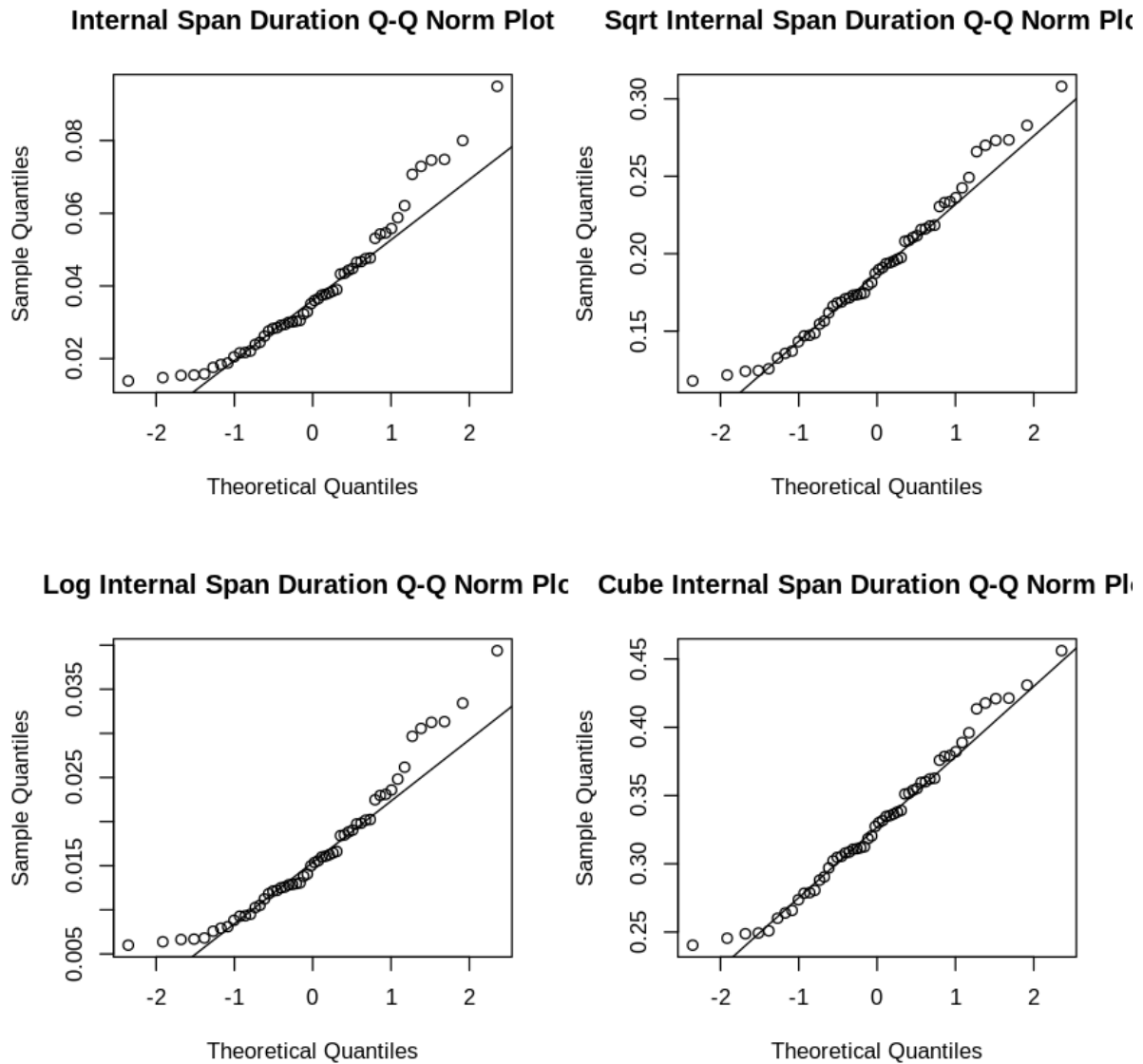
distributed population.

