Normalization - Variance and Bias

Nate Olson

2016-07-20

Objective:

Evaluate the bias and variance of different normalization methods used the marker gene two-sample titration study dataset.

Loading data

```
proj_dir <- "~/Projects/16S_etec_mix_study"
pipe_dir <- file.path(proj_dir, "analysis","pipelines")
qiime_dir <- file.path(pipe_dir, "qiime")
qiime_std <- readRDS(file.path(qiime_dir, "otus_uc_fast","mrexp_obj.rds"))
qiime_std <- qiime_std[,which(pData(qiime_std)$sampleID != "NTC" &
                              pData(qiime_std)$id != "B4_M6_P1_L1_S1")]
```

Tidy Data

Focusing on biological replicate 1 no normalization or scaling

```
sam_dat <- pData(qiime_std) %>% rownames_to_column(var = "samID") %>%
      mutate(sample_total = colSums(qiime_std))
```

```
qiime_std_df <- assayData(qiime_std)[['counts']] %>% as.data.frame() %>%
        rownames_to_column(var = "otu") %>%
        gather("samID","count", -otu) %>%
        left_join(sam_dat) %>% select(-pos_ns, -barcode_lab, -seq_lab, -samID)
```

```
## Joining, by = "samID"
```

Log transforming counts.

```
qiime_log2_df <- MRcounts(qiime_std,norm = FALSE, log = FALSE) %>% as.data.frame() %>%
        rownames_to_column(var = "otu") %>%
        gather("samID","count", -otu) %>% mutate(count = log2(count + 1)) %>%
        left_join(sam_dat) %>% select(-pos_ns, -barcode_lab, -seq_lab, -samID)
```

```
## Joining, by = "samID"
```

```
qiime_css_df <- cumNormMat(qiime_std, p = 0.75) %>% as.data.frame() %>%
        rownames_to_column(var = "otu") %>%
        gather("samID","count", -otu) %>%
        left_join(sam_dat) %>% select(-pos_ns, -barcode_lab, -seq_lab, -samID)
```

```
## Joining, by = "samID"
```

```
qiime_csslog2_df <- cumNormMat(qiime_std, p = 0.75) %>% as.data.frame() %>%
        rownames_to_column(var = "otu") %>%
```

```
        gather("samID","count", -otu) %>% mutate(count = log2(count + 1)) %>%
        left_join(sam_dat) %>% select(-pos_ns, -barcode_lab, -seq_lab, -samID)
```

```
## Joining, by = "samID"
```

Mean count vs. Standard Deviation count

Variability between technical replicates. The variability is correlated with the counts.

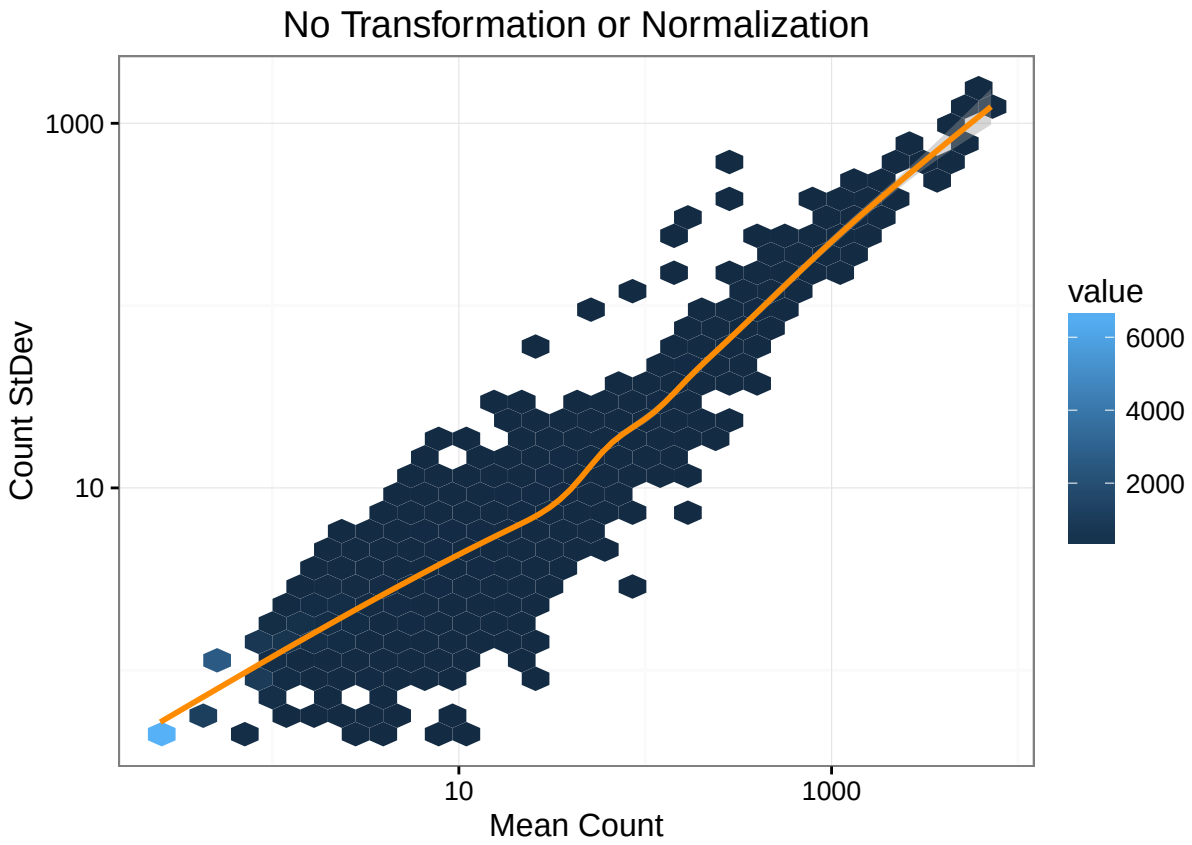
```
get_var_df <- function(count_df){
      count_df %>% filter(sampleID =="E01JH0004") %>%
            group_by(dilution) %>%
                  mutate(sample_max = max(sample_total),
                         sample_min = min(sample_total),
                         sample_range = sample_max - sample_min) %>%
            group_by(otu,dilution) %>%
                  summarise(total_count = sum(count),
                            mean_count = mean(count),
                            var_count = var(count),
                            sd_count = sd(count),
                            sample_range = median(sample_range),
                            n_range = length(unique(sample_range)))
}
```

```
bio1_var <- qiime_std_df %>% get_var_df()
bio1_log2_var <- qiime_log2_df %>% get_var_df()
bio1_css_var <- qiime_css_df %>% get_var_df()
bio1_csslog2_var <- qiime_csslog2_df %>% get_var_df()
```

```
bio1_var %>% filter(total_count != 0) %>%
      ggplot() + geom_hex(aes(x = mean_count, y = sd_count)) +
      geom_smooth(aes(x = mean_count, y = sd_count), color = "darkorange") +
      scale_y_log10() +
      scale_x_log10() +
      theme_bw() +
      labs(x = "Mean Count", y = "Count StDev",
           title = "No Transformation or Normalization")
```

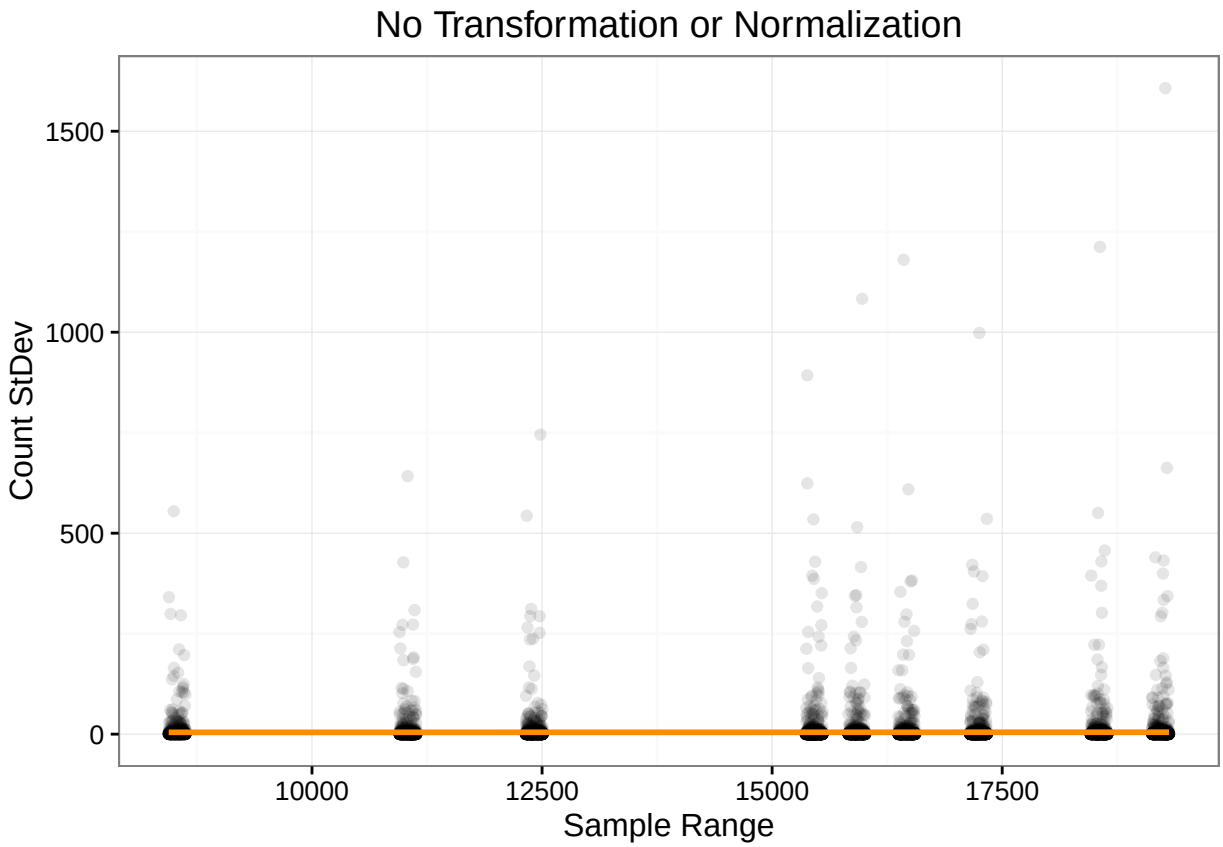
```
## Warning: Removed 29 rows containing non-finite values (stat_binhex).
```

```
## Warning: Removed 29 rows containing non-finite values (stat_smooth).
```



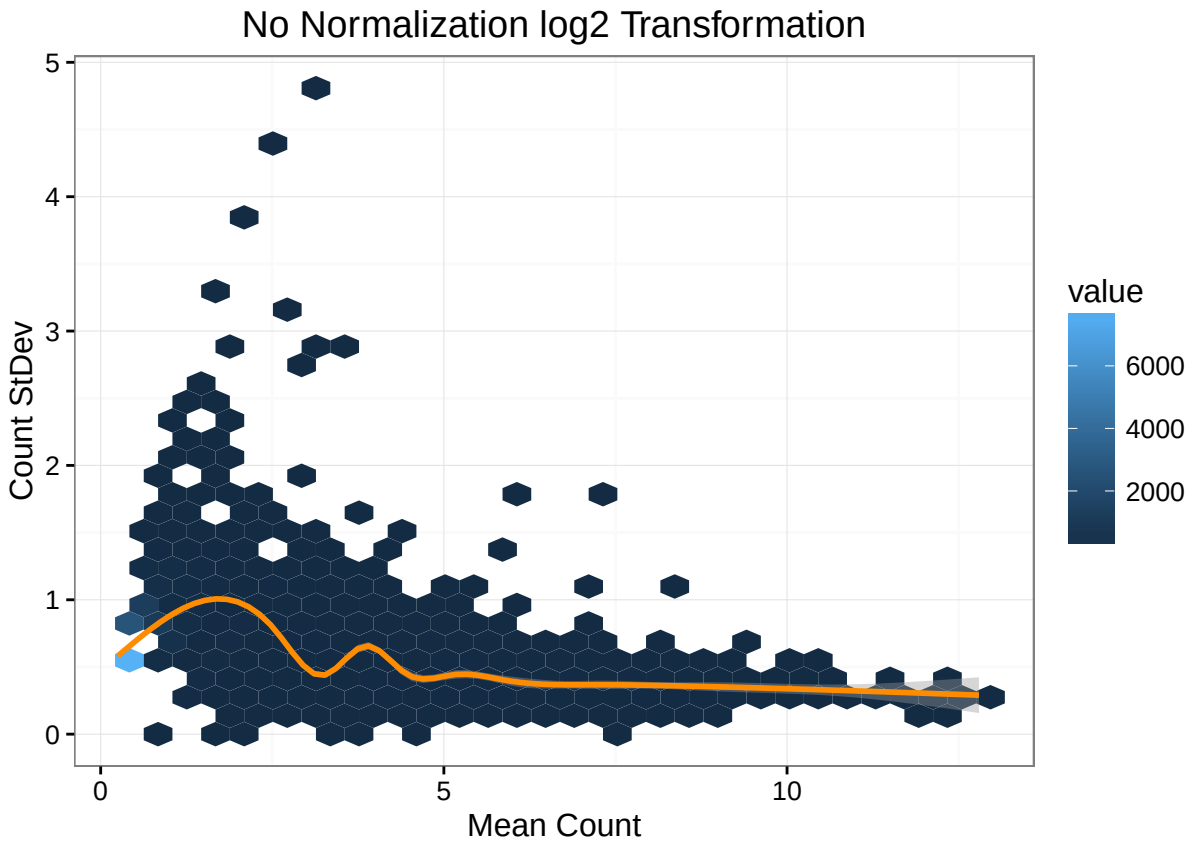
Overall the OTU count standard deviation between replicates does not increase with difference in total counts per sample. Though the count standard deviations outliers increase with sample range

```
bio1_var %>% filter(total_count != 0) %>%
      ggplot(aes(x = jitter(sample_range), y = sd_count)) +
      geom_point(alpha = 0.1) +
      geom_smooth(color = "darkorange") +
      theme_bw() +
      labs(x = "Sample Range", y = "Count StDev", title = "No Transformation or Normalization")
```

