# EPVS_EDA and bootstraping

Chunlin Liu

2025-11-03

File caa_all_radii_40um_donut_13Oct2025

```
library(knitr)
library(kableExtra)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(stringr)
library(ggplot2)
library(rstanarm)
```

```
## Loading required package: Rcpp
## This is rstanarm version 2.32.1
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
## - For execution on a local, multicore CPU with excess RAM we recommend calling
##   options(mc.cores = parallel::detectCores())
```

```
library(tidyr)
library(readr)
library(purrr)
library(dplyr)
library(foreign)
library(readxl)


scattering <- read_excel("/Users/liuchunlin/Desktop/BU/MA 675 Client EPVS/caa_all_radii_40um_donut_13Oc
retardance <- read_excel("/Users/liuchunlin/Desktop/BU/MA 675 Client EPVS/caa_all_radii_40um_donut_13Oc
orientation <- read_excel("/Users/liuchunlin/Desktop/BU/MA 675 Client EPVS/caa_all_radii_40um_donut_13O
```

```r
scattering$property <- "scattering"
retardance$property <- "retardance"
orientation$property <- "orientation"

summary(scattering)
```

```
##     Groups             Region              subID          distance
##  Length:330787      Length:330787      Min.   :1.000   Min.   : 40.0
##  Class :character   Class :character   1st Qu.:3.000   1st Qu.: 40.0
##  Mode  :character   Mode  :character   Median :4.000   Median :120.0
##                                        Mean   :4.671   Mean   :144.4
##                                        3rd Qu.:7.000   3rd Qu.:200.0
##                                        Max.   :9.000   Max.   :480.0
##  OpticalProperty     property
##  Min.   : 0.5275   Length:330787
##  1st Qu.:10.8702   Class :character
##  Median :12.1511   Mode  :character
##  Mean   :11.8954
##  3rd Qu.:13.0886
##  Max.   :82.9646
```

```r
summary(retardance)
```

```
##     Groups             Region              subID          distance
##  Length:330787      Length:330787      Min.   :1.000   Min.   : 40.0
##  Class :character   Class :character   1st Qu.:3.000   1st Qu.: 40.0
##  Mode  :character   Mode  :character   Median :4.000   Median :120.0
##                                        Mean   :4.671   Mean   :144.4
##                                        3rd Qu.:7.000   3rd Qu.:200.0
##                                        Max.   :9.000   Max.   :480.0
##  OpticalProperty   property
##  Min.   : 0.00   Length:330787
##  1st Qu.:24.36   Class :character
##  Median :28.35   Mode  :character
##  Mean   :28.11
##  3rd Qu.:32.07
##  Max.   :49.30
```

```r
summary(orientation)
```

```
##     Groups             Region              subID          distance
##  Length:330787      Length:330787      Min.   :1.000   Min.   : 40.0
##  Class :character   Class :character   1st Qu.:3.000   1st Qu.: 40.0
##  Mode  :character   Mode  :character   Median :4.000   Median :120.0
##                                        Mean   :4.671   Mean   :144.4
##                                        3rd Qu.:7.000   3rd Qu.:200.0
##                                        Max.   :9.000   Max.   :480.0
##  OpticalProperty    property
##  Min.   :0.0000   Length:330787
##  1st Qu.:0.1968   Class :character
##  Median :0.4423   Mode  :character
```

```
##  Mean   :0.5070
##  3rd Qu.:0.7381
##  Max.   :2.3976
```

```
#check for NAs
colSums(is.na(scattering))
```

```
##          Groups          Region           subID        distance OpticalProperty
##               0               0               0               0               0
##        property
##               0
```

```
colSums(is.na(retardance))
```

```
##          Groups          Region           subID        distance OpticalProperty
##               0               0               0               0               0
##        property
##               0
```

```
colSums(is.na(orientation))
```

```
##          Groups          Region           subID        distance OpticalProperty
##               0               0               0               0               0
##        property
##               0
```

```
#merge data
merged_data <- bind_rows(scattering, retardance, orientation)

merged_data$Groups <- factor(merged_data$Groups, levels = c("control", "experimental"))
merged_data$Region <- factor(merged_data$Region, levels = c("front", "occip"))

#check all data
str(merged_data)
```

```
## tibble [992,361 x 6] (S3: tbl_df/tbl/data.frame)
##  $ Groups         : Factor w/ 2 levels "control","experimental": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Region         : Factor w/ 2 levels "front","occip": 1 1 1 1 1 1 1 1 1 1 ...
