Coarse woody debris: Pre- and post-thinning

Morgan Gray

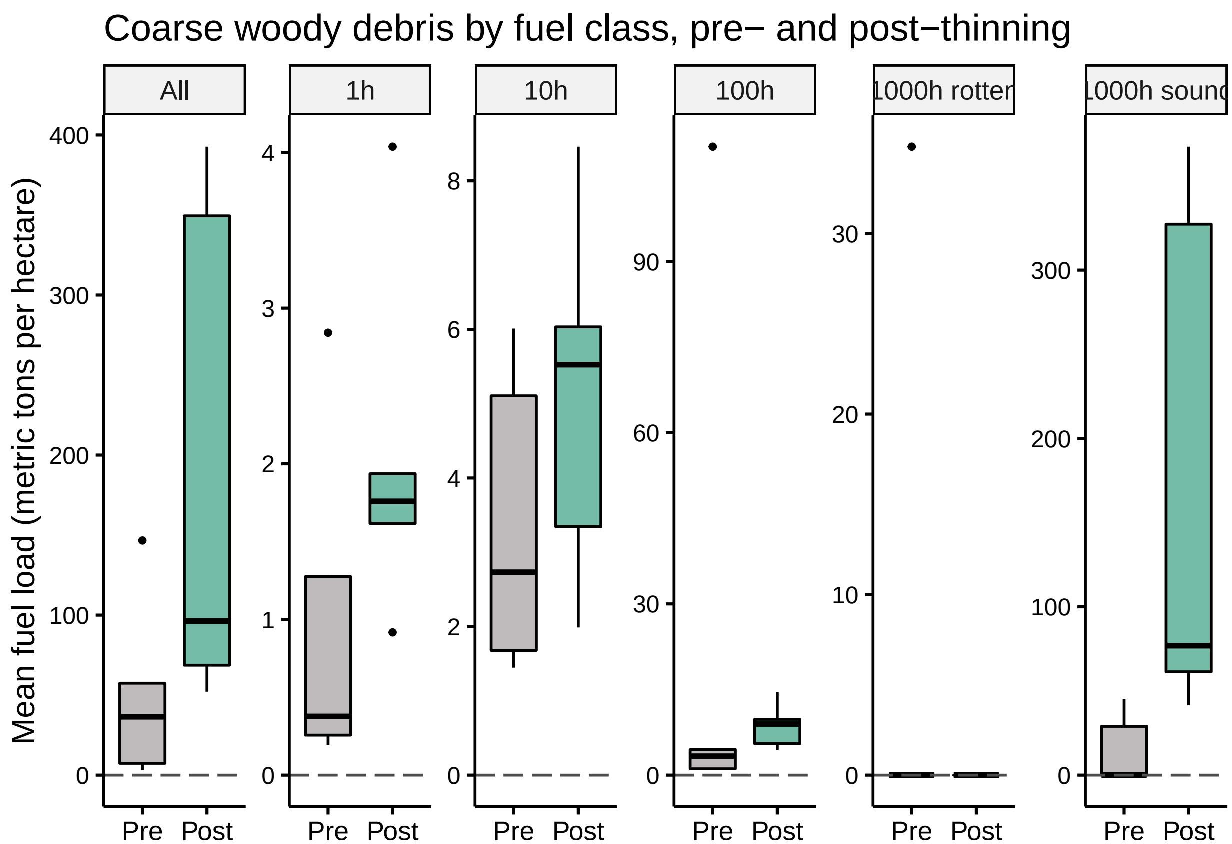
2022-11-01

Read and summarize coarse woody debris data (plot-level total, plot-level mean)

```
# Read data tables for derived total and mean, with metric units
input_wd <-
  read_csv(here(path_derived, "thin_total-by-plot-type-trmt.csv")) %>%
  bind_rows(read_csv(here(path_derived, "thin_mean-by-plot-type-class-trmt.csv"))) %>%
  arrange(survey, plot_id, data_type, fuel_class) %>%
  mutate_if(is.character, as_factor)  %>%
  mutate(value = fxn_digit(value_si),
         timing = ifelse(survey %in% "cont", "survey1", "survey2")) %>%
  select(-value_si,
         -units) %>%
  rename(units = units_si,
         metric = statistic) %>%
  relocate(c(metric, subset), .after = value)  %>%
  filter(data_type %in% "wd")

# # Create subset for mean by fuel class
# wd_class <-
#   input_wd %>%
#   filter(data_type %in% "wd",
#          metric %in% "mean")
#
# # Create subset for total (all fuel classes combined)
# wd_total <-
#   input_wd %>%
#   filter(data_type %in% "wd",
#          metric %in% "total") %>%
#   remove_empty("cols")
```

```
# Visualize coarse woody debris by fuel type and treatment
# Total fuel load (all CWD classes combined) increased after thinning
# Mean fuel load increased after thinning for all fuel classes except 1000-hr rotten
#   Data for 1000-hr rotten were mostly 0; difficult to interpret (see below)

input_wd %>%
  arrange(fuel_class) %>%
  mutate(lab_thin = str_remove_all(lab_thin, "-thinning"),
         lab_fuel = as_factor(str_replace_all(lab_fuel, "-hr", "h"))) %>%
  ggboxplot(x = "lab_thin",
```

```
            y = "value",
            fill = "lab_thin",
            outlier.size = 0.8,
            palette = colors_thin_faded) +
  geom_hline(yintercept = 0,
             linetype = "longdash",
             color = "gray30") +
  theme(legend.position = "none",
        axis.title.x = element_blank(),
        axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 9)) +
  labs(title = paste0(index_wd, " by fuel class, pre- and post-thinning"),
       y = index_units_lab_wd,
       fill = "Timing") +
  facet_wrap(~lab_fuel, nrow = 1, scales = "free")
```




```
# Pre- and post-thinning fuel load by fuel class
input_wd %>%
  group_by(lab_type, lab_fuel, lab_thin, metric, units) %>%