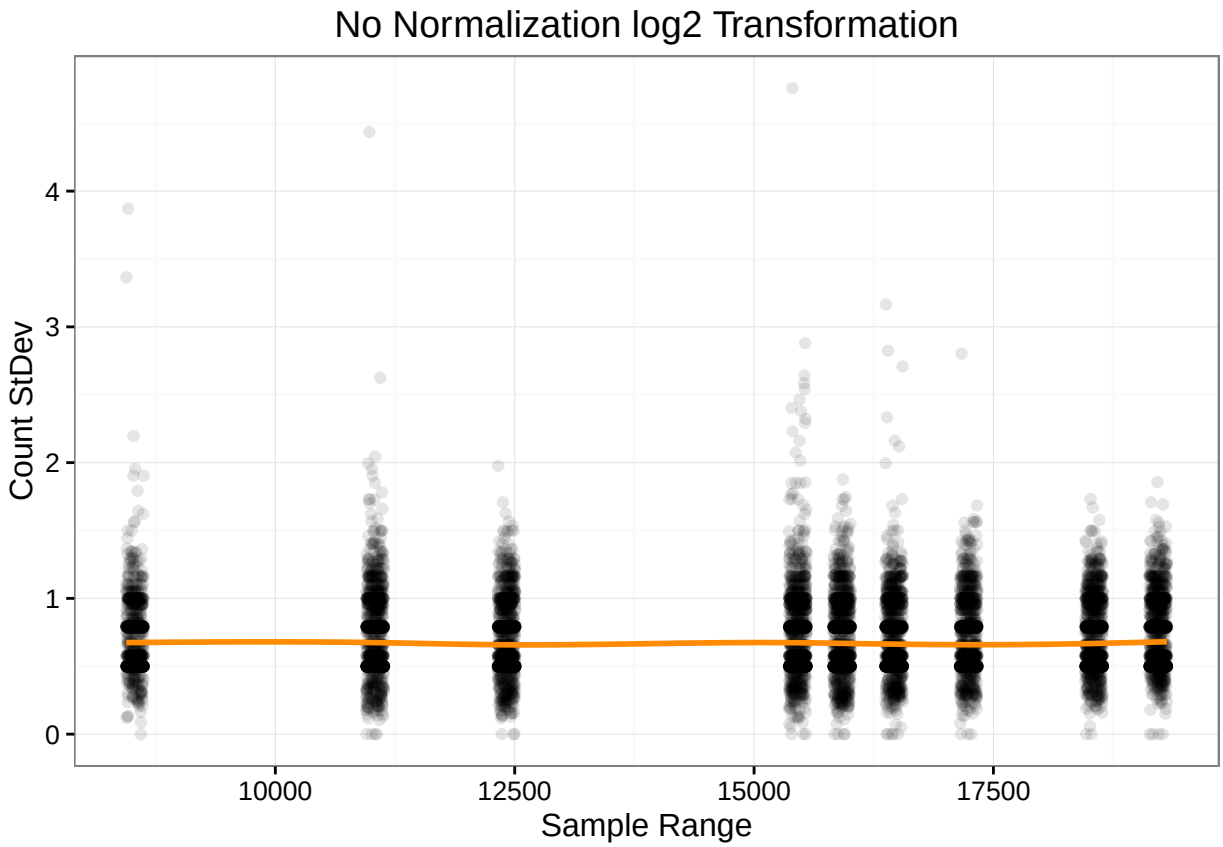


Variability between technical replicates with log2 transformation

```
bio1_log2_var %>% filter(total_count != 0) %>%
      ggplot() + geom_hex(aes(x = mean_count, y = sd_count)) +
      geom_smooth(aes(x = mean_count, y = sd_count), color = "darkorange") +
      #scale_y_log10() +
      #scale_x_log10() +
      theme_bw() +
      labs(x = "Mean Count", y = "Count StDev", title = "No Normalization log2 Transformation")
```

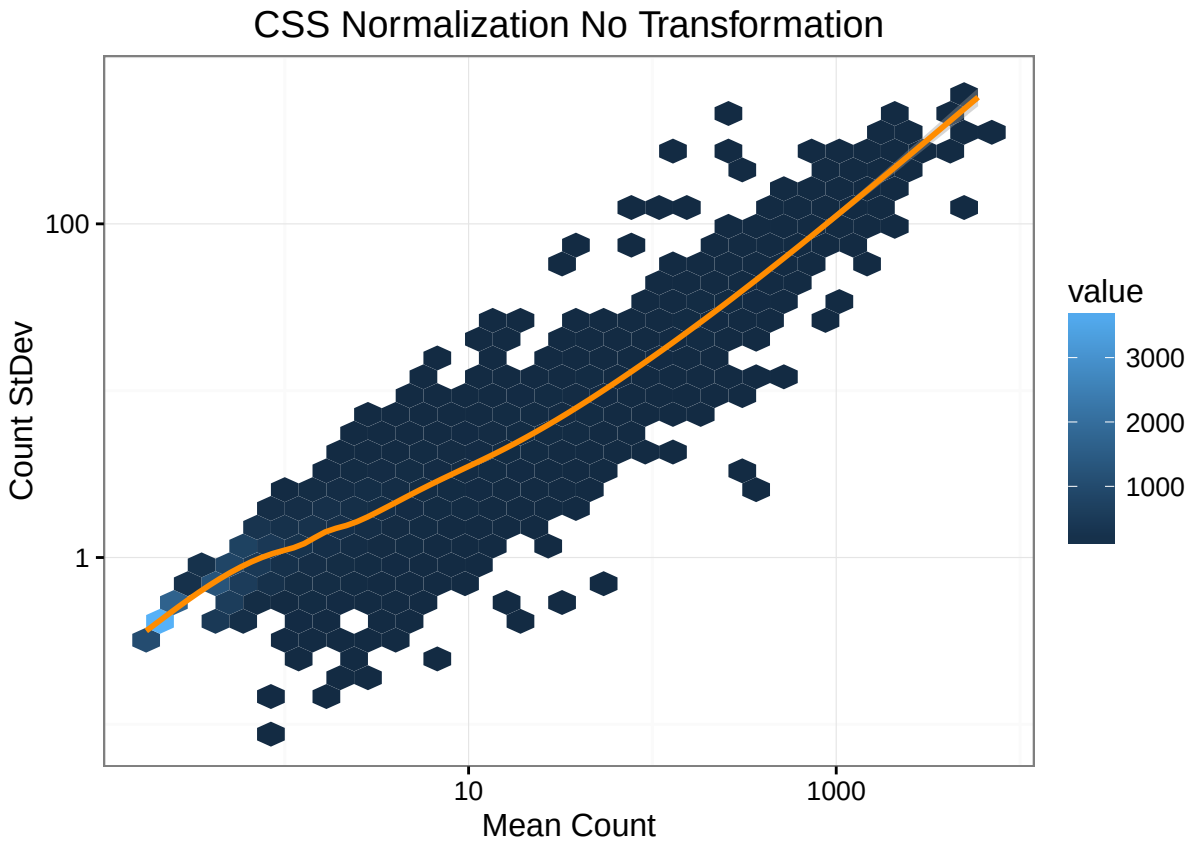



```
bio1_log2_var %>% filter(total_count != 0) %>%
      ggplot(aes(x = jitter(sample_range), y = sd_count)) +
      geom_point(alpha = 0.1) +
      geom_smooth(color = "darkorange") +
      theme_bw() +
      labs(x = "Sample Range", y = "Count StDev", title = "No Normalization log2 Transformation")
```

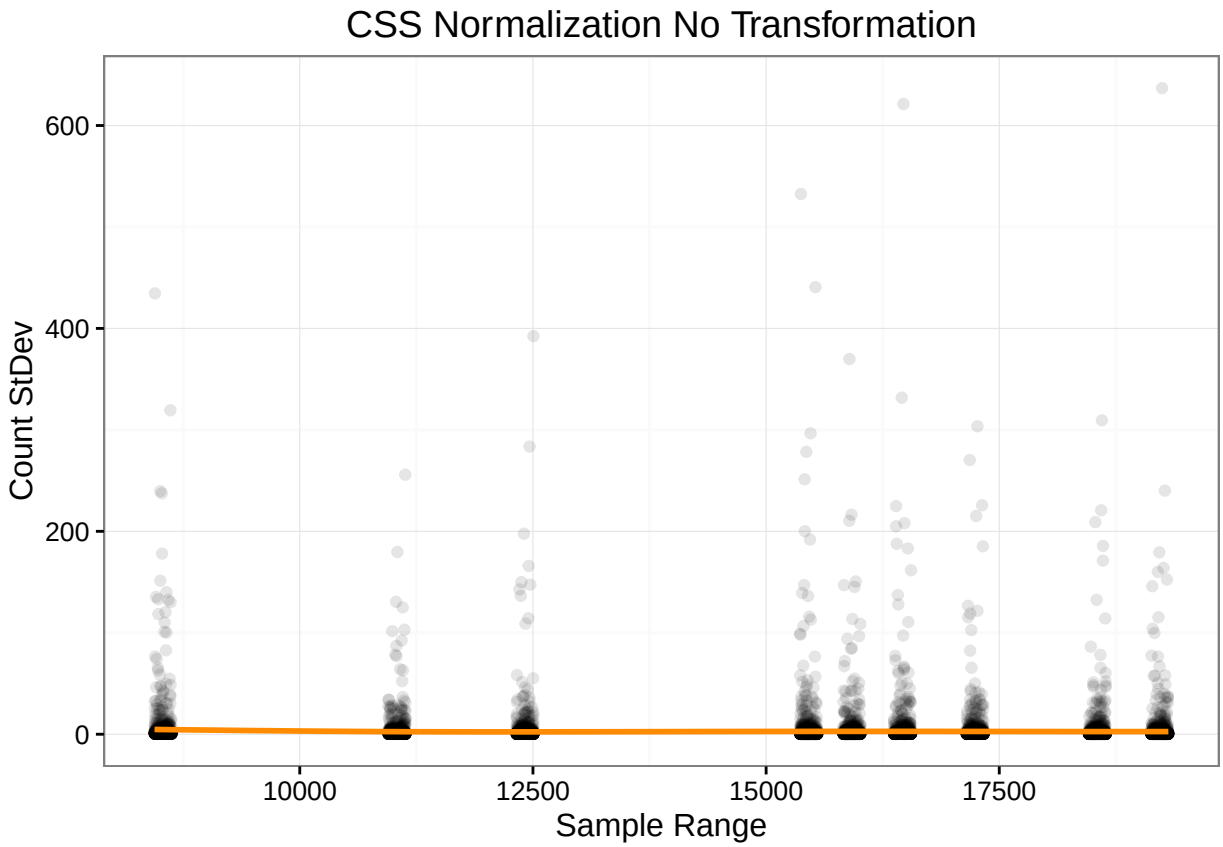


Variability between technical replicates with CSS

```
bio1_css_var %>% filter(total_count != 0) %>%
      ggplot() + geom_hex(aes(x = mean_count, y = sd_count)) +
      geom_smooth(aes(x = mean_count, y = sd_count), color = "darkorange") +
      scale_y_log10() + scale_x_log10() + theme_bw() +
      labs(x = "Mean Count", y = "Count StDev", title = "CSS Normalization No Transformation")
```

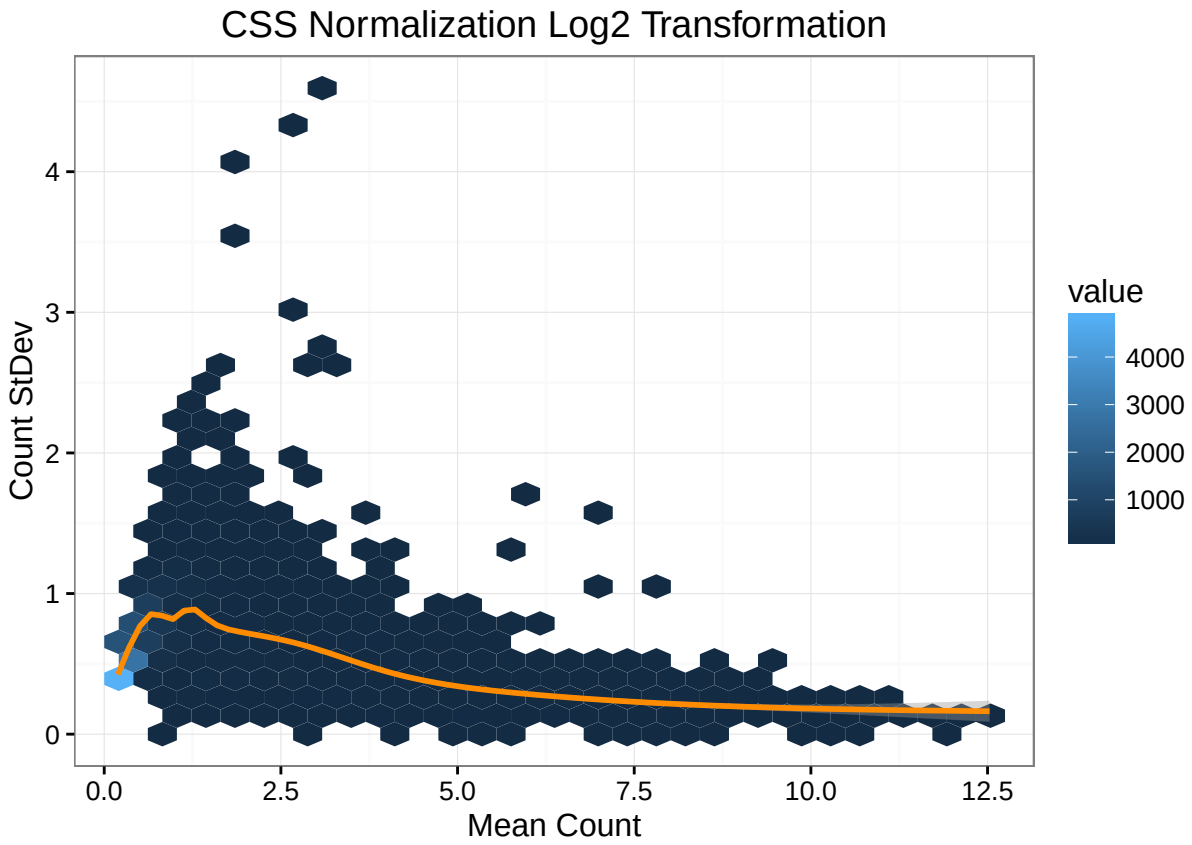



```
bio1_css_var %>% filter(total_count != 0) %>%
      ggplot(aes(x = jitter(sample_range), y = sd_count)) +
      geom_point(alpha = 0.1) +
      geom_smooth(color = "darkorange") +
      theme_bw() +
      labs(x = "Sample Range", y = "Count StDev", title = "CSS Normalization No Transformation")
```