##  $ subID          : num [1:992361] 1 1 1 1 1 1 1 1 1 1 ...
##  $ distance       : num [1:992361] 40 40 40 40 40 40 40 40 40 40 ...
##  $ OpticalProperty: num [1:992361] 11.4 12.9 13.1 12.9 12.6 ...
##  $ property       : chr [1:992361] "scattering" "scattering" "scattering" "scattering" ...
```

```
summary(merged_data)
```

```
##           Groups         Region           subID          distance
##  control     :962130   front:239985   Min.   :1.000   Min.   : 40.0
##  experimental: 30231   occip:752376   1st Qu.:3.000   1st Qu.: 40.0
##                                       Median :4.000   Median :120.0
##                                       Mean   :4.671   Mean   :144.4
```

```
##                                     3rd Qu.:7.000   3rd Qu.:200.0
##                                     Max.   :9.000   Max.   :480.0
##  OpticalProperty     property
##  Min.   : 0.0000   Length:992361
##  1st Qu.: 0.7371   Class :character
##  Median :12.1472   Mode  :character
##  Mean   :13.5044
##  3rd Qu.:24.4075
##  Max.   :82.9646
```

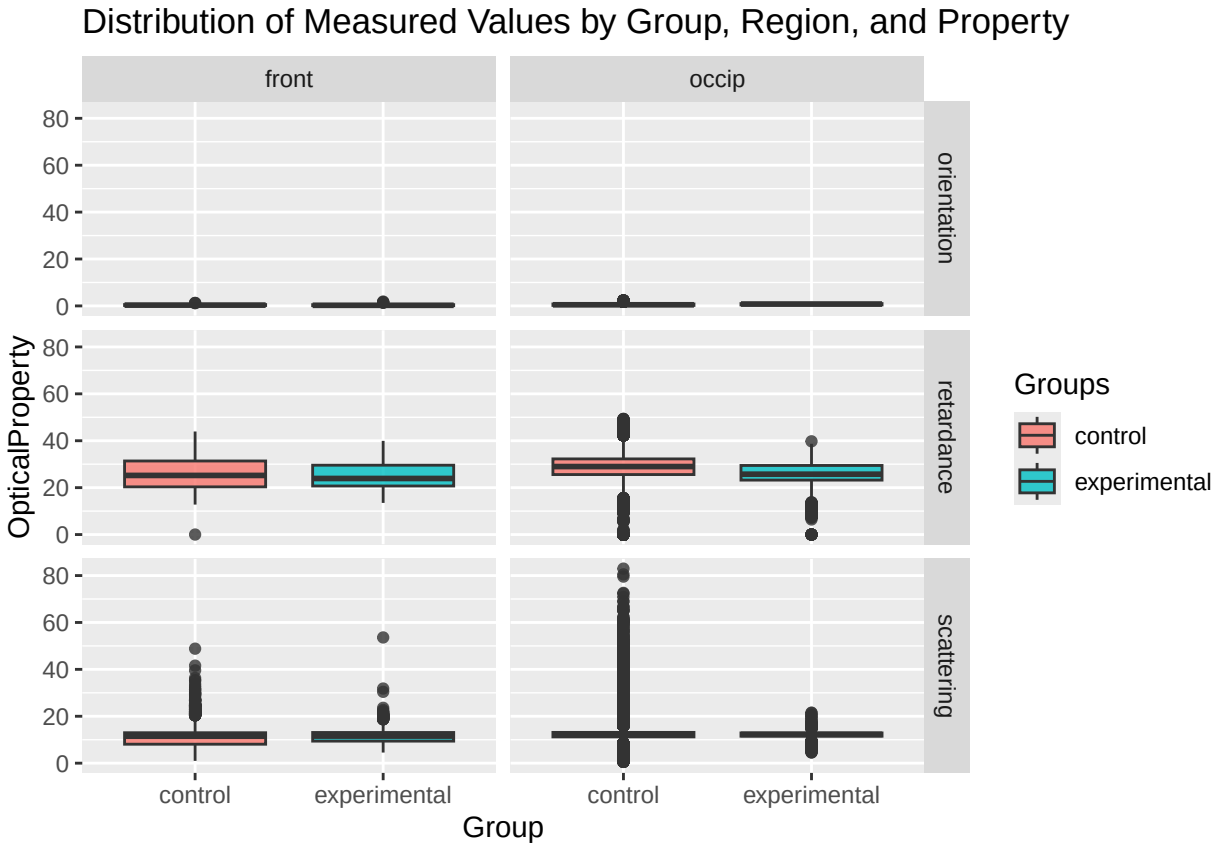
```
group_summary <- merged_data %>%
  group_by(property, Groups, Region) %>%
  summarise(across(where(is.numeric), list(mean = mean, sd = sd, min = min, max = max), na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'property', 'Groups'. You can override
## using the `.groups` argument.
```

```
print(group_summary)
```

