

# Signals packet

*Brian Cohn*

*11/28/2017*

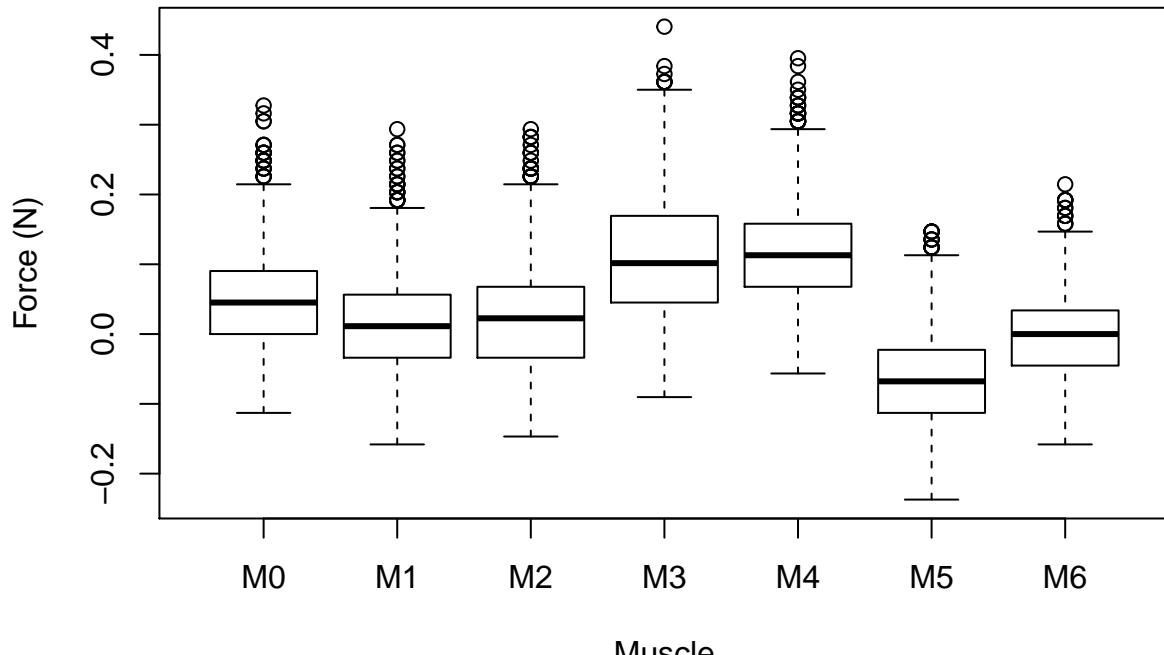
## Contents

<b>Basement Muscles</b>	<b>1</b>
Spectral Properties of the muscle forces on load cells . . . . .	2
<b>Akira BICE JR3 Sensor SysID</b>	<b>7</b>
<b>Akira's JR3 Before and after calibration &amp;/or mean-centering</b>	<b>7</b>
Akira JR3 Before and after mean-centering. TODO: after calibration & calibration+mean-centering	7
Mean-Centered Akira-JR3 Voltages . . . . .	10
Calibrated & Mean-Centered Akira-JR3 Voltages . . . . .	18
<b>Basement JR3 Signal SysID</b>	<b>26</b>
BasementJR3 Before and after calibration &/or mean-centering . . . . .	26
Mean-Centered Basement JR3 Voltages . . . . .	30
Calibrated and Mean-Centered Basement JR3 Forces . . . . .	33
How does Basement JR3 respond when pressing on the 20mm-displaced mount port where the fingertip will rest? . . . . .	41
How does Basement JR3 respond when fingertip presses near the JR3 baseplate? . . . . .	42
How does Basement JR3 respond when muscles motor-collars are twisted, 3x ea, by hand? . . . . .	43
How does Basement JR3 respond when tensioned by a static 500 g force hung from the kevlar? . . .	45
Low pass filtered output porcupine of MXYZ VOLTAGES at a 50Hz critical frequency . . . . .	49
<b>TODO create FFS from generators via n_binary_combinations</b>	<b>50</b>

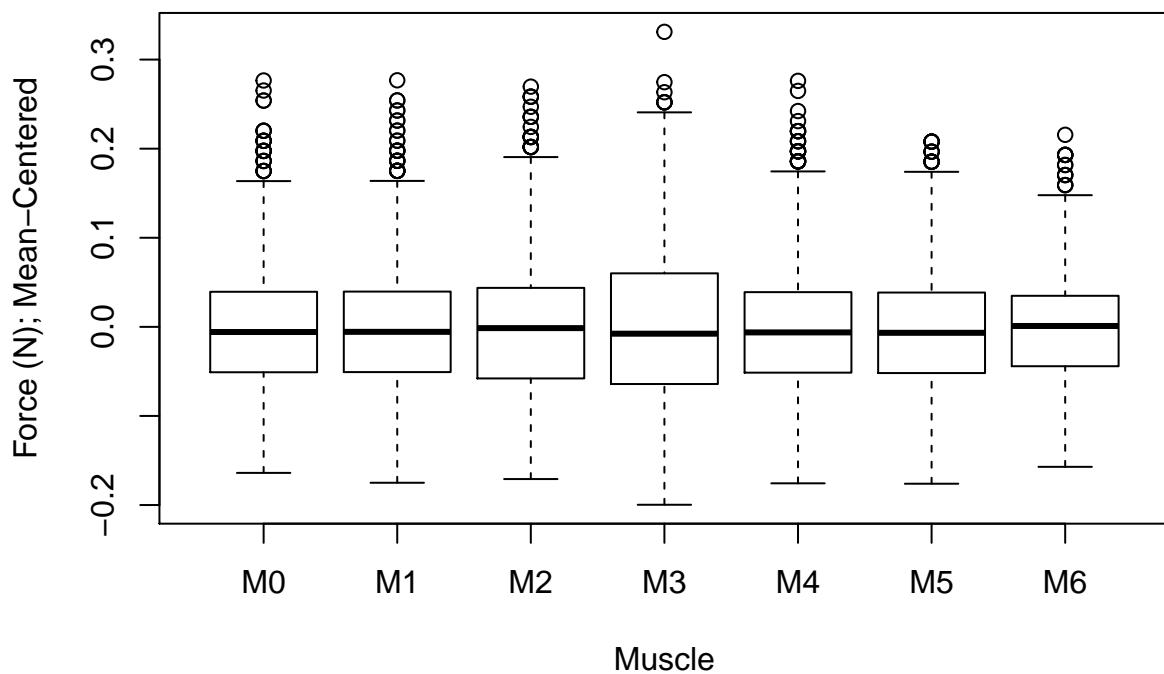
## Basement Muscles

noiseResponse2017\_11\_28\_15\_55\_30.txt features 1534 samples, over 3.333 seconds. Empirical sampling frequency is 460.2460246

```
boxplot(measured_signals, xlab="Muscle", ylab="Force (N)", names=muscle_names())
```



```
boxplot(measured_signals_zeroed, xlab="Muscle", ylab="Force (N); Mean-Centered", names=muscle_names())
```



### Spectral Properties of the muscle forces on load cells

```
time_series_to_harmonics_psd <- function(ts, acq_freq=1000,...){
library(GeneCycle)
f.data <- GeneCycle::periodogram(ts)
harmonics <- 1:(acq_freq/2)
plot(f.data$freq[harmonics]*length(ts),
```

```

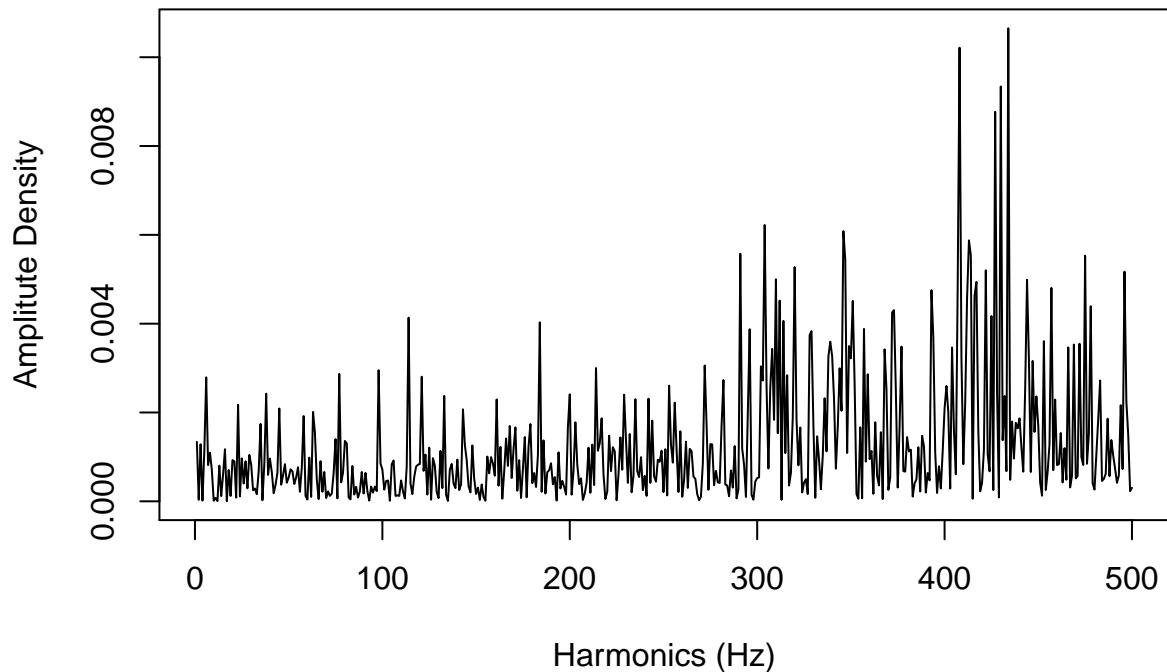
f.data$spec[harmonics] / sum(f.data$spec),
xlab="Harmonics (Hz)", ylab="Amplitude Density", type="l", ...)
}

muscle_sample <- as.data.frame(measured_signals_zeroed)
m_names <- measured(muscle_names())
time_series_to_harmonics_psd(muscle_sample[,1], acq_freq=1000, main="measured_M0")

## Loading required package: longitudinal
## Loading required package: corpcor
## Loading required package: fdrtool

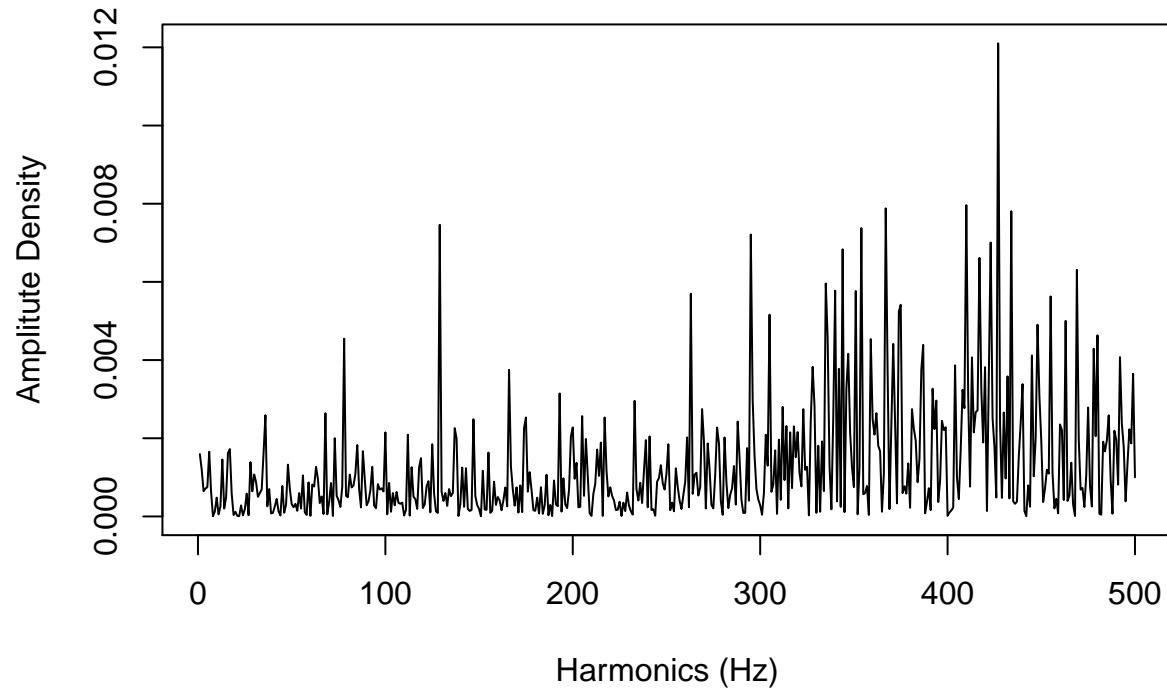
```

## measured\_M0



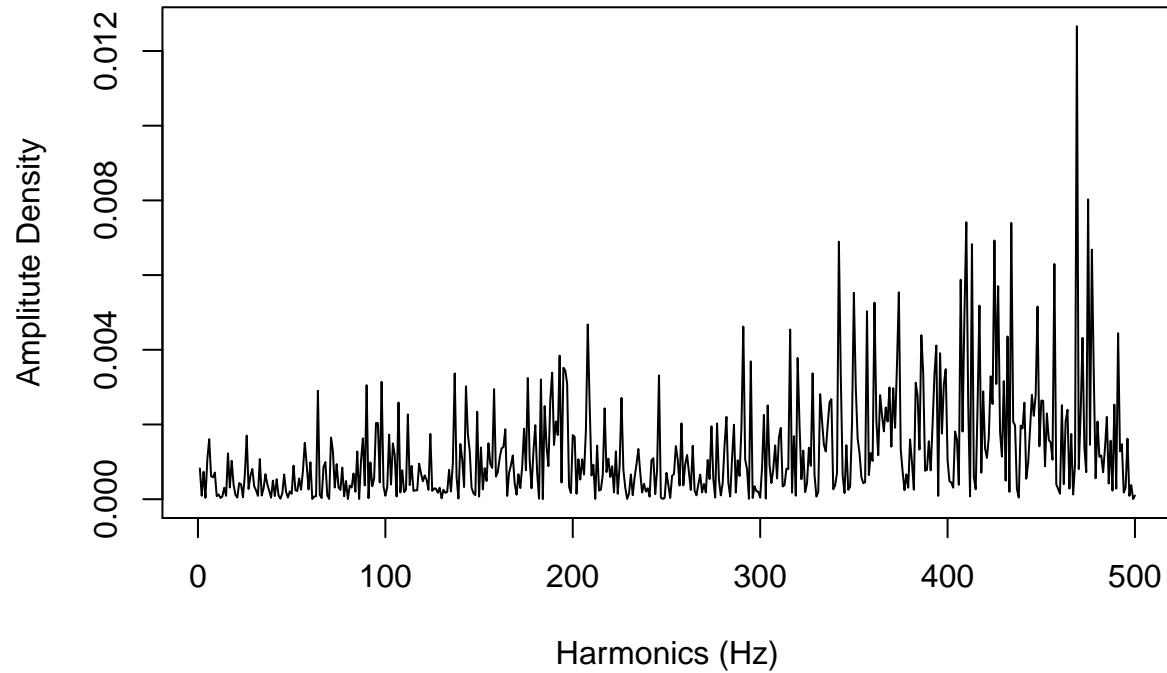
```
time_series_to_harmonics_psd(muscle_sample[,2], acq_freq=1000, main="measured_M1")
```

**measured\_M1**



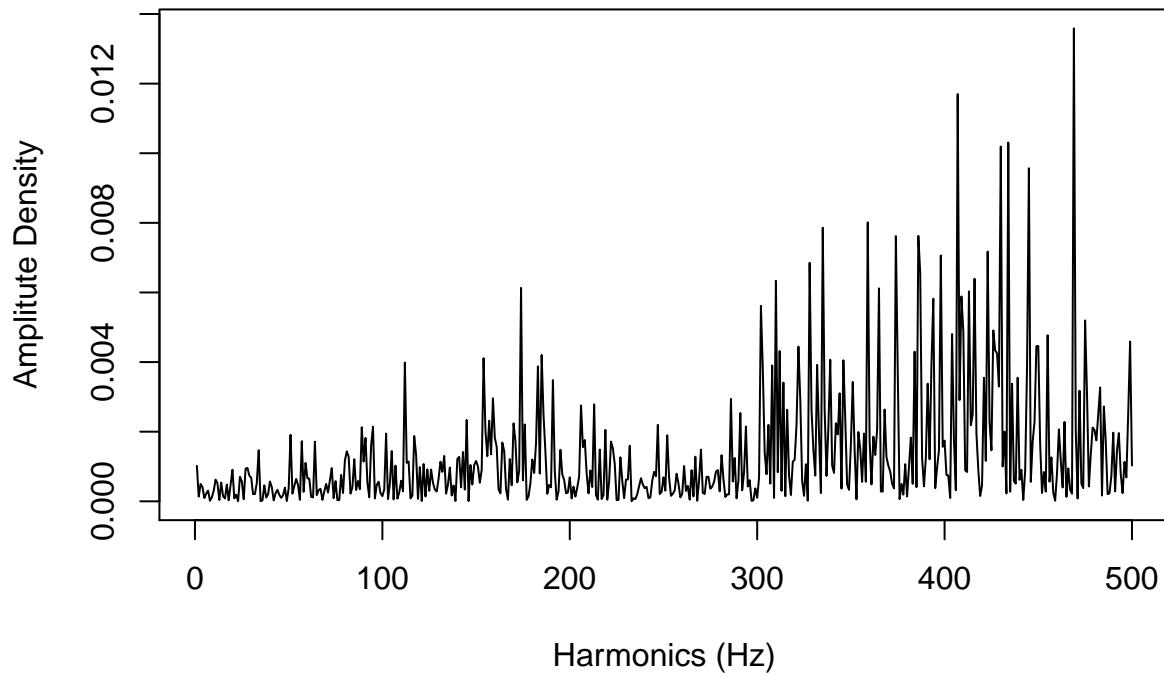
```
time_series_to_harmonics_psd(muscle_sample[,3], acq_freq=1000, main="measured_M2")
```

**measured\_M2**



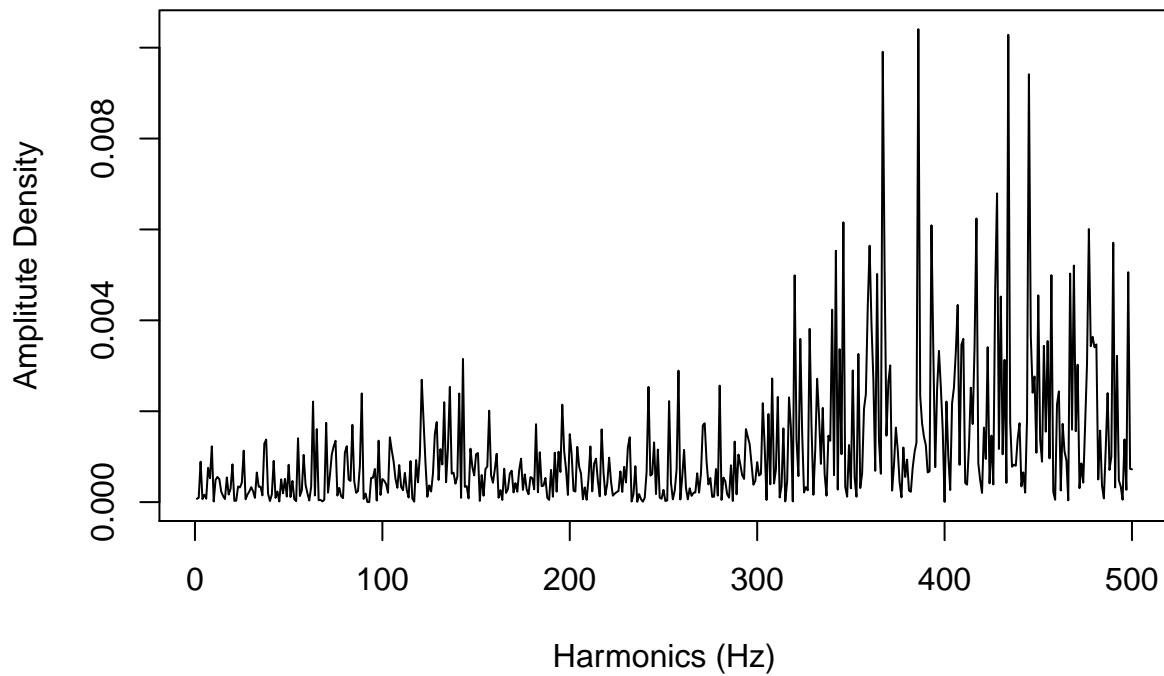
```
time_series_to_harmonics_psd(muscle_sample[,4], acq_freq=1000, main="measured_M3")
```

**measured\_M3**



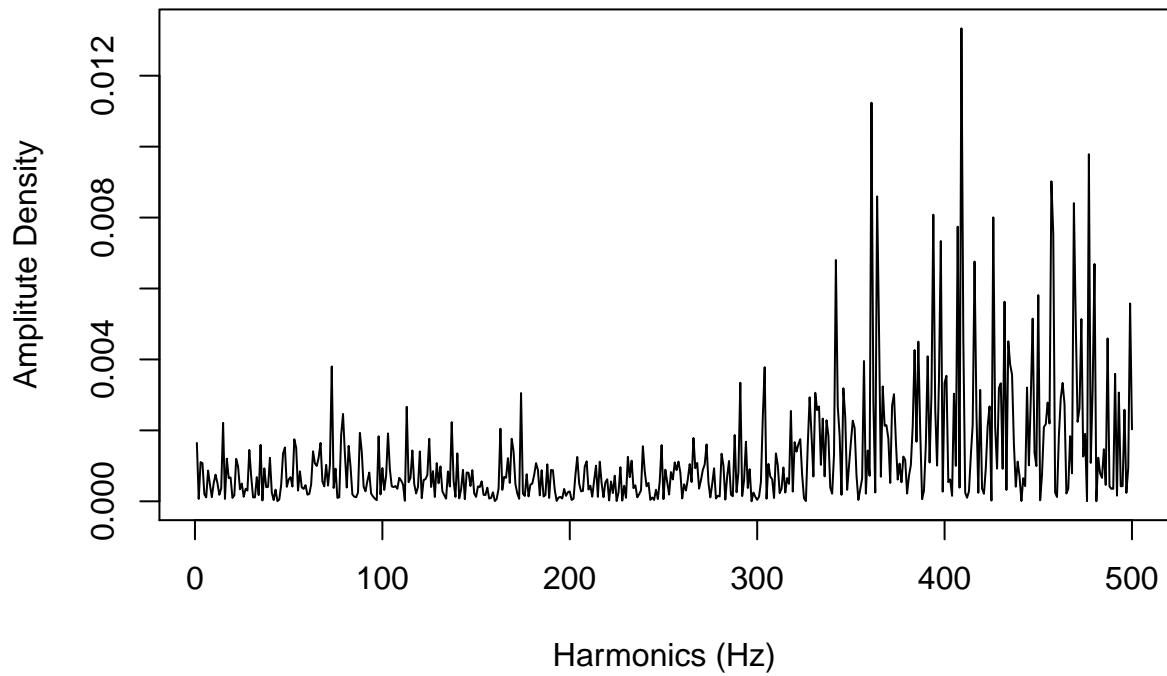
```
time_series_to_harmonics_psd(muscle_sample[,5], acq_freq=1000, main="measured_M4")
```

**measured\_M4**



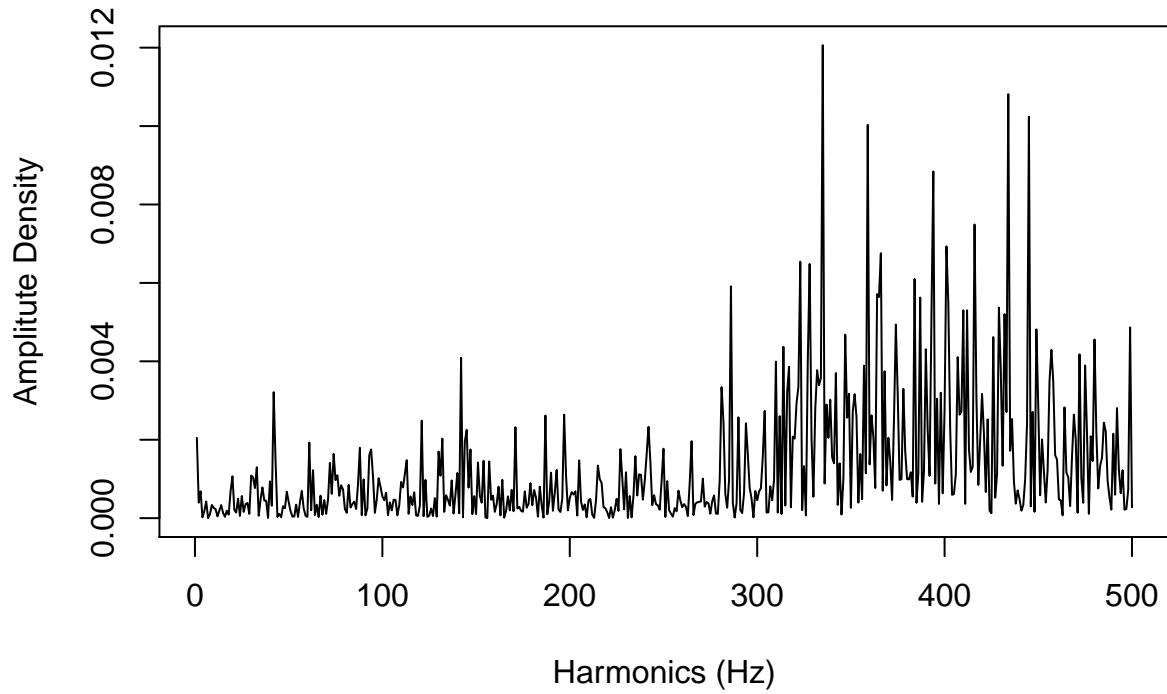
```
time_series_to_harmonics_psd(muscle_sample[,6], acq_freq=1000, main="measured_M5")
```

### **measured\_M5**



```
time_series_to_harmonics_psd(muscle_sample[,7], acq_freq=1000, main="measured_M6")
```

### **measured\_M6**



# Akira BICE JR3 Sensor SysID

## Akira's JR3 Before and after calibration &/or mean-centering

NOTE: Akira does not know moment labels and calibration matrix has not arrived yet.