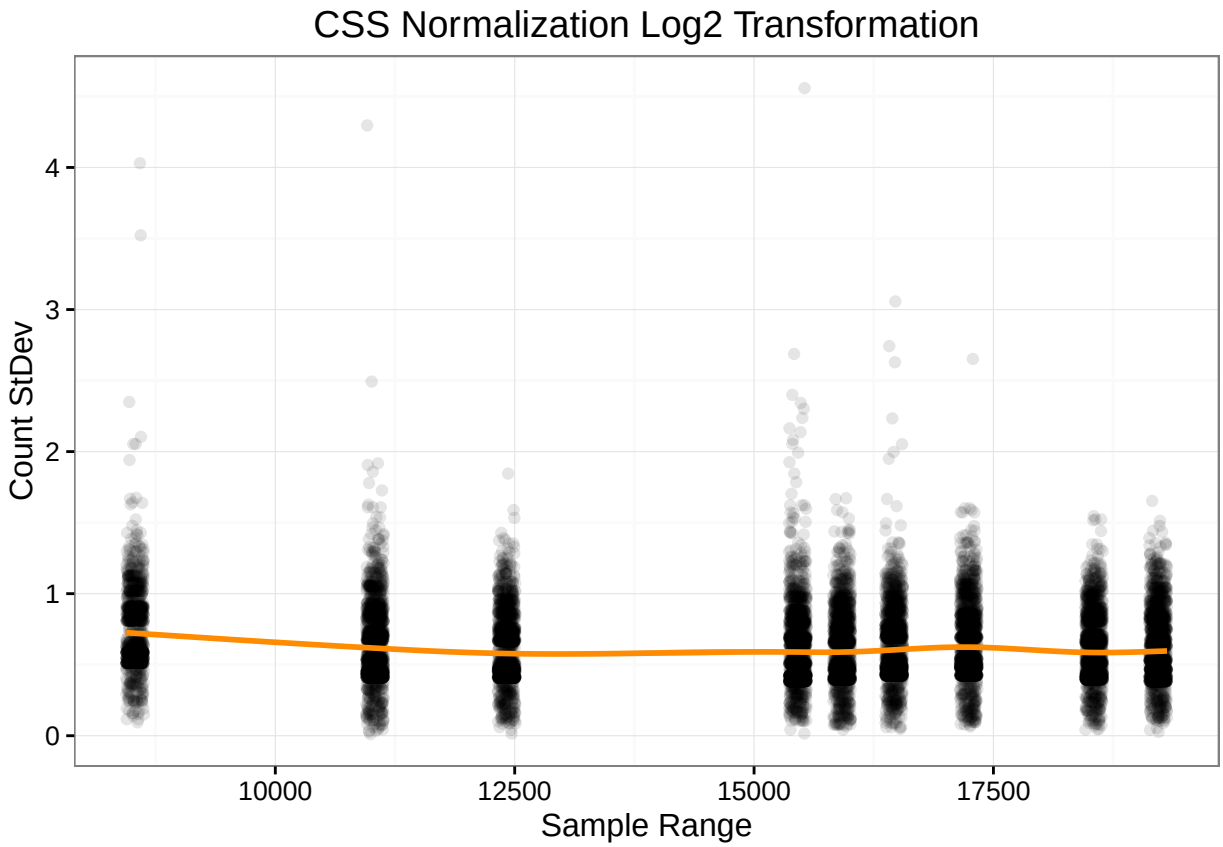


Variability between technical replicates with CSS and log2 transformation

```
bio1_csslog2_var %>% filter(total_count != 0) %>%
      ggplot() + geom_hex(aes(x = mean_count, y = sd_count)) +
      geom_smooth(aes(x = mean_count, y = sd_count), color = "darkorange") +
      theme_bw() +
      labs(x = "Mean Count", y = "Count StDev", title = "CSS Normalization Log2 Transformation")
```




```
bio1_csslog2_var %>% filter(total_count != 0) %>%
      ggplot(aes(x = jitter(sample_range), y = sd_count)) +
      geom_point(alpha = 0.1) +
      geom_smooth(color = "darkorange") +
      theme_bw() +
      labs(x = "Sample Range", y = "Count StDev", title = "CSS Normalization Log2 Transformation")
```



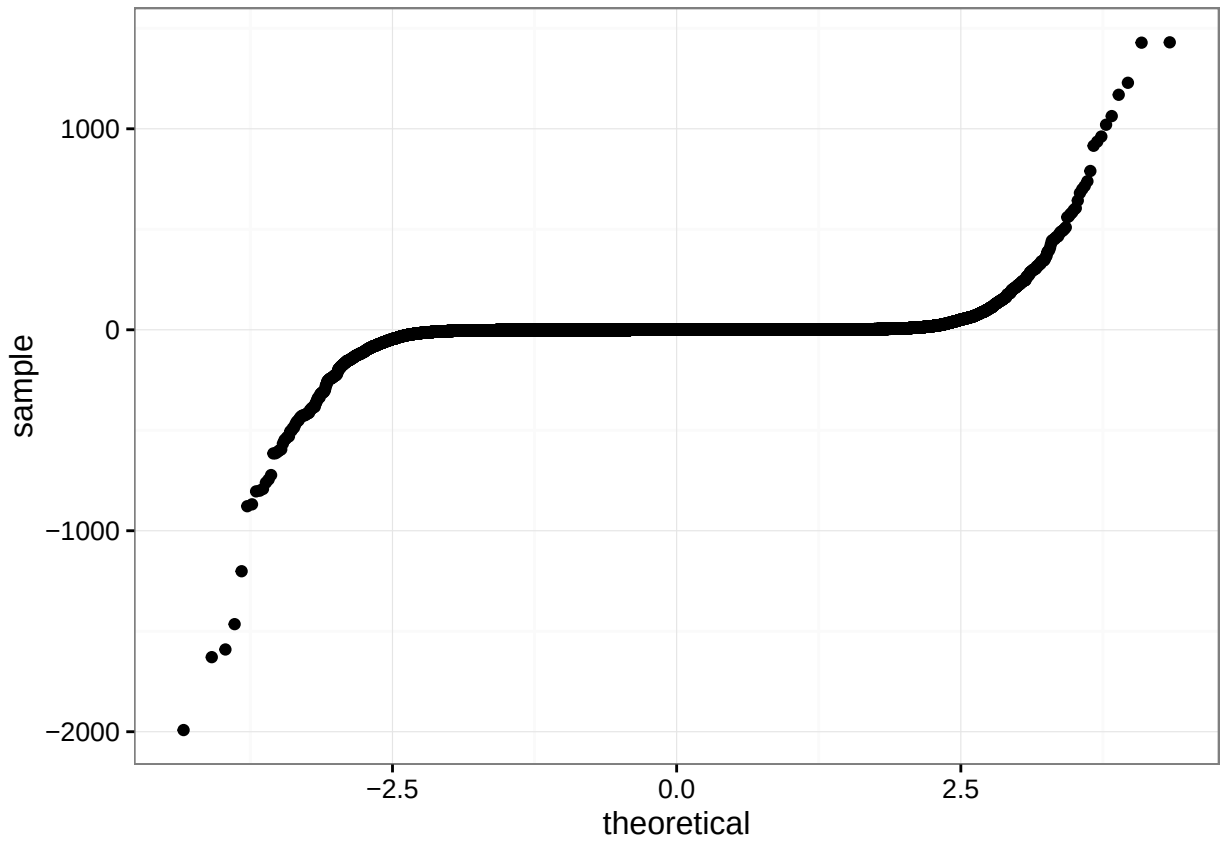
Characterization of residuals

```
get_resid_df <- function(count_df, min_total = 1){
      count_df %>% filter(sampleID == "E01JH0004") %>%
      group_by(otu, dilution) %>%
      mutate(total_count = sum(count),
            mean_count = mean(count),
            resid_count = count - mean_count,
            abs_resid_count = abs(resid_count)) %>%
      filter(total_count >= min_total) %>%
      separate(pos,c("plate_row","plate_col"),sep = "_",remove = FALSE)
}
```

```
bio1_resid <- qiime_std_df %>% get_resid_df()
```

Residuals are not normally distributed due to the correlation between the count value and standard deviation.

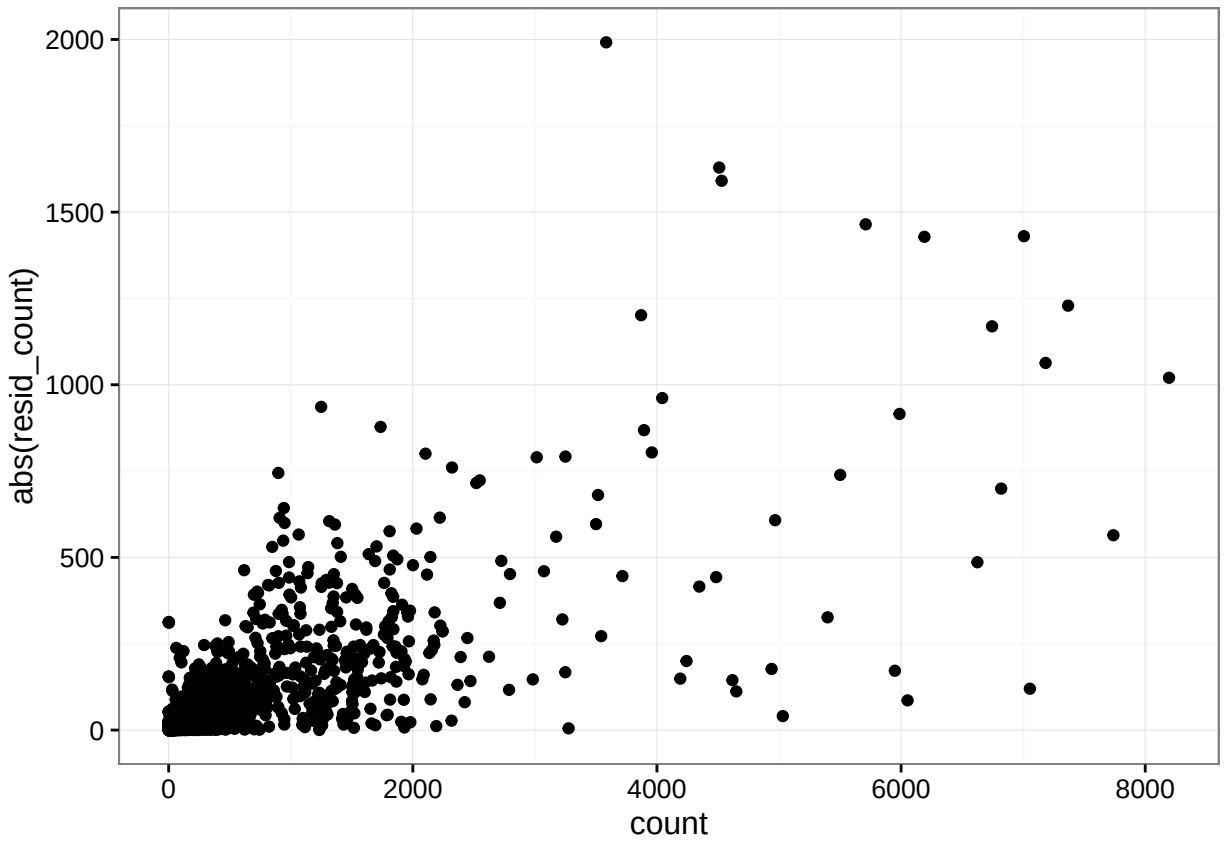
```
ggplot(bio1_resid) + geom_qq(aes(sample = resid_count)) + theme_bw()
```



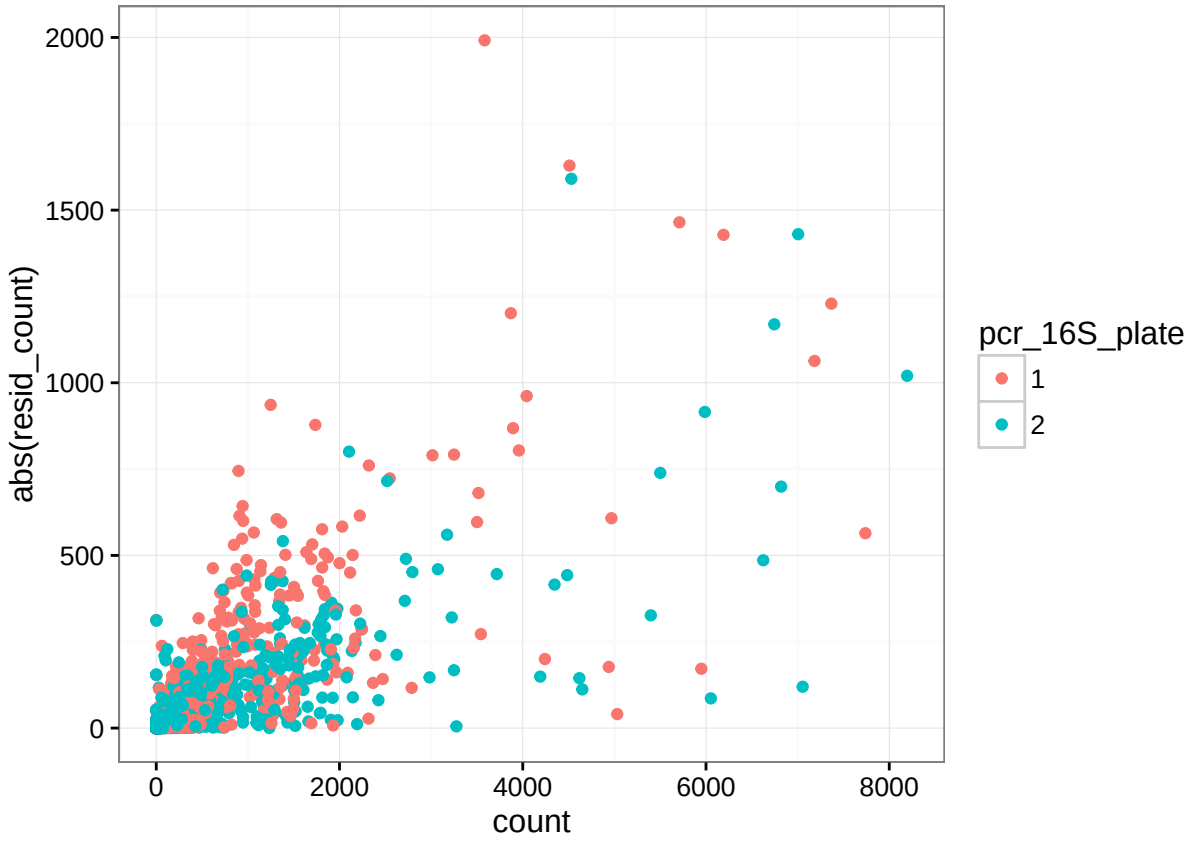
Visualizing Parameters Accounting for Variance

```
p <- ggplot(bio1_resid, aes(x = count, y = abs(resid_count))) + theme_bw()
```

