Goals

We have a dataset contain axon diameters of neurons from the optic nerve of control and mutant zebrafish. We'd like to know if the mean axon diameter, or the distribution of axon diameters, differs between groups. We want to implement tests that take account of within and between animal variance.

Here, we focus on the myRF data set.

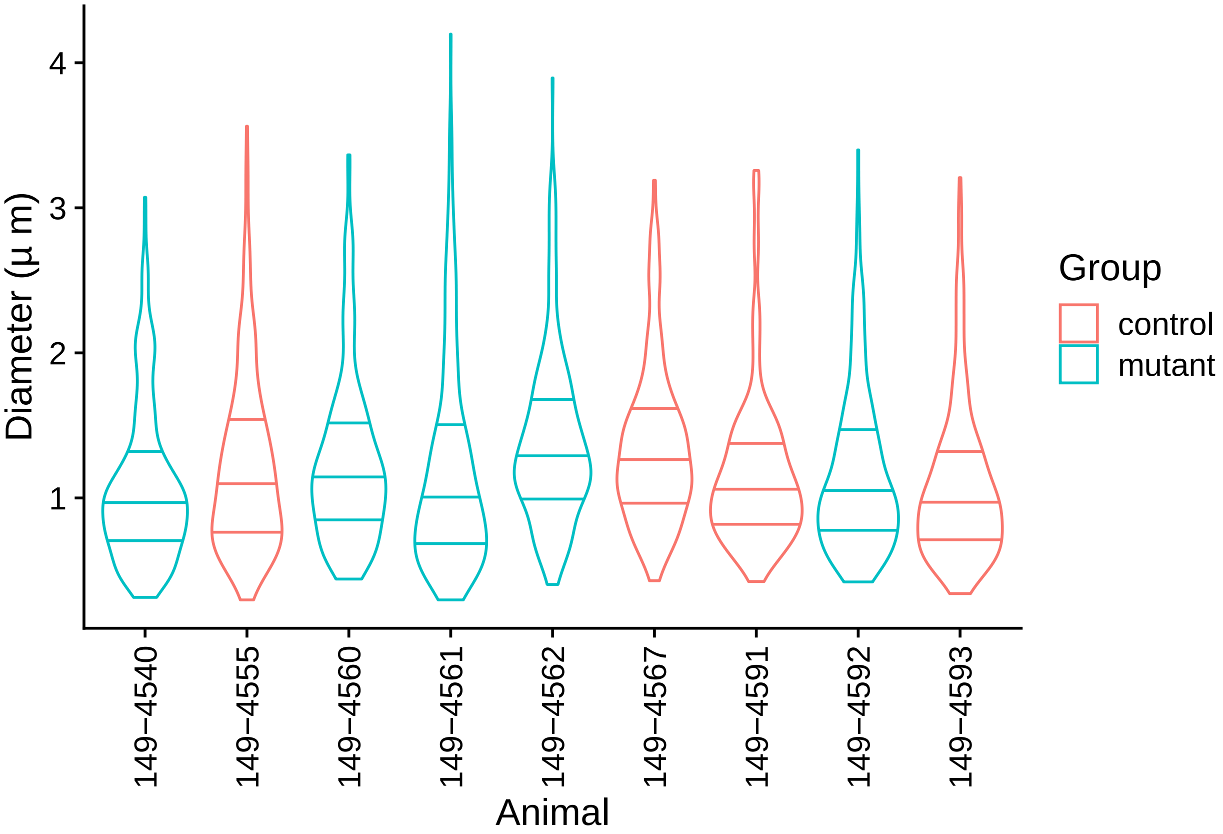
Load and format data

```
df <- read_excel("Shiverer Axon Diameter.xlsx", range = "A2:J293") %>%
    pivot_longer(cols = 1:10) %>%
    drop_na() %>%
    mutate(group = as_factor(ifelse(name %in% c("149-4555", "149-4567", "149-4591", "149-4593"), "contr
```

Plot the data

Focus here on plot of individual mice, colour coded by group.