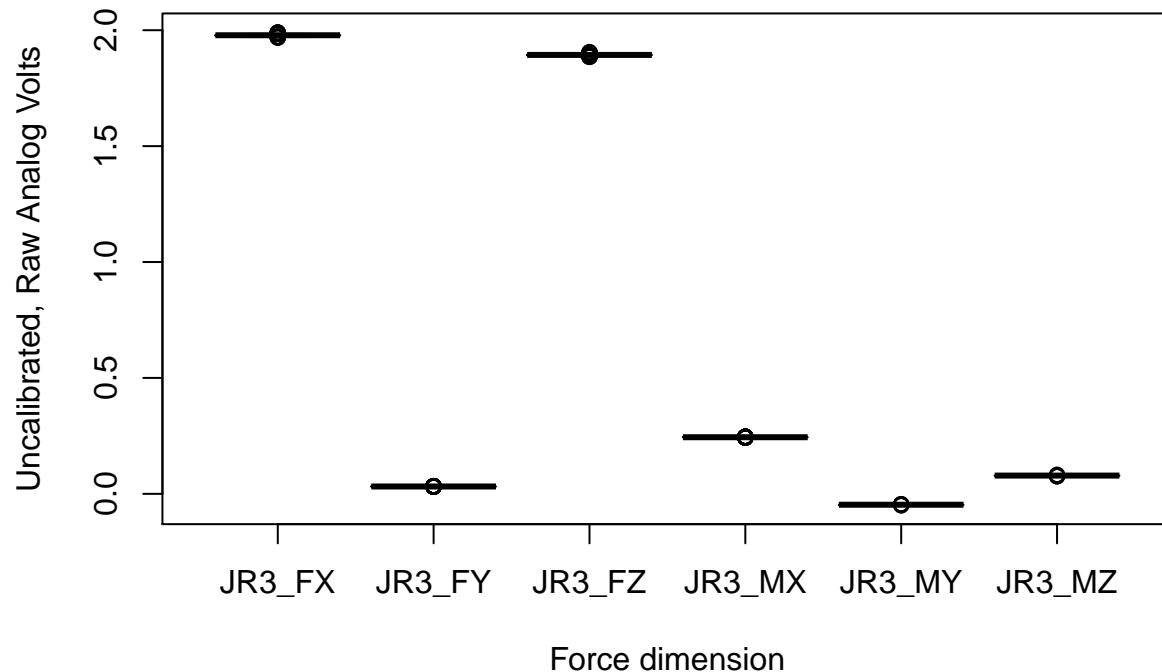
```
akira_jr3_at_rest.csv features 10000 samples, over 9.999 seconds. Empirical sampling frequency is 1000.10001
##Raw Akira JR3 signals over 1s from 2s to 3s
summary(just_jr3_cols_akira[2000:3000,])
```

```
##          JR3_FX           JR3_FY           JR3_FZ           JR3_MX
##  Min.   :1.967   Min.   :0.03039   Min.   :1.885   Min.   :0.2426
##  1st Qu.:1.977   1st Qu.:0.03138   1st Qu.:1.892   1st Qu.:0.2439
##  Median :1.978   Median :0.03170   Median :1.893   Median :0.2442
##  Mean   :1.978   Mean   :0.03179   Mean   :1.894   Mean   :0.2443
##  3rd Qu.:1.980   3rd Qu.:0.03203   3rd Qu.:1.894   3rd Qu.:0.2445
##  Max.   :1.991   Max.   :0.03335   Max.   :1.906   Max.   :0.2462
##          JR3_MY           JR3_MZ
##  Min.   :-0.04922   Min.   :0.07612
##  1st Qu.:-0.04758   1st Qu.:0.07809
##  Median :-0.04692   Median :0.07875
##  Mean   :-0.04701   Mean   :0.07873
##  3rd Qu.:-0.04659   3rd Qu.:0.07941
##  Max.   :-0.04462   Max.   :0.08171
```

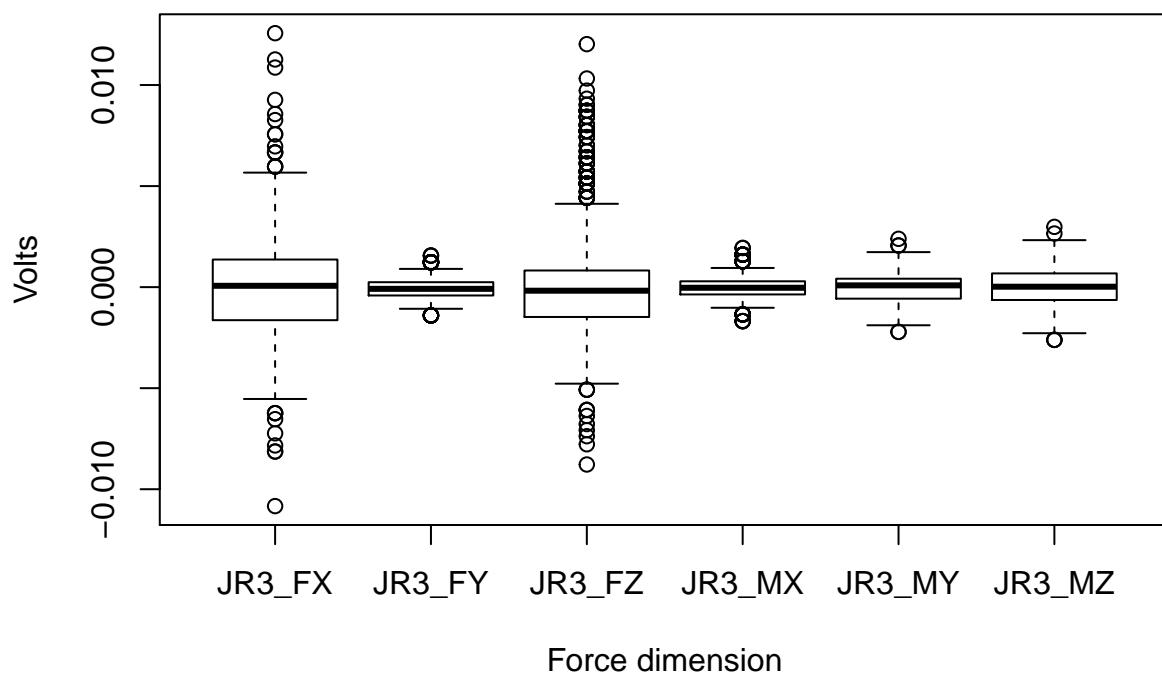
Akira JR3 Before and after mean-centering. TODO: after calibration & calibration+mean-centering

```
boxplot(just_jr3_cols_akira[2000:3000,], xlab="Force dimension", ylab="Uncalibrated, Raw Analog Volts",
```

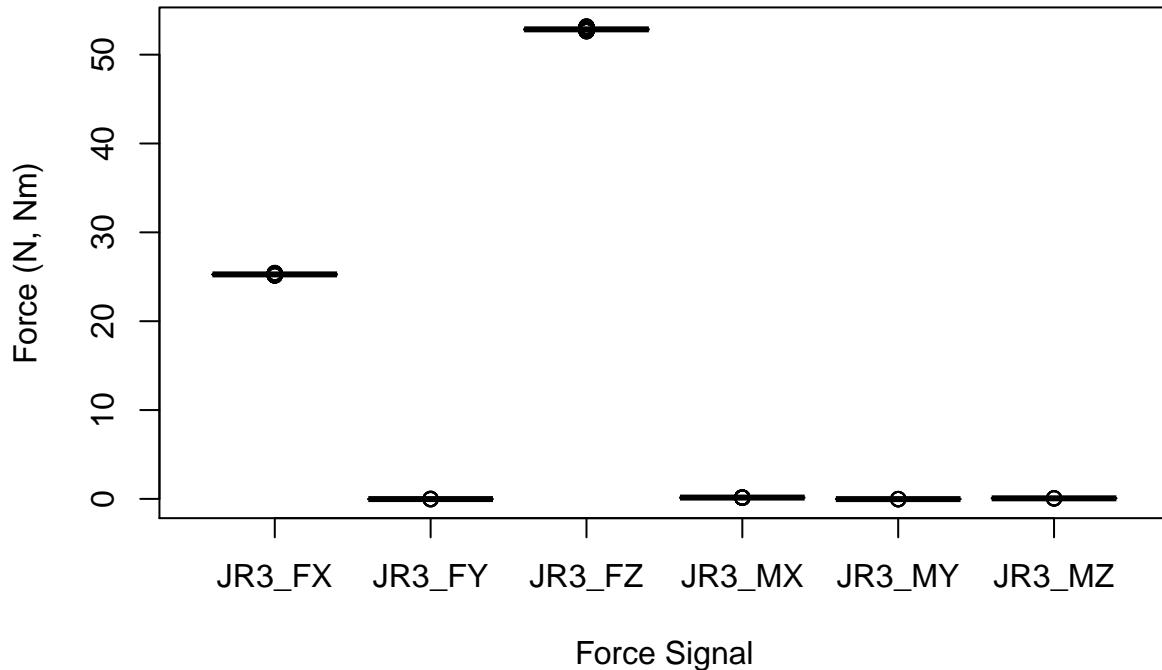
### JR3 Uncalibrated, Uncentered



### JR3 Uncalibrated & Mean-Centered



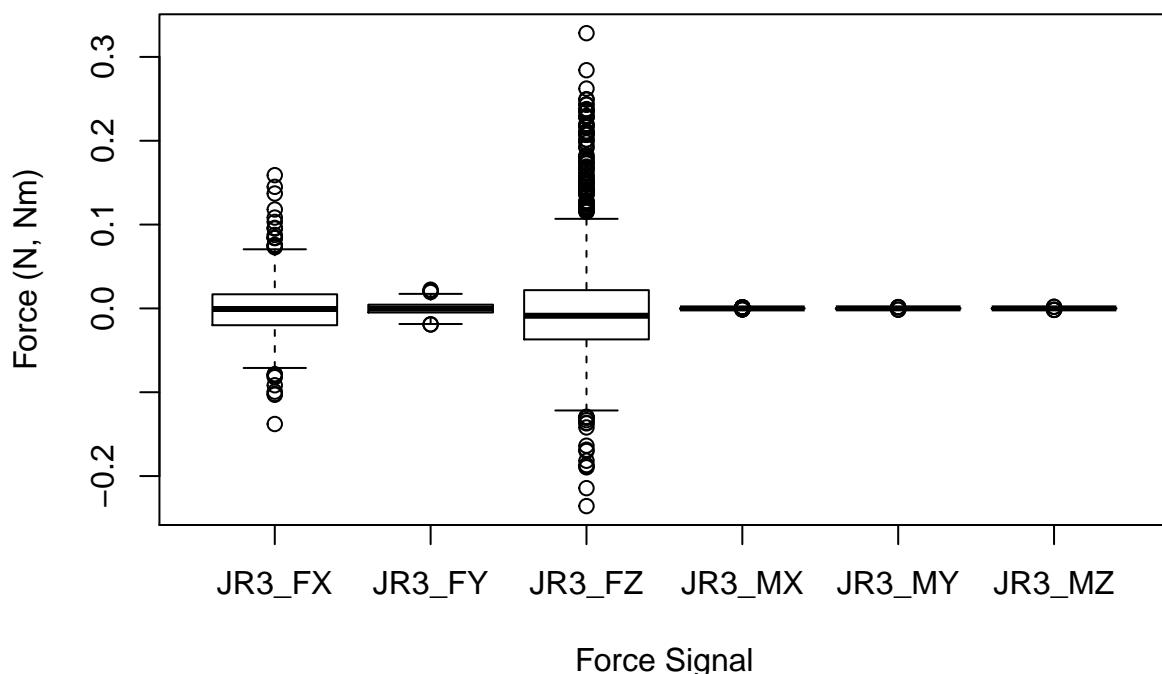
## JR3 Calibrated, Uncentered



Force Signal

```
boxplot(scale(just_jr3_cols_calibrated_akira[2000:3000,], scale=FALSE), xlab="Force Signal", ylab="Force")
```

## JR3 Calibrated & Mean-Centered



Force Signal

```
mean_centered_jr3 <- as.data.frame(scale(akira_jr3_rest, scale=FALSE))
one_second_sample <- mean_centered_jr3[mean_centered_jr3$time < 3.0 & mean_centered_jr3$time > 2.0,]
summary(one_second_sample)
```

```

##      JR3_FX          JR3_FY          JR3_FZ
##  Min. :-0.0125033  Min. :-1.759e-03  Min. :-8.530e-03
##  1st Qu.:-0.0020033 1st Qu.:-4.428e-04 1st Qu.:-1.005e-03
##  Median :0.0007033  Median :-1.138e-04  Median : 6.957e-05
##  Mean   :-0.0003554  Mean  :-2.963e-07  Mean  : 1.453e-04
##  3rd Qu.: 0.0012967 3rd Qu.: 5.442e-04 3rd Qu.: 1.070e-03
##  Max.   : 0.0160967  Max.  : 1.531e-03  Max.  : 1.187e-02
##      JR3_MX          JR3_MY          JR3_MZ
##  Min. :-1.986e-03  Min. :-2.230e-03  Min. :-2.584e-03
##  1st Qu.:-3.356e-04 1st Qu.:-5.852e-04 1st Qu.:-6.098e-04
##  Median :-5.584e-06  Median : 7.277e-05  Median : 4.720e-05
##  Mean   : 2.048e-05  Mean  : 2.803e-05  Mean  :-4.647e-06
##  3rd Qu.: 3.144e-04 3rd Qu.: 7.308e-04 3rd Qu.: 7.052e-04
##  Max.   : 1.964e-03  Max.  : 2.376e-03  Max.  : 2.679e-03
##      time
##  Min.   :2.001
##  1st Qu.:2.250
##  Median :2.500
##  Mean   :2.500
##  3rd Qu.:2.750
##  Max.   :2.999

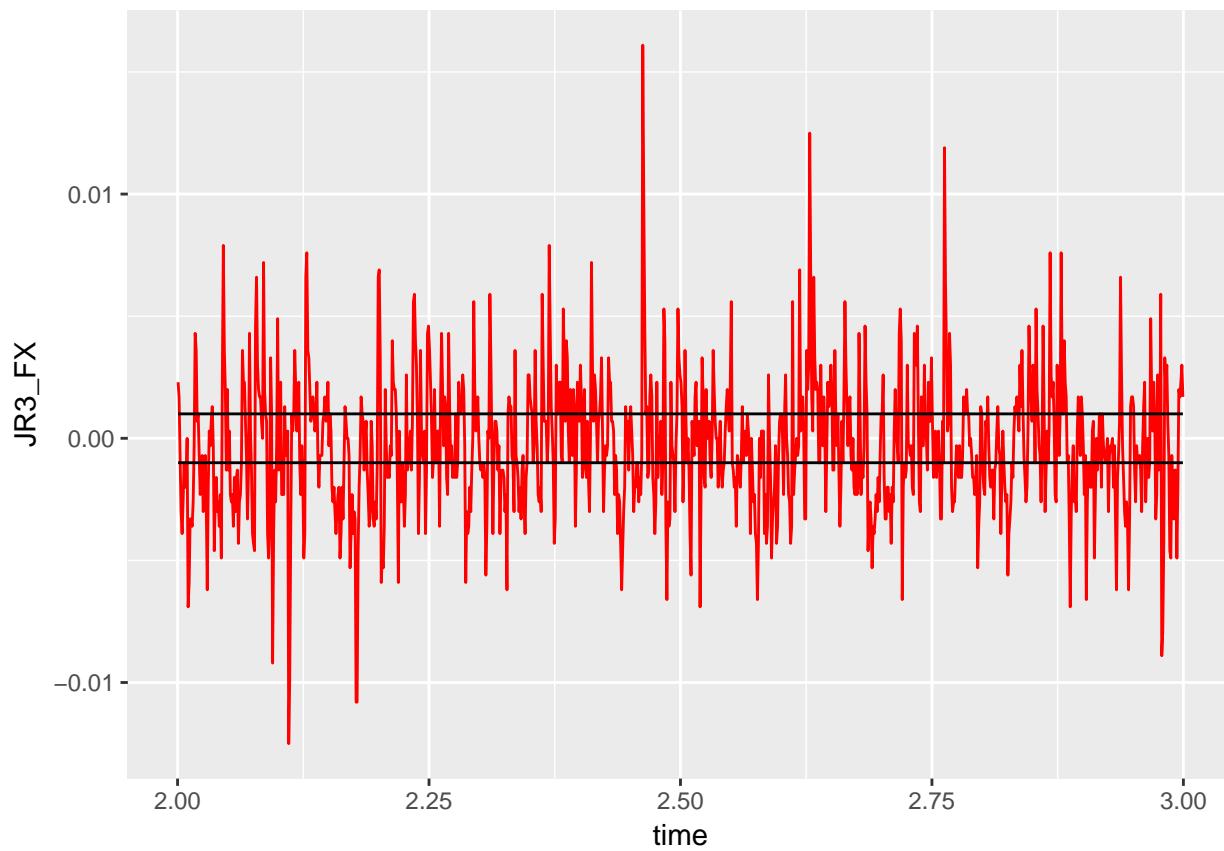
column_ranges(one_second_sample)

##             min         max
## JR3_FX -0.012503280 0.016096720
## JR3_FY -0.001758801 0.001531199
## JR3_FZ -0.008530430 0.011869570
## JR3_MX -0.001985584 0.001964416
## JR3_MY -0.002230225 0.002375775
## JR3_MZ -0.002583804 0.002679196
## time     2.000500000 2.999500000

```

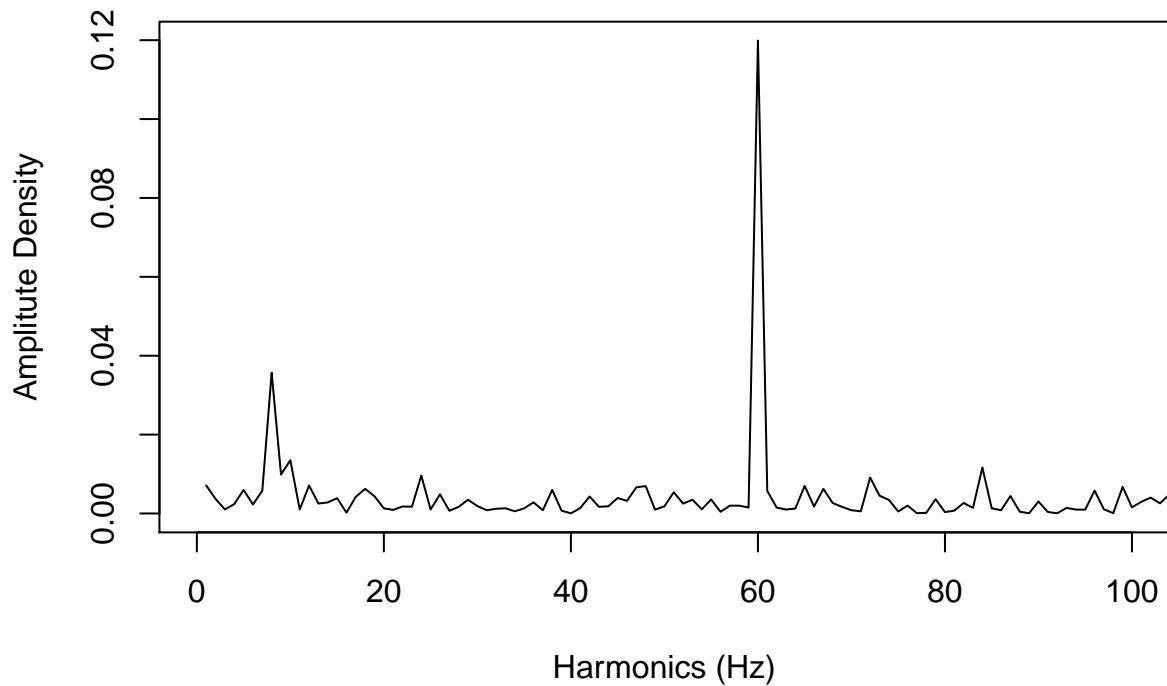
## Mean-Centered Akira-JR3 Voltages

```
ggplot(one_second_sample) + geom_line(aes(time, JR3_FX), col="red") + geom_line(aes(time, 1e-3)) + geom
```

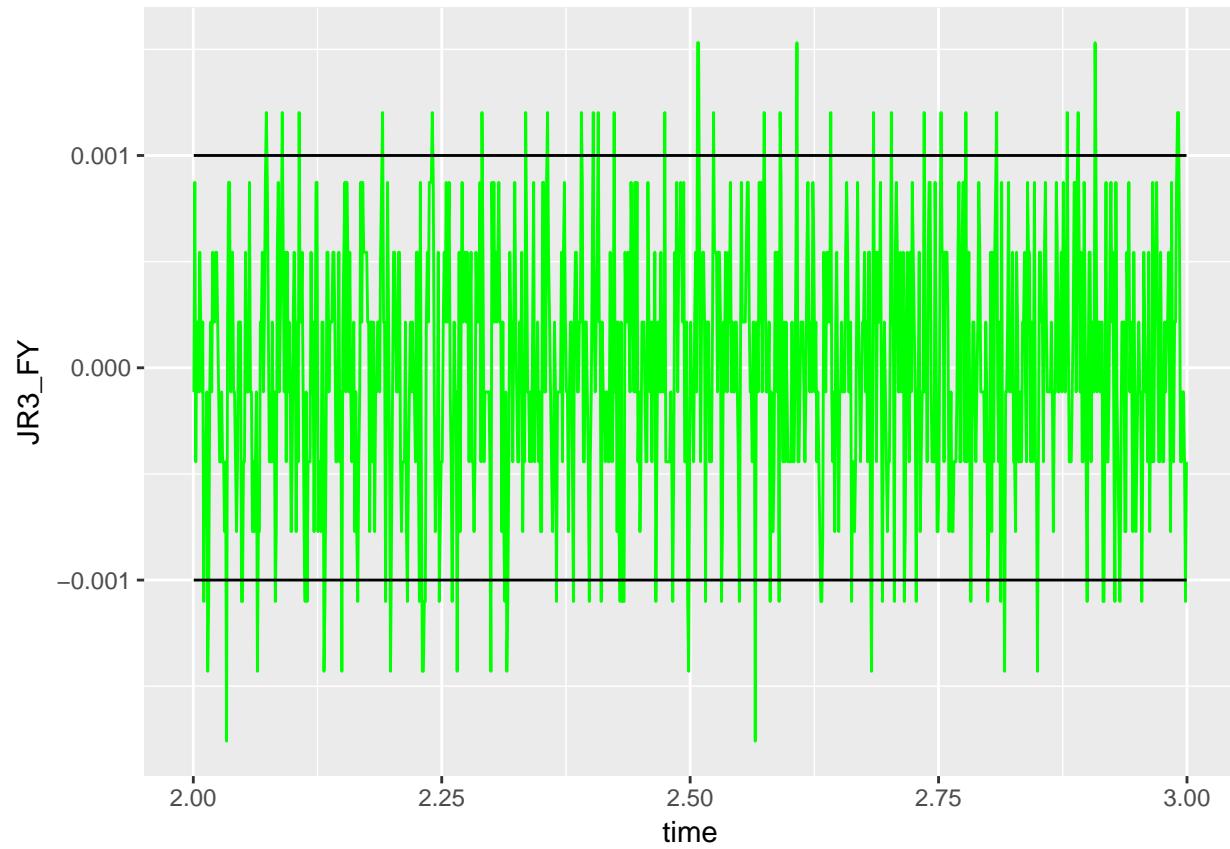


```
time_series_to_harmonics_psd(one_second_sample[, "JR3_FX"], acq_freq=1000, main="JR3_FX", xlim=c(0,100))
```

**JR3\_FX**

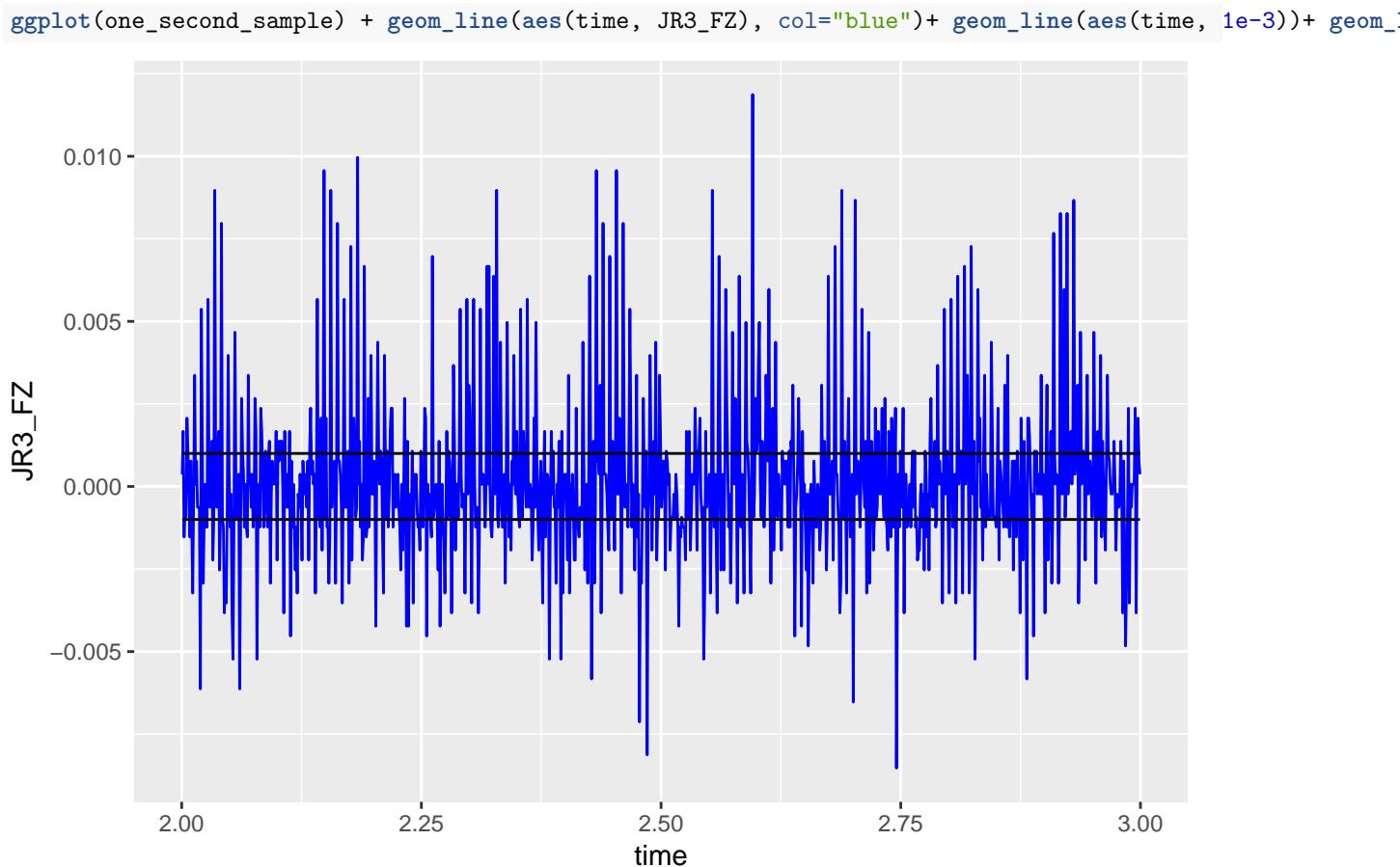
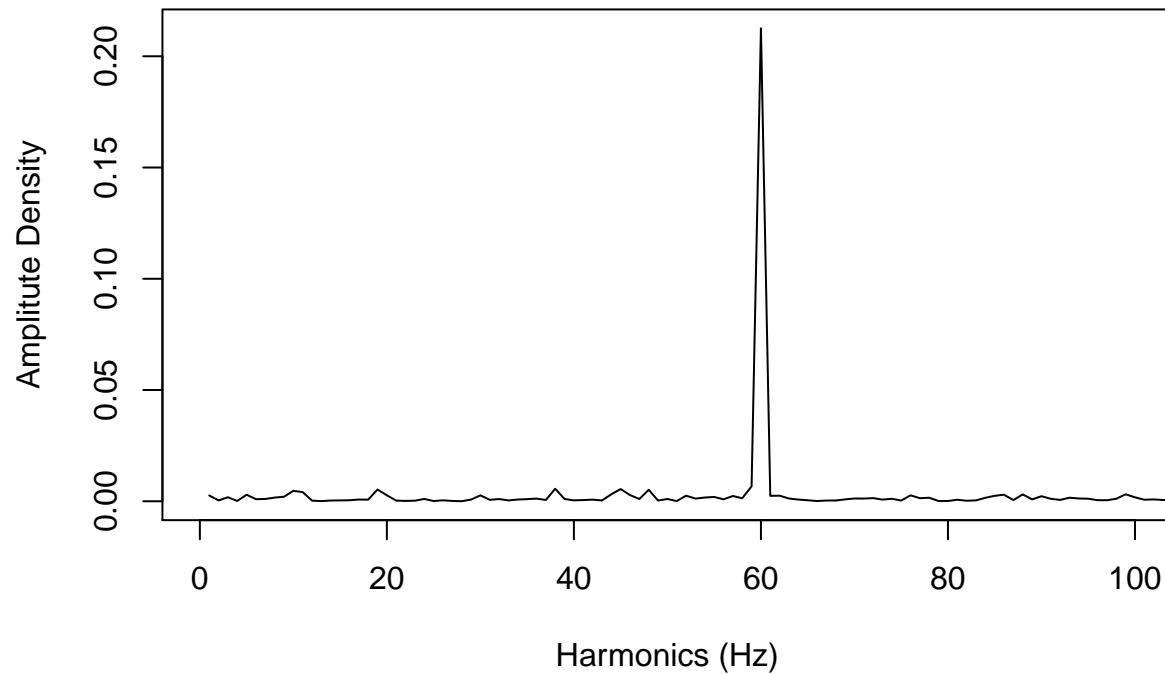


```
ggplot(one_second_sample) + geom_line(aes(time, JR3_FY), col="green") + geom_line(aes(time, 1e-3)) + geom
```