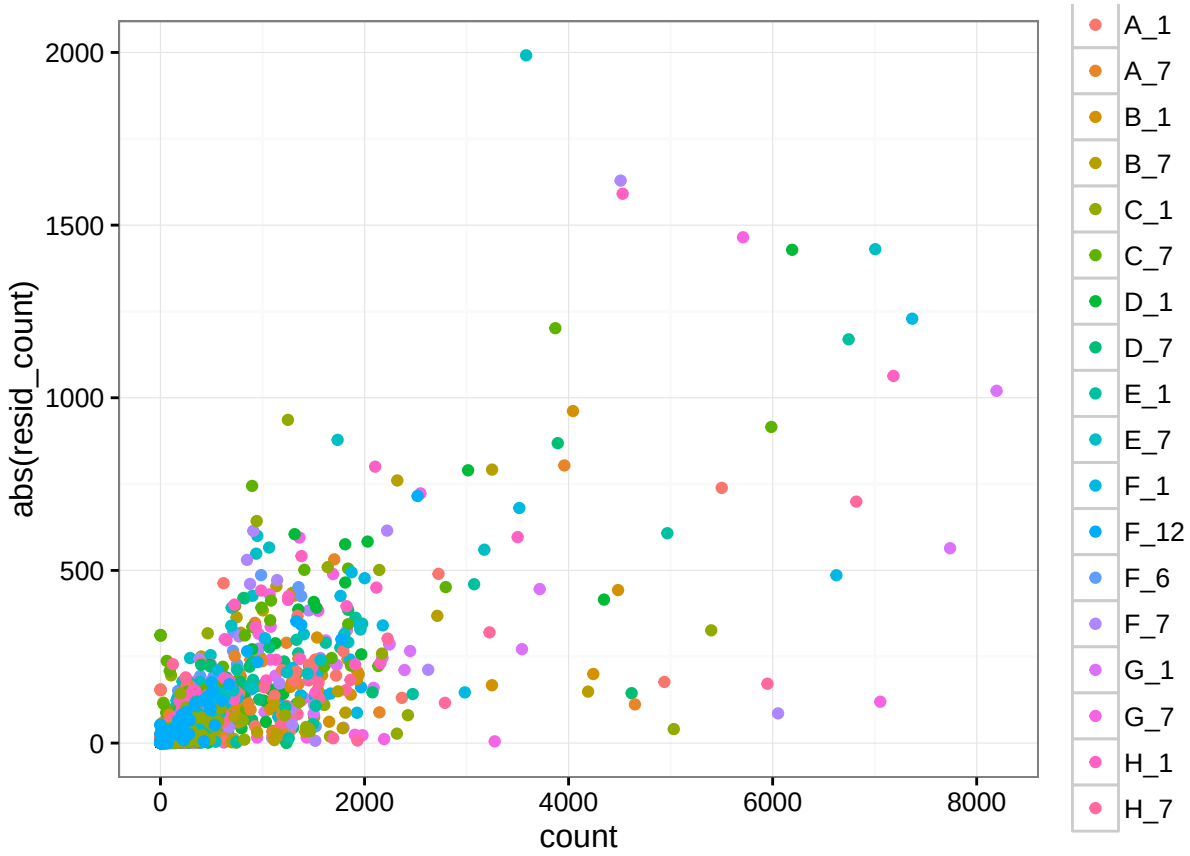
```
p + geom_point()
```



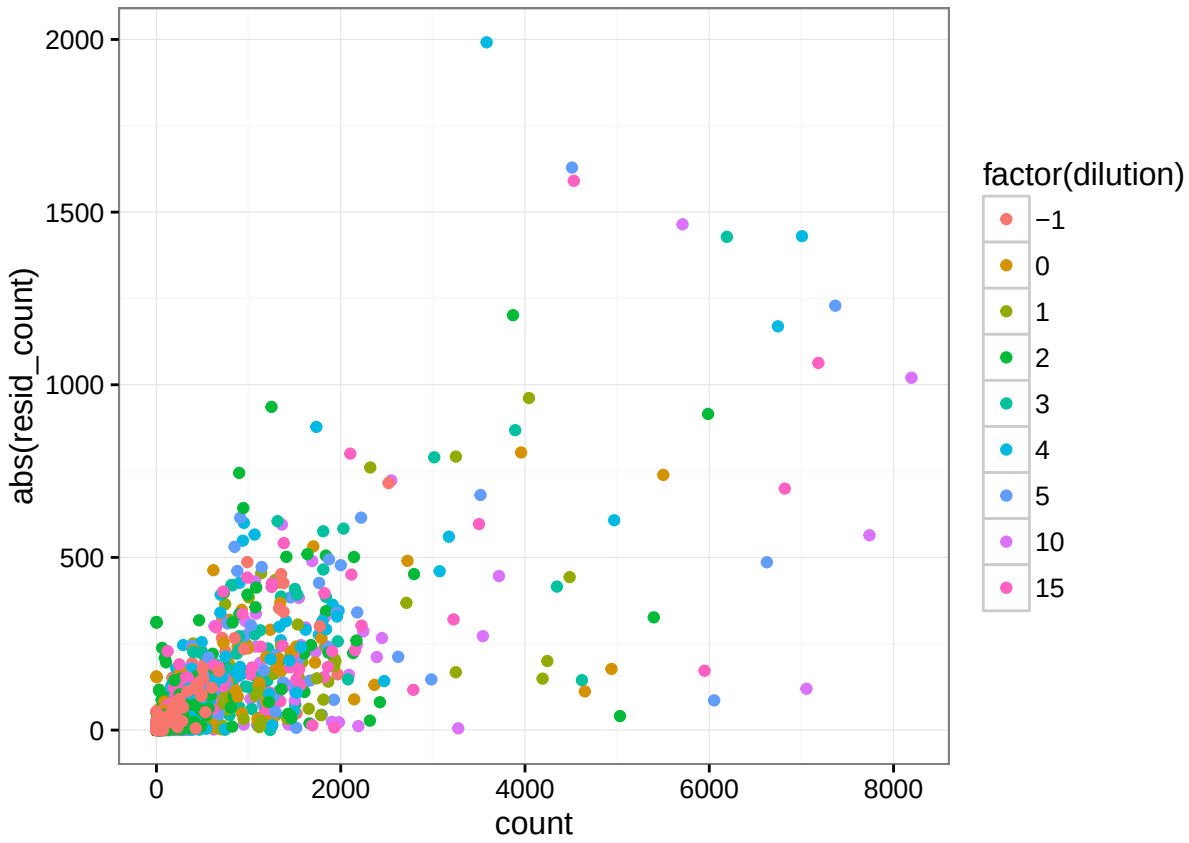

```
p + geom_point(aes(color = pcr_16S_plate))
```



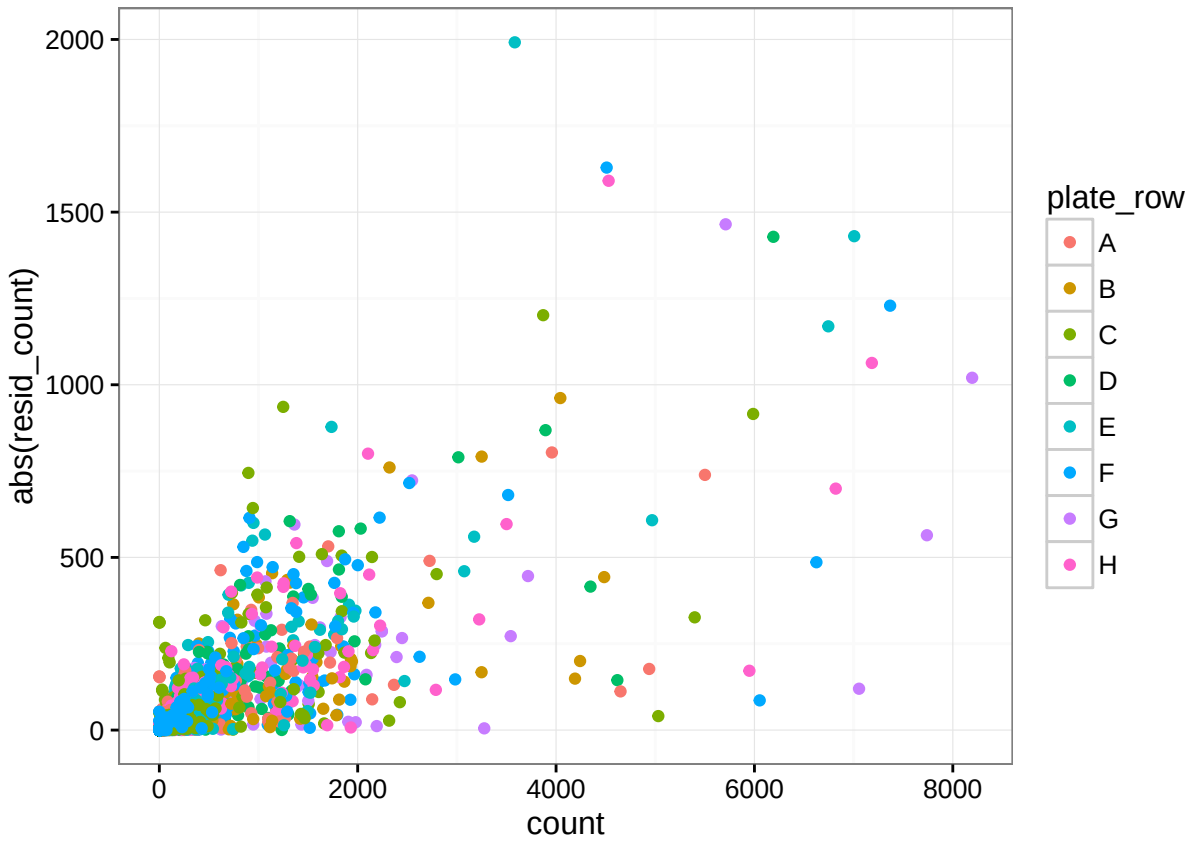

```
p + geom_point(aes(color = pos))
```



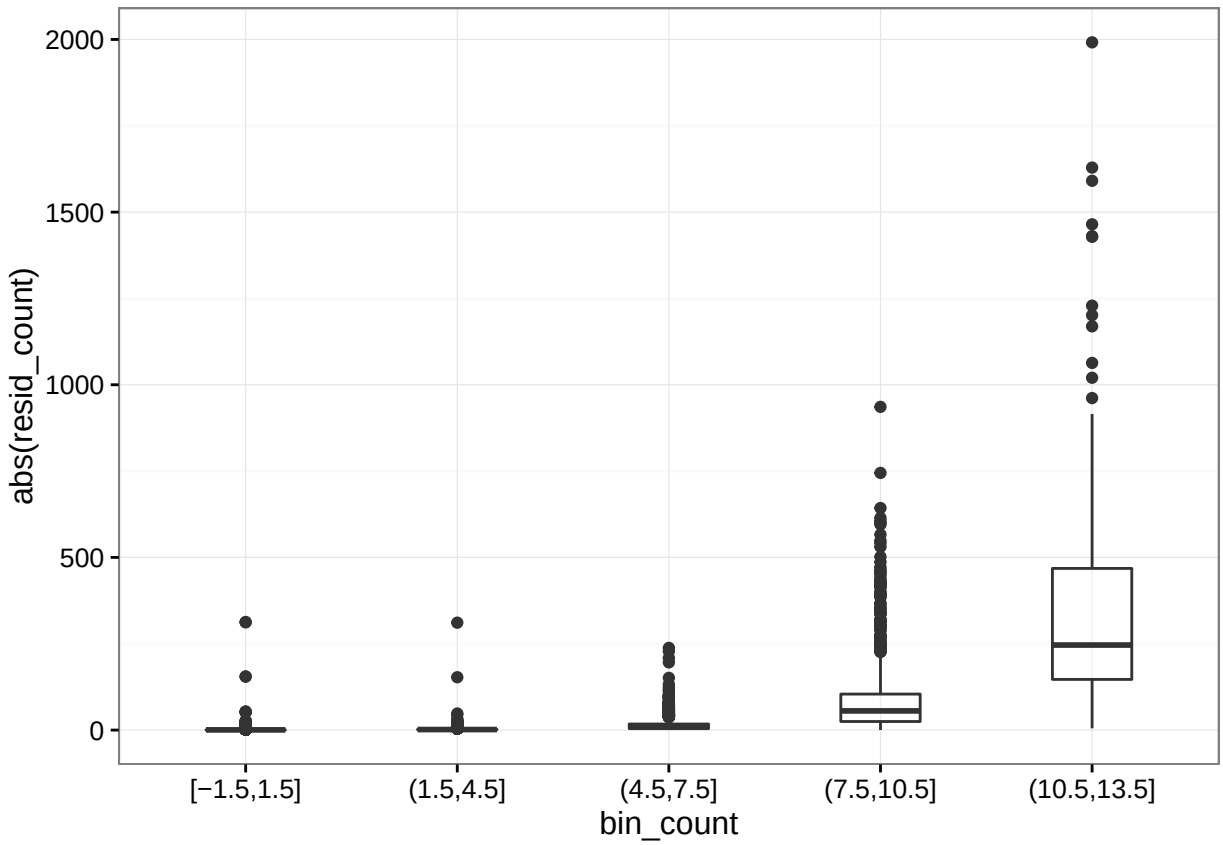

```
p + geom_point(aes(color = factor(dilution)))
```




```
p + geom_point(aes(color = plate_row))
```




```
bio1_resid %>% ungroup() %>%
      mutate(bin_count = cut_width(log2(count + 1), width = 3)) %>%
      ggplot() +
      geom_boxplot(aes(x = bin_count, y = abs(resid_count)),
                   alpha = 0.25, width = 0.5) + theme_bw()
```