  get_summary_stats(value, type = "mean_sd") %>%
  select(lab_type,
         lab_fuel,
         lab_thin,
         metric,
         units,
```

```
        mean,
        sd,
        n) %>%
  fxn_kable()
```

lab_type	lab_fuel	lab_thin	metric	units	mean	sd	n
Coarse woody debris	All	Pre-thinning	total	tons_per_hectare	50.214	58.286	5
Coarse woody debris	All	Post-thinning	total	tons_per_hectare	191.854	165.076	5
Coarse woody debris	1-hr	Pre-thinning	mean	tons_per_hectare	0.989	1.125	5
Coarse woody debris	1-hr	Post-thinning	mean	tons_per_hectare	2.053	1.174	5
Coarse woody debris	10-hr	Pre-thinning	mean	tons_per_hectare	3.395	2.059	5
Coarse woody debris	10-hr	Post-thinning	mean	tons_per_hectare	5.071	2.505	5
Coarse woody debris	100-hr	Pre-thinning	mean	tons_per_hectare	24.011	48.154	5
Coarse woody debris	100-hr	Post-thinning	mean	tons_per_hectare	8.640	3.985	5
Coarse woody debris	1000-hr rotten	Pre-thinning	mean	tons_per_hectare	6.962	15.568	5
Coarse woody debris	1000-hr rotten	Post-thinning	mean	tons_per_hectare	0.000	0.000	5
Coarse woody debris	1000-hr sound	Pre-thinning	mean	tons_per_hectare	14.856	21.144	5
Coarse woody debris	1000-hr sound	Post-thinning	mean	tons_per_hectare	176.090	160.365	5

```
# 1000-hr rotten (hr1000r) is not normally distributed
#   29 of the 30 transect measurements were 0
#   9 of 10 plot-level means were 0 (pre-thin: 4/5, post-thin: 5/5)
input_wd %>%
  filter(value == 0) %>%
  group_by(lab_type, lab_fuel, lab_thin) %>%
  count() %>%
  fxn_kable()
```

lab_type	lab_fuel	lab_thin	n
Coarse woody debris	1000-hr rotten	Pre-thinning	4
Coarse woody debris	1000-hr rotten	Post-thinning	5
Coarse woody debris	1000-hr sound	Pre-thinning	3

Check assumptions

Subset data subset by fuel class and timing because these are the groupings that will be evaluated with statistical tests.

```
# Determine if there are extreme outliers
#
# No extreme outliers for total CWD
# Outlier for total pre-thinning in FOR08
#
# Three extreme outliers by fuel class :
#   Pre-thinning hr0100 and hr1000r in FOR08
#   Post-thinning hr0001 in FOR06
#
# Overall it looks like something was different about FOR08 during pre- and post-thinning surveys
#
input_wd %>%
  group_by(lab_type, lab_fuel, lab_thin, metric, units) %>%
  identify_outliers(value) %>%
```