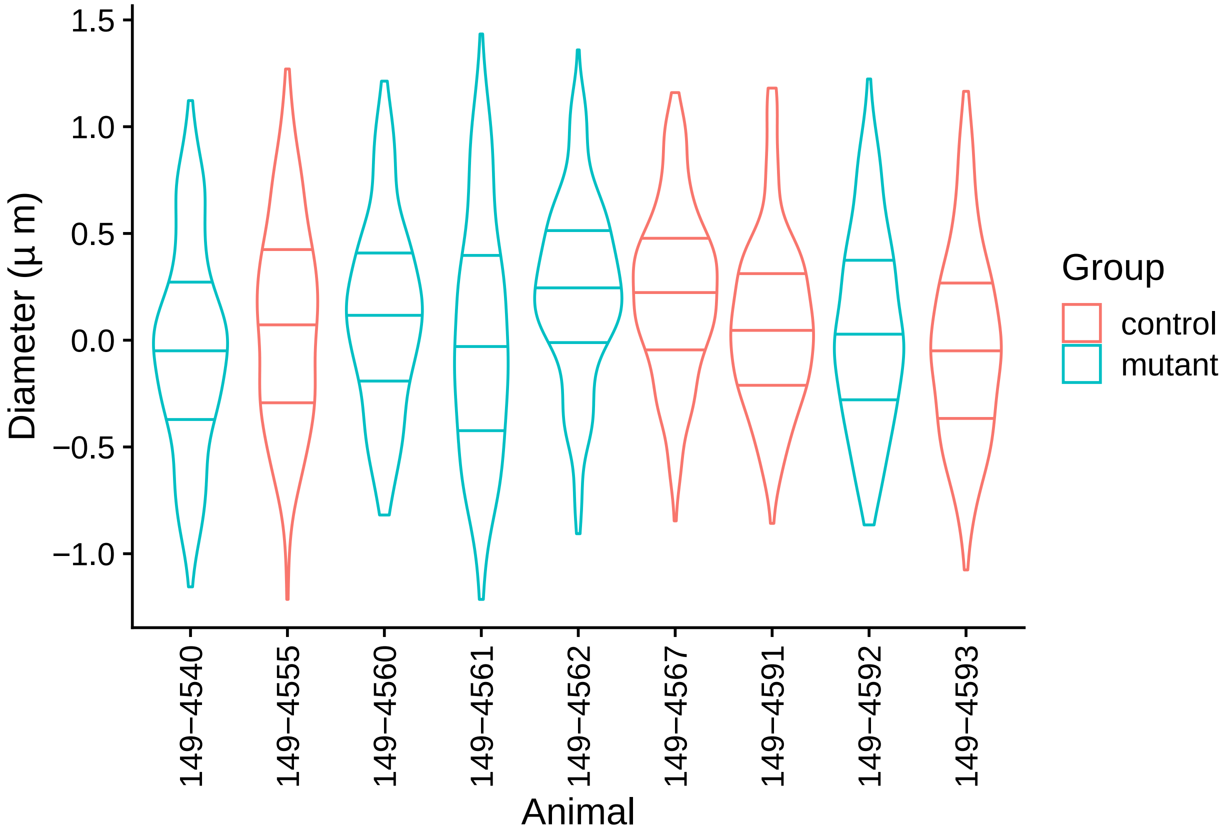
```
(plot_by_id <- ggplot(data = df, aes(name, value)) +
     geom_violin(aes(colour = group), draw_quantiles = c(0.25, 0.5, 0.75)) +
     theme_cowplot(font_size = 14) +
     theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
    labs(x = 'Animal', y = 'Diameter (\u00B5 m)', colour = "Group"))
```



```
ggsave('Plots/violins_shiv.jpeg', plot_by_id)
```

```
## Saving 6.5 x 4.5 in image
```

```
(plot_by_id <- ggplot(data = df, aes(name, log(value))) +
     geom_violin(aes(colour = group), draw_quantiles = c(0.25, 0.5, 0.75)) +
     theme_cowplot(font_size = 14) +
     theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
    labs(x = 'Animal', y = 'Diameter (\u00B5 m)', colour = "Group"))
```



```
ggsave('Plots/violins_log_shiv.jpeg', plot_by_id)
```

```
## Saving 6.5 x 4.5 in image
```

Tests for differences in means

```
mm_t <- lmer(value ~ group + (1 | name), data = df)
mm_t_null <- lmer(value ~ (1 | name), data = df)
summary(mm_t)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: value ~ group + (1 | name)
##    Data: df
##
## REML criterion at convergence: 4273.8
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6545 -0.7044 -0.2186  0.4247  5.1681
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  name     (Intercept) 0.01154  0.1074
##  Residual             0.33984  0.5830
## Number of obs: 2415, groups:  name, 9
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 1.206465   0.056751   21.26
## groupmutant 0.005307   0.075981    0.07
##
## Correlation of Fixed Effects:
##             (Intr)
## groupmutant -0.747
```

```
anova(mm_t, mm_t_null)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df
## Models:
## mm_t_null: value ~ (1 | name)
## mm_t: value ~ group + (1 | name)
##           npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mm_t_null    3 4271.6 4288.9 -2132.8   4265.6
## mm_t         4 4273.5 4296.7 -2132.8   4265.5 0.0058  1     0.9394
```

