Untitled

```
library(ggplot2)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.1.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(ggthemes)
library(directlabels)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(gridExtra)

setwd('/Users/brendo/repos/ggplot2-brendo61-byte')

center_speaker <- read.csv("fft_data/center_speaker.csv")
right_speaker <- read.csv("fft_data/right_speaker.csv")
```

```
# Data Organization
size = 200

# given that I used a pretty cheap mic 200 seems to be a fair number
# also the speaker also does not have a lot of output below 100Hz - this is known from the manufacturer

# Rolling AVG to smooth noise in recording
rolling_freq = rollmean(center_speaker$freq, size)

freqency_val = c(rolling_freq, rolling_freq, rolling_freq, rolling_freq)

applitudes = c(
  rollmean(center_speaker$X0, size),
  rollmean(center_speaker$X15, size),
  rollmean(center_speaker$X30, size),
  rollmean(center_speaker$X45, size))

applitudes = 20 * log10(applitudes)

len = length(rolling_freq)
off_axis_degree = c(rep('0°', len),
                    rep('15°', len),
                    rep('30°', len),
                    rep('45°', len)
                    )


rolled_center_speaker = data.frame(freqency = freqency_val,
                                   amplitude = applitudes,
                                   degrees_off_axis = off_axis_degree
                                   )
```

```
center_plot = ggplot(data = rolled_center_speaker,
              mapping = aes(x = freqency, y = amplitude, color = degrees_off_axis)) +
  geom_line(size = 0.4, show.legend = FALSE) +
  directlabels::geom_dl(aes(label = degrees_off_axis), method = "first.qp") +
  theme_classic() +
  theme_tufte(base_size=10, base_family = "sans") +
  scale_y_continuous(limits = c(80, 121)) +
  scale_x_continuous(trans = "log10", limits = c(100, 20000),
                     breaks = c(200, 2000, 20000),
                     minor_breaks = c(100, 300, 400, 500, 600, 700, 800, 900, 1000,
                                      3000, 4000, 5000, 6000, 7000, 8000, 9000,
                                      10000)
                     ) +
  theme(panel.grid.major.x = element_line(color = "grey", size = 0.25, linetype = 2)) +
  theme(panel.grid.minor = element_line(color = "grey", size = 0.1, linetype = 1)) +
  ylab("Amplitude (dB)") +
  xlab("Frequency (Hz)") +
  labs(title = "Center Channel") +
  theme(plot.title = element_text(size = 10.5)) +
    geom_vline(xintercept = 1000, linetype="solid", size = 0.3, show.legend = FALSE) +
  geom_text(aes(x=1000, label="\nX-Over", y=90, size = 4), colour="black", angle=90, show.legend = FALSE)


# center_plot
```

```
# Data Organization
size = 200

# again setting rolling mean size to 200
# the right channel will have more output under 100Hz - but the subwoofers are doing most of the heavy lifting th
ere
# I don't care too much abouve < 100 Hz

# Rolling AVG to smooth noise in recording
rolling_freq = rollmean(right_speaker$freq, size)