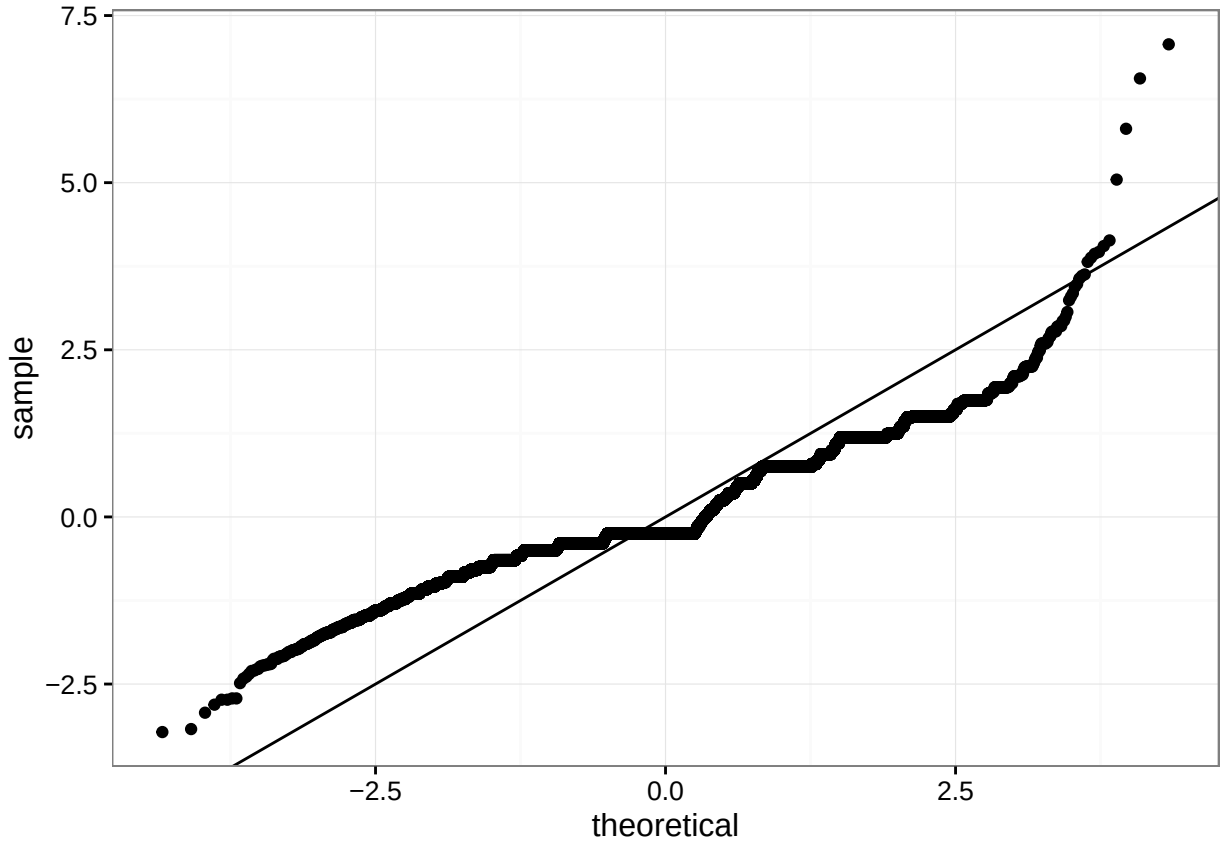
Normalized Residuals

```
bio1_log2_resid <- qiime_log2_df %>% get_resid_df()
bio1_css_resid <- qiime_css_df %>% get_resid_df()
bio1_csslog2_resid <- qiime_csslog2_df %>% get_resid_df()
```

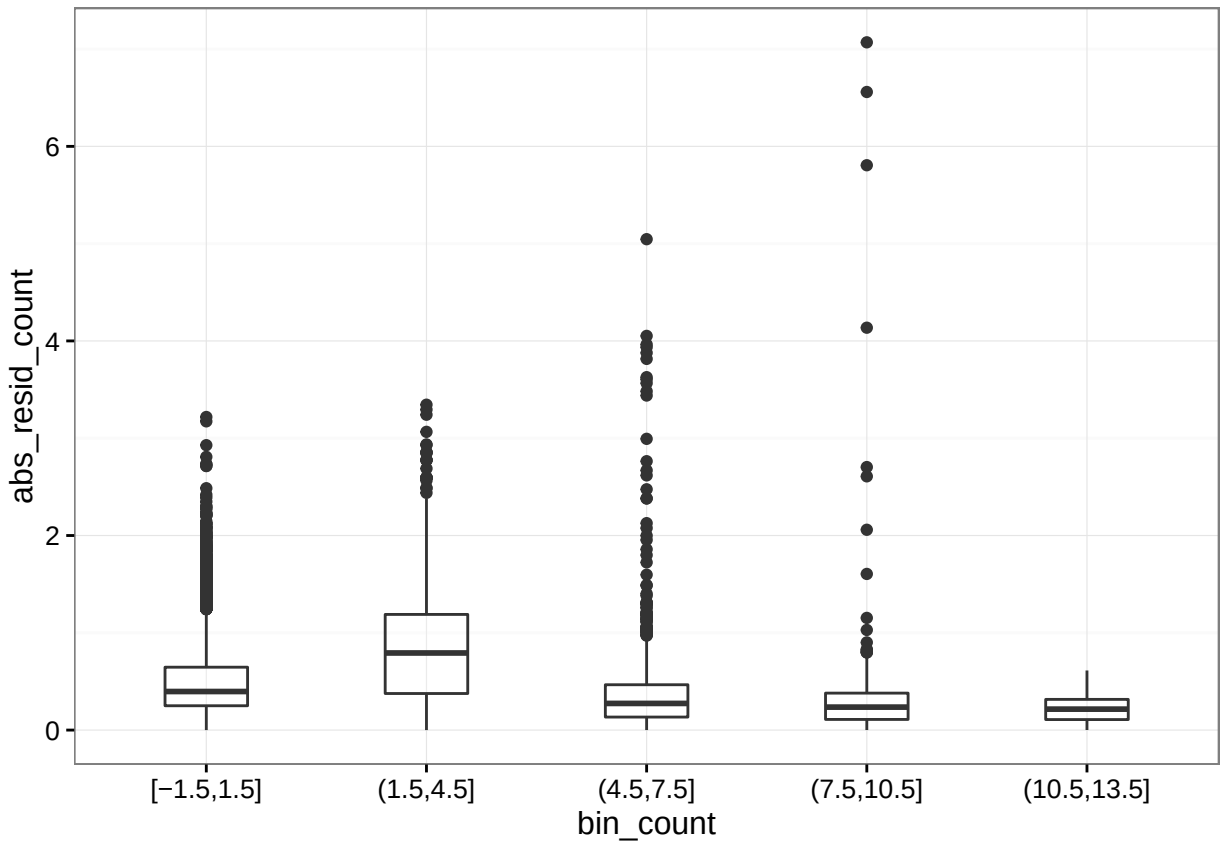
```
bio1_log2_min10_resid <- qiime_log2_df %>% get_resid_df(min_total = 10)
bio1_css_min10_resid <- qiime_css_df %>% get_resid_df(min_total = 10)
bio1_csslog2_min10_resid <- qiime_csslog2_df %>% get_resid_df(min_total = 10)
```

Residuals Log2 Transformed Counts

```
ggplot(bio1_log2_resid) + geom_qq(aes(sample = resid_count)) + geom_abline(aes(slope = 1, intercept = 0)
```

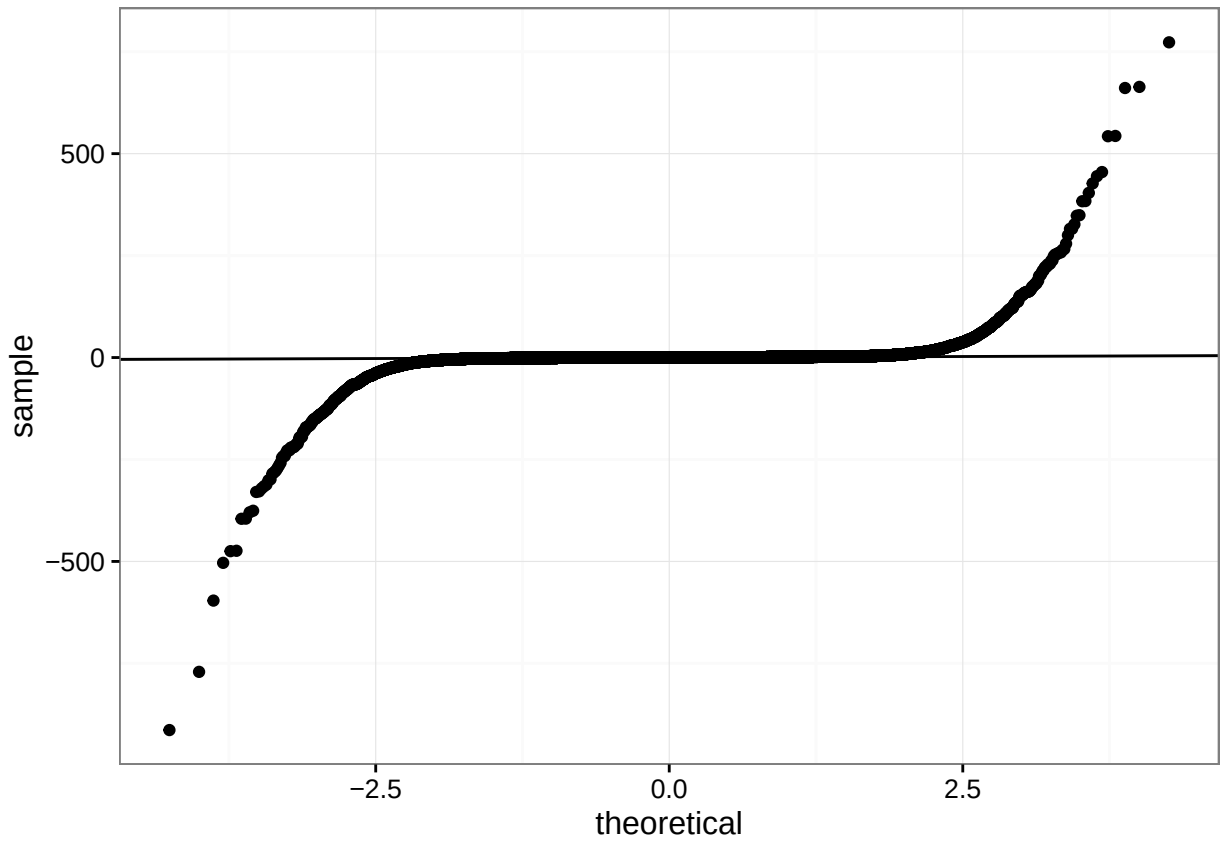



```
bio1_log2_resid %>% ungroup() %>%
      mutate(bin_count = cut_width(count, width = 3)) %>%
      ggplot() +
      geom_boxplot(aes(x = bin_count, y = abs_resid_count),
                   alpha = 0.25, width = 0.5) + theme_bw()
```

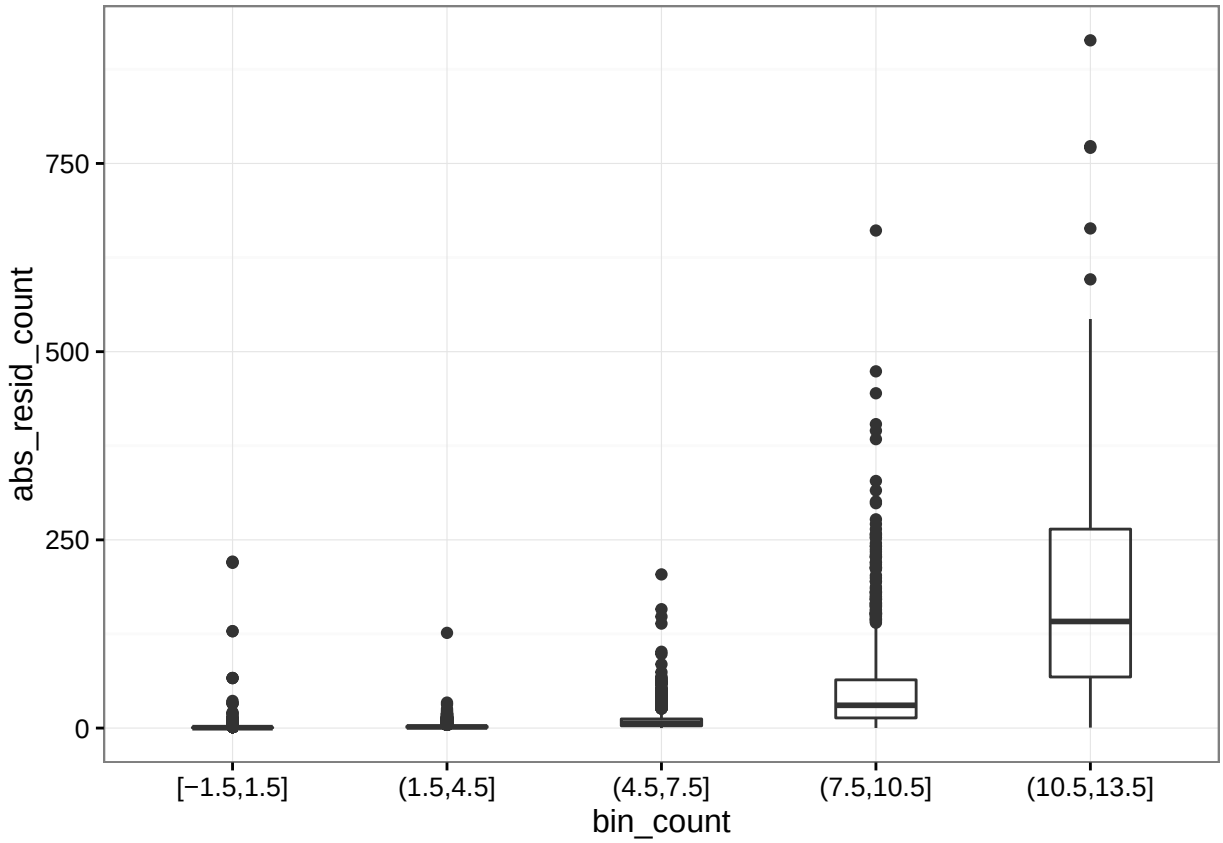


Residuals CSS Normalization Counts

```
ggplot(bio1_css_resid) + geom_qq(aes(sample = resid_count)) + geom_abline(aes(slope = 1, intercept = 0))
```