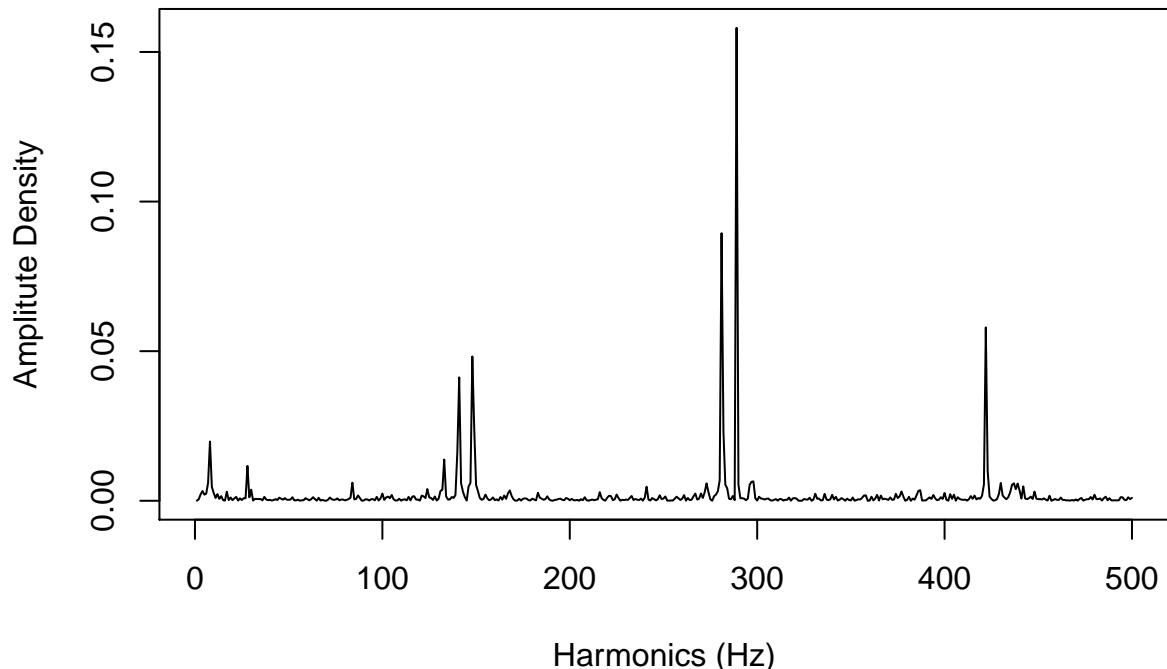
```
time_series_to_harmonics_psd(one_second_sample[, 'JR3_FY'], acq_freq=1000, main="JR3_FY", xlim=c(0,100))
```

## JR3\_FY

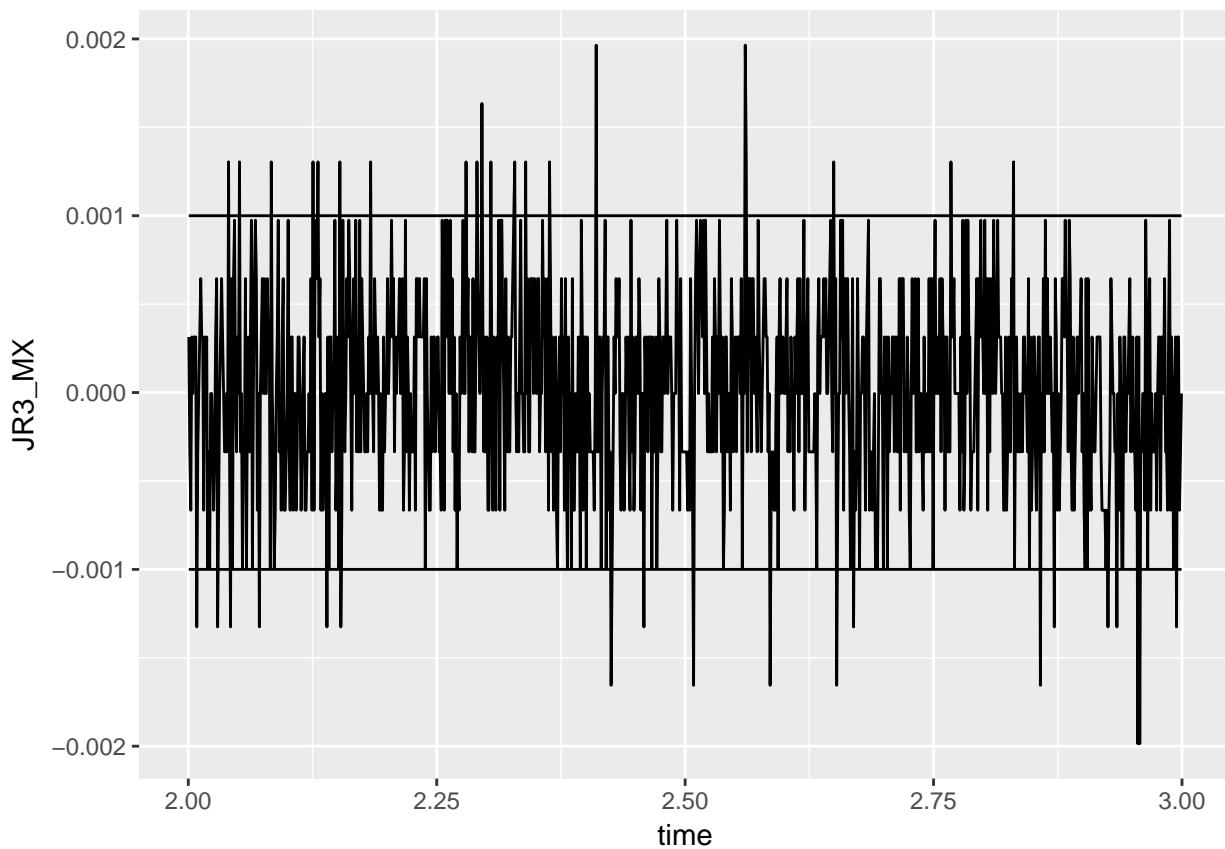


```
time_series_to_harmonics_psd(one_second_sample[, 'JR3_FZ'], acq_freq=1000, main="JR3_FZ")
```

JR3\_FZ

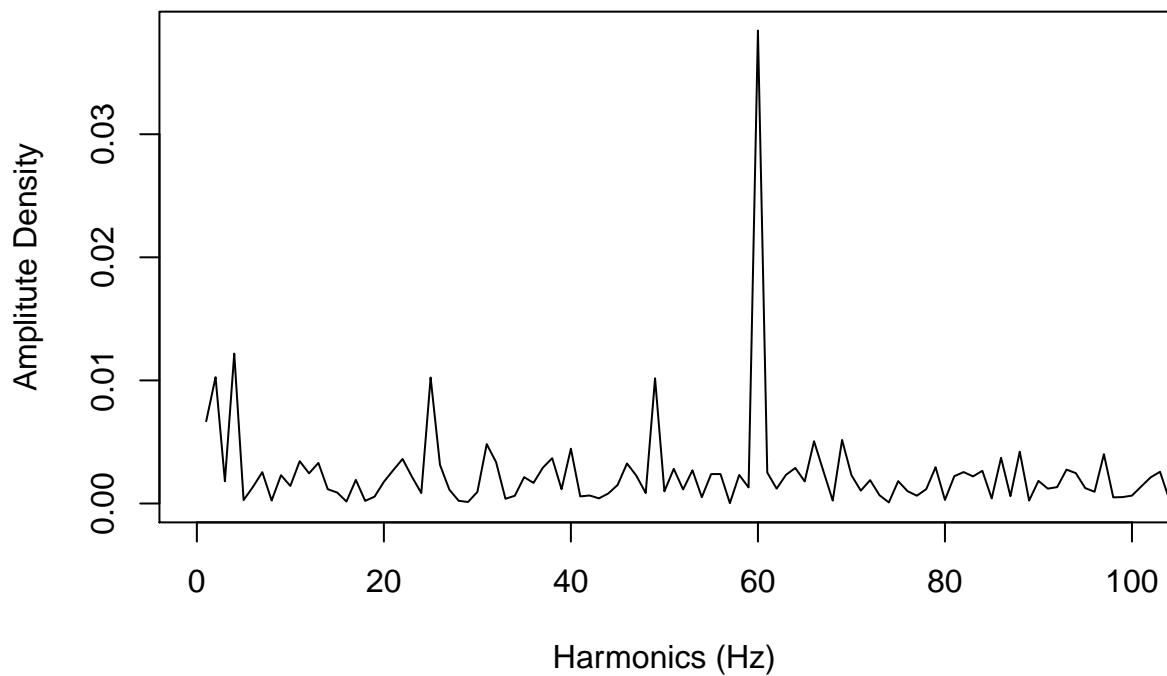


```
ggplot(one_second_sample) + geom_line(aes(time, JR3_MX)) + geom_line(aes(time, 1e-3))+ geom_line(aes(tim
```

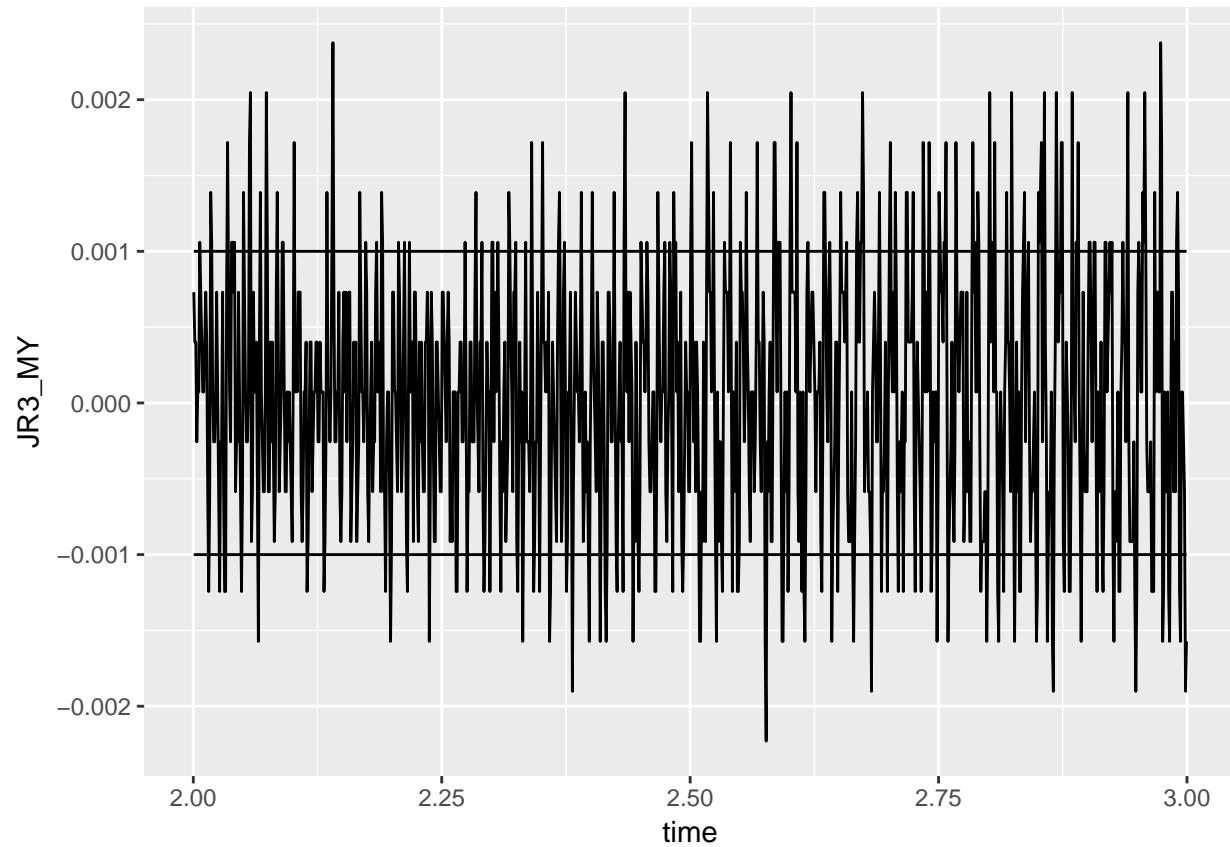


```
time_series_to_harmonics_psd(one_second_sample[, 'JR3_MX'], acq_freq=1000, main="JR3_MX", xlim=c(0,100))
```

### JR3\_MX

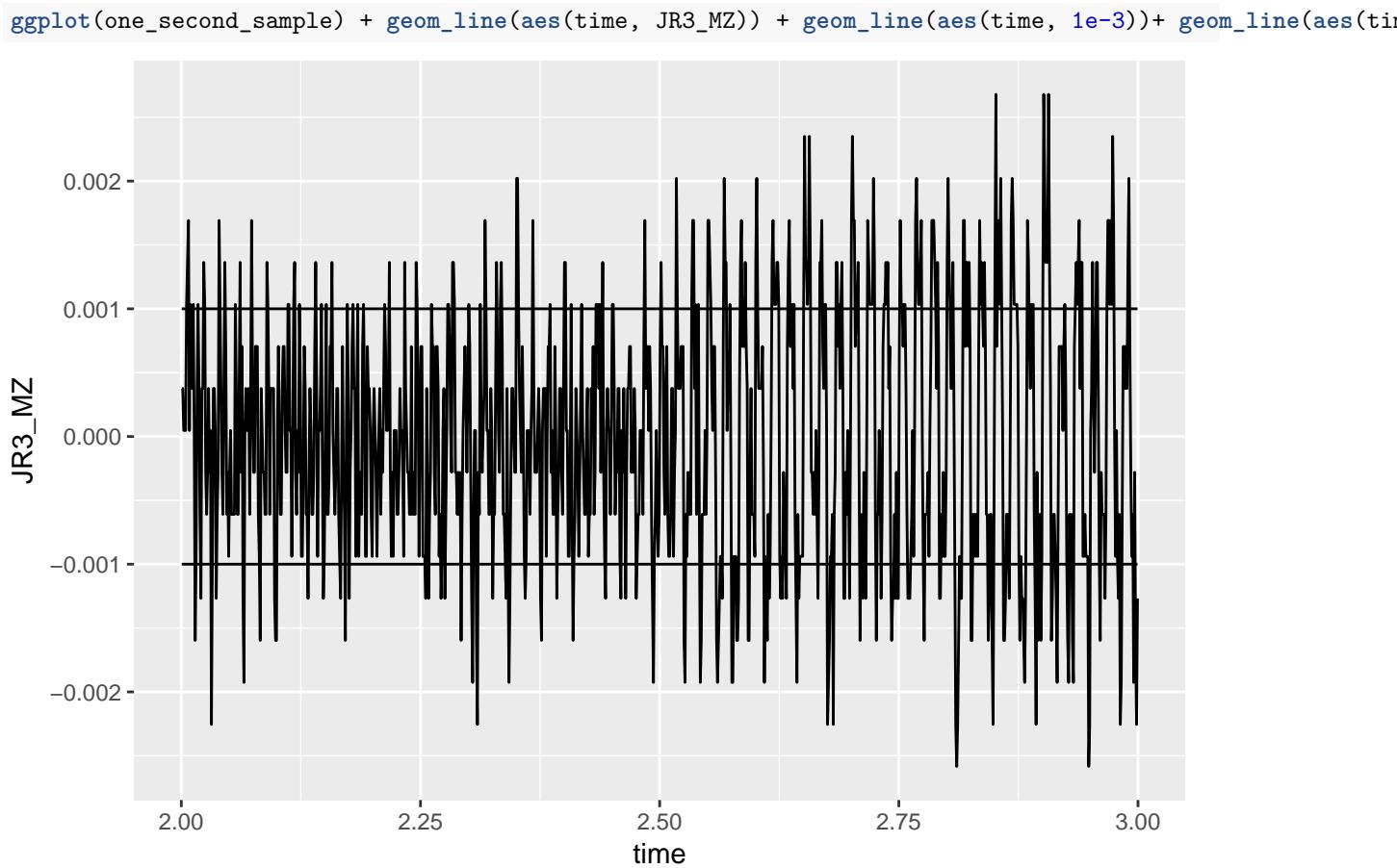
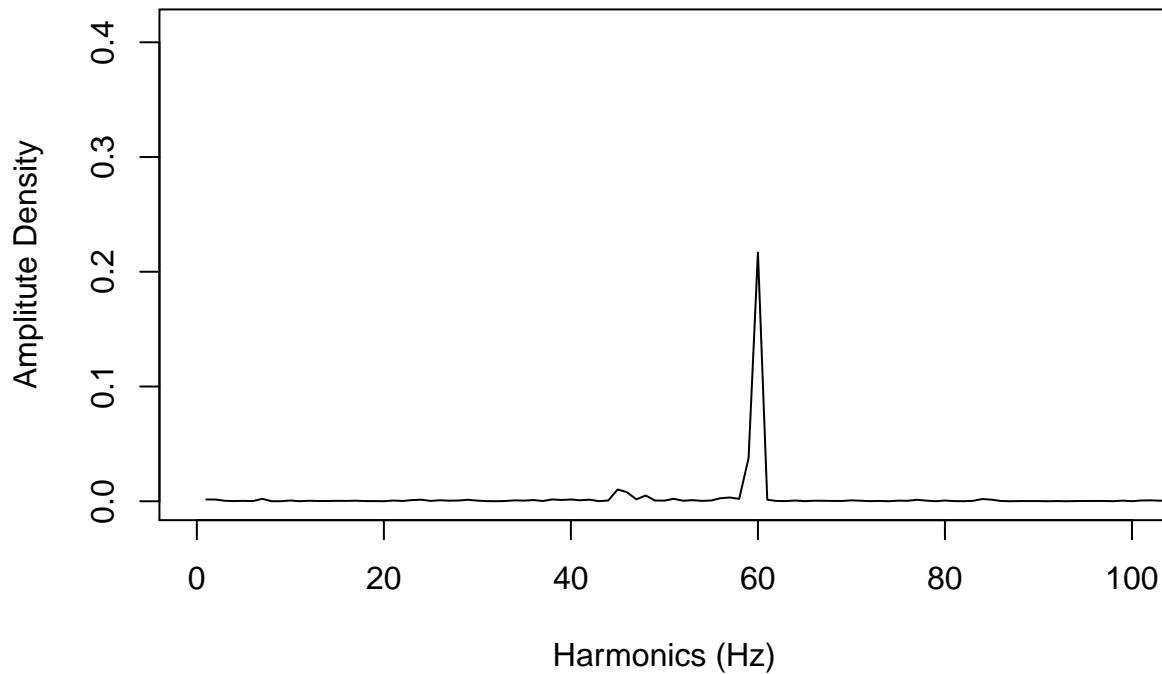


```
ggplot(one_second_sample) + geom_line(aes(time, JR3_MY)) + geom_line(aes(time, 1e-3))+ geom_line(aes(ti
```



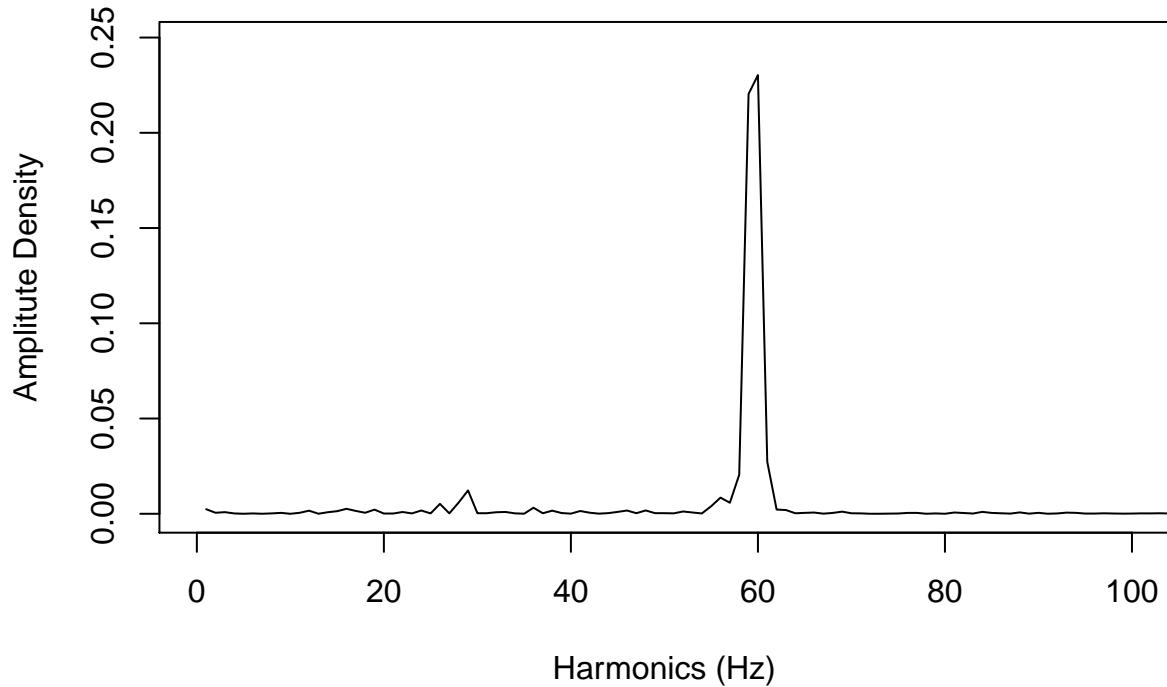
```
time_series_to_harmonics_psd(one_second_sample[, 'JR3_MY'], acq_freq=1000, main="JR3_MY", xlim=c(0,100))
```

## JR3\_MY



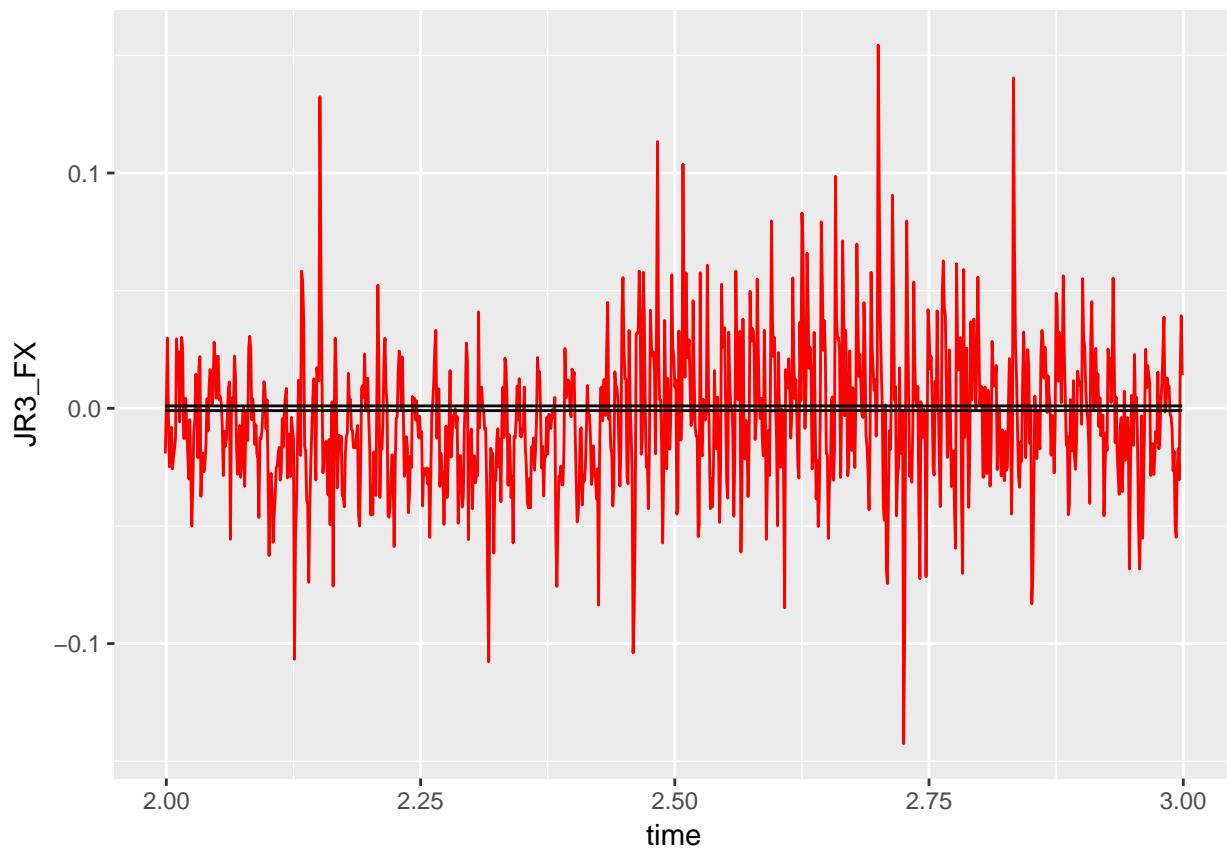
```
time_series_to_harmonics_psd(one_second_sample[, 'JR3_MZ'], acq_freq=1000, main="JR3_MZ", xlim=c(0,100))
```