```
## # A tibble: 12 x 15
## # Groups:   property, Groups [6]
##    property  Groups Region subID_mean subID_sd subID_min subID_max distance_mean
##    <chr>     <fct>  <fct>       <dbl>    <dbl>     <dbl>     <dbl>         <dbl>
##  1 orientat~ contr~ front        4.70     2.55         1         8          143.
##  2 orientat~ contr~ occip        4.65     2.49         2         9          140.
##  3 orientat~ exper~ front        5.26     1.72         1         8          270.
##  4 orientat~ exper~ occip        4.67     1.82         2         9          230.
##  5 retardan~ contr~ front        4.70     2.55         1         8          143.
##  6 retardan~ contr~ occip        4.65     2.49         2         9          140.
##  7 retardan~ exper~ front        5.26     1.72         1         8          270.
##  8 retardan~ exper~ occip        4.67     1.82         2         9          230.
##  9 scatteri~ contr~ front        4.70     2.55         1         8          143.
## 10 scatteri~ contr~ occip        4.65     2.49         2         9          140.
## 11 scatteri~ exper~ front        5.26     1.72         1         8          270.
## 12 scatteri~ exper~ occip        4.67     1.82         2         9          230.
## # i 7 more variables: distance_sd <dbl>, distance_min <dbl>,
## #   distance_max <dbl>, OpticalProperty_mean <dbl>, OpticalProperty_sd <dbl>,
## #   OpticalProperty_min <dbl>, OpticalProperty_max <dbl>
```