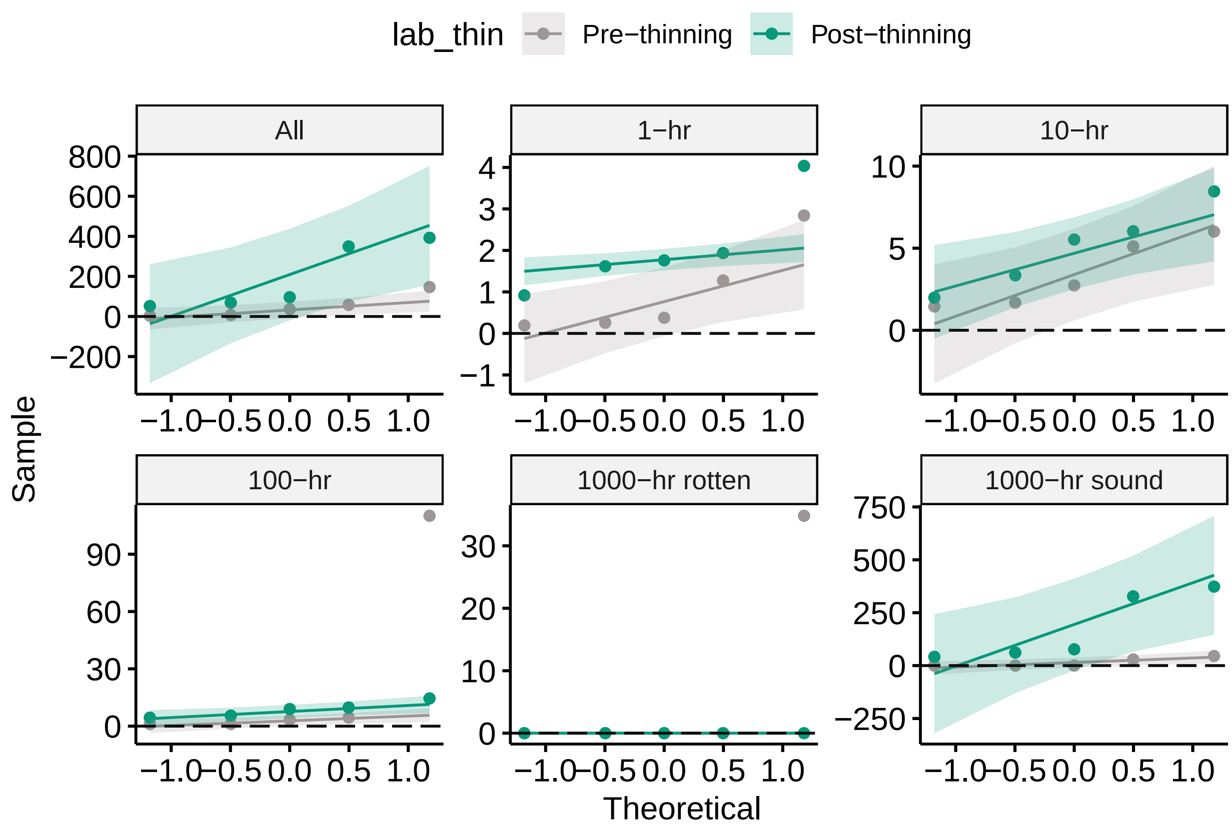
```r
  clean_names() %>%
  select(lab_type,
         lab_fuel,
         lab_thin,
         plot_id,
         value,
         metric,
         units,
         starts_with("is")) %>%
  arrange(desc(is_extreme), metric, lab_fuel) %>%
  # filter(is_extreme == TRUE) %>%
  fxn_kable()
```

lab_type	lab_fuel	lab_thin	plot_id	value	metric	units	is_outlier	is_
Coarse woody debris	1-hr	Post-thinning	FOR06	4.037	mean	tons_per_hectare	TRUE	T
Coarse woody debris	100-hr	Pre-thinning	FOR08	110.112	mean	tons_per_hectare	TRUE	T
Coarse woody debris	1000-hr rotten	Pre-thinning	FOR08	34.811	mean	tons_per_hectare	TRUE	T
Coarse woody debris	All	Pre-thinning	FOR08	146.628	total	tons_per_hectare	TRUE	FA
Coarse woody debris	1-hr	Pre-thinning	FOR07	2.842	mean	tons_per_hectare	TRUE	FA
Coarse woody debris	1-hr	Post-thinning	FOR08	0.917	mean	tons_per_hectare	TRUE	FA

```r
# Evaluate normality
#
# Shapiro test results indicated total CWD was normally distributed at each time point
#
# Shapiro test results by fuel class indicated 100-hr, 1000-hr sound (pre-thin) were not normally distr
#
input_wd %>%
  # Exclude hr1000r because we already know it's not normal
  filter(fuel_class %nin% "hr1000r") %>%
  # Rename columns for shapiro test
  select(id = plot_id,
         time = lab_thin,
         score = value,
         lab_type,
         lab_fuel,
         metric) %>%
  group_by(lab_type, lab_fuel, time, metric) %>%
  shapiro_test(score) %>%
  clean_names() %>%
  mutate(statistic = fxn_digit(statistic),
         # p = fxn_digit(p),
         is_normal = p>0.05) %>%
  select(lab_type,
         lab_fuel,
         lab_thin = time,
         metric,
         is_normal,
         p,
         statistic) %>%
  arrange(is_normal, metric, lab_fuel) %>%
  filter(is_normal == FALSE) %>%
  fxn_kable()
```