"if the p value is greater than the chosen alpha level, then the null hypothesis (that the data came from a normally distributed population) can not be rejected"

5.1.4 Data Transformations

5.1.4.1 Sqrt-Log-Cube Transformations

Histogram of iSpan$Duration

Histogram of sqrt_iSpan$Duration

Histogram of log_iSpan$Duration

Histogram of cube_iSpan$Duration

5.1.4.2 Q-Q Norm Sqrt-Log-Cube

Internal Span Duration Q-Q Norm Plot

Sqrt Internal Span Duration Q-Q Norm Plo

Log Internal Span Duration Q-Q Norm Plo

Cube Internal Span Duration Q-Q Norm Pl

5.1.4.3 Shapiro-Wilk Testing Sqrt-Log-Cube

```
	Shapiro-Wilk normality test

data:  sqrt_iSpan$Duration
W = 0.9683, p-value = 0.1621
```

```
     Shapiro-Wilk normality test

data:  log_iSpan$Duration
W = 0.92922, p-value = 0.003398


     Shapiro-Wilk normality test

data:  cube_iSpan$Duration
W = 0.97633, p-value = 0.3593
```

The **cube transformation** seems to provide the best q-q plot fit. With a p-value of 0.3593 > 0.05 we fail to reject the null hypothesis and assume we now have a normal distribution.

"if the p value is greater than the chosen alpha level, then the null hypothesis (that the data came from a normally distributed population) can not be rejected"

5.1.5 Autocorrelation

Autocorrelation plots are a commonly-used tool for checking randomness in a data set. This randomness is ascertained by computing autocorrelations for data values at varying time lags. If random, such autocorrelations should be near zero for any and all time-lag separations. If non-random, then one or more of the autocorrelations will be significantly non-zero.