```
#check dataset as a whole
ggplot(merged_data, aes(x = Groups, y = OpticalProperty, fill = Groups)) +
  geom_boxplot(alpha = 0.8) +
  facet_grid(property ~ Region) +
  labs(
    title = "Distribution of Measured Values by Group, Region, and Property",
    x = "Group",
    y = "OpticalProperty"
  )
```




```
# check for outlier
outlier_data <- merged_data %>%
  group_by(Groups, property, Region) %>%
  mutate(
    Q1 = quantile(OpticalProperty, 0.25, na.rm = TRUE),
    Q3 = quantile(OpticalProperty, 0.75, na.rm = TRUE),
    IQR = Q3 - Q1,
    lower = Q1 - 1.5 * IQR,
    upper = Q3 + 1.5 * IQR,
    is_outlier = OpticalProperty < lower | OpticalProperty > upper
  ) %>%
  ungroup()

table(outlier_data$is_outlier)
```

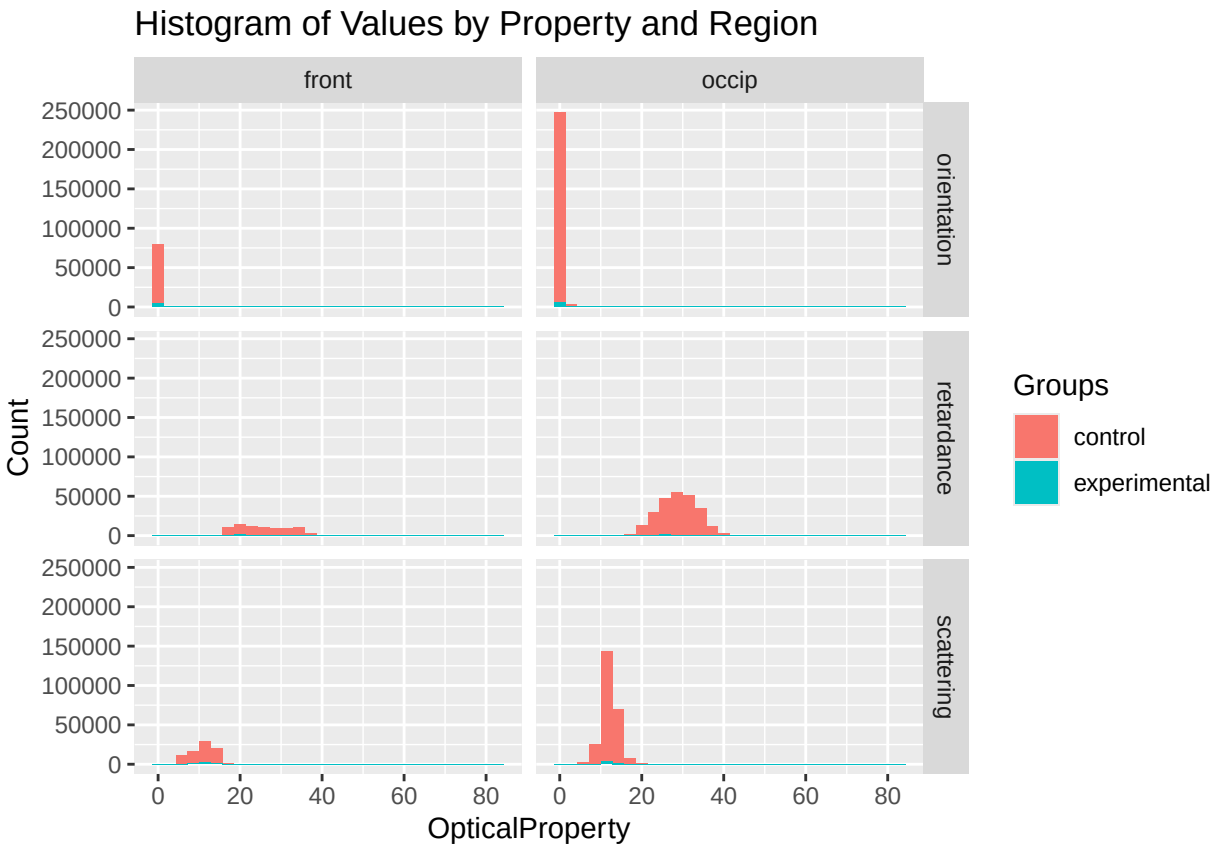
```
##
##  FALSE   TRUE
## 974138  18223
```

```
# checking numbers of control and experinemntal group

summary_table <- merged_data %>%
  group_by(property, Groups) %>%
  summarise(count = n(), .groups = "drop") %>%
  tidyr::pivot_wider(names_from = Groups, values_from = count, values_fill = 0)
```

```r
print(summary_table)
```

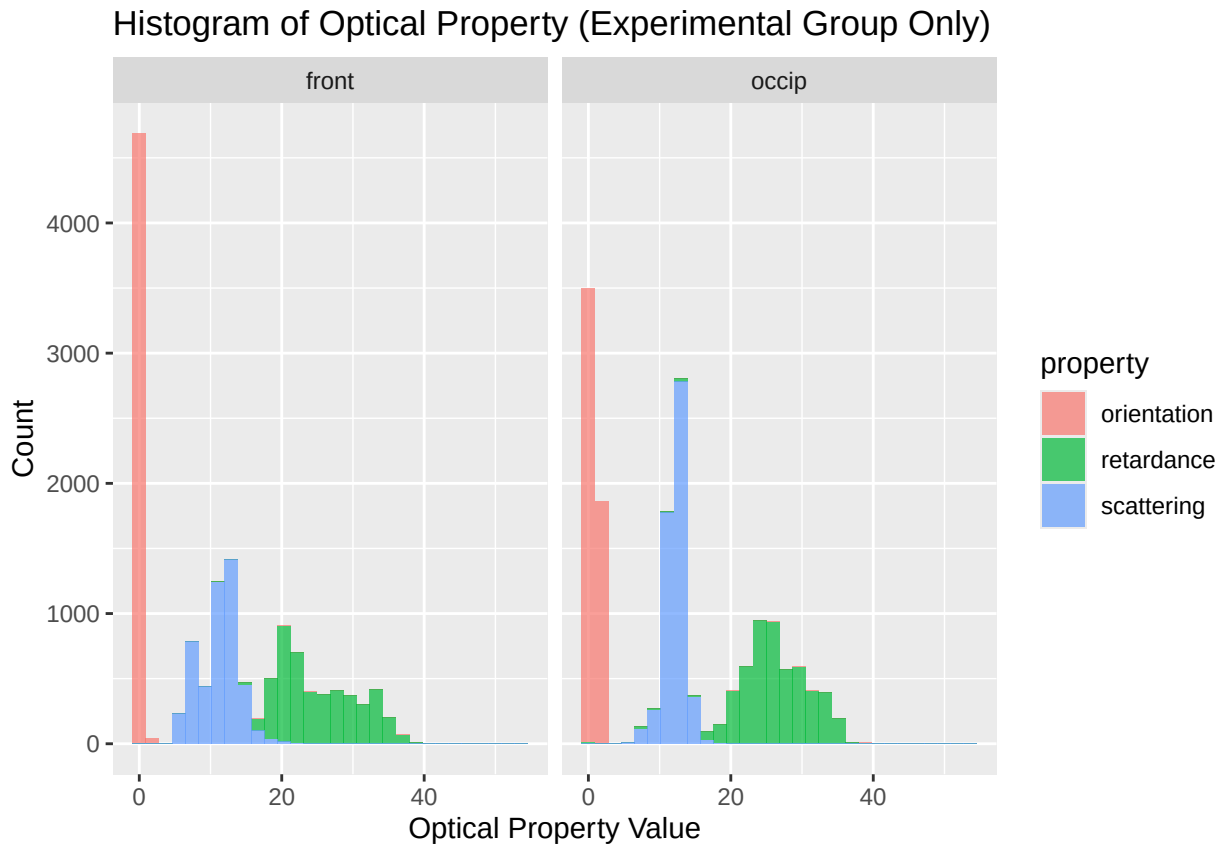
```