**JR3\_MZ**



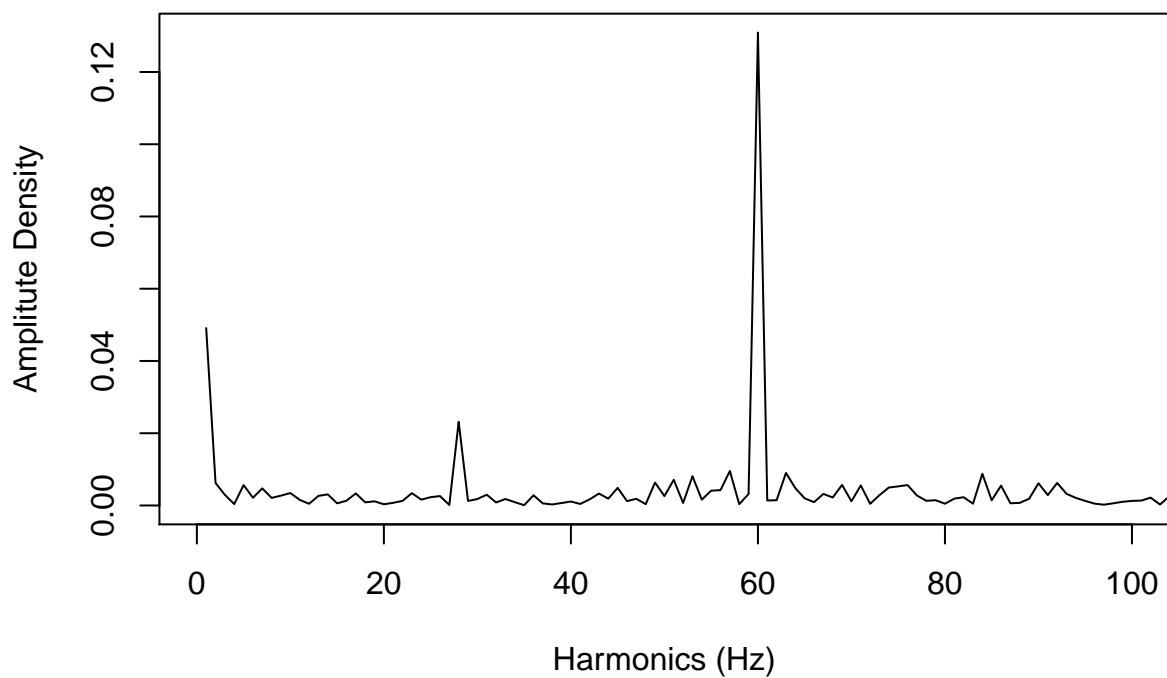
### Calibrated & Mean-Centered Akira-JR3 Voltages

```
one_second_sample_akira_cal <- as.data.frame(cbind(time = akira_jr3_rest$time, center_scale(just_jr3_co...  
ggplot(one_second_sample_akira_cal) + geom_line(aes(time, JR3_FX), col="red") + geom_line(aes(time, 1e-...
```

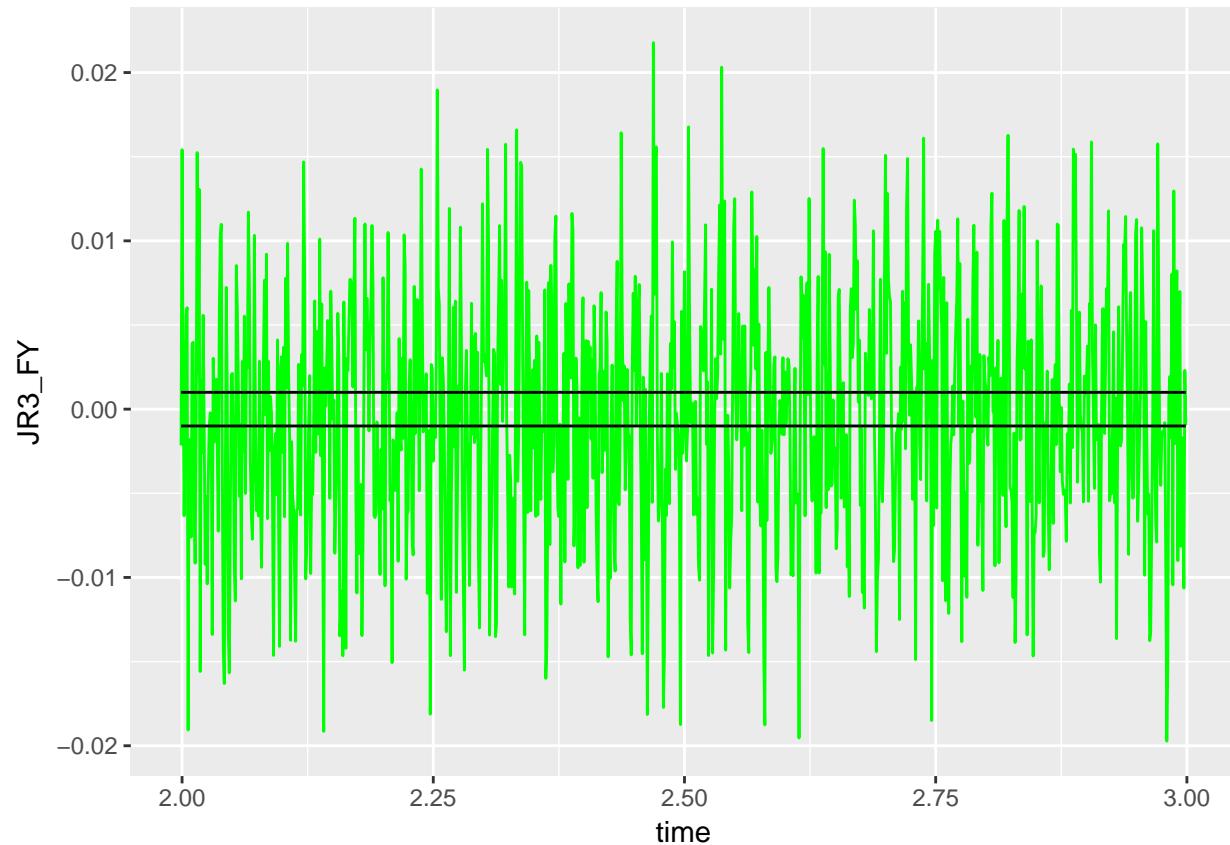


```
time_series_to_harmonics_psd(one_second_sample_akira_cal[, "JR3_FX"], acq_freq=1000, main="JR3_FX", xlim=
```

### JR3\_FX

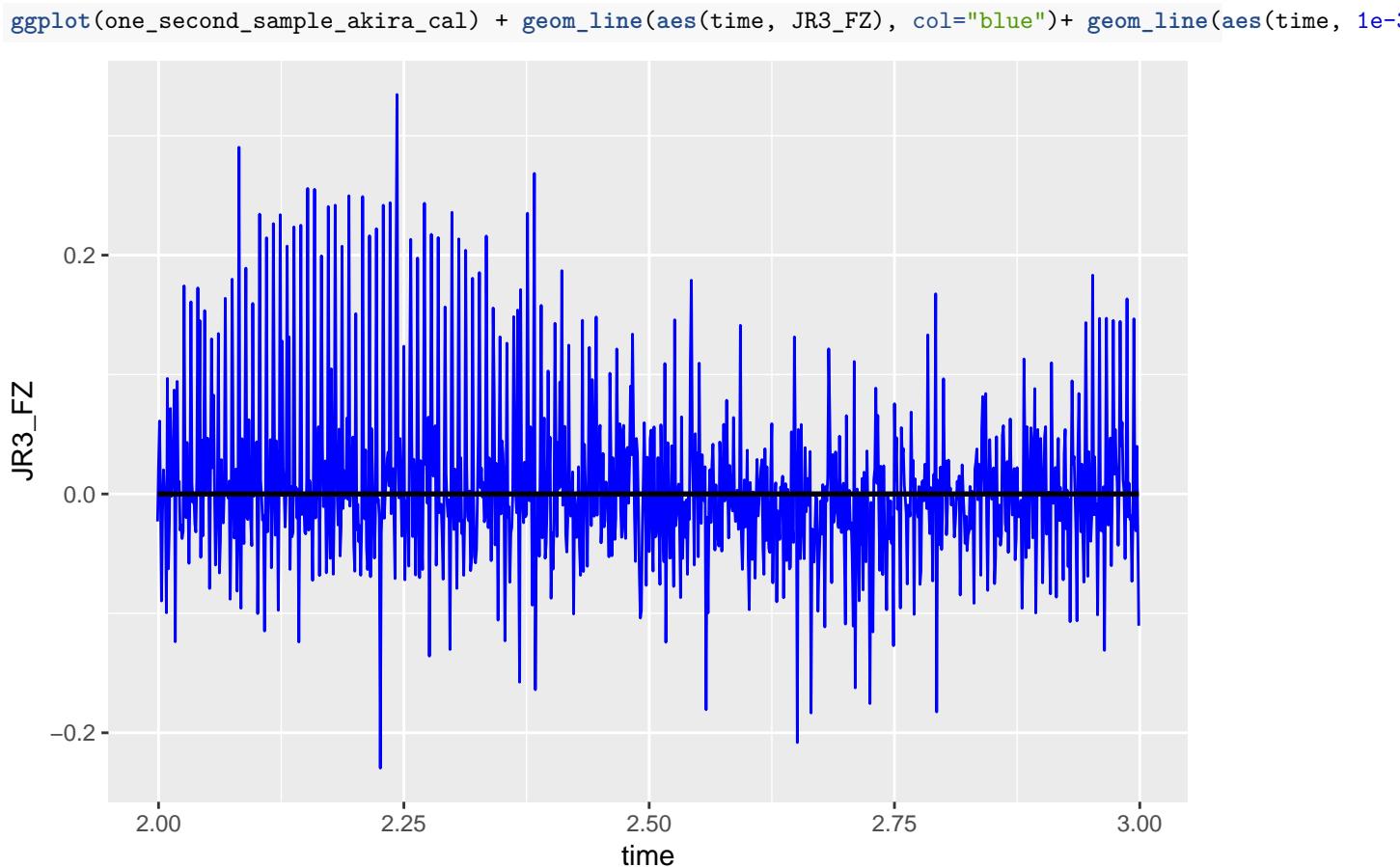
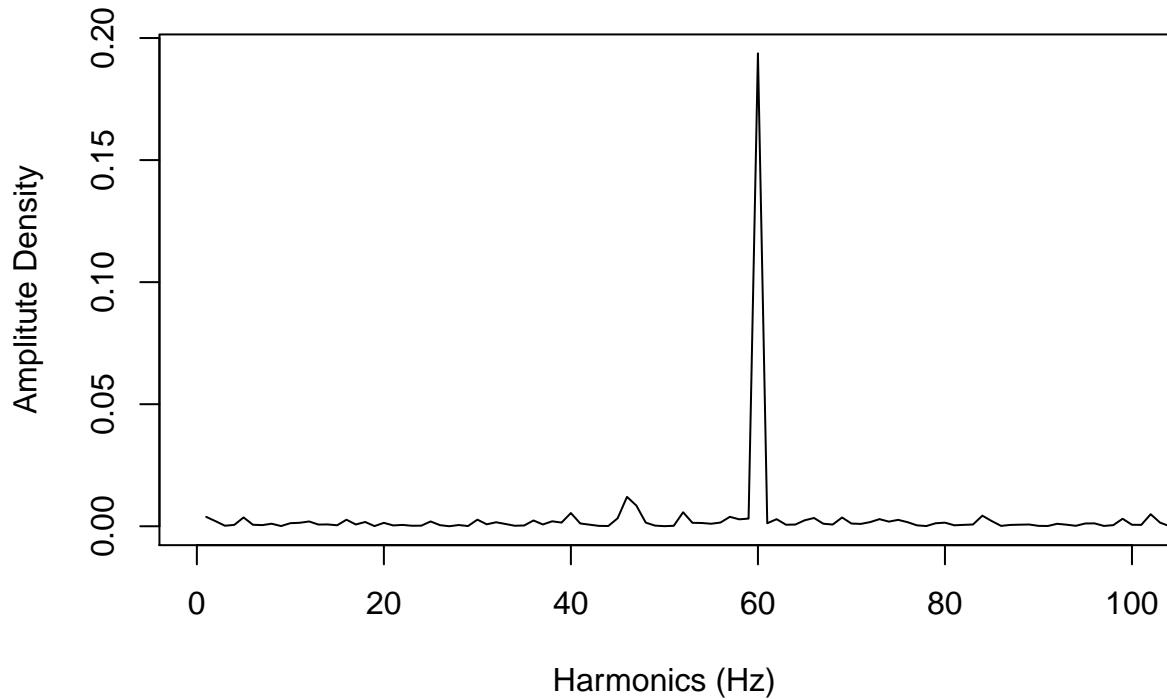


```
ggplot(one_second_sample_akira_cal) + geom_line(aes(time, JR3_FY), col="green") + geom_line(aes(time, 1e-05))
```



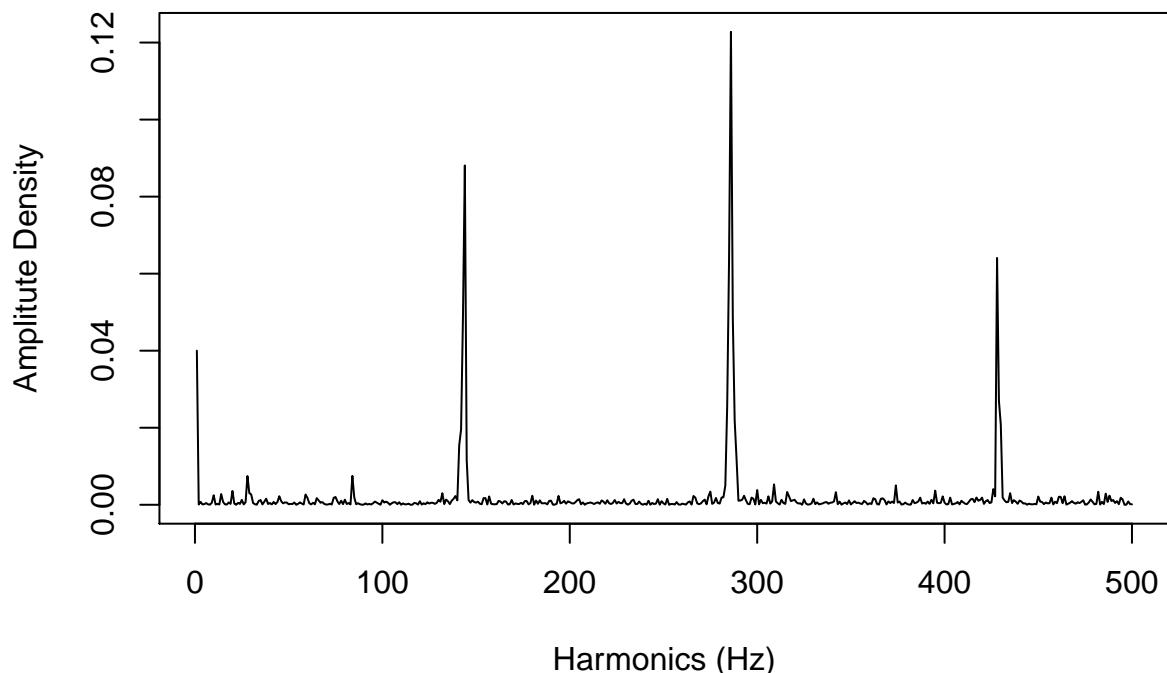
```
time_series_to_harmonics_psd(one_second_sample_akira_cal[, 'JR3_FY'], acq_freq=1000, main="JR3_FY", xlim=c(2.0, 3.0))
```

## JR3\_FY

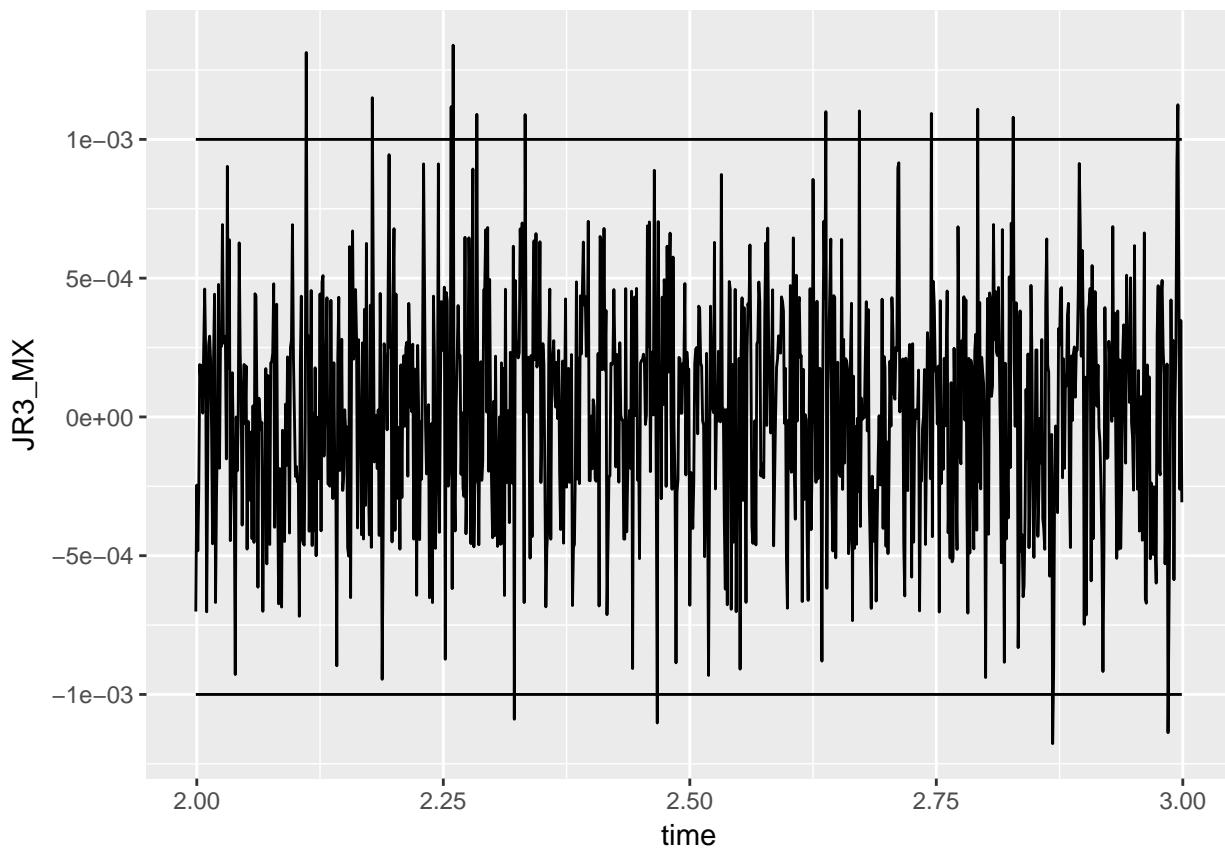


```
time_series_to_harmonics_psd(one_second_sample_akira_cal[, 'JR3_FZ'], acq_freq=1000, main="JR3_FZ")
```

JR3\_FZ

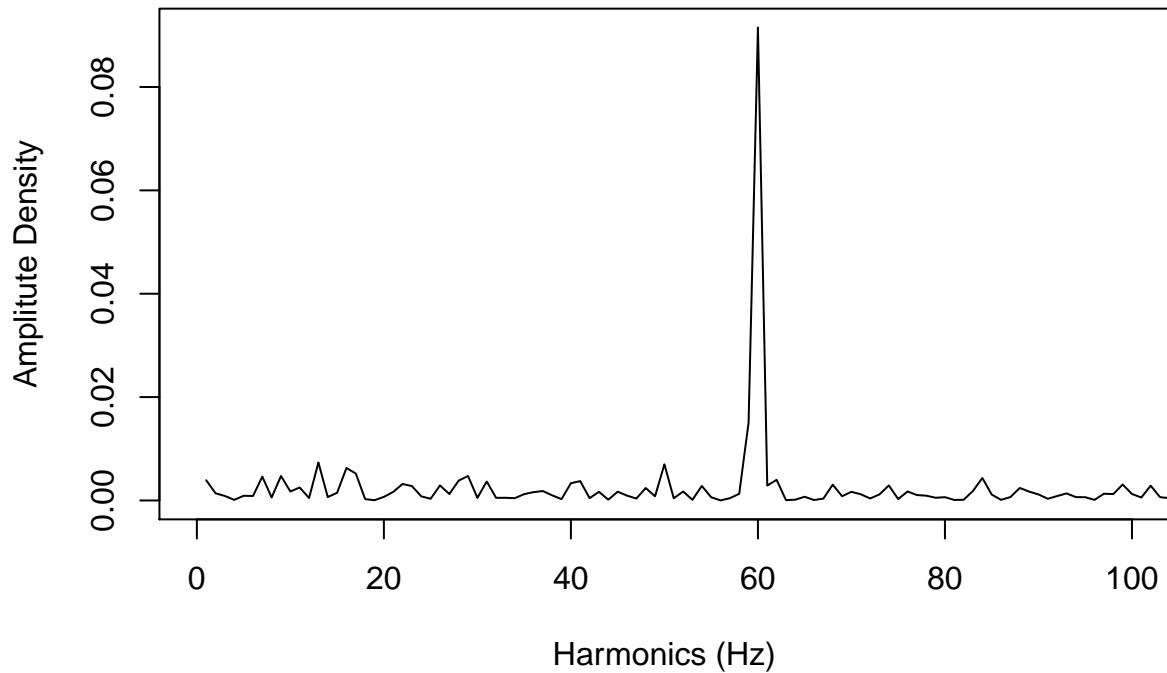


```
ggplot(one_second_sample_akira_cal) + geom_line(aes(time, JR3_MX)) + geom_line(aes(time, 1e-3))+ geom_1
```

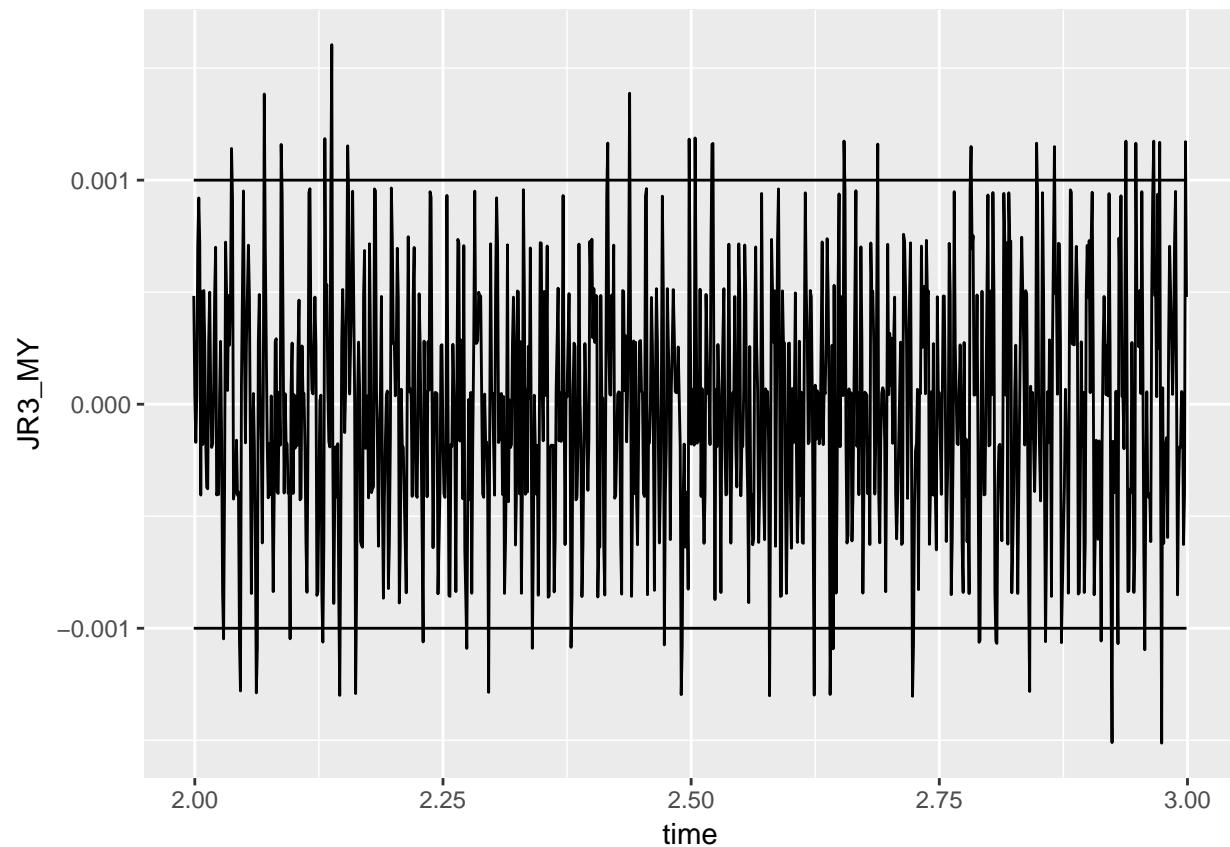


```
time_series_to_harmonics_psd(one_second_sample_akira_cal[, "JR3_MX"], acq_freq=1000, main="JR3_MX", xlim=[2.0, 3.0])
```