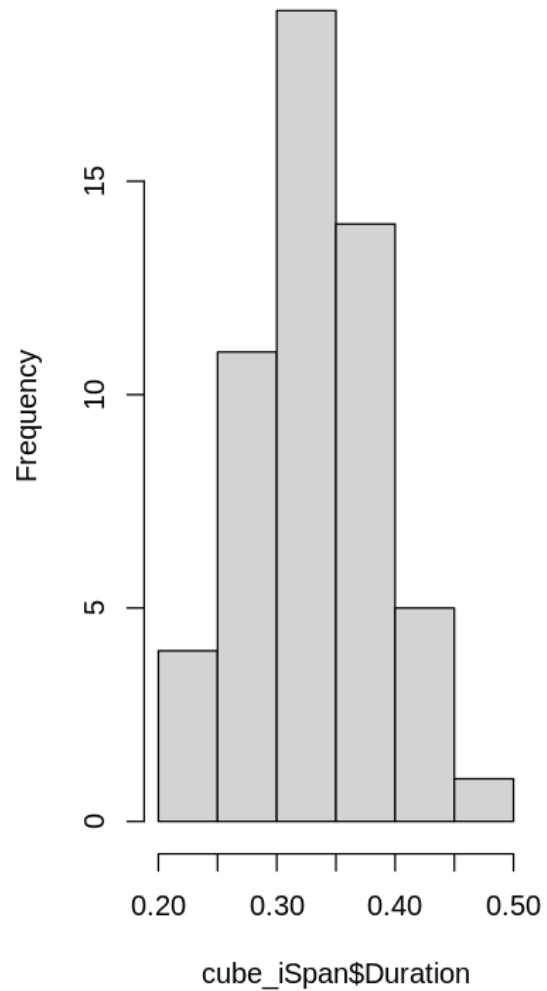## ACF of Cube Transformed Internal Span Duration
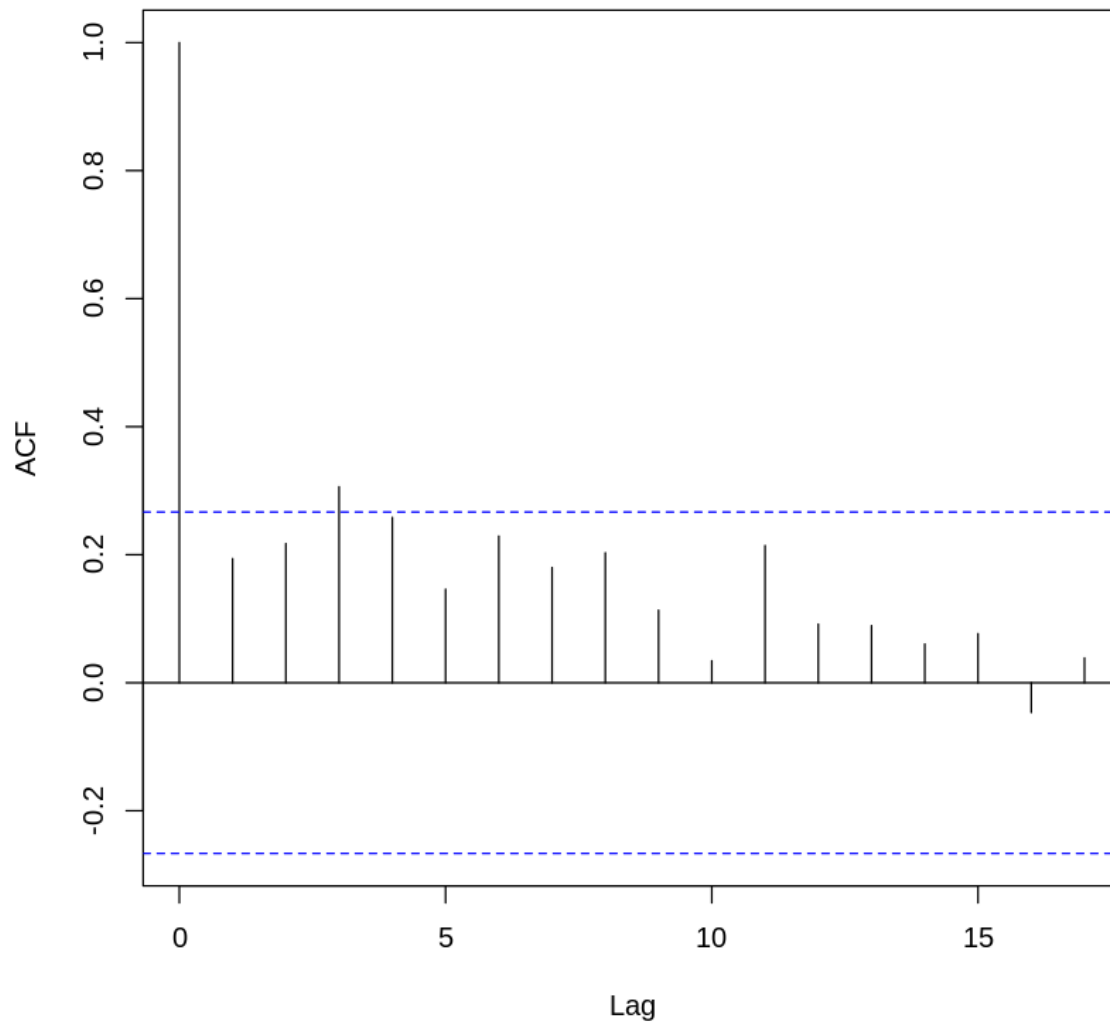
The ACF indicates that