## # A tibble: 3 x 3
##   property    control experimental
##   <chr>         <int>        <int>
## 1 orientation  320710        10077
## 2 retardance   320710        10077
## 3 scattering   320710        10077
```

```r
#check the OpticalProperty of different groups by region
ggplot(merged_data, aes(x = OpticalProperty, fill = Groups)) +
  geom_histogram(bins = 30, alpha = 1) +
  facet_grid(property ~ Region) +
  labs(
    title = "Histogram of Values by Property and Region",
    x = "OpticalProperty",
    y = "Count"
  )
```

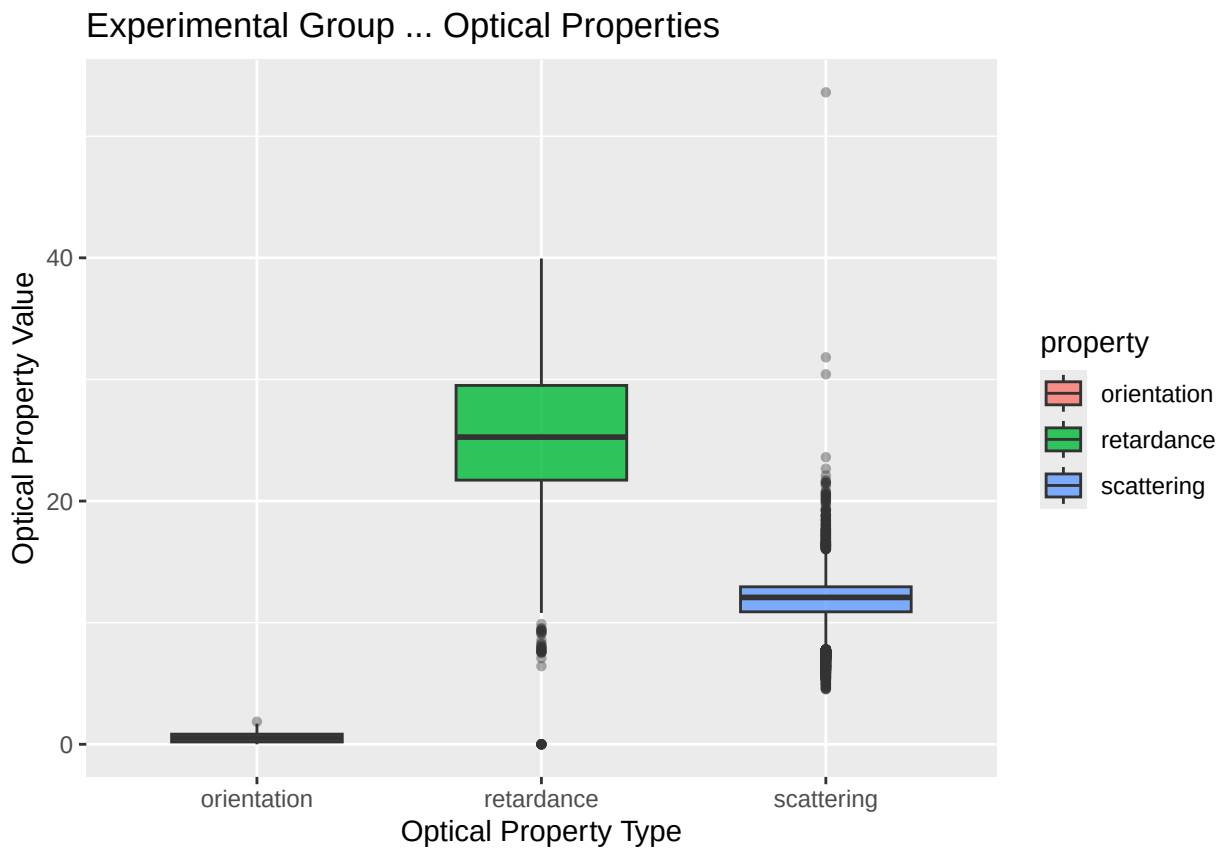



```r
#check the OpticalProperty in 3 groups (experimental only)
ggplot(
  data = merged_data %>% filter(Groups == "experimental"),
  aes(x = OpticalProperty, fill = property)
) +
```

```
  geom_histogram(bins = 30, alpha = 0.7) +
  facet_wrap(~Region) +
  labs(
    title = "Histogram of Optical Property (Experimental Group Only)",
    x = "Optical Property Value",
    y = "Count"
  )
```




```
# check for boxplot
ggplot(
  data = merged_data %>% filter(Groups == "experimental"),
  aes(x = property, y = OpticalProperty, fill = property)
) +
  geom_boxplot(alpha = 0.8, width = 0.6, outlier.shape = 16, outlier.alpha = 0.4) +
  labs(
    title = "Experimental Group - Optical Properties",
    x = "Optical Property Type",
    y = "Optical Property Value"
  )
```




```
#check for outlier: scattering