lab_type	lab_fuel	lab_thin	metric	is_normal	p	statistic
Coarse woody debris	100-hr	Pre-thinning	mean	FALSE	0.0003236	0.579
Coarse woody debris	1000-hr sound	Pre-thinning	mean	FALSE	0.0448490	0.770

```
# Create QQ plots for raw data subset by fuel_class and treatment
ggqqplot(input_wd,
         "value",
         palette = colors_thin_bright,
         color = "lab_thin") +
  facet_wrap(~lab_fuel,
             scales = "free") +
  theme(legend.position = "top") +
  geom_hline(yintercept = 0,
             linetype = "longdash",
             color = "gray5")
```



Identify appropriate transformation

Apply a series of transformations to each subset, then evaluate impact on outliers and normality.

```
# Use four normalization functions from the bestNormalize package to create transformed subsets:
#   arcsinh_x
#   log_x
#   orderNorm
#   sqrt_x
```

```r
#
# Most important for subsets that had extreme outliers or were not normally distributed:
# Outlier for total:
#   Pre-thin in FOR08
#
# Extreme outliers by fuel class:
#   Pre-thin hr0100 in FOR08
#   Pre-thin hr1000r in FOR08
#   Post-thin hr0001 in FOR06
#
# Not normally distributed:
#   Pre-thin hr0100
#   Pre-thin hr1000s
#
wd_transform_eval <-
  fxn_tranform_eval(index_data = input_wd,
                    index_list = list_classes_wd)  %>%
  gather(transform, value, value_raw:value_sqrt)
```

```r
# Check outliers for each transformation
check_transform_outlier <-
  wd_transform_eval %>%
  group_by(timing,
           fuel_class,
           transform) %>%
  identify_outliers(value) %>%
  clean_names() %>%
  select(transform,
         fuel_class,
         timing,
         plot_id,
         value,
         starts_with("is")) %>%
  arrange(desc(is_extreme), transform, fuel_class)

# Pre-thin total in FOR08: No extreme outliers for orderNorm, log_x, arcsinh_x, sqrt_x
check_transform_outlier %>%
  filter(fuel_class %in% "all" & timing %in% "survey1" & plot_id %in% "FOR08") %>%
  fxn_kable()
```

transform	fuel_class	timing	plot_id	value	is_outlier	is_extreme
value_raw	all	survey1	FOR08	146.6280000	TRUE	FALSE
value_std	all	survey1	FOR08	0.1847408	TRUE	FALSE

```r
# Pre-thin hr0100 in FOR08: No extreme outliers for orderNorm, log_x, arcsinh_x
check_transform_outlier %>%
  filter(fuel_class %in% "hr0100" & timing %in% "survey1" & plot_id %in% "FOR08") %>%
  fxn_kable()
```

transform	fuel_class	timing	plot_id	value	is_outlier	is_extreme
value_raw	hr0100	survey1	FOR08	110.112000	TRUE	TRUE
value_sqrt	hr0100	survey1	FOR08	2.697155	TRUE	TRUE
value_std	hr0100	survey1	FOR08	2.823576	TRUE	TRUE
value_arcsine	hr0100	survey1	FOR08	2.231336	TRUE	FALSE
value_log	hr0100	survey1	FOR08	2.180986	TRUE	FALSE
value_ordnorm	hr0100	survey1	FOR08	1.644854	TRUE	FALSE

```r
# Post-thin hr0001 in FOR06: No extreme outliers for orderNorm
check_transform_outlier %>%
  filter(fuel_class %in% "hr0001" & timing %in% "survey2" & plot_id %in% "FOR06") %>%
  fxn_kable()
```

transform	fuel_class	timing	plot_id	value	is_outlier	is_extreme
value_arcsine	hr0001	survey2	FOR06	1.617466	TRUE	TRUE
value_log	hr0001	survey2	FOR06	1.320270	TRUE	TRUE
value_raw	hr0001	survey2	FOR06	4.037000	TRUE	TRUE
value_sqrt	hr0001	survey2	FOR06	1.706900	TRUE	TRUE
value_std	hr0001	survey2	FOR06	2.061163	TRUE	TRUE
value_ordnorm	hr0001	survey2	FOR06	1.644854	TRUE	FALSE

```r
# Pre-thin hr1000r in FOR08: No improvement with any transformation
check_transform_outlier %>%
  filter(fuel_class %in% "hr1000r" & timing %in% "survey1" & plot_id %in% "FOR08") %>%
  fxn_kable()
```