the data is random since the results are near zero.

Histogram of iSpan$Duration
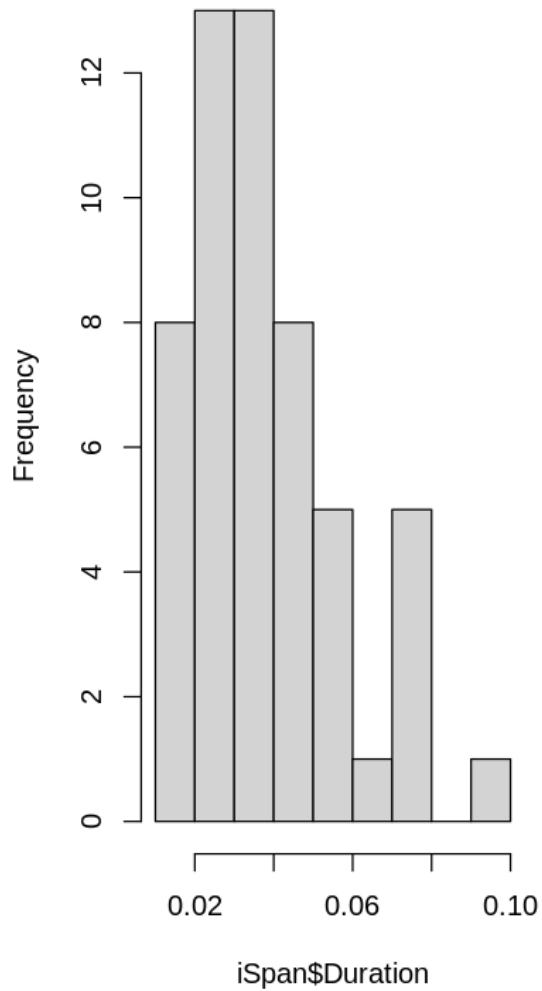
Histogram of cube_iSpan$Duratio