Same test but with log transformed data

```
mm_t <- lmer(log(value) ~ group + (1 | name), data = df)
mm_t_null <- lmer(log(value) ~ (1 | name), data = df)
summary(mm_t)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(value) ~ group + (1 | name)
##    Data: df
##
## REML criterion at convergence: 3086.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.84380 -0.68071 -0.02389  0.63836  3.12308
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  name     (Intercept) 0.01067  0.1033
##  Residual             0.20754  0.4556
## Number of obs: 2415, groups:  name, 9
##
## Fixed effects:
```

```
##             Estimate Std. Error t value
## (Intercept)  0.08933    0.05360   1.667
## groupmutant -0.01302    0.07181  -0.181
##
## Correlation of Fixed Effects:
##             (Intr)
## groupmutant -0.746
```

```r
anova(mm_t, mm_t_null)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df
## Models:
## mm_t_null: log(value) ~ (1 | name)
## mm_t: log(value) ~ group + (1 | name)
##           npar  AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## mm_t_null    3 3084 3101.4  -1539     3078
## mm_t         4 3086 3109.2  -1539     3078 0.0429  1      0.836
```

Generate histograms for all animals

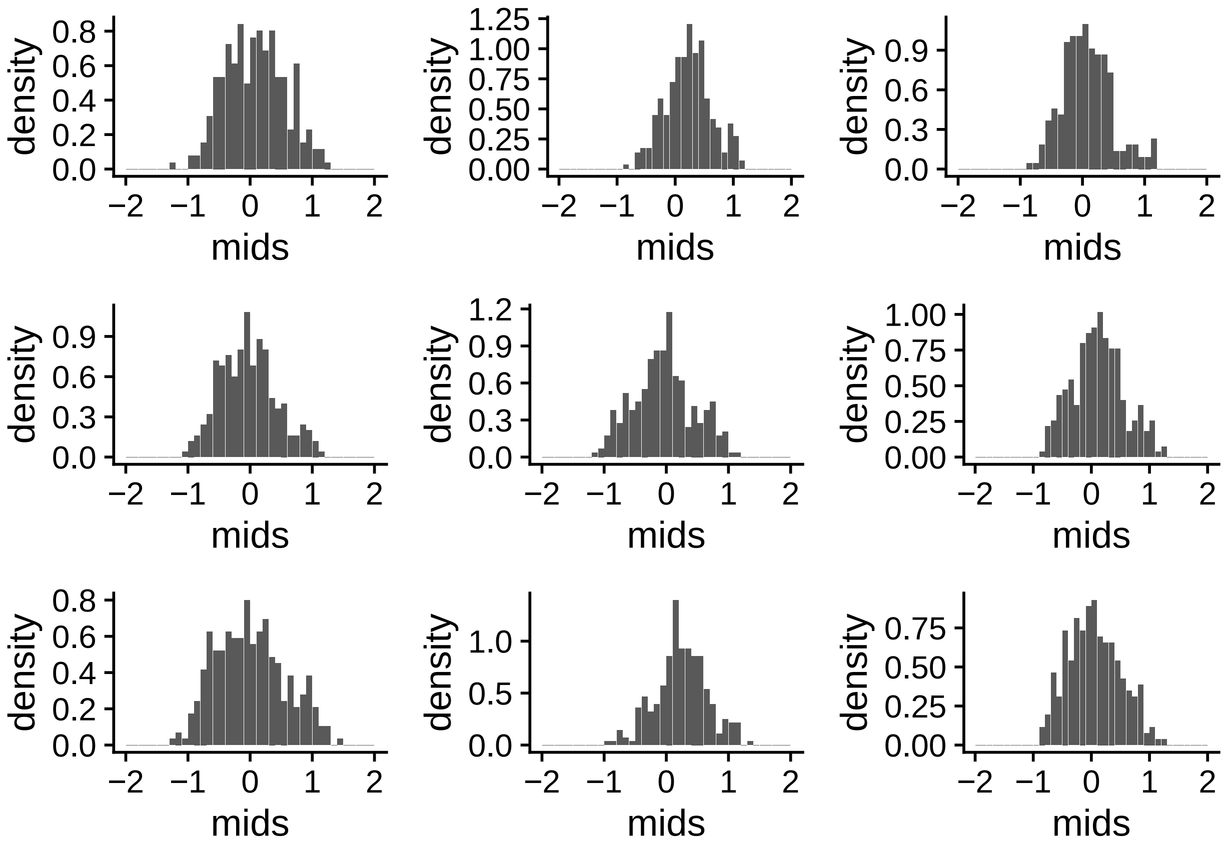
```r
### Make histograms for each animal
### Use log transformed data
names = unique(df$name)
### If submean = 1 then will substract means before making histograms
histfun <- function(name, df, submean = 0) {
    sub <- ifelse(submean == 1, mean(df$value[df$name==name]), 0)
    hist(log(df$value[df$name==name]-sub), seq(-2,2,0.1), plot = FALSE)
}
hists <- sapply(names, histfun, df, submean=0)

### Convert results to tibble for use with tidyverse functions
rns <- rownames(hists)
hists_tib <- as_tibble(hists) %>%
    rownames_to_column(var = "rowname") %>%
    pivot_longer(-rowname, names_to = "column", values_to = "value") %>%
    pivot_wider(names_from = rowname, values_from = value)
colnames(hists_tib) <- c("animal", rns)


ggconvfun <- function(mids, density) {
    gghist <- cbind(mids, density)
    ggplot(gghist, aes(mids, density)) +
        geom_col() +
        theme_cowplot()
}

gghists <- map2(hists_tib$mids, hists_tib$density, ggconvfun)

plot_grid(plotlist = gghists)
```