freqency_val = c(rolling_freq, rolling_freq, rolling_freq, rolling_freq)

applitudes = c(
  rollmean(right_speaker$X0, size),
  rollmean(right_speaker$X15, size),
  rollmean(right_speaker$X30, size),
  rollmean(right_speaker$X45, size))

applitudes = 20 * log10(applitudes)

len = length(rolling_freq)
off_axis_degree = c(rep('0°', len),
                    rep('15°', len),
                    rep('30°', len),
                    rep('45°', len)
                    )


rolled_right_speaker = data.frame(freqency = freqency_val,
                                  amplitude = applitudes,
                                  degrees_off_axis = off_axis_degree
                                  )
```

```
right_plot = ggplot(data = rolled_right_speaker,
              mapping = aes(x = freqency, y = amplitude, color = degrees_off_axis)) +
  geom_line(size = 0.4, show.legend = FALSE) +
  directlabels::geom_dl(aes(label = degrees_off_axis), method = "first.qp") +
  theme_classic() +
  theme_tufte(base_size=10, base_family = "sans") +
  scale_y_continuous(limits = c(80, 121)) +
  scale_x_continuous(trans = "log10", limits = c(100, 20000),
                     breaks = c(200, 2000, 20000),
                     minor_breaks = c(100, 300, 400, 500, 600, 700, 800, 900, 1000,
                                      3000, 4000, 5000, 6000, 7000, 8000, 9000,
                                      10000)
                     ) +
  theme(panel.grid.major.x = element_line(color = "grey", size = 0.25, linetype = 2)) +
  theme(panel.grid.minor = element_line(color = "grey", size = 0.1, linetype = 1)) +
  ylab("Amplitude (dB)") +
  xlab("Frequency (Hz)") +
  labs(title = "Right Channel",
              caption = "Frequency sweeps taken 1 meter from speaker at intevals of 15 degress off-axis relative
 to forward direction of speaker.\nSingnal below cross-over point is produced via woofer and above x-over via the
tweeter.") +
  theme(plot.title = element_text(size = 10.5), plot.caption = element_text(size = 8.5, hjust = 0)) +
  geom_vline(xintercept = 800, linetype="solid", size = 0.3, show.legend = FALSE) +
  geom_text(aes(x=800, label="\nX-Over", y=90, size = 4), colour="black", angle=90, show.legend = FALSE)

# right_plot
```

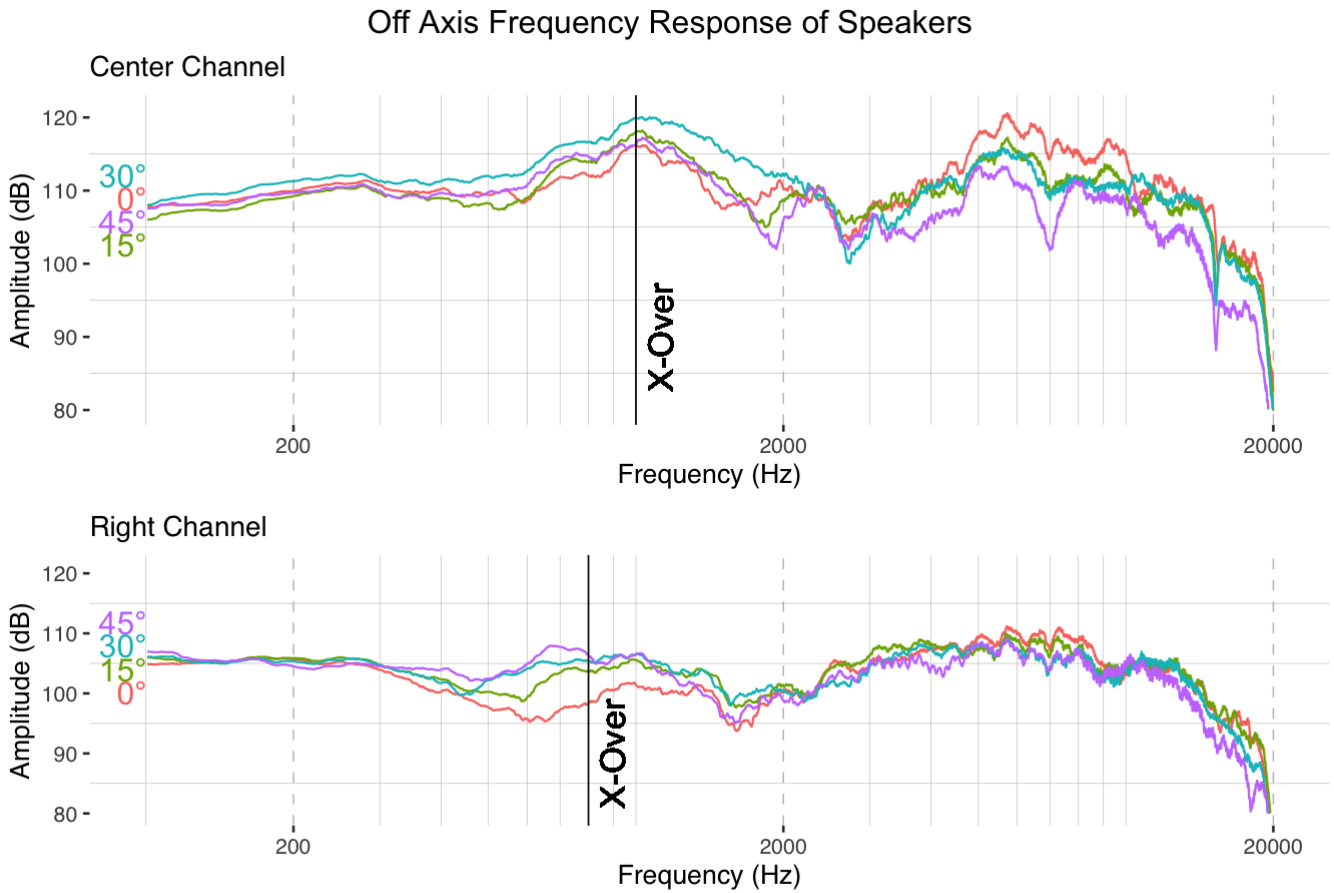
```
grid.arrange(center_plot, right_plot,
             ncol=1,
             top = "Off Axis Frequency Response of Speakers")
```

```
## Warning: Removed 8324 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 8326 rows containing missing values (geom_dl).
```

```
## Warning: Removed 9419 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 9419 rows containing missing values (geom_dl).
```



Frequency sweeps taken 1 meter from speaker at intevals of 15 degress off-axis relative to forward direction of speaker.
Singnal below cross-over point is produced via woofer and above x-over via the tweeter.