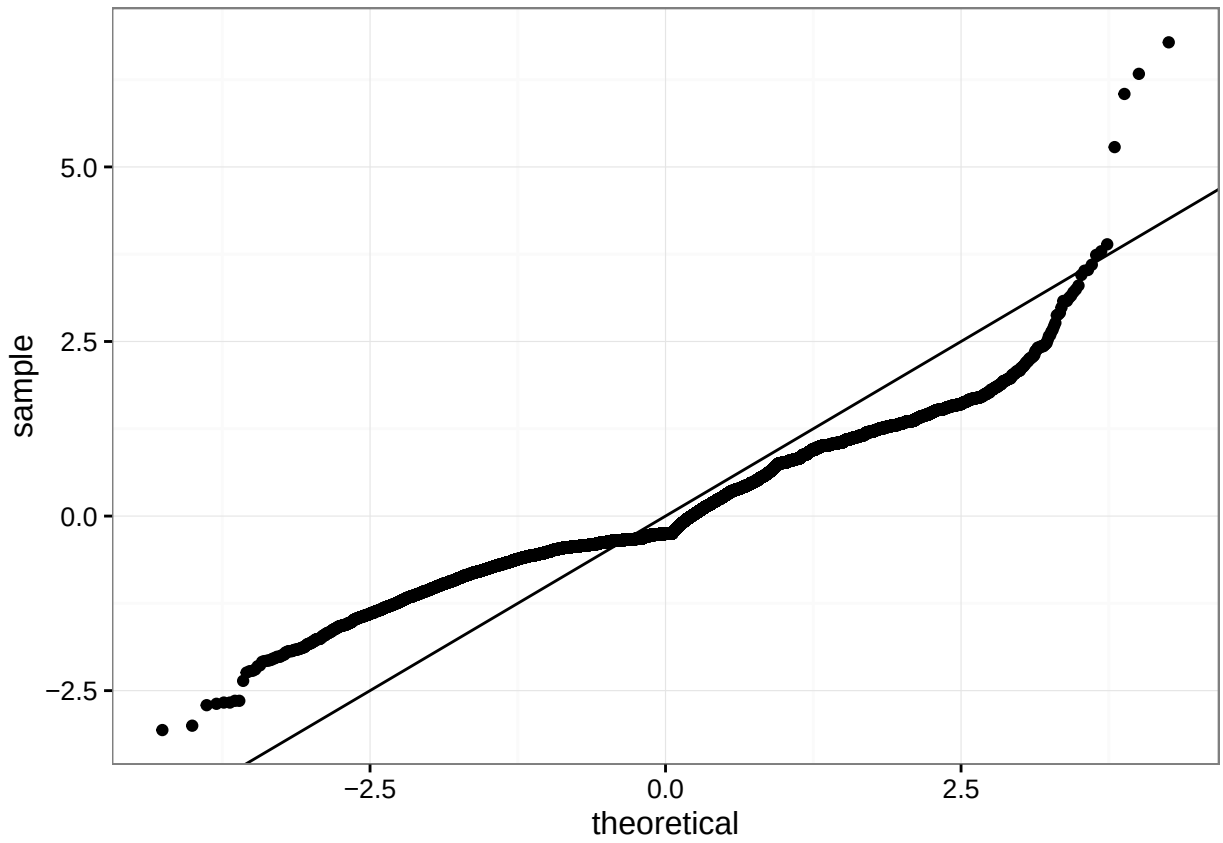

```
bio1_css_resid %>% ungroup() %>%
      mutate(bin_count = cut_width(log2(count+1), width = 3)) %>%
      ggplot() +
      geom_boxplot(aes(x = bin_count, y = abs_resid_count),
                   alpha = 0.25, width = 0.5) + theme_bw()
```



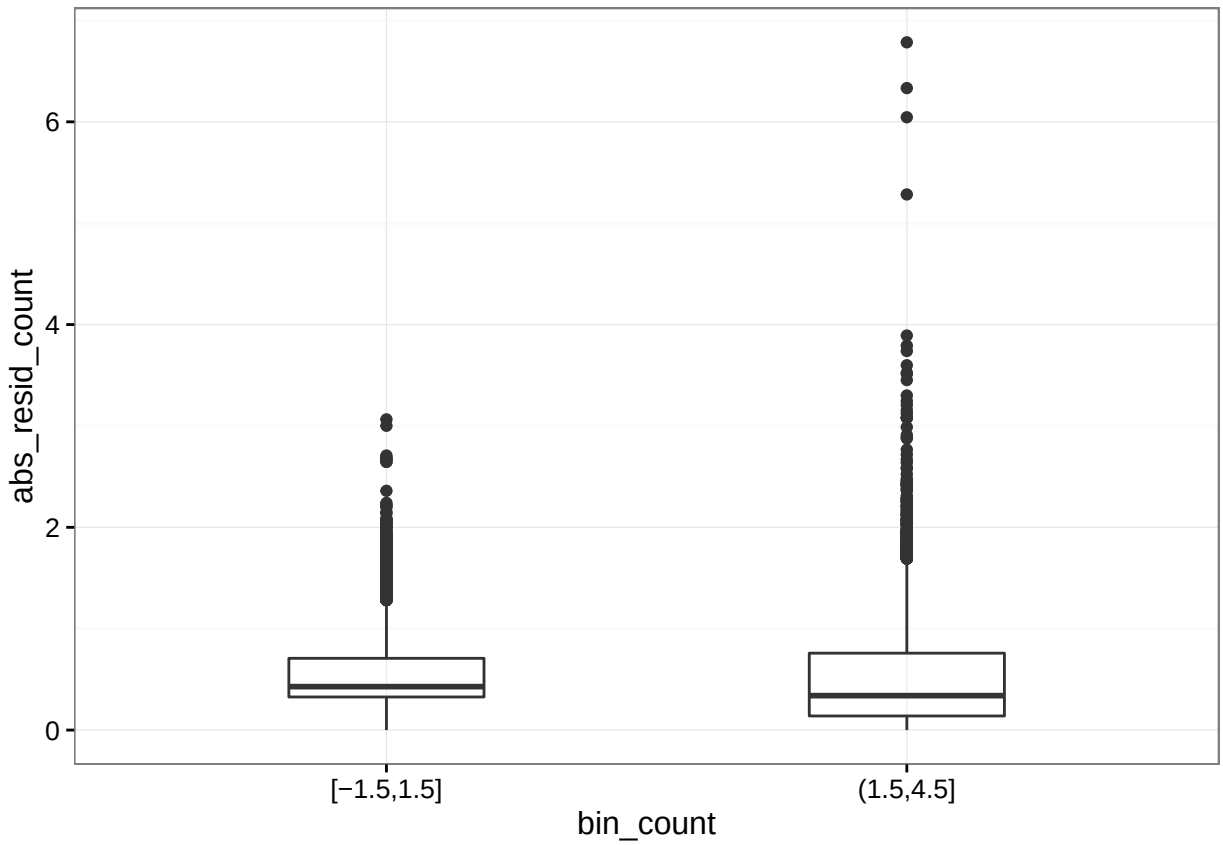
Residuals CSS Normalization and Log2 Transformed Counts

Residuals are still not normally distributed.

```
ggplot(bio1_csslog2_resid) + geom_qq(aes(sample = resid_count)) + geom_abline(aes(slope = 1, intercept =
```

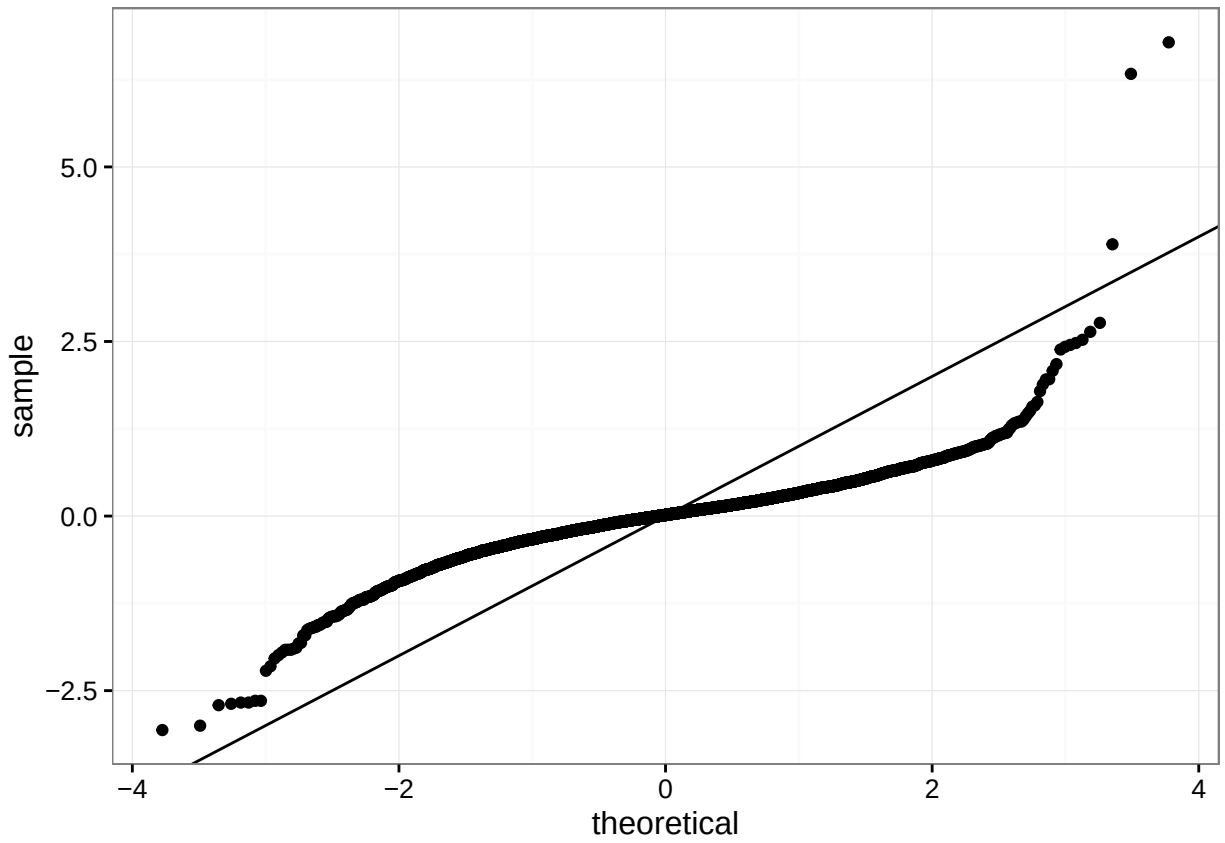



```
bio1_csslog2_resid %>% ungroup() %>%
      mutate(bin_count = cut_width(log2(count+1), width = 3)) %>%
      ggplot() +
      geom_boxplot(aes(x = bin_count, y = abs_resid_count),
                   alpha = 0.25, width = 0.5) + theme_bw()
```



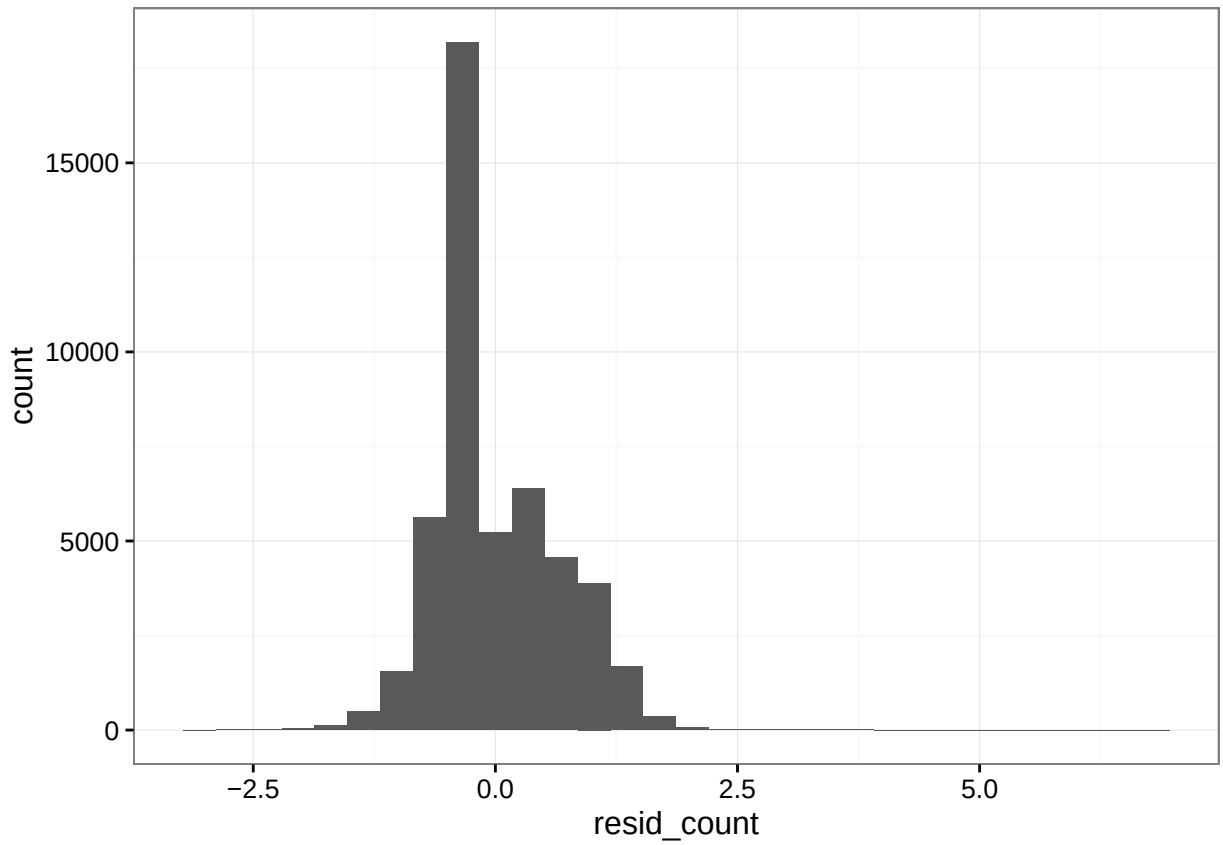
Excluding OTUs with total counts < 10

```
ggplot(bio1_csslog2_min10_resid) + geom_qq(aes(sample = resid_count)) + geom_abline(aes(slope = 1, inte
```