transform	fuel_class	timing	plot_id	value	is_outlier	is_extreme
value_arcsine	hr1000r	survey1	FOR08	2.846050	TRUE	TRUE
value_log	hr1000r	survey1	FOR08	2.846050	TRUE	TRUE
value_ordnorm	hr1000r	survey1	FOR08	1.644854	TRUE	TRUE
value_raw	hr1000r	survey1	FOR08	34.811000	TRUE	TRUE
value_sqrt	hr1000r	survey1	FOR08	2.846050	TRUE	TRUE
value_std	hr1000r	survey1	FOR08	2.846050	TRUE	TRUE

```r
# Check normality for each transformation
check_transform_normal <-
  wd_transform_eval %>%
  # Exclude hr1000r because we already know it's not normal
  filter(fuel_class %nin% "hr1000r") %>%
  # Rename columns for shapiro test
  select(id = plot_id,
         time = timing,
         score = value,
         fuel_class,
         transform) %>%
  group_by(fuel_class,
           time,
           transform) %>%
  shapiro_test(score) %>%
  clean_names() %>%
  mutate(statistic = fxn_digit(statistic),
         is_normal = p>0.05) %>%
  select(fuel_class,
```

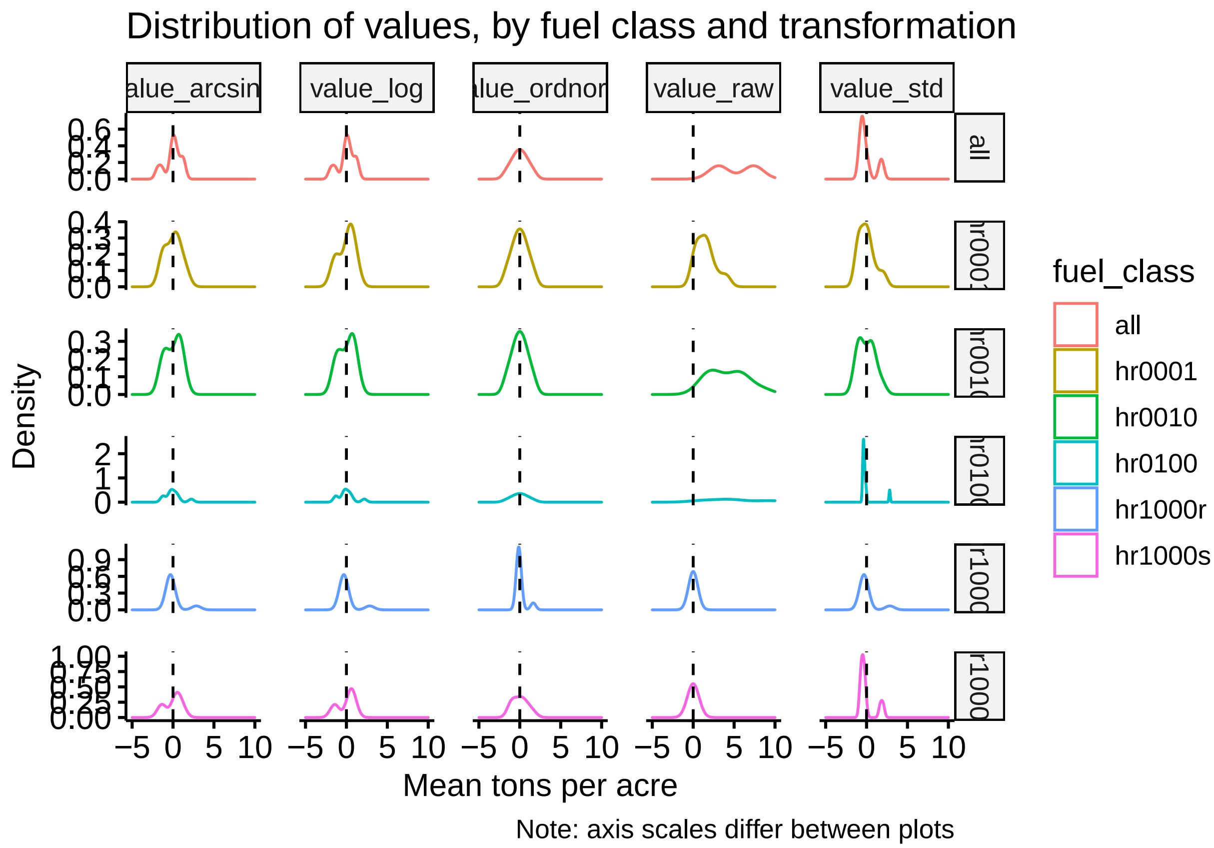
```
         timing = time,
         transform,
         is_normal,
         p,
         statistic)

# Pre-thin hr0100: Normal distribution with orderNorm, log_x, arcsinh_x
check_transform_normal %>%
  filter(fuel_class %in% "hr0100" & timing %in% "survey1",
         is_normal == TRUE) %>%
  fxn_kable()
```