scattering_data <- merged_data %>%
  filter(Groups == "experimental", property == "scattering")

Q1_s <- quantile(scattering_data$OpticalProperty, 0.25, na.rm = TRUE)
Q3_s <- quantile(scattering_data$OpticalProperty, 0.75, na.rm = TRUE)
IQR_s <- Q3_s - Q1_s

lower_s <- Q1_s - 1.5 * IQR_s
upper_s <- Q3_s + 1.5 * IQR_s

scattering_outlier <- scattering_data %>%
  mutate(is_outlier = OpticalProperty < lower_s | OpticalProperty > upper_s)

scattering_outlier %>%
  filter(is_outlier) %>%
  select(Region, OpticalProperty)
```

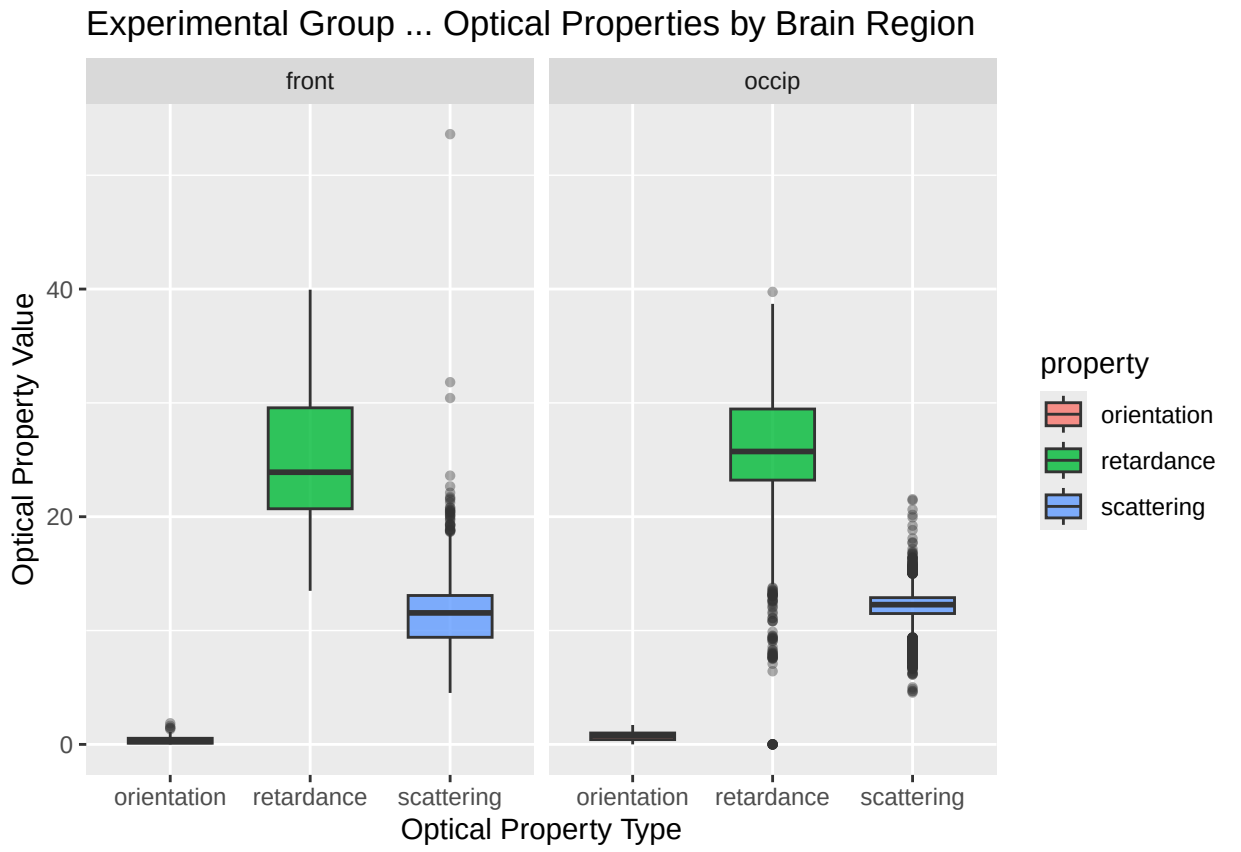
```
## # A tibble: 1,158 x 2
##    Region OpticalProperty
##    <fct>            <dbl>
##  1 occip             7.18
##  2 occip            17.8
##  3 occip            16.4
##  4 occip            17.7
##  5 occip            16.1
```

```
##  6 occip            18.1
##  7 occip            19.2
##  8 occip            21.4
##  9 occip            16.7
## 10 occip            16.9
## # i 1,148 more rows
```

```r
#check for outlier: orientation
orientation_data <- merged_data %>%
  filter(Groups == "experimental", property == "orientation")

Q1_o <- quantile(orientation_data$OpticalProperty, 0.25, na.rm = TRUE)
Q3_o <- quantile(orientation_data$OpticalProperty, 0.75, na.rm = TRUE)
IQR_o <- Q3_o - Q1_o

lower_o <- Q1_o - 1.5 * IQR_o
upper_o <- Q3_o + 1.5 * IQR_o