**JR3\_MX**

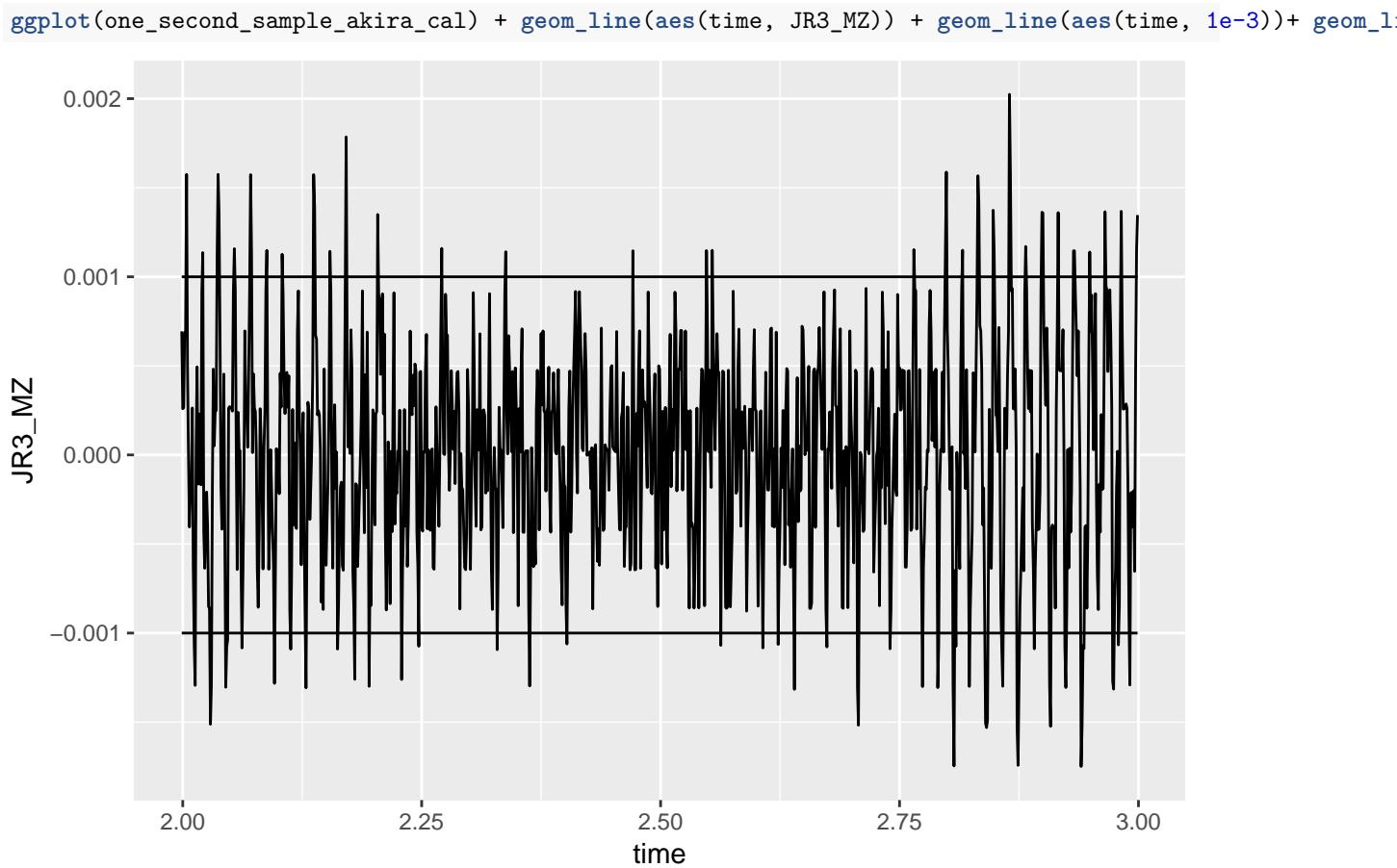
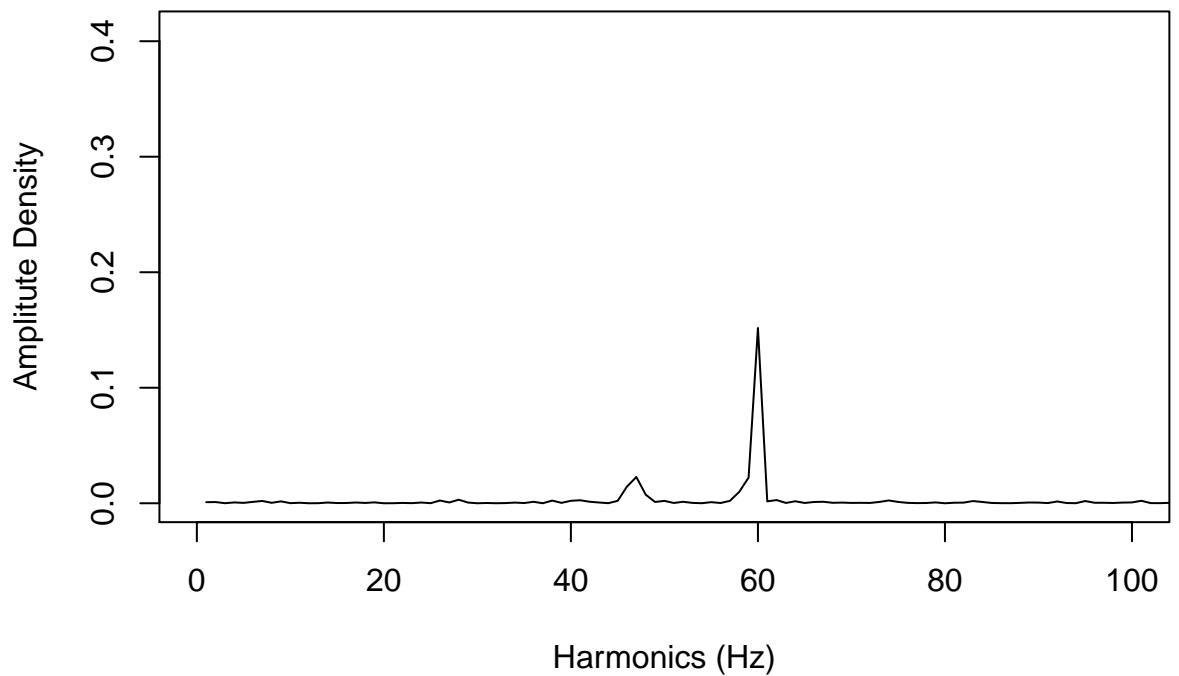


```
ggplot(one_second_sample_akira_cal) + geom_line(aes(time, JR3_MY)) + geom_line(aes(time, 1e-3))+ geom_
```



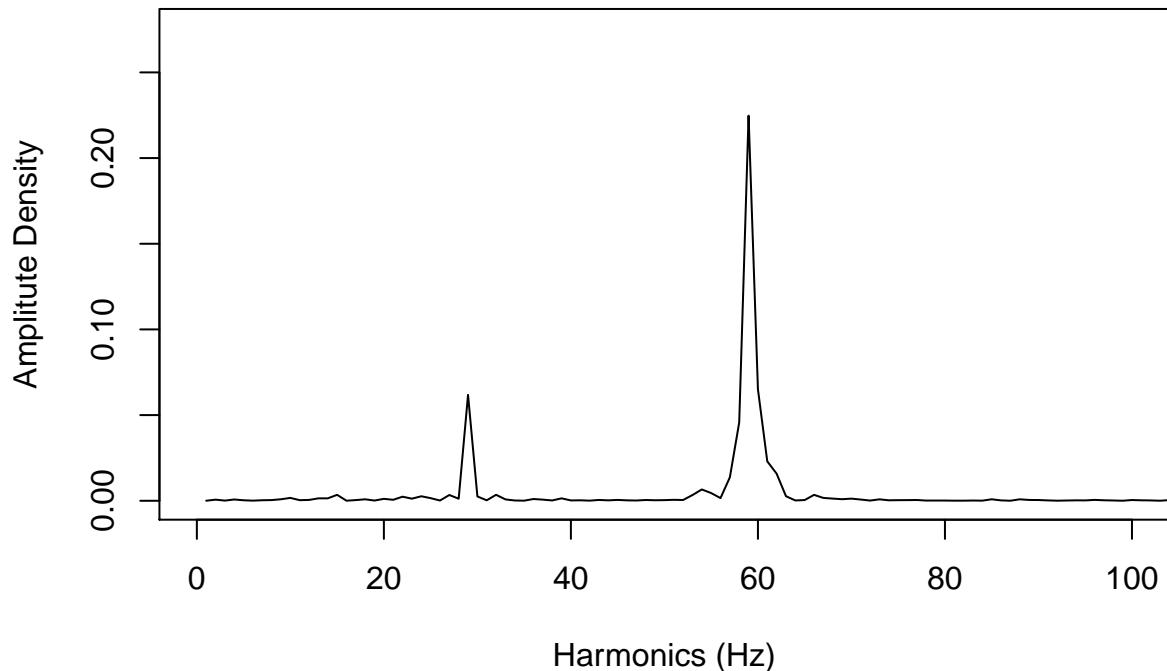
```
time_series_to_harmonics_psd(one_second_sample_akira_cal[, 'JR3_MY'], acq_freq=1000, main="JR3_MY", xlim
```

## JR3\_MY



```
time_series_to_harmonics_psd(one_second_sample_akira_cal[, 'JR3_MZ'], acq_freq=1000, main="JR3_MZ", xlim
```

## JR3\_MZ

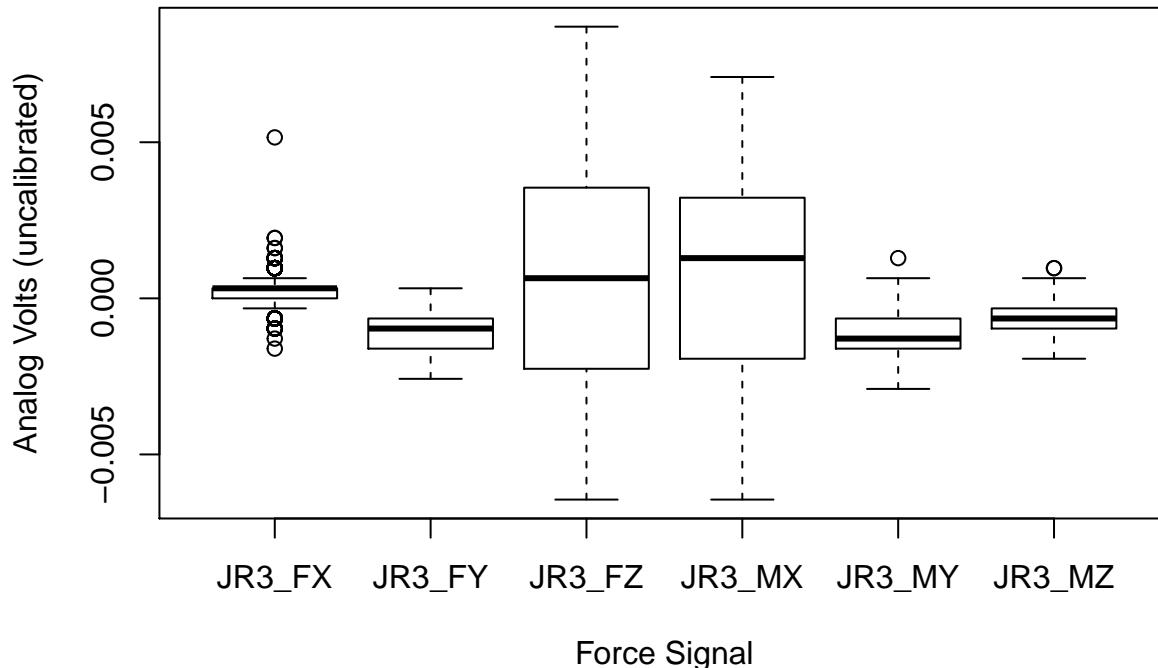


## Basement JR3 Signal SysID

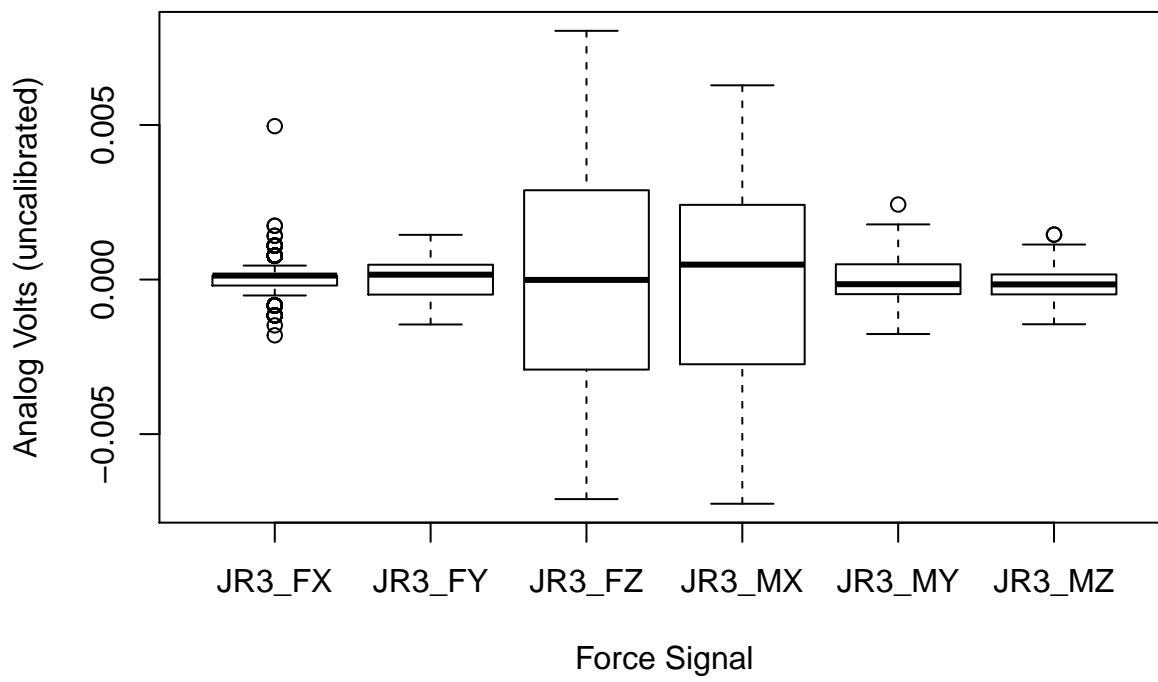
```
##      JR3_FX          JR3_FY          JR3_FZ
##  Min. :-0.0016110  Min. :-0.002578  Min. :-0.0064460
##  1st Qu.: 0.0000000  1st Qu.:-0.001611  1st Qu.:-0.0022560
##  Median : 0.0003220  Median :-0.000967  Median : 0.0006450
##  Mean   : 0.0001931  Mean  :-0.001125  Mean   : 0.0006559
##  3rd Qu.: 0.0003220  3rd Qu.:-0.000645  3rd Qu.: 0.0035450
##  Max.   : 0.0051560  Max.  : 0.000322  Max.   : 0.0087020
##      JR3_MX          JR3_MY          JR3_MZ
##  Min. :-0.0064460  Min. :-0.002901  Min. :-0.0019340
##  1st Qu.:-0.0019340 1st Qu.:-0.001611  1st Qu.:-0.0009670
##  Median : 0.0012890  Median :-0.001289  Median :-0.0006450
##  Mean   : 0.0008061  Mean  :-0.001139  Mean   :-0.0004879
##  3rd Qu.: 0.0032230  3rd Qu.:-0.000645  3rd Qu.:-0.0003220
##  Max.   : 0.0070900  Max.  : 0.001289  Max.   : 0.0009670
```

## BasementJR3 Before and after calibration &/or mean-centering

```
boxplot(just_jr3_cols, xlab="Force Signal", ylab="Analog Volts (uncalibrated)")
```

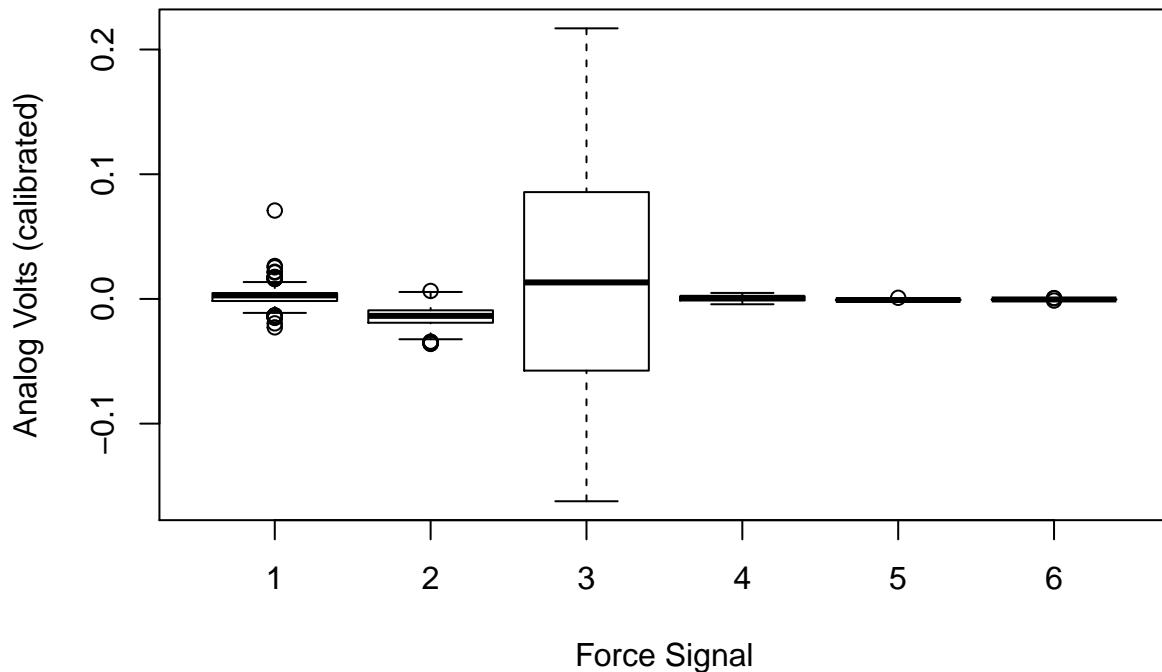


```
boxplot(scale(just_jr3_cols, scale=FALSE), xlab="Force Signal", ylab="Analog Volts (uncalibrated)")
```



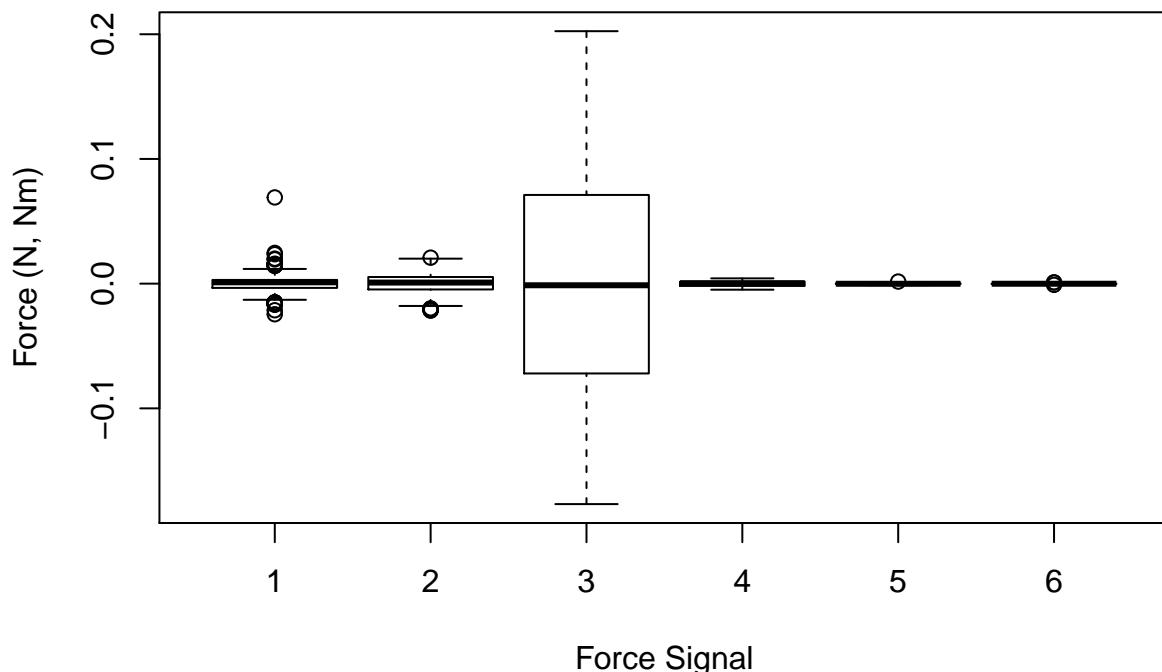
```
boxplot(jr3_cols_calibrated, xlab="Force Signal", ylab="Analog Volts (calibrated)", main="JR3 Calibrated")
```

## JR3 Calibrated



```
boxplot(scale(jr3_cols_calibrated, scale=FALSE), xlab="Force Signal", ylab="Force (N, Nm)", main="JR3 C")
```