5.1.6 Hypothesis Testing

We will use a Student's t-Test to test the hypothesis on **normal** internal span data. Our mean is 500 ms (e.g. $\mu = 0.5$ seconds) and our null hypthesis is less than 500 ms.

```
    One Sample t-test

data:  x
```

```
t = -64.323, df = 53, p-value = 1
alternative hypothesis: true mean is greater than 0.7937005
95 percent confidence interval:
 0.3178723        Inf
sample estimates:
mean of x
0.3299424


    One Sample t-test

data:  x
t = -180.44, df = 53, p-value = 1
alternative hypothesis: true mean is greater than 0.5
95 percent confidence interval:
 0.03440894        Inf
sample estimates:
 mean of x
0.03868889
```
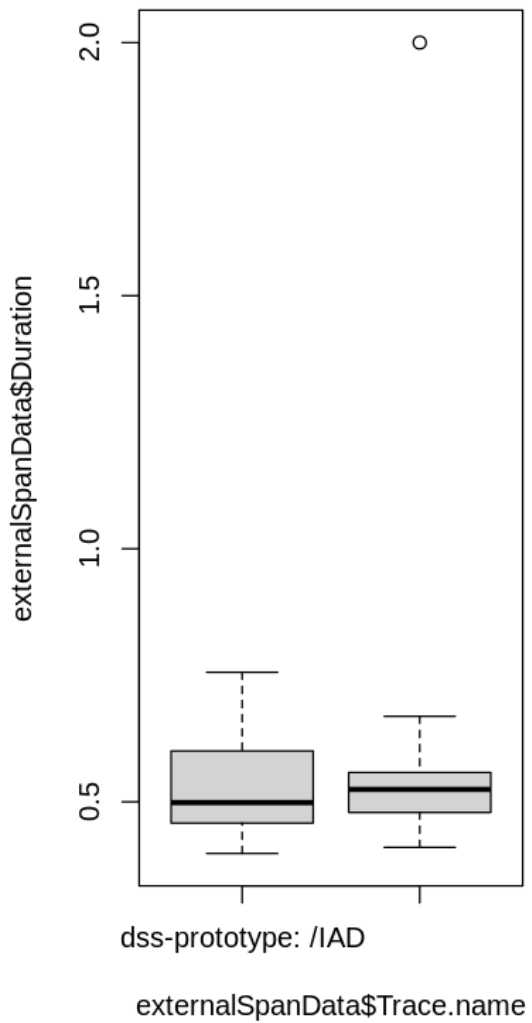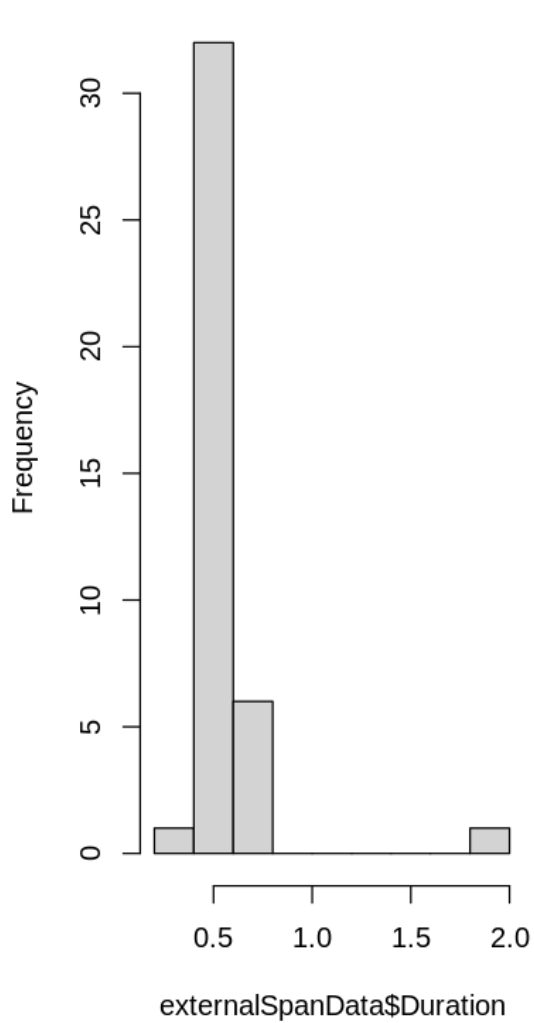
With a original and transformation with a p-value of $1 > 0.05$ we fail to reject the null hypothesis, i.e. we assume that latency will be less than 500 ms.