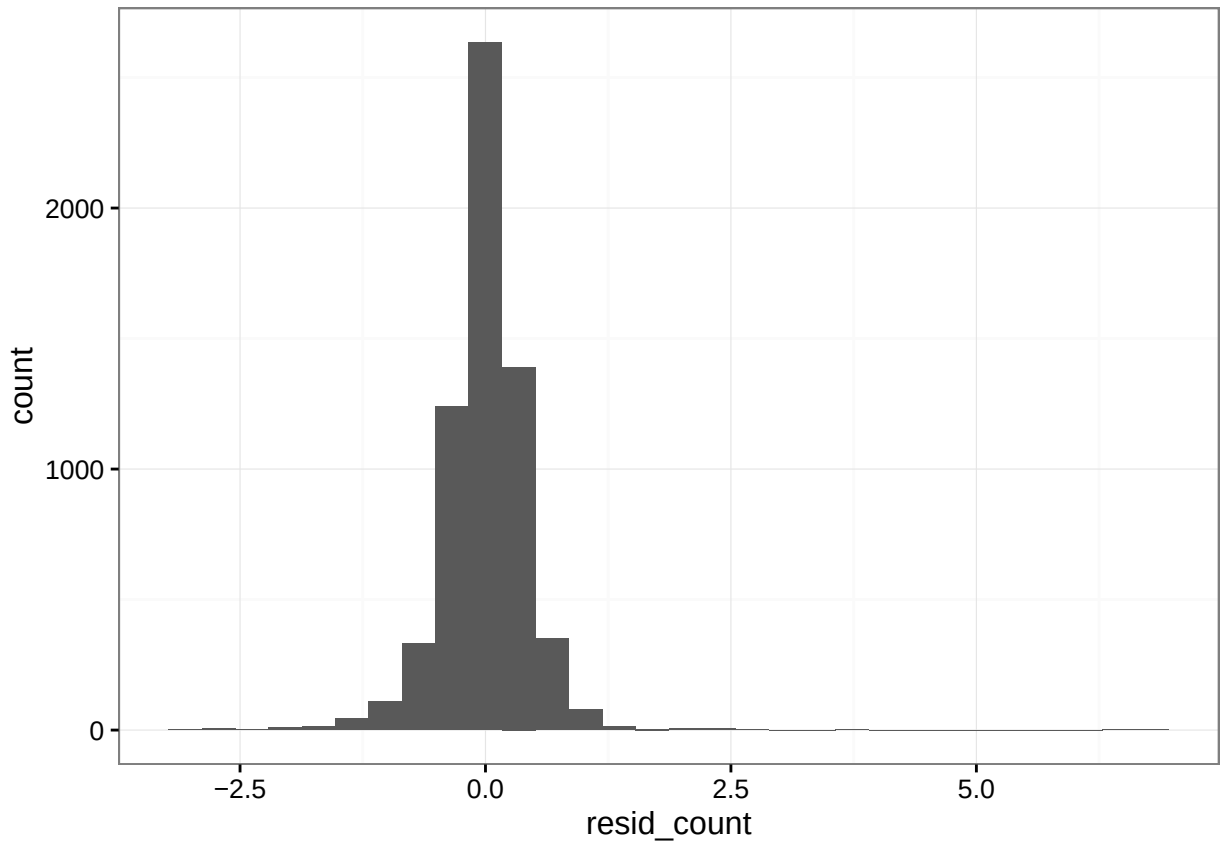

```
ggplot(bio1_csslog2_resid) + geom_histogram(aes(x = resid_count)) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



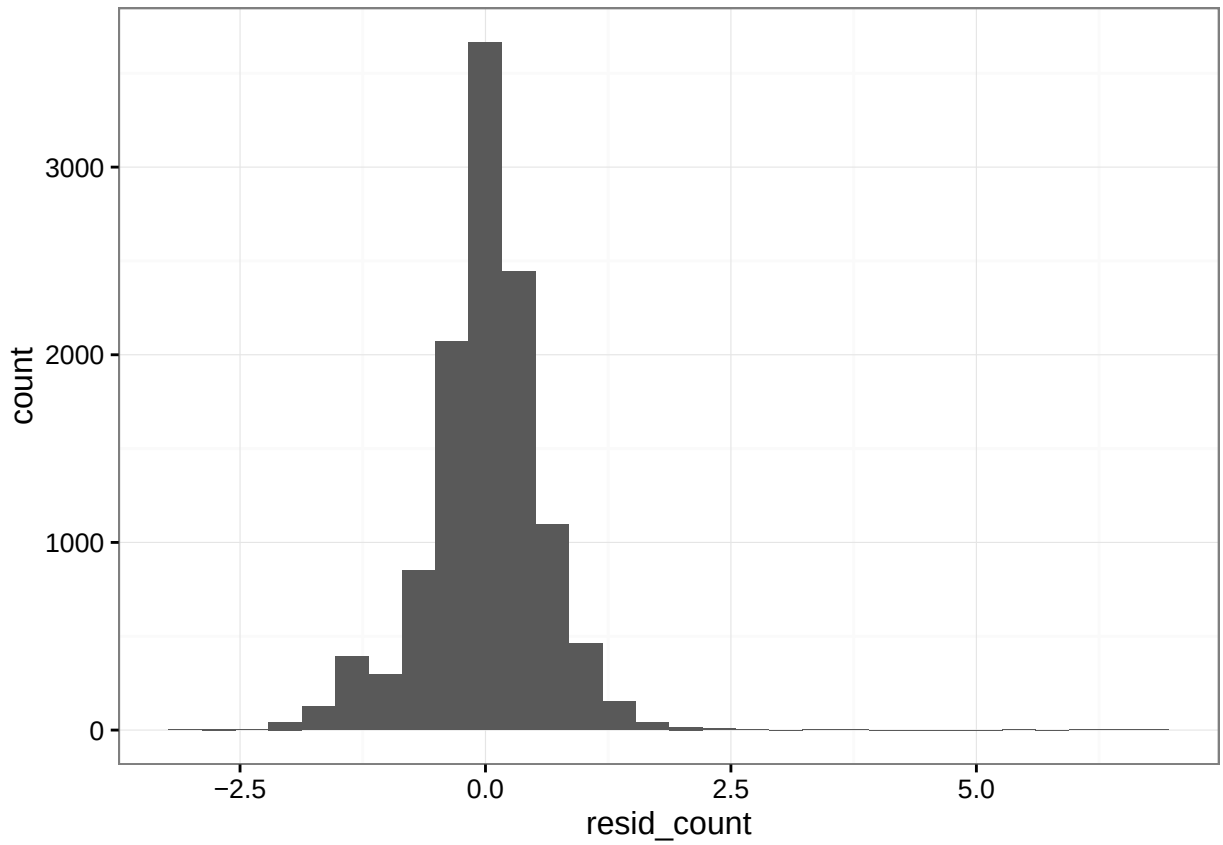

```
ggplot(bio1_csslog2_min10_resid) + geom_histogram(aes(x = resid_count)) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



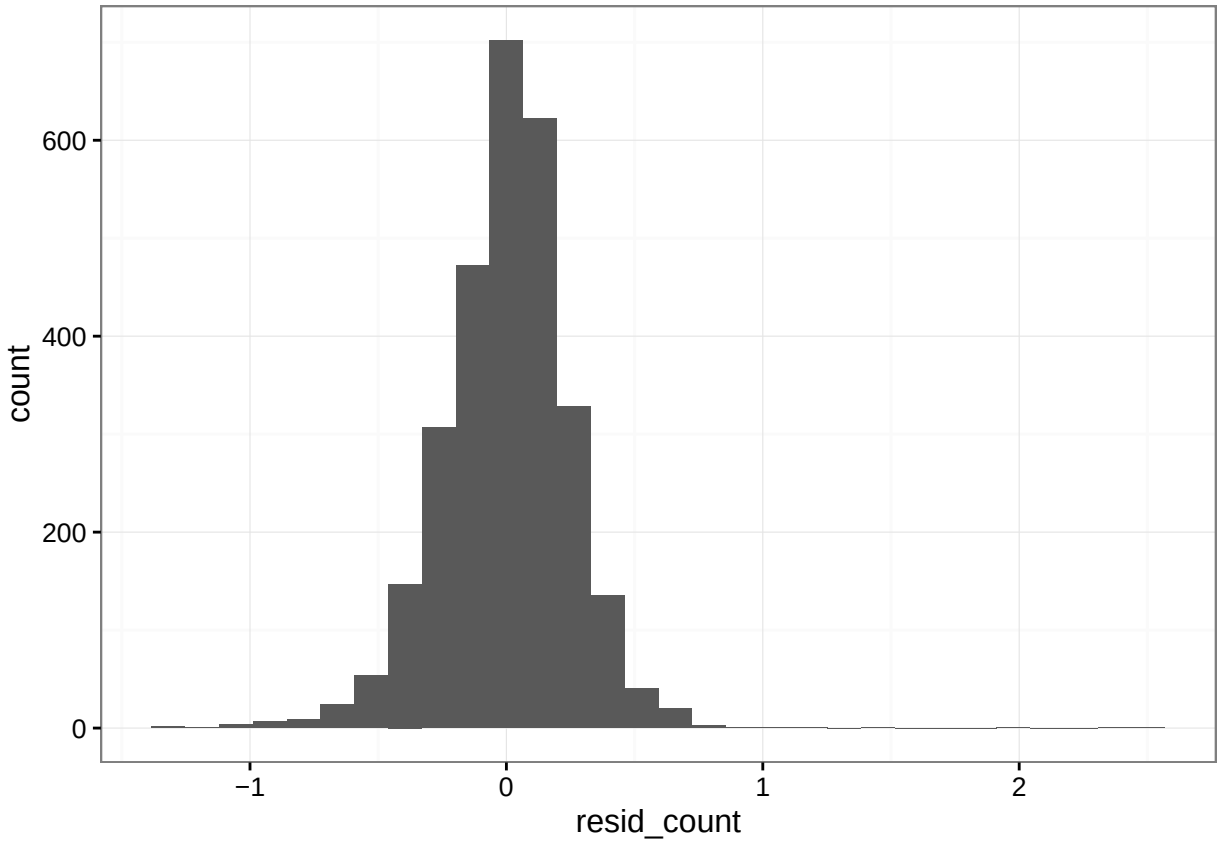

```
qiime_csslog2_df %>% get_resid_df(min_total = 5) %>%
      ggplot() + geom_histogram(aes(x = resid_count)) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qiime_csslog2_df %>% get_resid_df(min_total = 20) %>%
      ggplot() + geom_histogram(aes(x = resid_count)) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Characterizing sources of variance

Non-transformed counts, the raw count value accounts for most of the observed variance.

```
fit <- lm(abs_resid_count ~ 0 + count,bio1_resid)
```

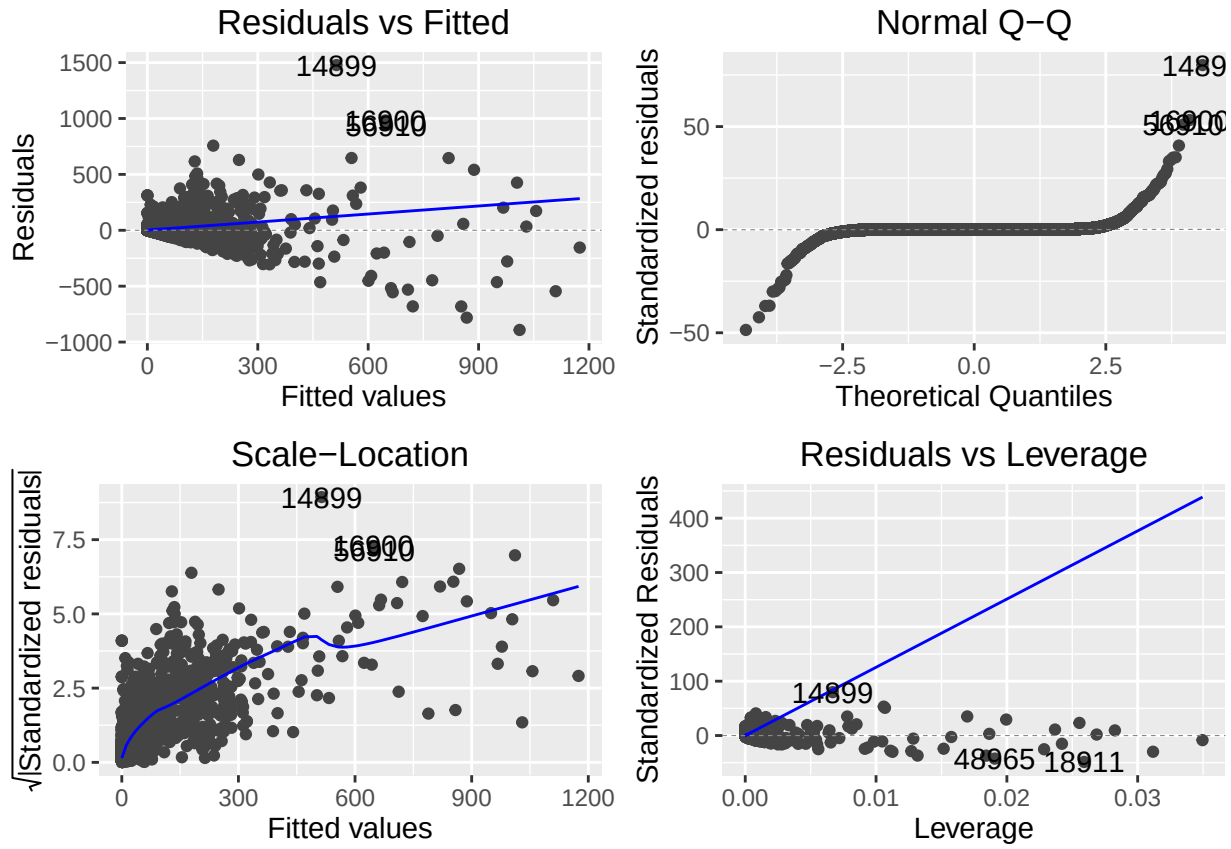
```
summary(fit)
```