## JR3 Calibrated & Mean-Centered



```
##Snapshots of BasementJR3 force signals at rest, mean-centered, uncalibrated
```

```
mean_centered_jr3 <- as.data.frame(scale(just_jr3_cols, scale=FALSE))
colnames(mean_centered_jr3) <- dots_to_underscores(force_column_names)
```

```

mean_centered_jr3 <- cbind(time=jr3_test_no_input$time, mean_centered_jr3)
one_second_sample <- mean_centered_jr3[mean_centered_jr3$time < 3.0 & mean_centered_jr3$time > 2.0,]
summary(one_second_sample)

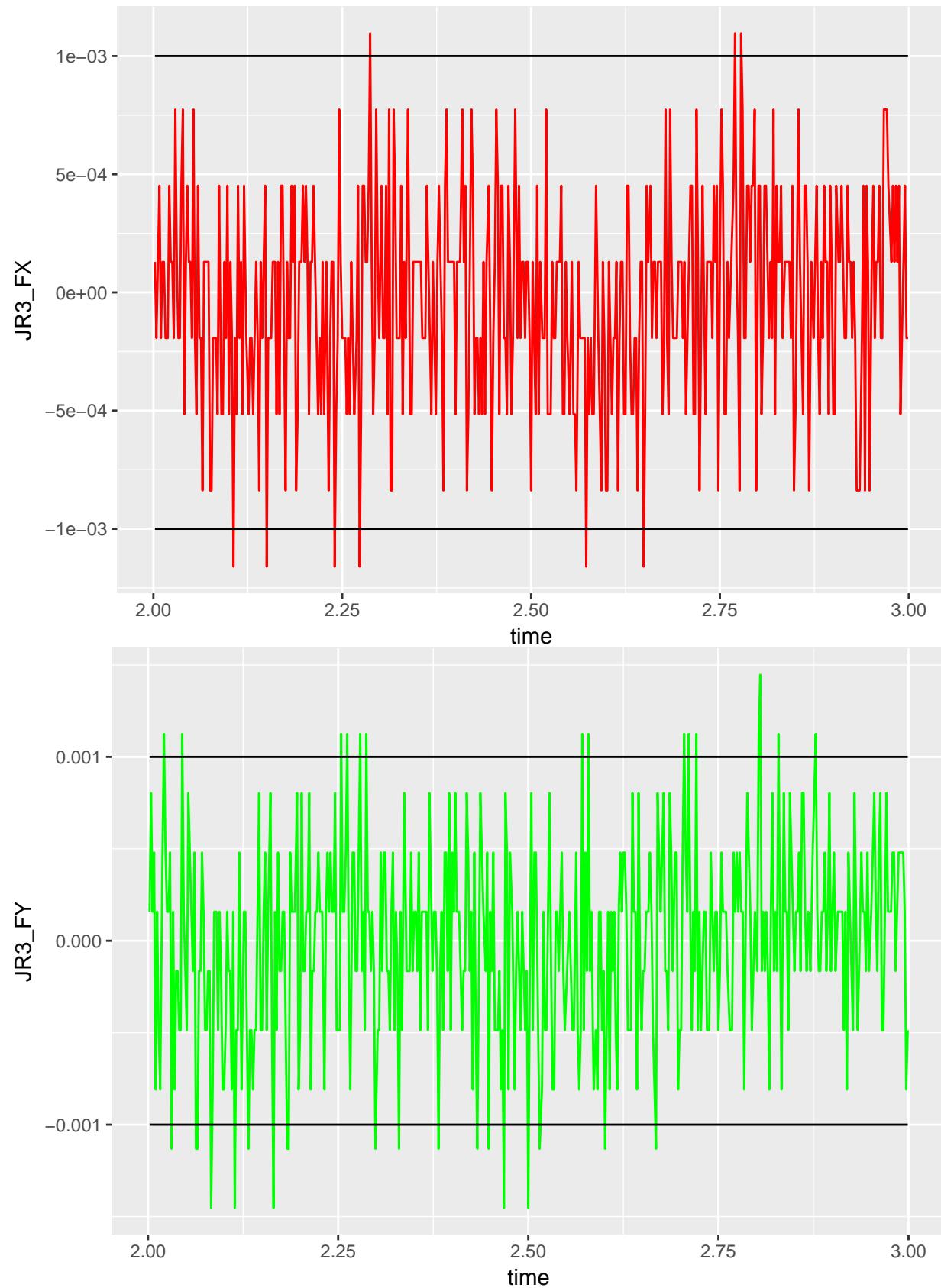
##          time      JR3_FX      JR3_FY
##  Min.   :2.002   Min.   :-1.160e-03   Min.   :-1.453e-03
##  1st Qu.:2.249   1st Qu.:-1.931e-04   1st Qu.:-4.861e-04
##  Median :2.498   Median : 1.289e-04   Median : 1.579e-04
##  Mean   :2.499   Mean   :-3.833e-05   Mean   :-6.843e-06
##  3rd Qu.:2.749   3rd Qu.: 1.289e-04   3rd Qu.: 4.799e-04
##  Max.   :2.999   Max.   : 1.096e-03   Max.   : 1.447e-03
##          JR3_FZ      JR3_MX      JR3_MY
##  Min.   :-6.457e-03   Min.   :-7.252e-03   Min.   :-1.762e-03
##  1st Qu.:-2.912e-03   1st Qu.:-2.740e-03   1st Qu.:-4.719e-04
##  Median :-1.092e-05   Median : 4.829e-04   Median :-1.499e-04
##  Mean   :-2.680e-05   Mean   :-3.102e-05   Mean   :-1.612e-05
##  3rd Qu.: 2.889e-03   3rd Qu.: 2.256e-03   3rd Qu.: 4.941e-04
##  Max.   : 7.079e-03   Max.   : 5.962e-03   Max.   : 1.784e-03
##          JR3_MZ
##  Min.   :-1.123e-03
##  1st Qu.:-1.571e-04
##  Median :-1.571e-04
##  Mean   : 6.768e-06
##  3rd Qu.: 1.659e-04
##  Max.   : 1.455e-03

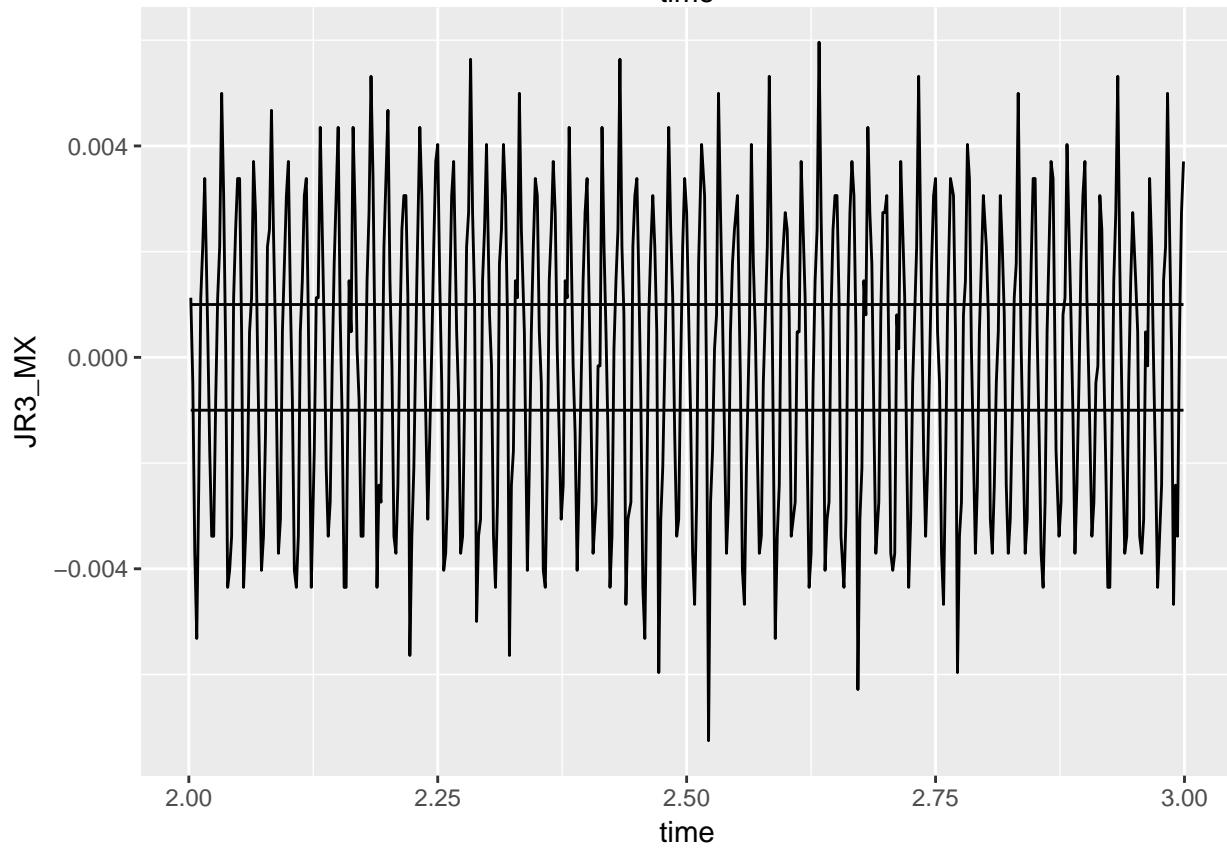
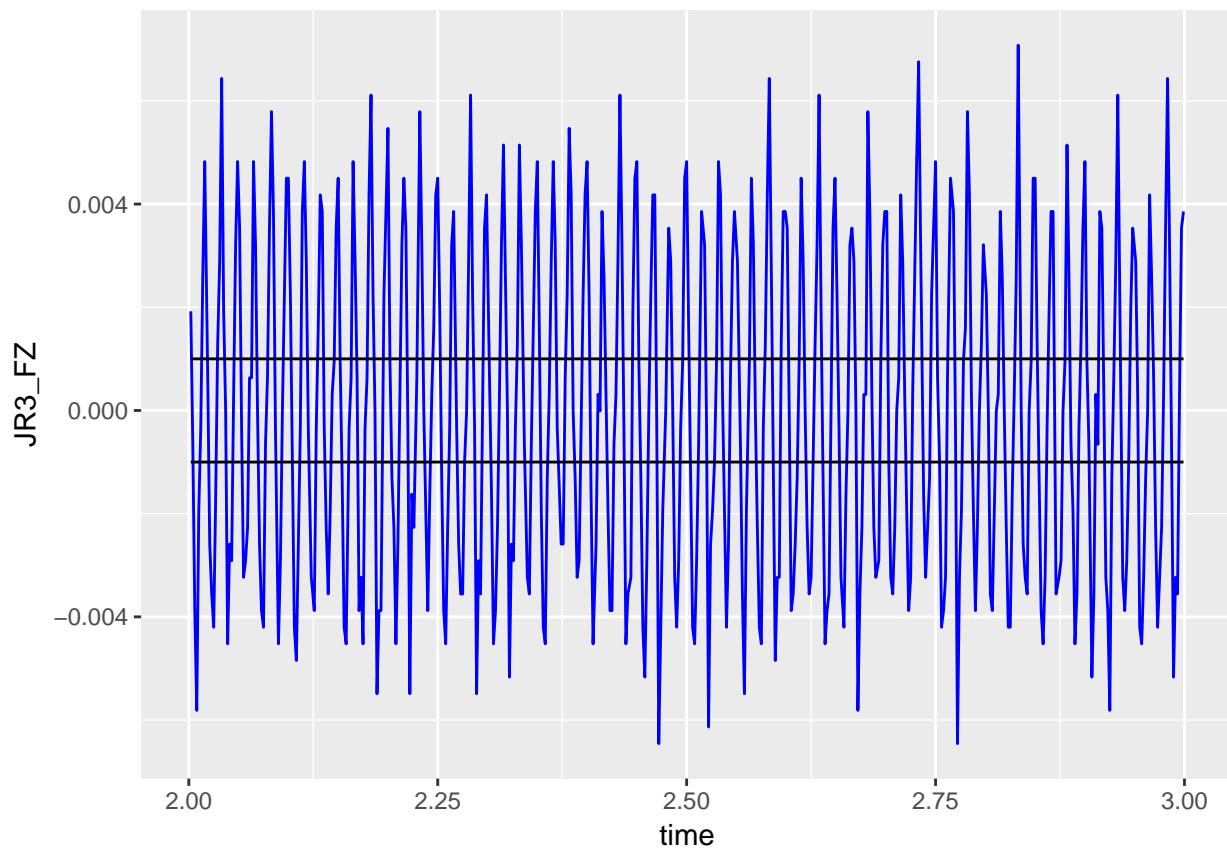
column_ranges(one_second_sample)

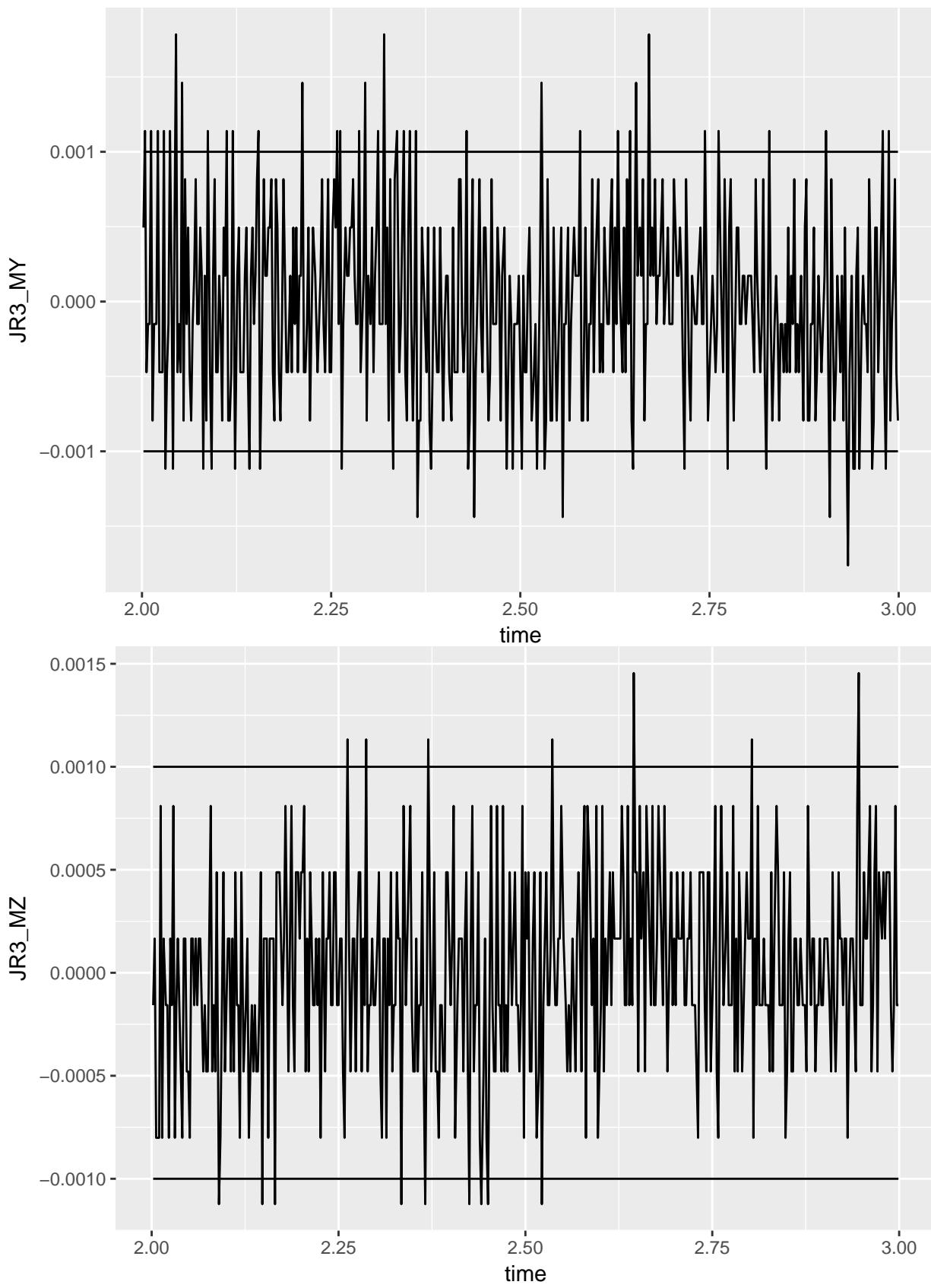
##             min         max
## time    2.002000000 2.999000000
## JR3_FX -0.001160087 0.001095913
## JR3_FY -0.001453140 0.001446860
## JR3_FZ -0.006456921 0.007079079
## JR3_MX -0.007252115 0.005961885
## JR3_MY -0.001761886 0.001784114
## JR3_MZ -0.001123098 0.001454902

```

### Mean-Centered Basement JR3 Voltages







##Summary of calibrated basement JR3 Forces (Newtons, N\*m)

```

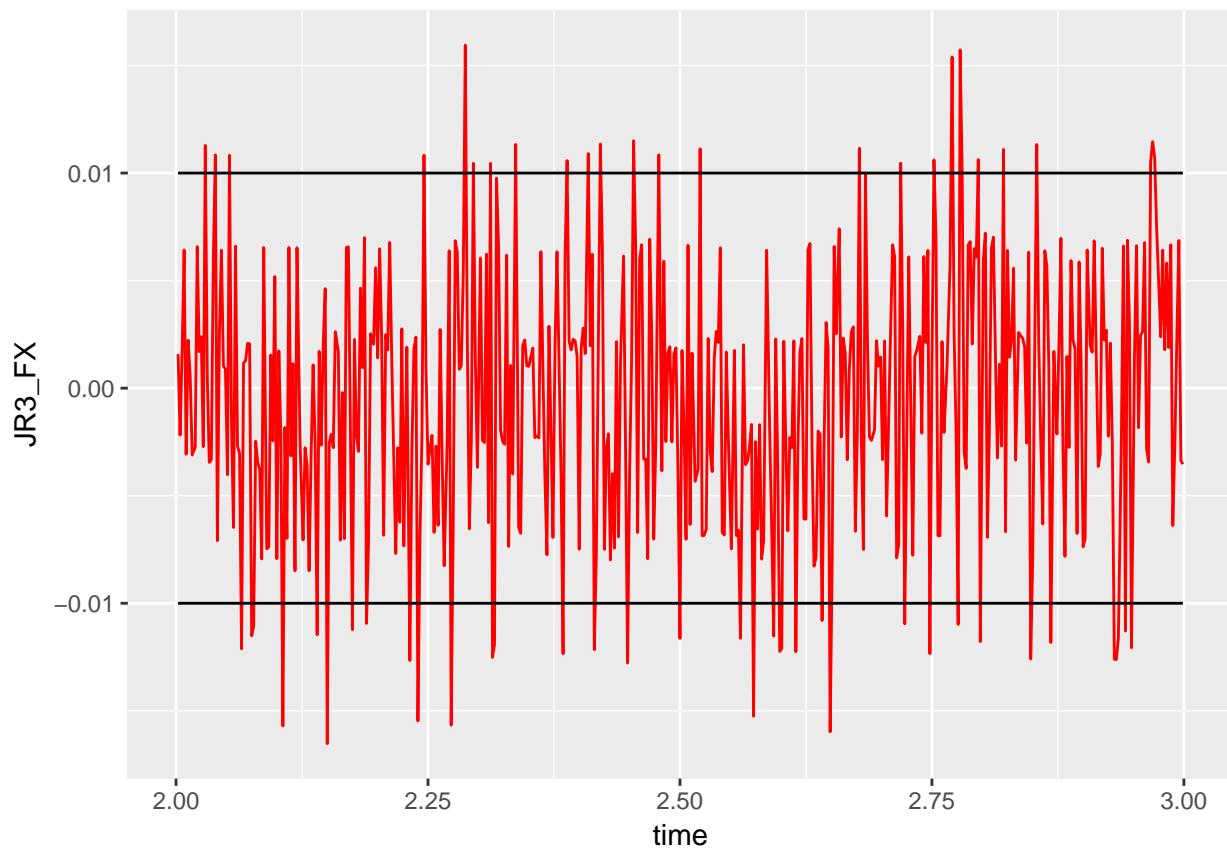
##      time          JR3_FX          JR3_FY
##  Min.   :2.002   Min.   :-0.0165256   Min.   :-2.163e-02
##  1st Qu.:2.249   1st Qu.:-0.0038236   1st Qu.:-4.634e-03
##  Median :2.498   Median : 0.0009173   Median : 8.234e-04
##  Mean   :2.499   Mean   :-0.0005150   Mean   :-8.235e-05
##  3rd Qu.:2.749   3rd Qu.: 0.0026141   3rd Qu.: 4.781e-03
##  Max.   :2.999   Max.   : 0.0159441   Max.   : 2.084e-02
##      JR3_FZ          JR3_MX          JR3_MY
##  Min.   :-0.1579035   Min.   :-4.846e-03   Min.   :-1.206e-03
##  1st Qu.:-0.0705616   1st Qu.:-1.835e-03   1st Qu.:-3.213e-04
##  Median :-0.0019114   Median : 3.071e-04   Median :-9.685e-05
##  Mean   :-0.0006729   Mean   :-2.088e-05   Mean   :-1.132e-05
##  3rd Qu.: 0.0699325   3rd Qu.: 1.527e-03   3rd Qu.: 3.330e-04
##  Max.   : 0.1769798   Max.   : 4.035e-03   Max.   : 1.223e-03
##      JR3_MZ
##  Min.   :-7.653e-04
##  1st Qu.:-1.361e-04
##  Median :-8.001e-05
##  Mean   : 4.223e-06
##  3rd Qu.: 1.288e-04
##  Max.   : 9.804e-04

##             min           max
## time    2.0020000000 2.9990000000
## JR3_FX -0.0165255558 0.0159440832
## JR3_FY -0.0216262512 0.0208428508
## JR3_FZ -0.1579034734 0.1769798326
## JR3_MX -0.0048463191 0.0040354559
## JR3_MY -0.0012055880 0.0012230990
## JR3_MZ -0.0007652588 0.0009803802

```

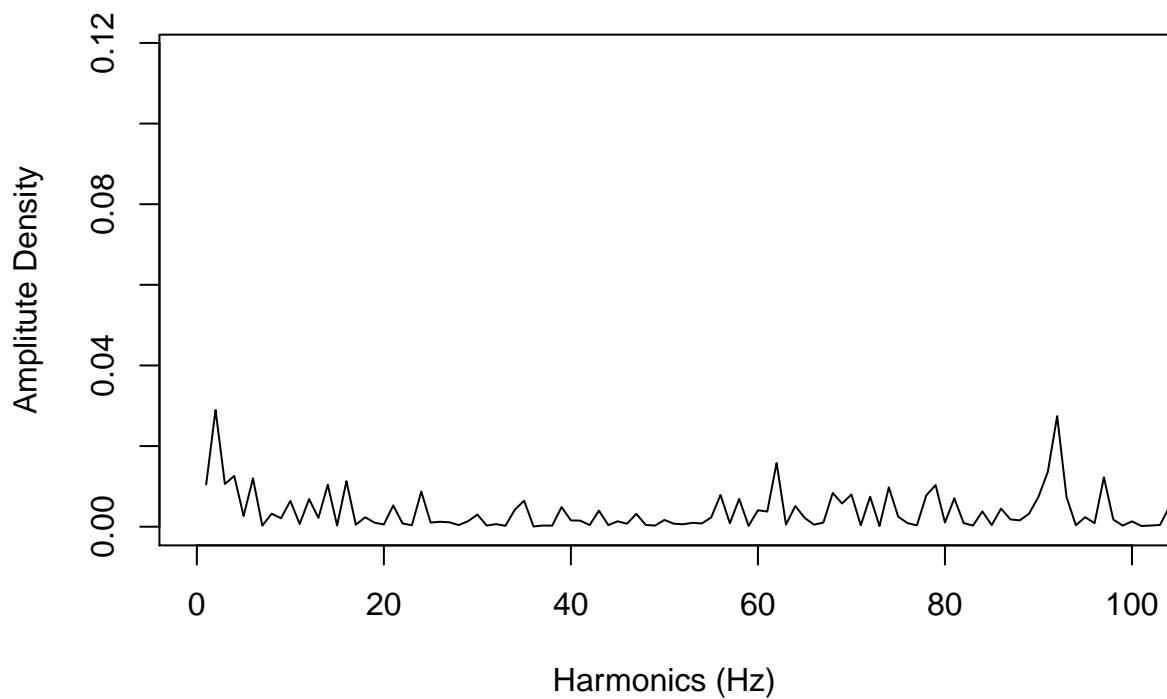
## Calibrated and Mean-Centered Basement JR3 Forces

```
ggplot(one_second_sample_cal) + geom_line(aes(time, JR3_FX), col="red") + geom_line(aes(time, 1e-2)) + geom
```

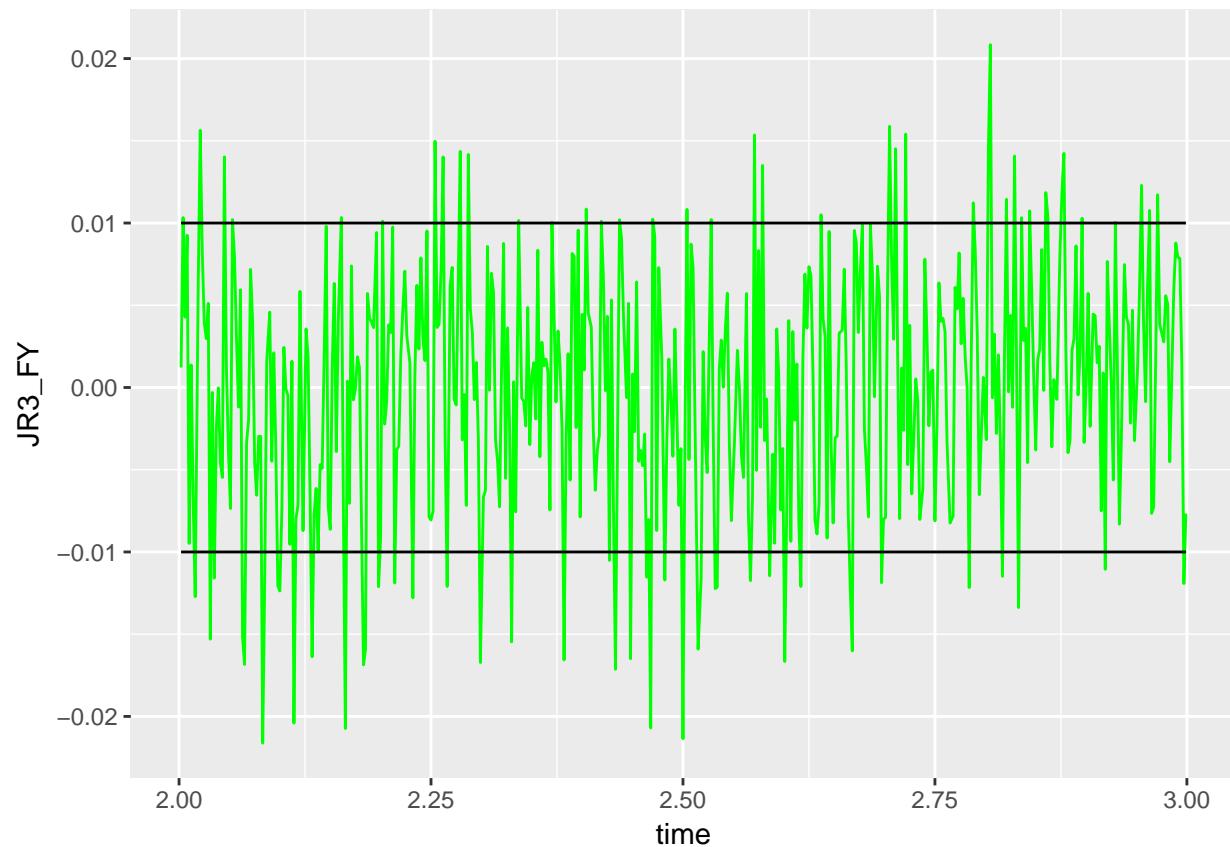


```
time_series_to_harmonics_psd(one_second_sample_cal[, 'JR3_FX'], acq_freq=1000, main="JR3_FX", xlim=c(0,100))
```

### JR3\_FX

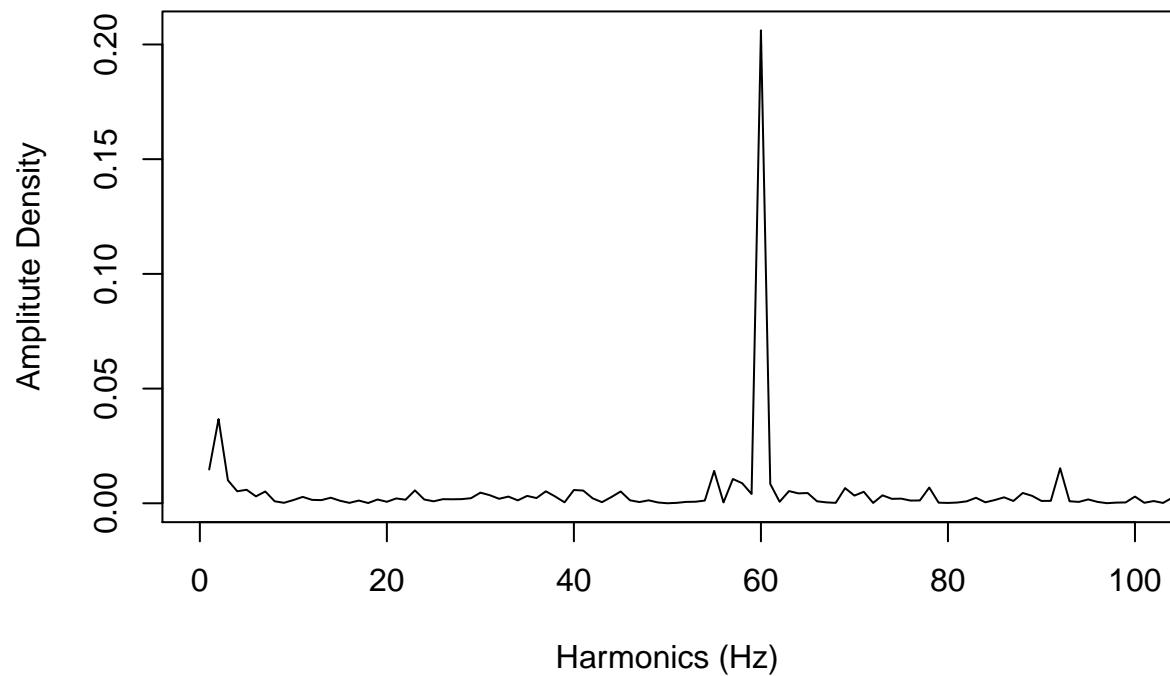


```
ggplot(one_second_sample_cal) + geom_line(aes(time, JR3_FY), col="green") + geom_line(aes(time, 1e-2)) +
```

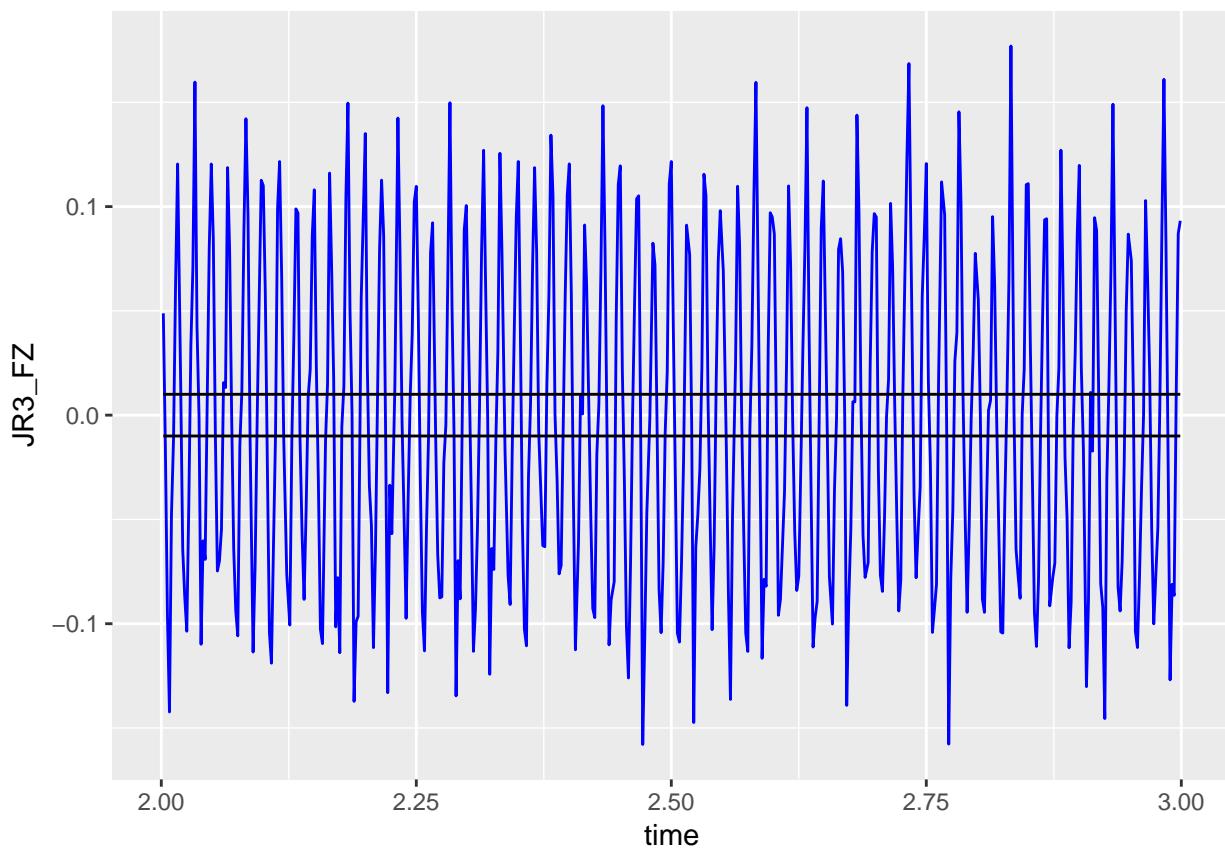


```
time_series_to_harmonics_psd(one_second_sample_cal[, 'JR3_FY'], acq_freq=1000, main="JR3_FY", xlim=c(0, 10))
```