fuel_class	timing	transform	is_normal	p	statistic
hr0100	survey1	value_arcsine	TRUE	0.0762655	0.797
hr0100	survey1	value_log	TRUE	0.1061219	0.815
hr0100	survey1	value_ordnorm	TRUE	0.5416692	0.922

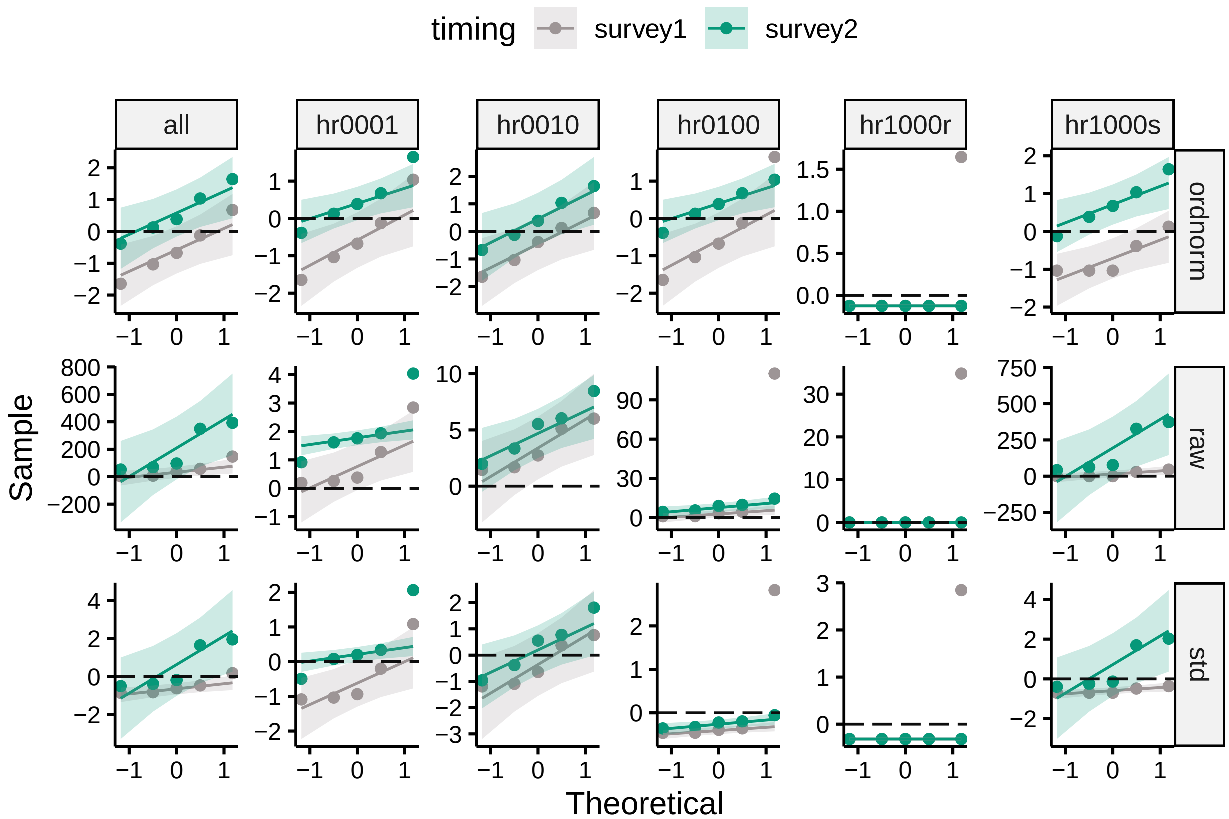
```
# Pre-thin hr1000s: No transformation normalized hr1000s
check_transform_normal %>%
  filter(fuel_class %in% "hr1000s" & timing %in% "survey1",
         is_normal == TRUE) %>%
  fxn_kable()
```