orientation_outlier <- orientation_data %>%
  mutate(is_outlier = OpticalProperty < lower_o | OpticalProperty > upper_o)

orientation_outlier %>%
  filter(is_outlier) %>%
  select(Region, OpticalProperty)
```

```
## # A tibble: 1 x 2
##   Region OpticalProperty
##   <fct>            <dbl>
## 1 front             1.86
```

```r
#check boxplot for OpticalProperty different by properties and regions
ggplot(
  data = merged_data %>% filter(Groups == "experimental"),
  aes(x = property, y = OpticalProperty, fill = property)
) +
  geom_boxplot(alpha = 0.8, width = 0.6, outlier.shape = 16, outlier.alpha = 0.4) +
  facet_wrap(~Region) +
  labs(
    title = "Experimental Group - Optical Properties by Brain Region",
    x = "Optical Property Type",
    y = "Optical Property Value"
  )
```




File about ring measurement

```
library(readxl)
library(dplyr)
library(ggplot2)
library(tidyr)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
ring_data <- read_excel("/Users/liuchunlin/Desktop/BU/MA 675 Client EPVS/histology_ring_measurements_15
mean_ring_data <- read_excel("/Users/liuchunlin/Desktop/BU/MA 675 Client EPVS/histology_mean_ring_measu


summary(ring_data)
```

```
##    baseName          radiusSize        radiusValue        measurementType
##  Length:6989        Length:6989        Length:6989        Length:6989
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   measurement