```
##
## Call:
## lm(formula = abs_resid_count ~ 0 + count, data = bio1_resid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -891.74    0.25    0.50    0.75 1477.91
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## count 0.1433718  0.0004238   338.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.58 on 69499 degrees of freedom
## Multiple R-squared:  0.6221, Adjusted R-squared:  0.6221
## F-statistic: 1.144e+05 on 1 and 69499 DF,  p-value: < 2.2e-16
```

```
aov(fit) %>% summary()
```

```
##               Df   Sum Sq  Mean Sq F value Pr(>F)
## count          1 39489707 39489707  114429 <2e-16 ***
## Residuals  69499 23984268      345
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
autoplot(fit)
```



Log2 Transformed Values

```
fit <- lm(abs_resid_count ~ 0 + count,bio1_log2_resid)
```

```
summary(fit)
```

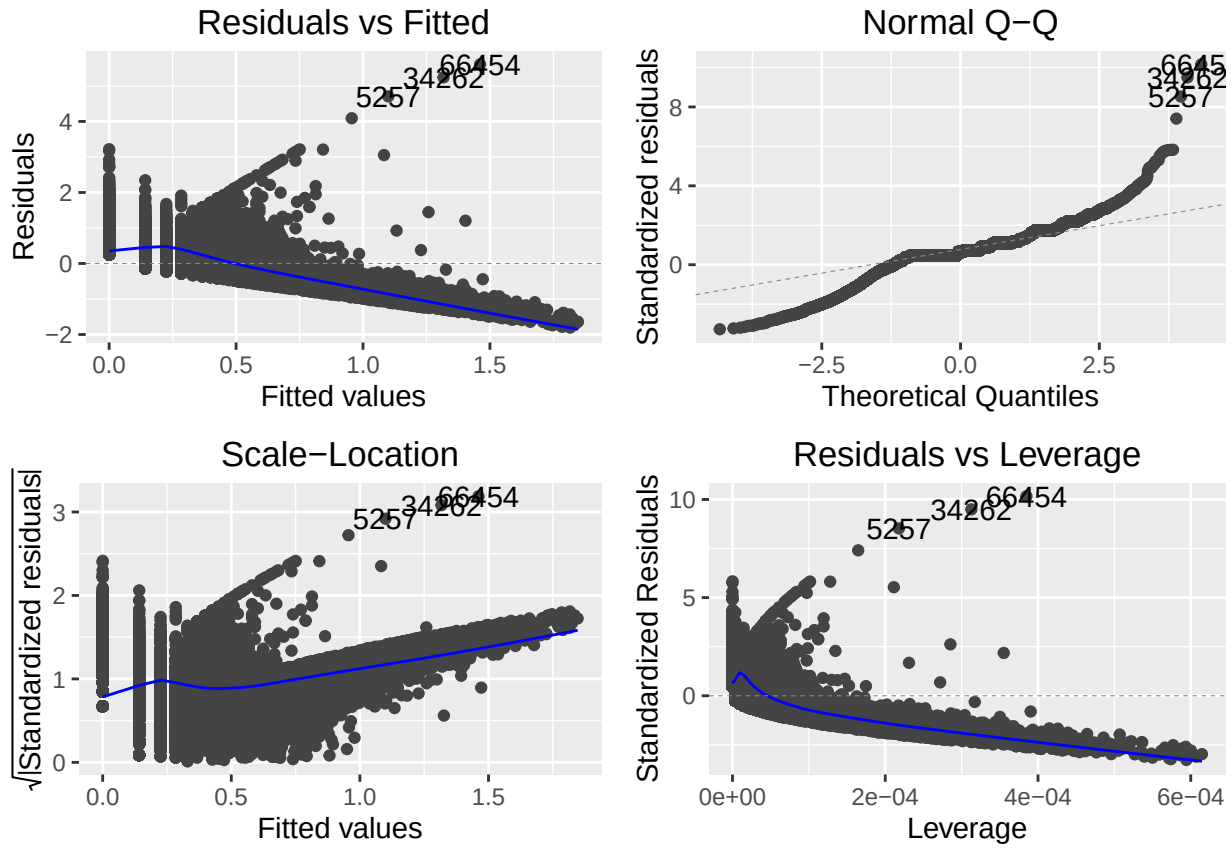
```
##
## Call:
## lm(formula = abs_resid_count ~ 0 + count, data = bio1_log2_resid)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.805  0.250  0.358  0.608  5.608
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## count 0.142038   0.001053   134.9   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5522 on 69499 degrees of freedom
## Multiple R-squared:  0.2075, Adjusted R-squared:  0.2075
## F-statistic: 1.82e+04 on 1 and 69499 DF,  p-value: < 2.2e-16
```

```
aov(fit) %>% summary()
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## count         1   5548    5548   18195 <2e-16 ***
## Residuals 69499  21192       0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
autoplot(fit)
```



CSS normalization

CSS normalization does not address issue regarding correlation between count and residuals.

```
fit <- lm(abs_resid_count ~ 0 + count,bio1_css_resid)
```

```
summary(fit)
```

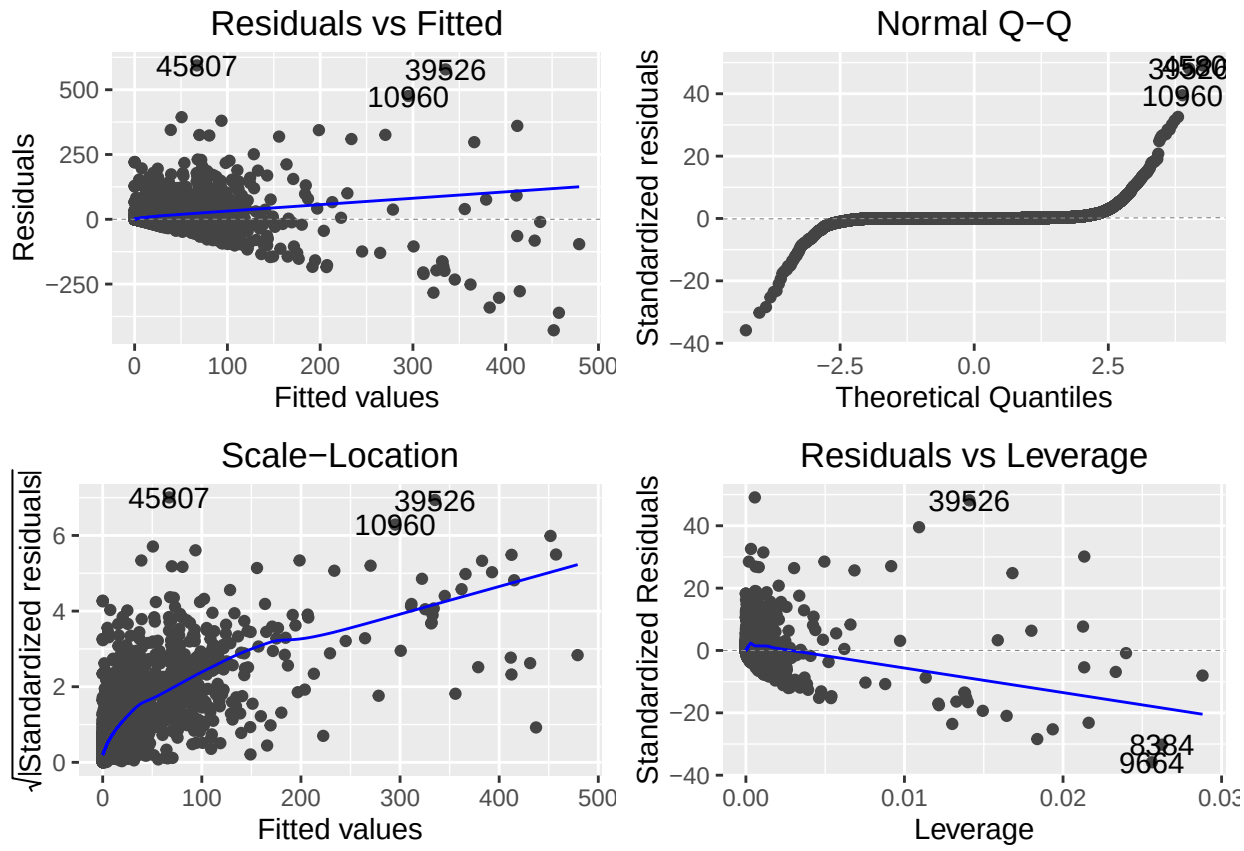
```
##
## Call:
## lm(formula = abs_resid_count ~ 0 + count, data = bio1_css_resid)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -428.25    0.34    0.55    1.05  593.85
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## count 0.0760693  0.0003259   233.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.1 on 48251 degrees of freedom
## Multiple R-squared:  0.5303, Adjusted R-squared:  0.5302
## F-statistic: 5.447e+04 on 1 and 48251 DF,  p-value: < 2.2e-16
```

```
aov(fit) %>% summary()
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## count         1 7972718 7972718   54467 <2e-16 ***
## Residuals 48251 7062868     146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
autoplot(fit)
```



Log2 Transformed and CSS Normalization

```r
fit <- lm(abs_resid_count ~ 0 + count, bio1_csslog2_resid)
```

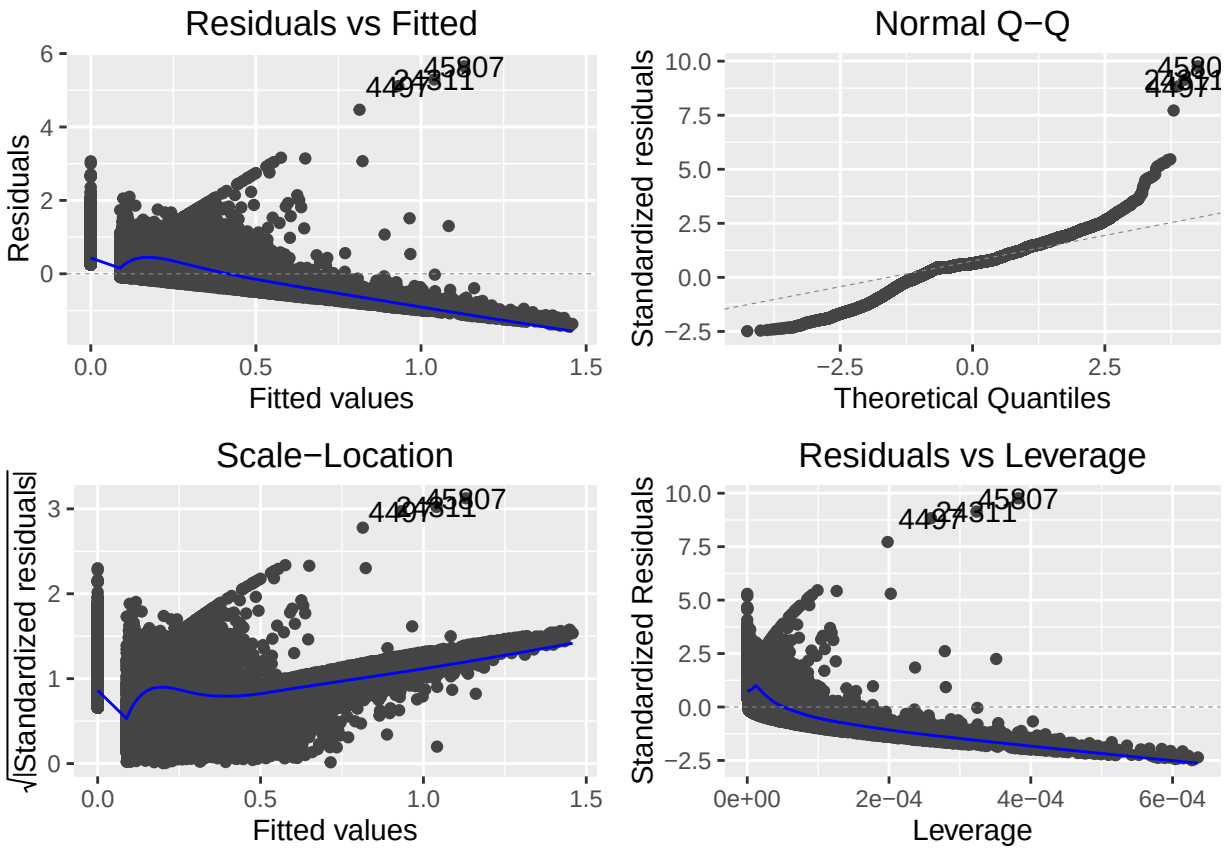
```r
summary(fit)
```

```
##
## Call:
## lm(formula = abs_resid_count ~ 0 + count, data = bio1_csslog2_resid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4404  0.2518  0.3717  0.6221  5.6539
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## count 0.115531   0.001157   99.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5792 on 48251 degrees of freedom
## Multiple R-squared:  0.1712, Adjusted R-squared:  0.1712
## F-statistic:  9970 on 1 and 48251 DF,  p-value: < 2.2e-16
```

```r
aov(fit) %>% summary()
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## count         1   3345    3345    9970 <2e-16 ***
## Residuals 48251  16187       0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
autoplot(fit)
```



Plot of raw bias - observed vs. expected based on dilutions