"If the p value is greater than the chosen alpha level, then the null hypothesis (that latency is < 500 ms) can not be rejected"

5.2 External Data

```
   Trace.ID            Trace.name           Start.time              Duration
 Length:40           Length:40          Min.   :1.651e+09    Min.   :0.3980
 Class :character    Class :character   1st Qu.:1.651e+09    1st Qu.:0.4670
 Mode  :character    Mode  :character   Median :1.651e+09    Median :0.5070
                                        Mean   :1.651e+09    Mean   :0.5565
                                        3rd Qu.:1.651e+09    3rd Qu.:0.5645
                                        Max.   :1.651e+09    Max.   :2.0000
   useCase            numContainers extNetworkHops
 Length:40           Min.   :3    Min.   :14
 Class :character    1st Qu.:3    1st Qu.:14
 Mode  :character    Median :3    Median :14
                     Mean   :3    Mean   :14
                     3rd Qu.:3    3rd Qu.:14
                     Max.   :3    Max.   :14
```

31

Histogram of externalSpanData$Dura

1. 2
2. 0.756

```
   Trace.ID          Trace.name          Start.time            Duration
Length:38          Length:38          Min.   :1.651e+09    Min.    :0.3980
Class :character   Class :character   1st Qu.:1.651e+09    1st Qu.:0.4650
Mode  :character   Mode  :character   Median :1.651e+09    Median :0.5020
                                      Mean   :1.651e+09    Mean    :0.5132
                                      3rd Qu.:1.651e+09    3rd Qu.:0.5537
```
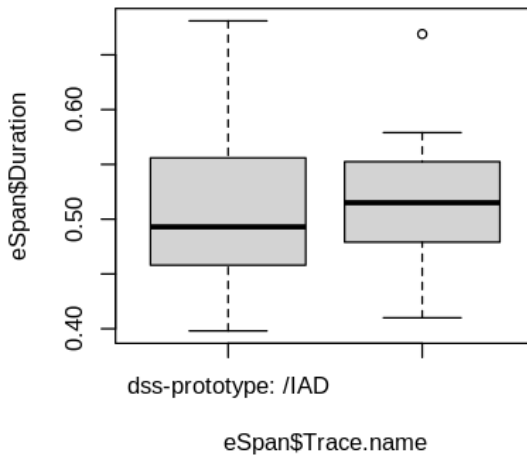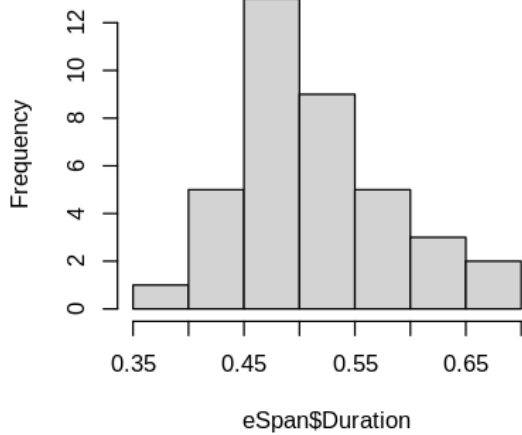
```
                                          Max.   :1.651e+09   Max.    :0.6810
    useCase           numContainers extNetworkHops
Length:38             Min.   :3     Min.   :14
Class :character      1st Qu.:3     1st Qu.:14
Mode  :character      Median :3     Median :14
                      Mean   :3     Mean   :14
                      3rd Qu.:3     3rd Qu.:14
                      Max.   :3     Max.   :14
```
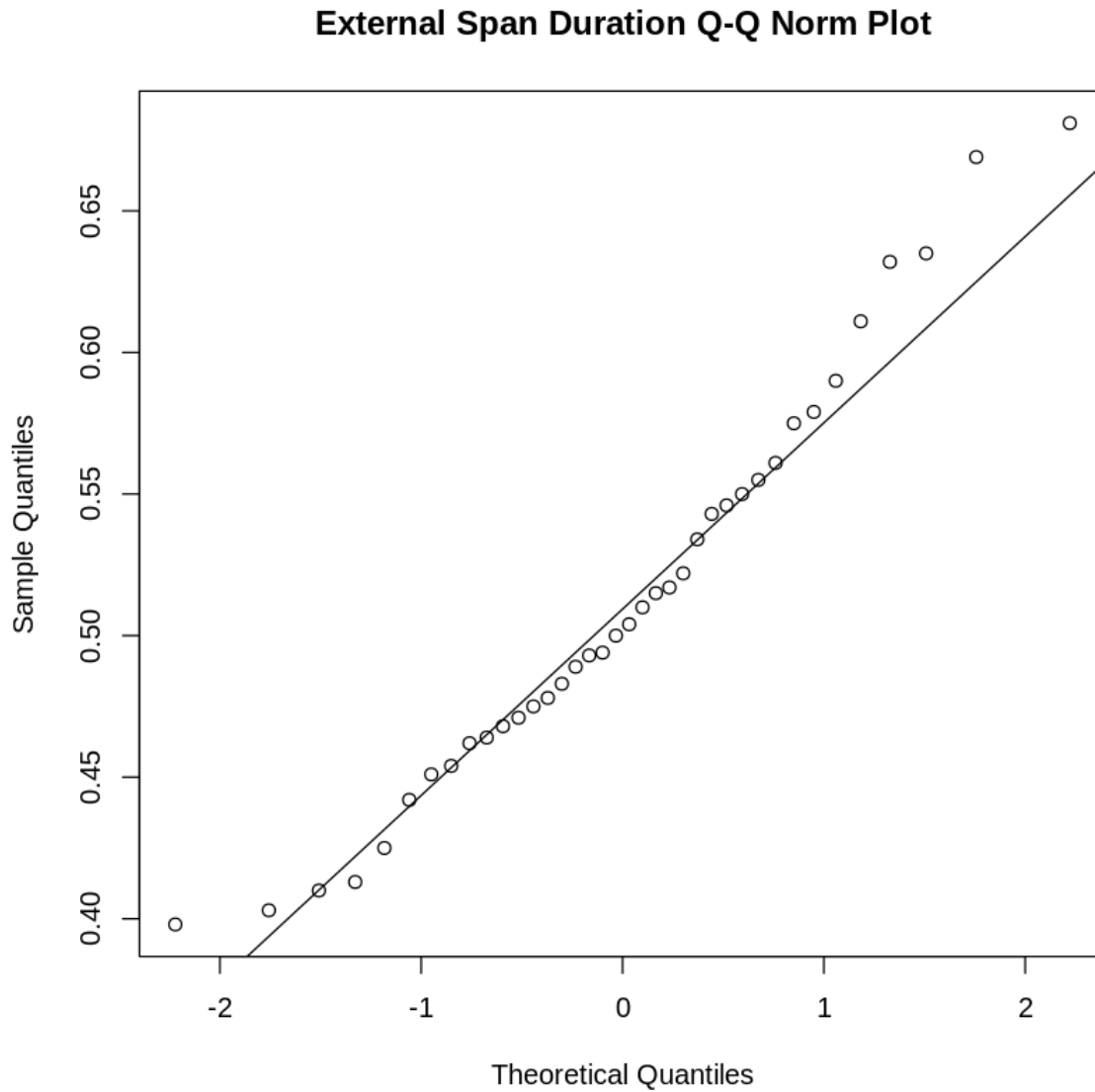
Histogram of externalSpanData$Duratior
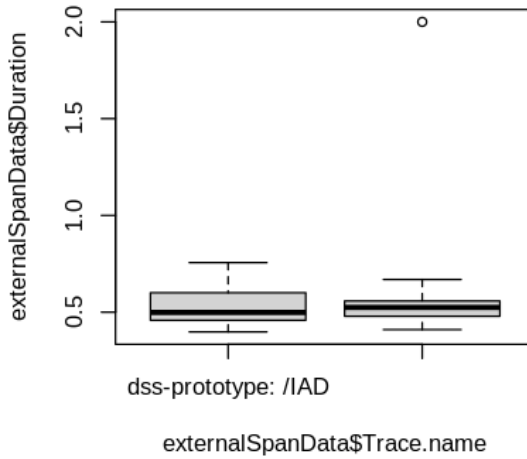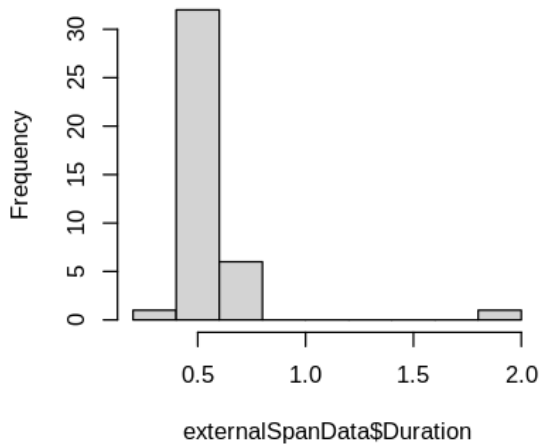
Histogram of eSpan$Duration