fuel_class	timing	transform	is_normal	p	statistic

```
# The orderNorm transformation looks pretty good for the total and mean subsets
#
wd_transform_eval %>%
  filter(transform %nin% "value_sqrt") %>%
  ggdensity(x = "value",
            color = "fuel_class") +
  geom_vline(xintercept = 0,
             linetype = "dashed") +
  facet_grid(fuel_class~transform,
             scales = "free") +
  theme(panel.spacing = unit(1, "lines"),
        legend.position = "right") +
  labs(title = "Distribution of values, by fuel class and transformation",
       caption = "Note: axis scales differ between plots",
       x = "Mean tons per acre",
       y = "Density") +
  xlim(-5, 10)
```




```
# The orderNorm transformed subsets look ok compared to the raw or standardized data
wd_transform_eval %>%
  filter(transform %in% c("value_raw", "value_std",  "value_ordnorm")) %>%
  mutate(transform = str_remove_all(transform, "value_")) %>%
  ggqqplot("value",
           palette = colors_thin_bright,
           color = "timing") +
  # To let scales on y-axis vary between faceted columns
  ggh4x::facet_grid2(transform ~ fuel_class,
                     scales="free",
                     independent = "all") +
  theme(legend.position = "top",
        axis.text = element_text(size = 9)) +
  geom_hline(yintercept = 0,
             linetype = "longdash",
             color = "gray5")  +
  scale_x_continuous(breaks = c(-1, 0, 1))
```



Create transformed data as input for statistical tests

```
wd_transform <-
  read_csv(here(path_derived, "thin_wd_transformed_metric-units.csv")) %>%
  arrange(timing, plot_id, data_type, fuel_class) %>%
  # Non-numeric variables must be factors for rstatix
  mutate_if(is.character, as_factor)

wd_transform_total <-
  wd_transform %>%
  filter(data_type %in% "wd",
         metric %in% "total")

wd_transform_class <-
  wd_transform %>%
  filter(data_type %in% "wd",
         metric %in% "mean")
```

Conduct statistical tests

```
# Test for an effect of treatment on total CWD load (all CWD fuel classes combined)
# We found no significant difference in total fuel load between measurements collected pre- and post-th
```

```r
wd_transform_total %>%
  # Use transformed values as input for statistical tests
  fxn_aov_me(index_value = "value_tran",
             index_id = "plot_id",
             index_time = "lab_thin")  %>%
  fxn_signif_adj() %>%
  mutate(data_type = index_wd,
         fuel_class = "All") %>%
  select(data_type,
         fuel_class,
         starts_with("p_adj"),
         statistic,
         starts_with("d_"),
         ges) %>%
  fxn_kable()
```

Did thinning treatment have a significant effect on total coarse woody debris fuel load?

data_type	fuel_class	p_adj	p_adj_sig	statistic	d_fn	d_fd	ges
Coarse woody debris	All	0.079	n.s.	5.474	1	4	0.358

```r
# Two-way interaction between thinning treatment and fuel class
# We wanted to know if the thinning treatment induced a significant change in fuel load among the five f
#   In other words, was there a significant interaction between thinning and fuel class on fuel load fo
# We conducted a two-way repeated measures ANOVA to evaluate the effect of thinning over different fuel
#
# There was no statistically significant interaction between thinning treatment and fuel class on CWD f
wd_transform_class  %>%
  fxn_aov2_me(index_value = "value_tran",
              index_id = "plot_id",
              index_time = "timing",
              index_variable = "fuel_class") %>%
  fxn_signif_adj() %>%
  mutate(data_type = index_wd) %>%
  select(data_type,
           effect,
             starts_with("p_adj"),
             statistic,
             starts_with("d_"),
             ges) %>%
  fxn_kable()
```

Did thinning treatment have a significant effect on the individual fuel classes?