Evaluate distributions

Use package lqmm (https://www.jstatsoft.org/article/view/v057i13) to evaluate potential differences in the distributions. Use Nelder-Mead optimization (derivative-free method) as gives fits with much narrower bounds on the intercept.

```r
fit.lqmm <- lqmm(fixed = value ~ group, random = ~1, group = name, tau = c(0.25,0.5, 0.75), nK = 7, typ
summary(fit.lqmm, R = 500, seed = 2)
```

```
## Call: lqmm(fixed = value ~ group, random = ~1, group = name, tau = c(0.25,
##     0.5, 0.75), nK = 7, type = "normal", data = df, control = c(method = "df",
##     LP_max_iter = 5000))
##
## tau = 0.25
##
## Fixed effects:
##                   Value Std. Error lower bound upper bound Pr(>|t|)
## (Intercept)  0.798005   0.086190    0.628666      0.9673   <2e-16 ***
## groupmutant -0.032016   0.098076   -0.224710      0.1607   0.7442
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## tau = 0.5
##
## Fixed effects:
##                   Value Std. Error lower bound upper bound Pr(>|t|)
## (Intercept)  1.094997   0.062838    0.971537      1.2185   <2e-16 ***
## groupmutant -0.028996   0.089871   -0.205567      0.1476   0.7471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## tau = 0.75
##
## Fixed effects:
##                  Value Std. Error lower bound upper bound Pr(>|t|)
## (Intercept)  1.461998   0.064582    1.335112      1.5889   <2e-16 ***
## groupmutant  0.034035   0.089207   -0.141233      0.2093    0.703
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## AIC:
## [1] 3768 (df = 4) 4214 (df = 4) 5318 (df = 4)
```

```
fit.log.lqmm <- lqmm(fixed = log(value) ~ group, random = ~1, group = name, tau = c(0.25,0.5, 0.75), nK
summary(fit.log.lqmm, R = 500, seed = 2)
```

```
## Call: lqmm(fixed = log(value) ~ group, random = ~1, group = name, tau = c(0.25,
##     0.5, 0.75), nK = 7, type = "normal", data = df, control = c(method = "df",
##     LP_max_iter = 5000))
##
## tau = 0.25
##
## Fixed effects:
##                  Value Std. Error lower bound upper bound Pr(>|t|)
## (Intercept) -0.225647   0.068187   -0.359615     -0.0917 0.001003 **
## groupmutant -0.040933   0.099849   -0.237110      0.1552 0.682019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## tau = 0.5
##
## Fixed effects:
##                  Value Std. Error lower bound upper bound Pr(>|t|)
## (Intercept)  0.090754   0.057507   -0.022231      0.2037   0.1152
## groupmutant -0.026838   0.082629   -0.189182      0.1355   0.7455
##
## tau = 0.75
##
## Fixed effects:
##                  Value Std. Error lower bound upper bound  Pr(>|t|)
## (Intercept)  0.379806   0.044760    0.291864      0.4677 2.464e-16 ***
## groupmutant  0.022990   0.061072   -0.097000      0.1430    0.7067
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## AIC:
## [1] 3709 (df = 4) 3445 (df = 4) 3790 (df = 4)
```