## JR3\_FY

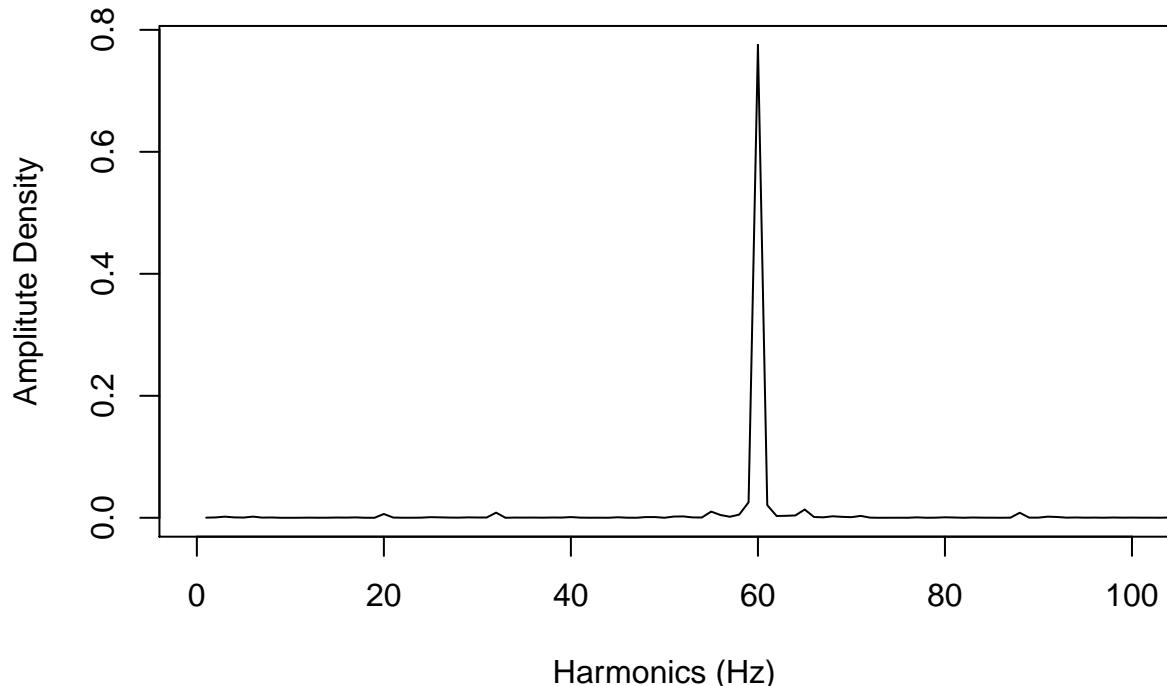


```
ggplot(one_second_sample_cal) + geom_line(aes(time, JR3_FZ), col="blue") + geom_line(aes(time, 1e-2)) + g
```

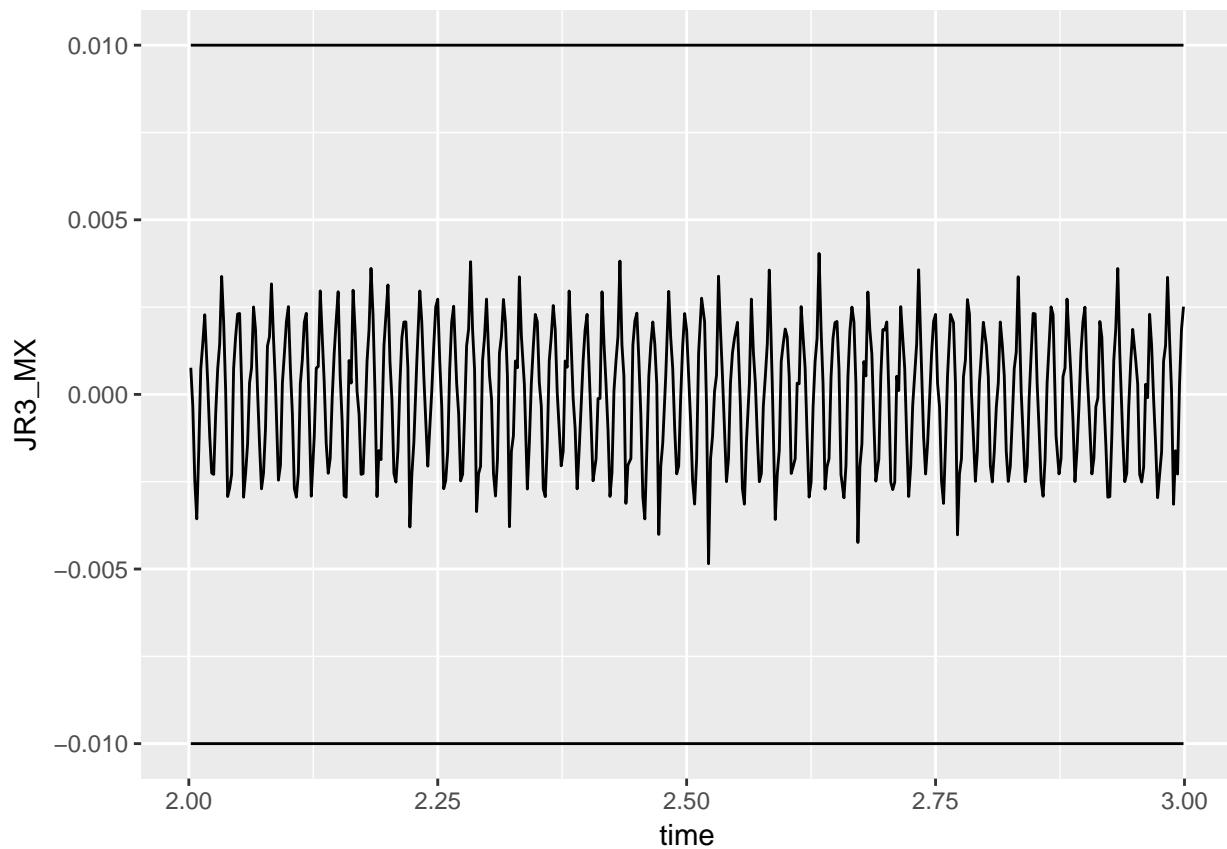


```
time_series_to_harmonics_psd(one_second_sample_cal[, 'JR3_FZ'], acq_freq=1000, main="JR3_FZ", xlim=c(0,100))
```

**JR3\_FZ**

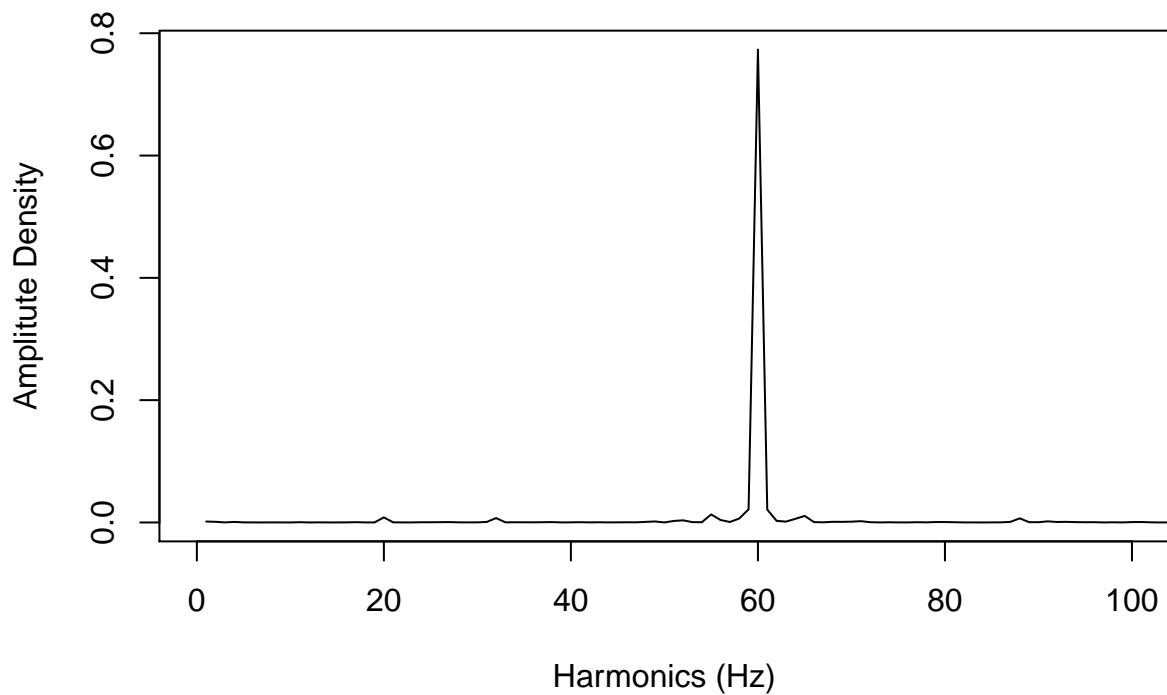


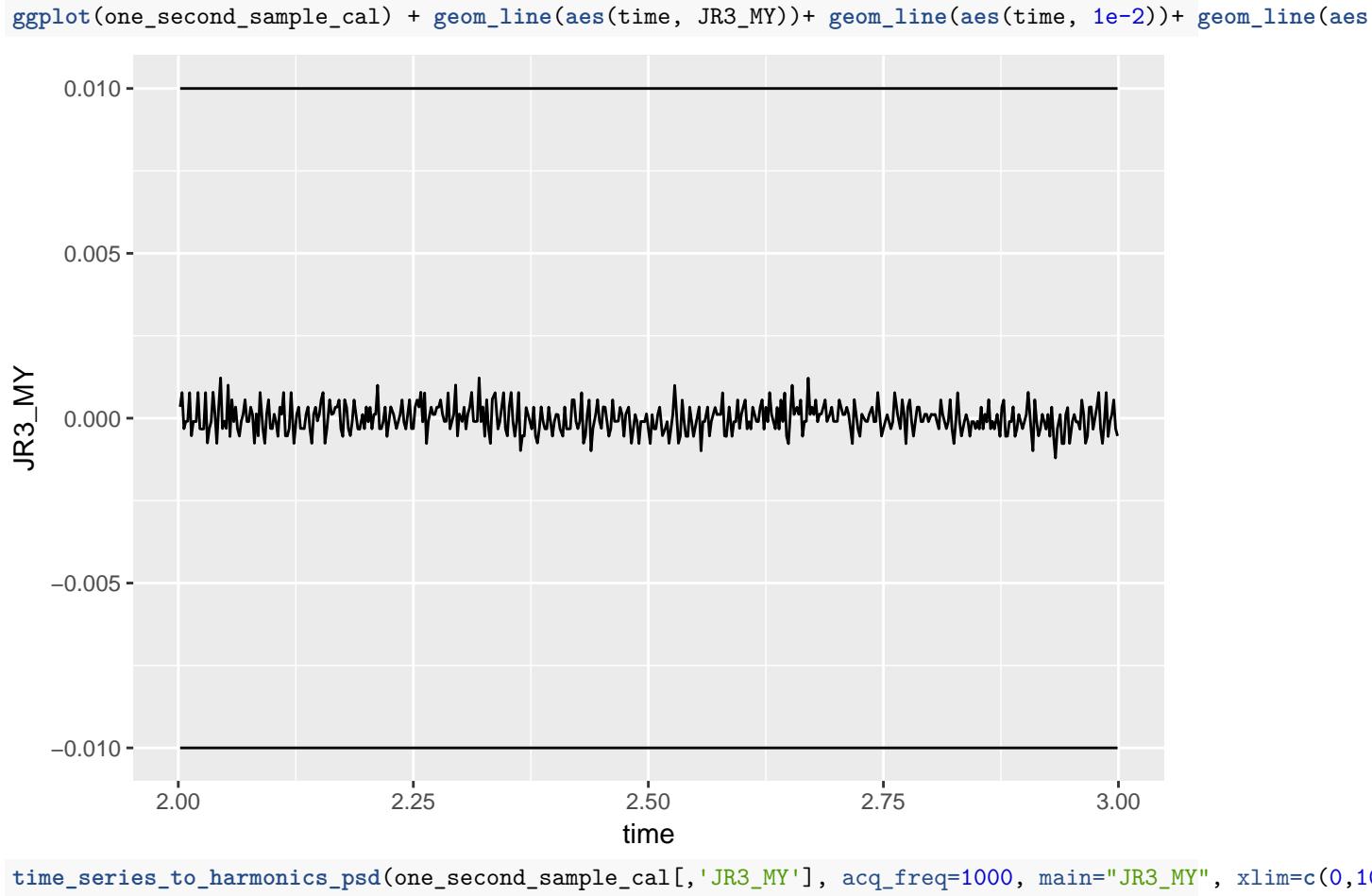
```
ggplot(one_second_sample_cal) + geom_line(aes(time, JR3_MX))+ geom_line(aes(time, 1e-2))+ geom_line(aes
```



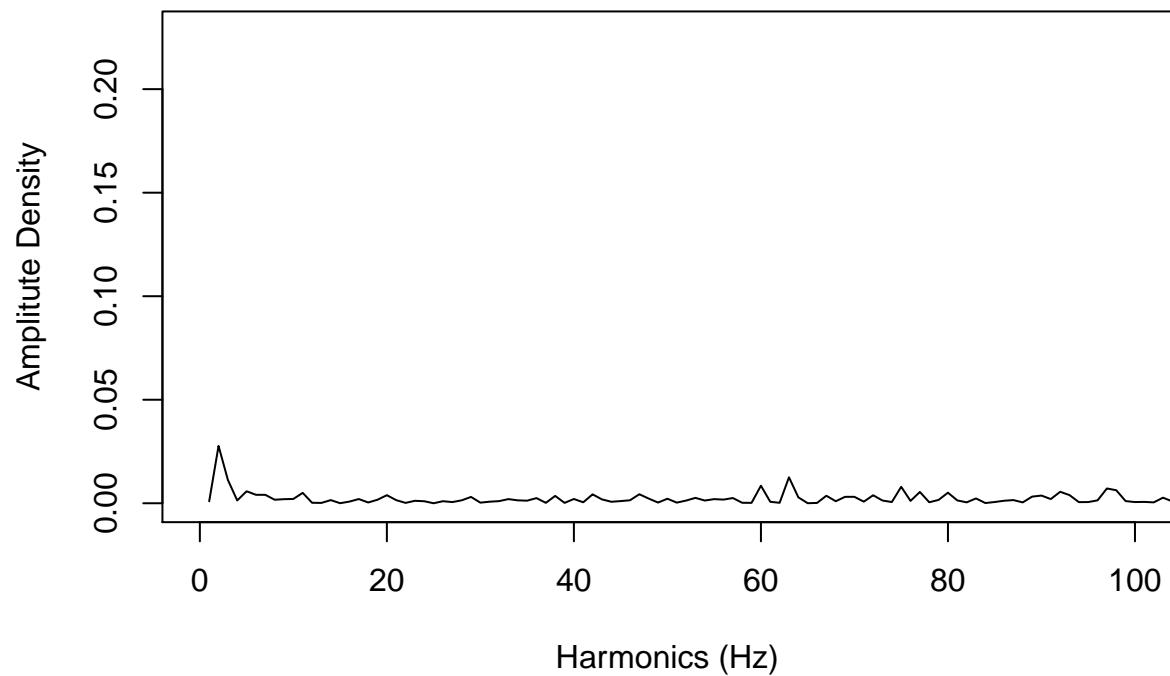
```
time_series_to_harmonics_psd(one_second_sample_cal[, 'JR3_MX'], acq_freq=1000, main="JR3_MX", xlim=c(0,100))
```

### JR3\_MX

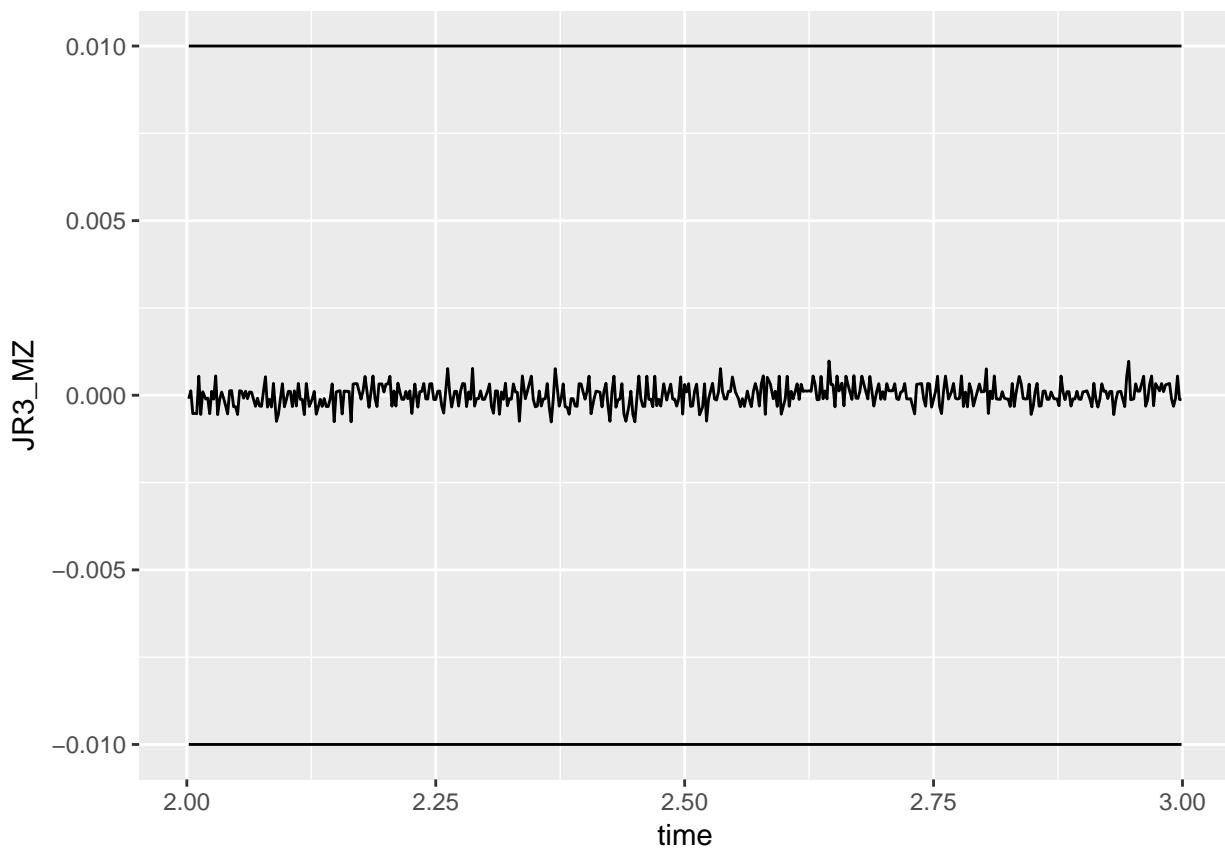




## JR3\_MY

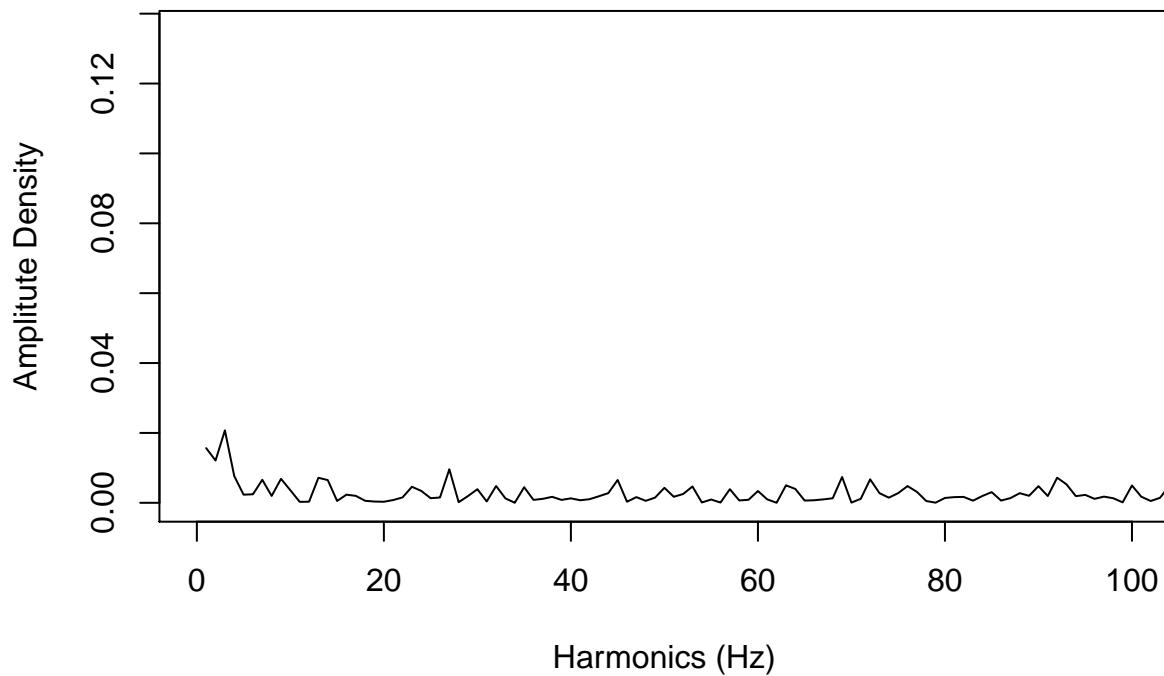


```
ggplot(one_second_sample_cal) + geom_line(aes(time, JR3_MZ))+ geom_line(aes(time, 1e-2))+ geom_line(aes
```



```
time_series_to_harmonics_psd(one_second_sample_cal[, 'JR3_MZ'], acq_freq=1000, main="JR3_MZ", xlim=c(0,100))
```

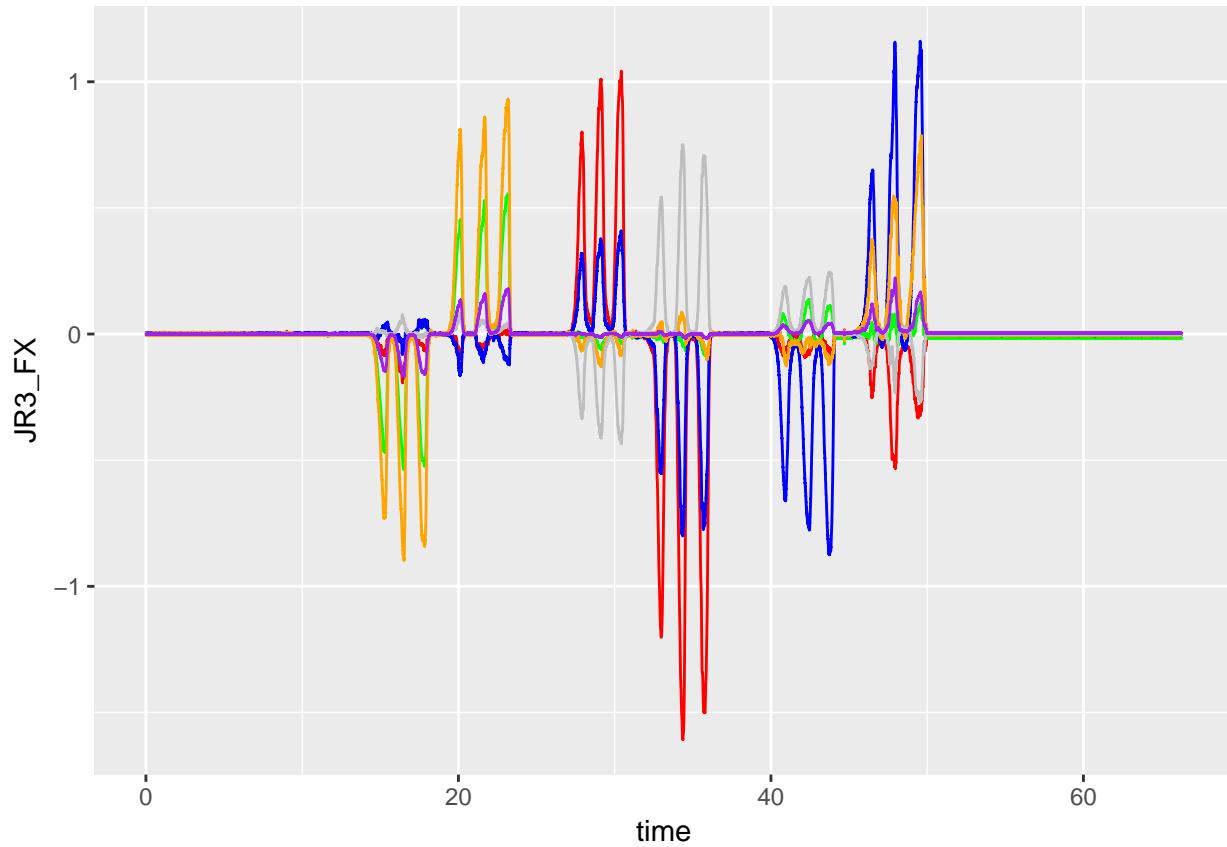
## JR3\_MZ



**How does Basement JR3 respond when pressing on the 20mm-displaced mount port where the fingertip will rest?**

noiseResponse2017\_11\_28\_17\_10\_48.txt features 32802 samples, over 66.249 seconds. Empirical sampling frequency is 495.132002