5.2.1 Q-Q Norm Plot of "Clean" External Span Data

We'll look a the Q-Q Norm Plot and Shapiro-Wilk Test
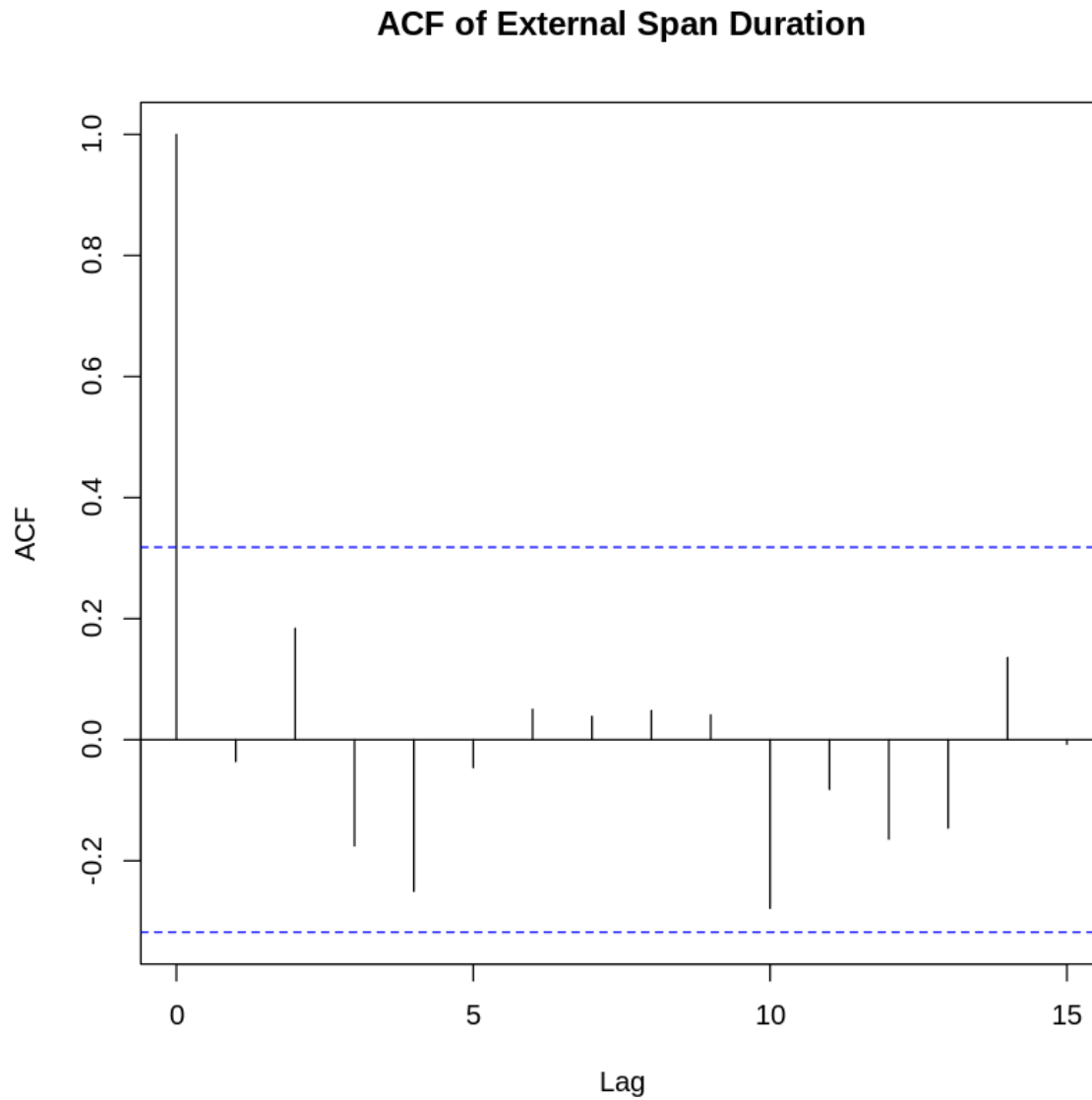
External Span Duration Q-Q Norm Plot

5.2.2 Shapiro-Wilk Normality Test

```
Shapiro-Wilk normality test
```

34

```
data:  eSpan$Duration
W = 0.96564, p-value = 0.2878
```

With a p-value of $0.2878 > 0.05$ we fail to reject the null hypothesis, i.e. we assume that we have a normal distribution.

5.2.3 Autocorrelation

ACF of External Span Duration

The ACF indicates that the data is random since the results are near zero.

5.2.4 Hypothesis Testing

We will use a Student's t-Test to test the hypothesis on external span data. Our mean is 500 ms (e.g. $\mu = 0.5$ seconds) and our null hypthesis is less than 500 ms.

```
    One Sample t-test

data:  x
t = 1.1267, df = 37, p-value = 0.1336
alternative hypothesis: true mean is greater than 0.5
95 percent confidence interval:
 0.4934287        Inf
sample estimates:
mean of x
0.5132105
```

With a p-value of $0.1336 > 0.05$ we fail to reject the null hypothesis, i.e. we assume that 500 ms can be maintained for external service requests.

"If the p value is greater than the chosen alpha level, then the null hypothesis (that latency is < 500 ms) can not be rejected"

6 Observations

6.1 General Discussion of Normality

It was required to separate external data from internal to establish normality of the data samples. The internal data set required transformation to establish normality, while the external data did not require a transformation.

6.2 Hypothesis Results

Hypothesis testing using the Student's t-Test indicates that latency constraints of 500 ms can be maintained internally and external. However, serveral external samples were greater than 500 ms. This is most likely due to the non-deterministic nature of internet (e.g. http) requests. Within the internal environment, data is directly routed between microservices within the Docker environment within a private network. The data shows that a container based microservice architecture can meet the requirement; however, care must be taken to manage processing per container that may increase container response times.