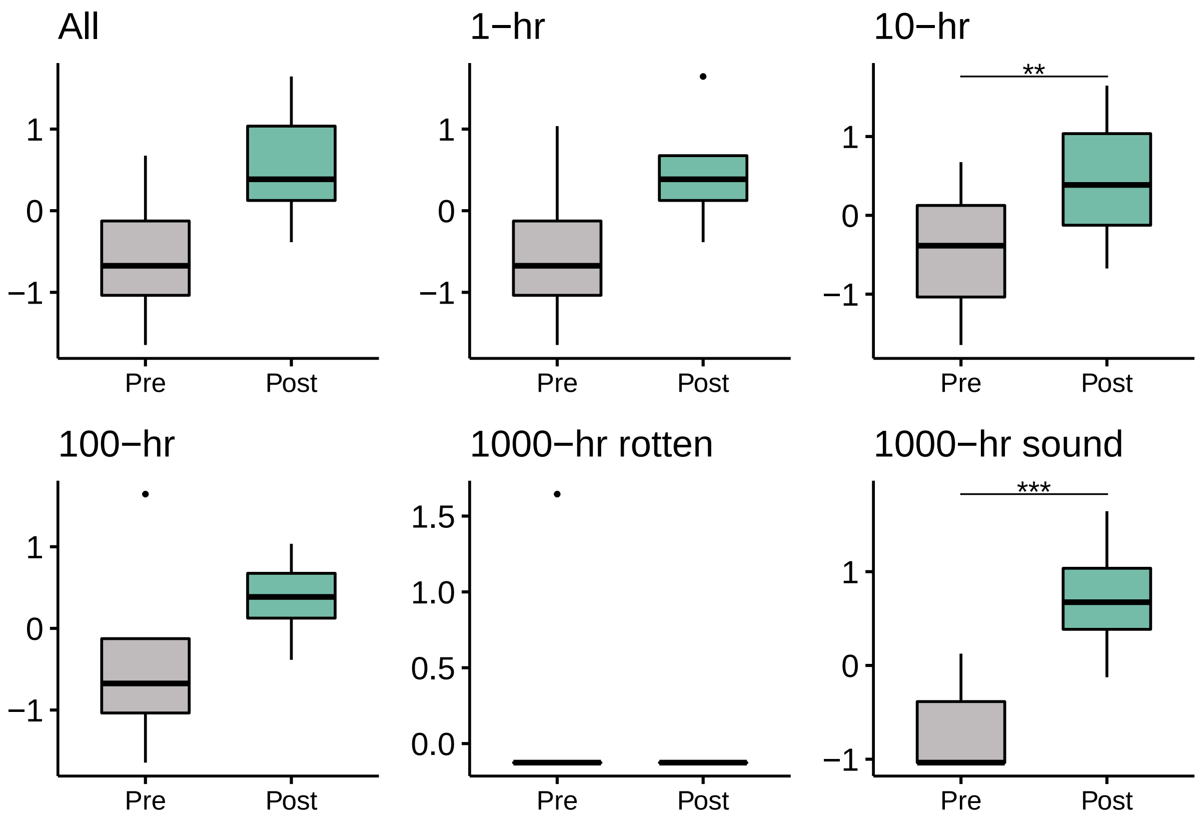
data_type	effect
Coarse woody debris	time
Coarse woody debris	time:varia
Coarse woody debris	variable

```r
# time:variable 0.249   n.s.    2.512   4.00    16.00   0.141000
```

```r
# Pairwise comparison between fuel load by thinning treatment, for each fuel class
# We analyzed the effect of thinning treatment for each fuel class. The Bonferroni adjustment was appli
# A post hoc pairwise comparison showed mean fuel load was significantly different between control and

wd_transform_class  %>%
  group_by(fuel_class) %>%
    fxn_pwc(index_value = "value_tran",
            index_id = "plot_id",
            index_time = "timing") %>%
  # fxn_aov_me(index_value = "value_tran",
  #            index_id = "plot_id",
  #            index_time = "timing") %>%
  fxn_signif_adj() %>%
  mutate(data_type = index_wd) %>%
  select(data_type,
         fuel_class,
         starts_with("p_adj"),
         statistic,
         starts_with("d")) %>%
  arrange(p_adj) %>%
  fxn_kable()
```

data_type	fuel_class	p_adj	p_adj_sig	statistic	df
Coarse woody debris	hr1000s	0.000458	***	-10.543	4
Coarse woody debris	hr0010	0.008000	**	-4.874	4
Coarse woody debris	hr0001	0.131000	n.s.	-1.893	4
Coarse woody debris	hr1000r	0.374000	n.s.	1.000	4
Coarse woody debris	hr0100	0.402000	n.s.	-0.937	4

```r
# Create individual plots
all <- fxn_boxplot_class("all")
hr0001 <- fxn_boxplot_class("hr0001")
hr0010 <- fxn_boxplot_class("hr0010")
hr0100 <- fxn_boxplot_class("hr0100")
hr1000r <- fxn_boxplot_class("hr1000r")
hr1000s <- fxn_boxplot_class("hr1000s")

# Join plots using patchwork
all + hr0001 + hr0010 + hr0100 + hr1000r + hr1000s
```


All
1
0
−1
Pre
Post
1−hr
1
0
−1
Pre
Post
10−hr
**
1
0
−1
Pre
Post
100−hr
1
0
−1
Pre
Post
1000−hr rotten
1.5
1.0
0.5
0.0
Pre
Post
1000−hr sound
***
1
0
−1
Pre
Post