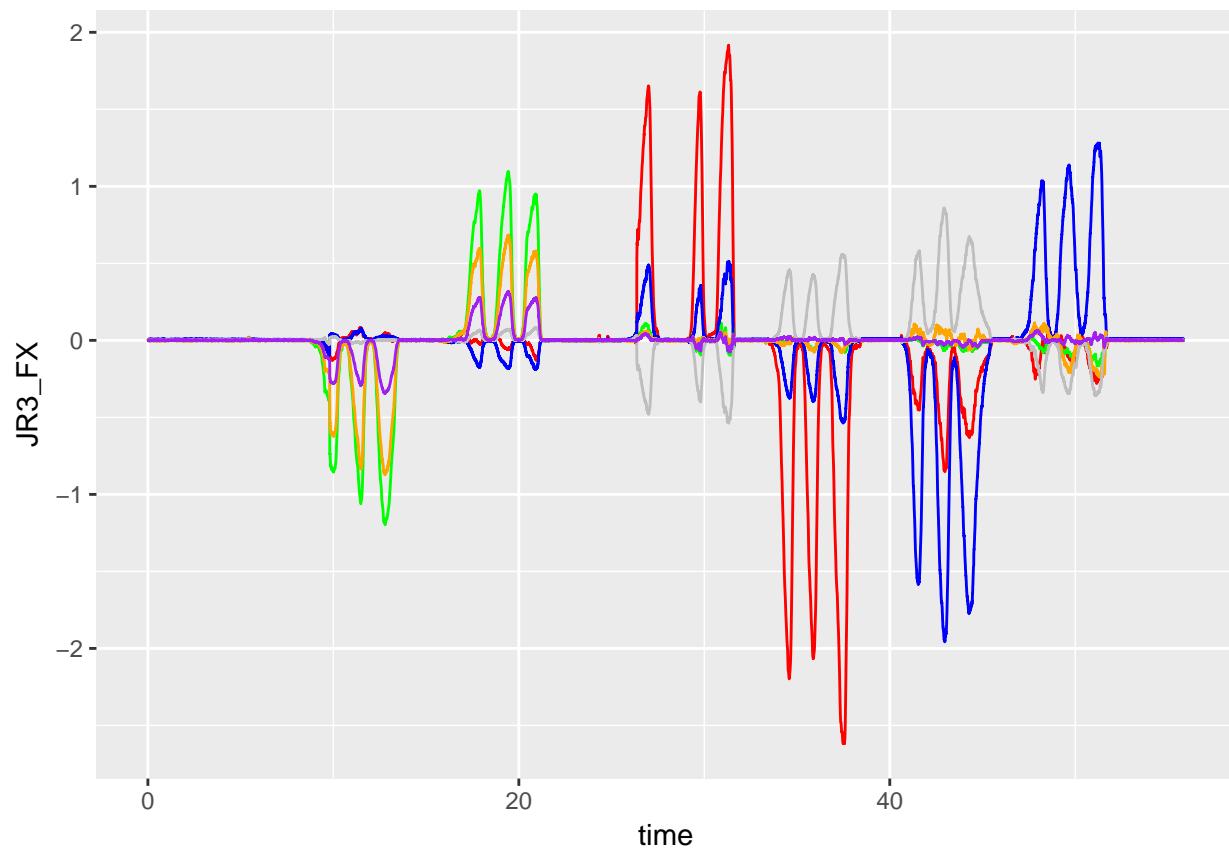
```
ggplot(jr3_brian_presses_directions) + geom_line(aes(time, JR3_FX), col="red") + geom_line(aes(time, JR3_MZ), col="blue")
```



**How does Basement JR3 respond when fingertip presses near the JR3 baseplate?**

noiseResponse2017\_11\_28\_18\_01\_19.txt features 25976 samples, over 55.812 seconds. Empirical sampling frequency is 465.419623

```
ggplot(jr3_brian_presses_directions_near_base) + geom_line(aes(time, JR3_FX), col="red") + geom_line(ae
```

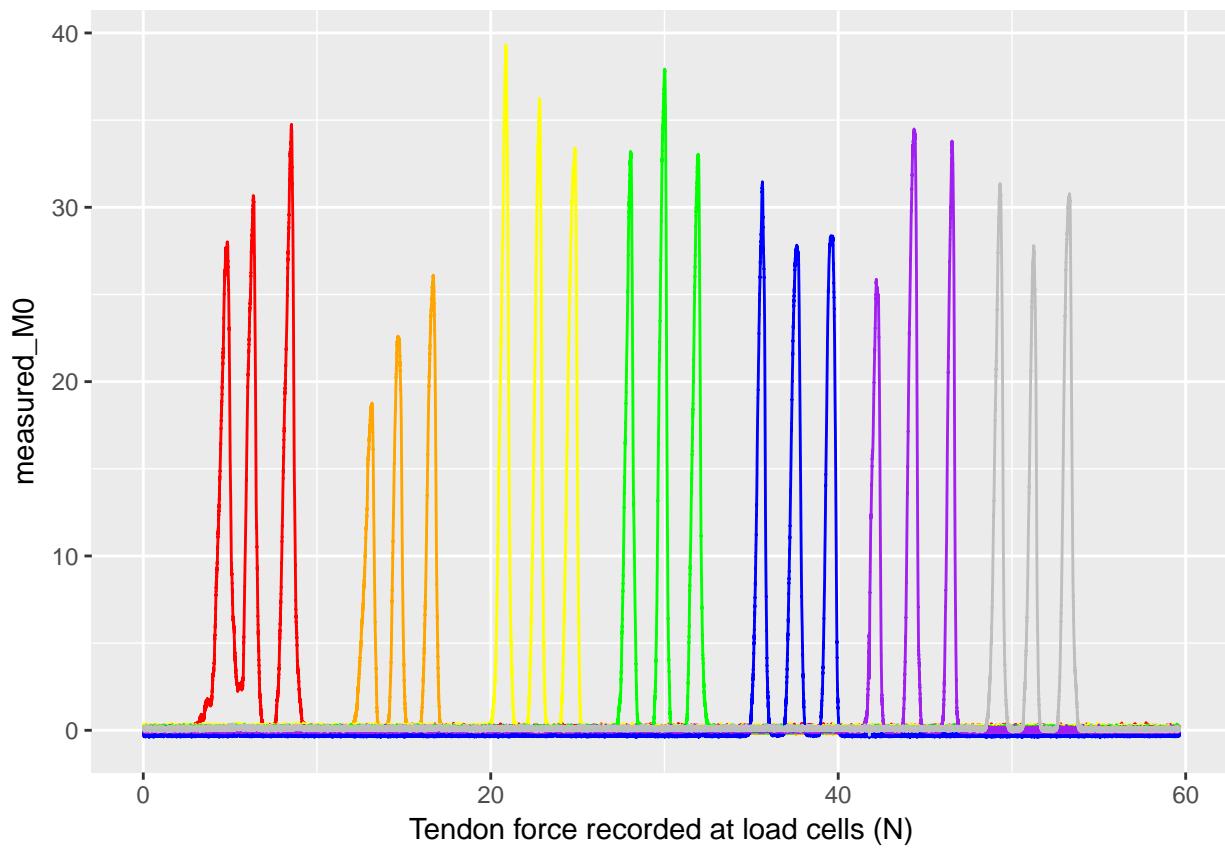


**How does Basement JR3 respond when muscles motor-collars are twisted, 3x ea, by hand?**

noiseResponse2017\_11\_28\_18\_10\_16.txt features 28093 samples, over 59.618 seconds. Empirical sampling frequency is 471.2167466

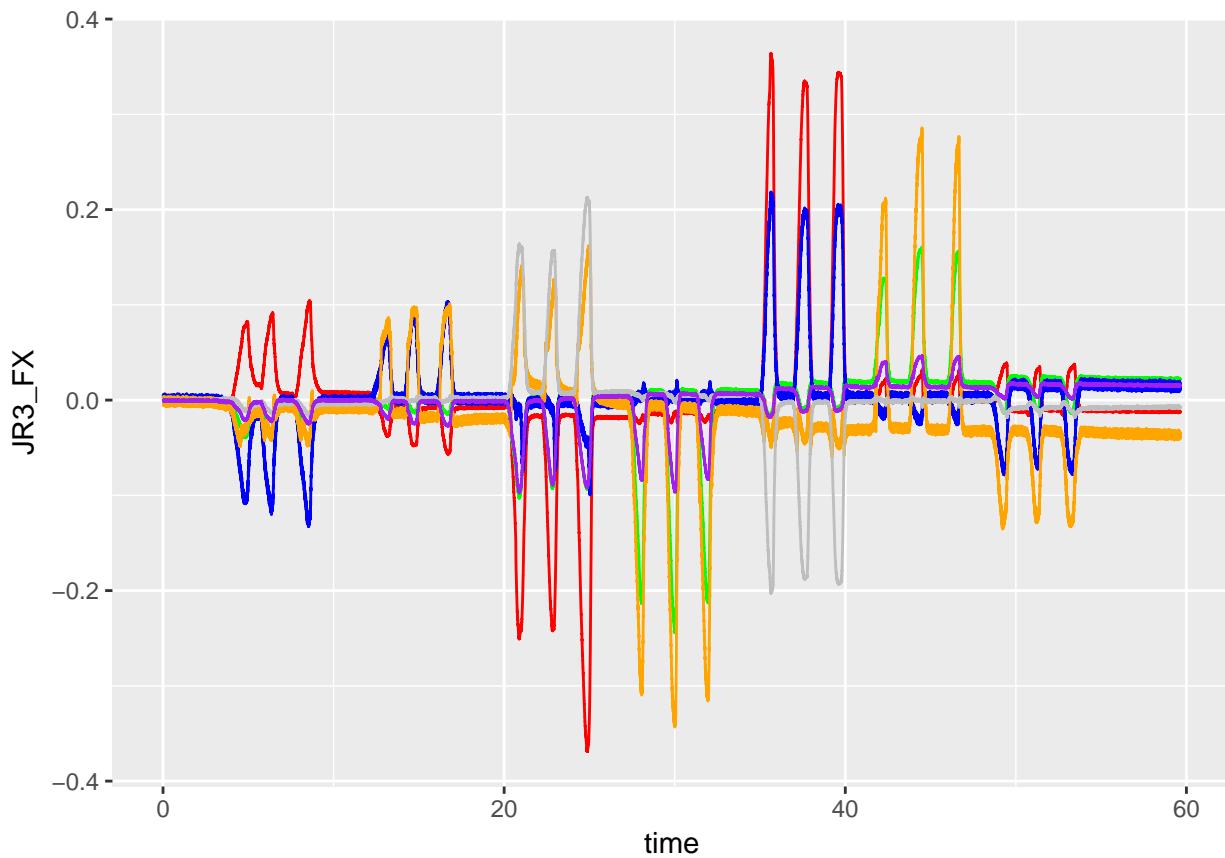
#### Muscle inputs

M0	red
M1	orange
M2	yellow
M3	green
M4	blue
M5	violet
M6	gray



### Muscle Outputs

```
ggplot(muscle_generator_twists) + geom_line(aes(time, JR3_FX), col="red") + geom_line(aes(time, JR3_FY)
```



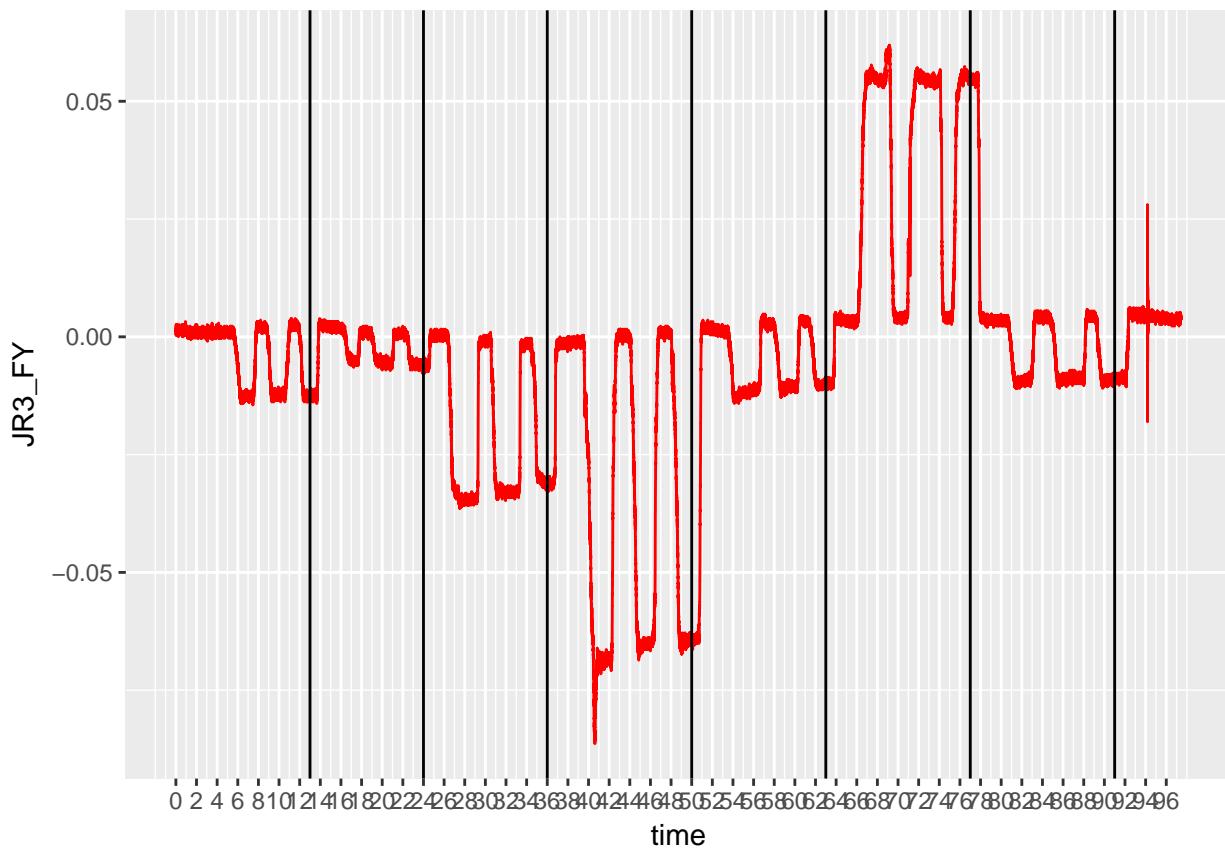
**How does Basement JR3 respond when tensioned by a static 500 g force hung from the kevlar?**

Importantly, all lengths of the muscles were approximately equal. Three replicates per muscle, in order from M0 to M6.

noiseResponse2017\_11\_28\_18\_15\_09.txt features 46274 samples, over 97.4 seconds. Empirical sampling frequency is 475.0924025

*#Small verison to get the seven hold indices*

```
ggplot(zeroed_500g_df) + geom_line(aes(time, JR3_FY), col="red") + geom_vline(xintercept=c(13,24,36,50))
  scale_x_continuous(breaks = round(seq(min(zeroed_500g_df$time), max(zeroed_500g_df$time), by = 2),1))
```

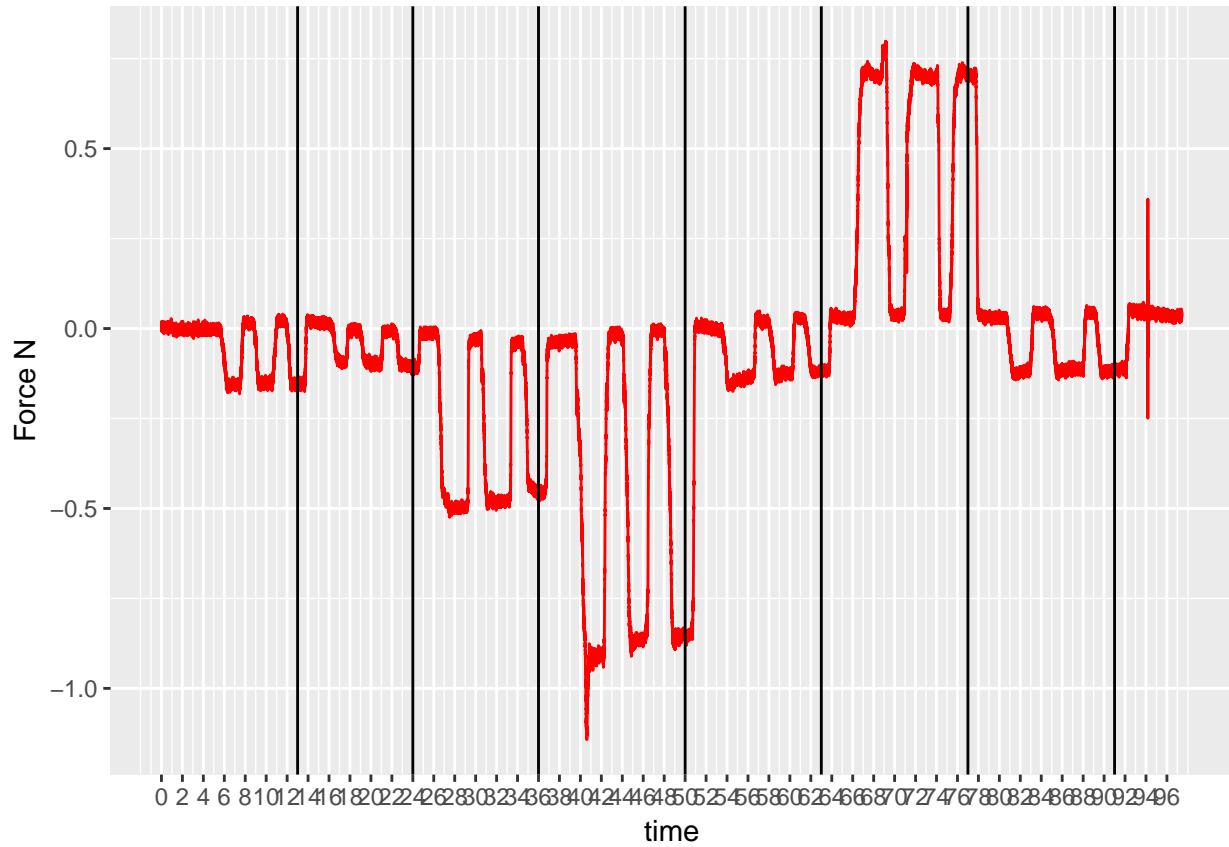


```

zeroed_500g_df_calibrated<- munge_JR3_data(zeroed_500g_df, input_are_voltages = TRUE, indices_for_null=1)

## Converting voltages into forces and torques
ggplot(zeroed_500g_df_calibrated) + geom_line(aes(time, JR3_FY), col="red") + geom_vline(xintercept=c(
  scale_x_continuous(breaks = round(seq(min(zeroed_500g_df_calibrated$time), max(zeroed_500g_df_calibrated$time), by=10)))
)

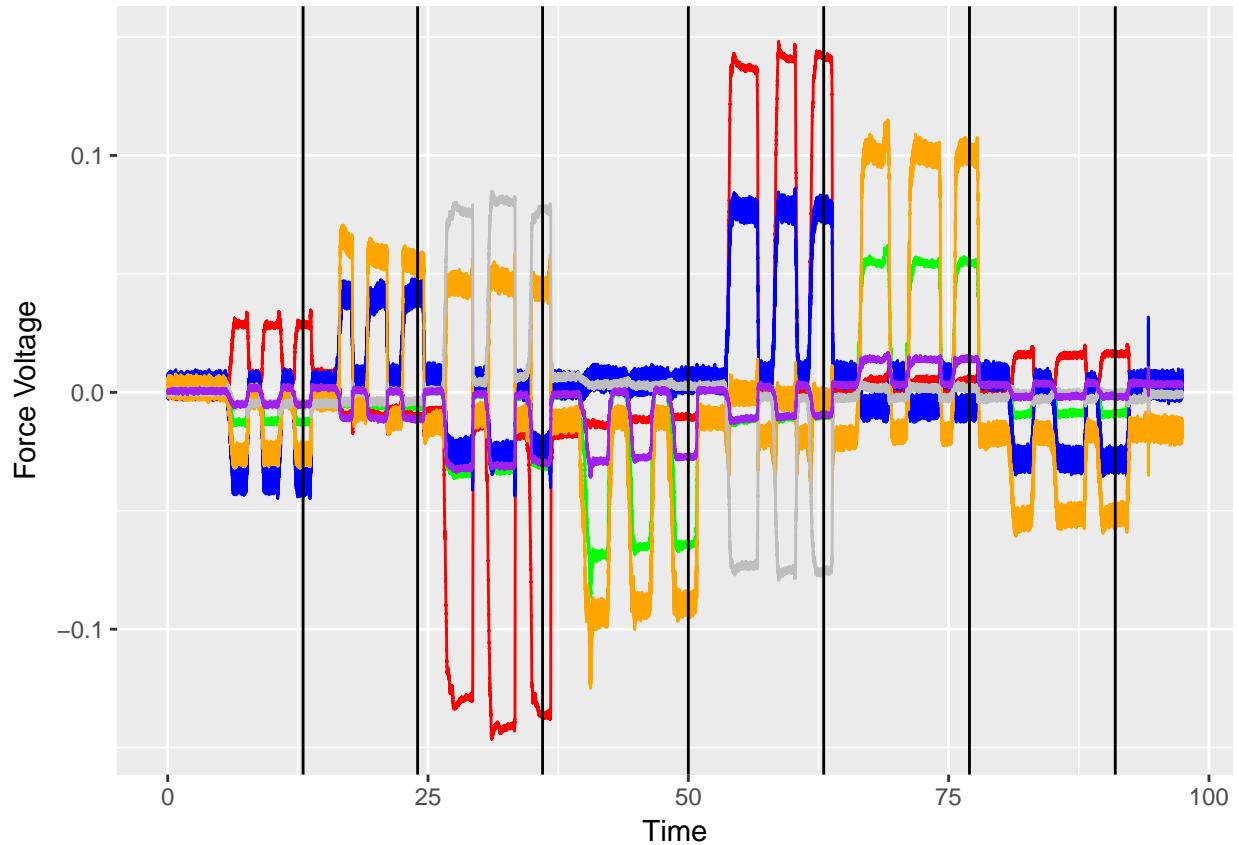
```



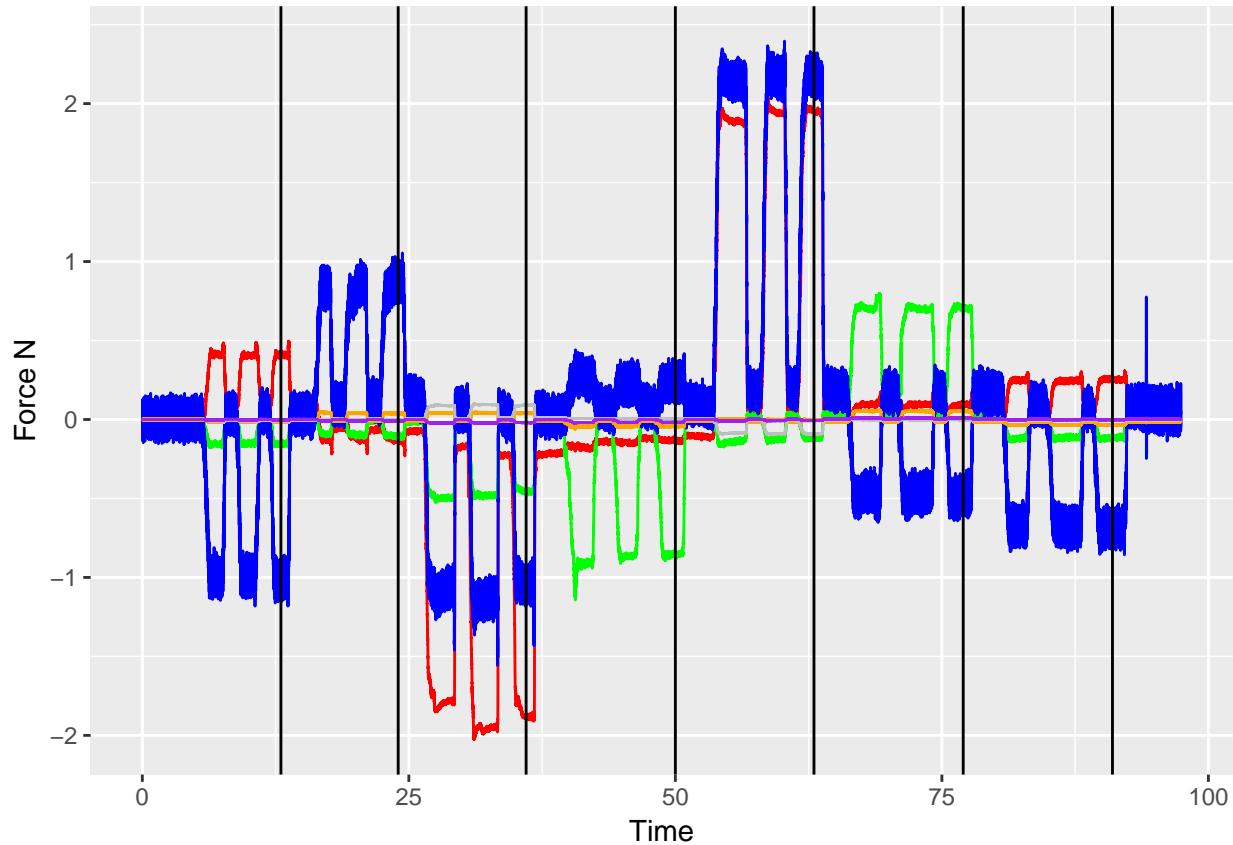
The vertical slices are the points from which I grabbed 30 datapoints, low pass filtered, got the mean, and used the resultant value as the generator.

```
generator_indices <- c(13, 24, 36, 50, 63, 77, 91)
```

```
ggplot(zeroed_500g_df) + geom_line(aes(time, JR3_FX), col="red") + geom_line(aes(time, JR3_FY), col="green")
```



```
ggplot(zeroed_500g_df_calibrated) + geom_line(aes(time, JR3_FX), col="red") + geom_line(aes(time, JR3_F
```



Muscle forces were recorded but should not have any change over time. Seems good. Notice how there are specific bands in Y that the signal hops to- that means we are pushing the precision boundary of the sensor. This means signal variance at rest is low.

```
## Low pass filtered output porcupine of FXYZ VOLTAGES at a
## 50Hz critical frequency

force_df <- zeroed_500g_df_calibrated[,dots_to_underscores(force_column_names)]
muscle_500g_generators <- take_running_mean_snapshots(force_df,zeroed_500g_df_calibrated$time, generators)
colnames(muscle_500g_generators) <- measured(muscle_names())
generators <- t(as.data.frame(as.matrix(muscle_500g_generators)))
axes_for_set(generators[,1:3], sizes=c(3,3,3), dimension_label="F")
apply(generators[,1:3], 1, function(x) arrow3d(c(0,0,0), x, type = "rotation", col = "#4daf4a", s=0.25)

##           measured_M0 measured_M1 measured_M2 measured_M3 measured_M4
## triangles          12          14          16          18          20
## quads              13          15          17          19          21
##           measured_M5 measured_M6
## triangles          22          24
## quads              23          25
```

Low pass filtered output porcupine of MXYZ VOLTAGES at a 50Hz critical frequency

```
axes_for_set(generators[,4:6], sizes=c(3,3,3), dimension_label="M")
apply(generators[,4:6], 1, function(x) arrow3d(c(0,0,0), x, type = "rotation", col = "#4daf4a", s=0.25)

##           measured_M0 measured_M1 measured_M2 measured_M3 measured_M4
```

```
## triangles      32      34      36      38      40
## quads         33      35      37      39      41
##               measured_M5 measured_M6
## triangles      42      44
## quads         43      45
```

**TODO** create FFS from generators via `n_binary_combinations`

```
#binary_combination_ffs_points <- custom_binary_combinations(7, c(0,1)) %*% generators
```