##  Min.   :-3.62590
```

```
##  1st Qu.:-0.14338
##  Median : 0.02371
##  Mean   : 0.01613
##  3rd Qu.: 0.17735
##  Max.   : 2.92249
```

```
summary(mean_ring_data)
```

```
##    baseName          radiusSize        radiusValue        measurementType
##  Length:306         Length:306         Length:306         Length:306
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   measurement
##  Min.   :-0.244640
##  1st Qu.:-0.046680
##  Median : 0.002240
##  Mean   : 0.006119
##  3rd Qu.: 0.051341
##  Max.   : 0.365451
```

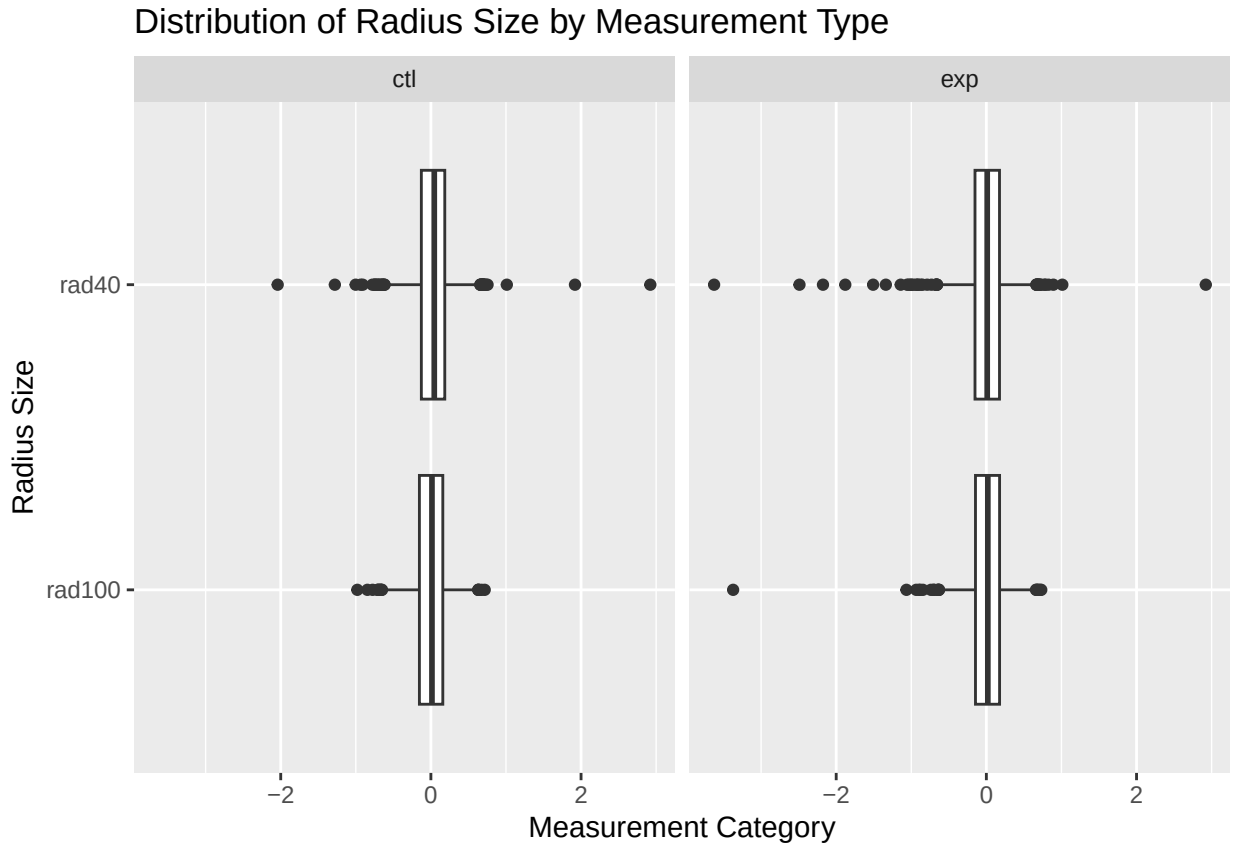
```
#check for NAs
colSums(is.na(ring_data))
```

```
##        baseName      radiusSize     radiusValue measurementType     measurement
##               0               0               0               0               0
```

```
colSums(is.na(mean_ring_data))
```

```
##        baseName      radiusSize     radiusValue measurementType     measurement
##               0               0               0               0               0
```

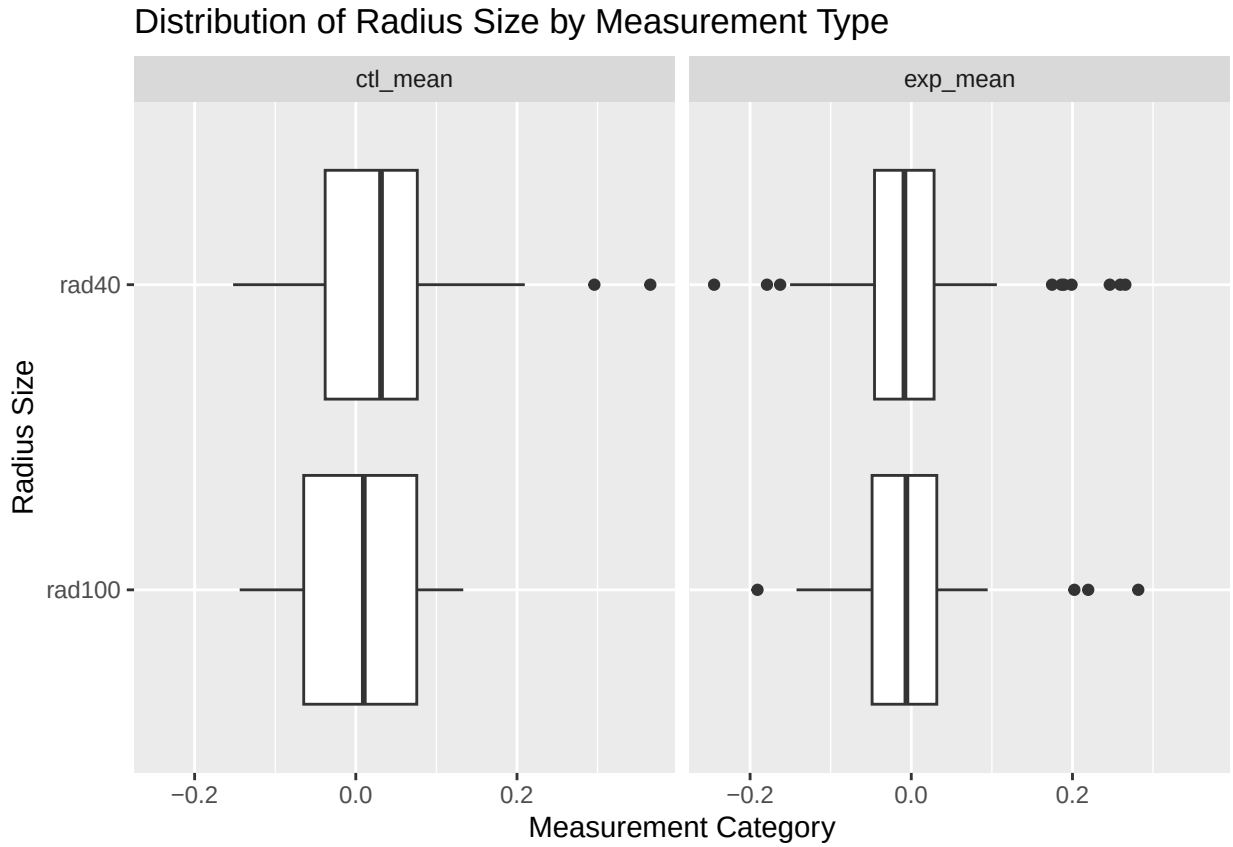
```
#check boxplot for ring data
ggplot(ring_data, aes(x = measurement, y = radiusSize, fill = measurement)) +
  geom_boxplot() +
  facet_wrap(~measurementType) +
  labs(
  title = "Distribution of Radius Size by Measurement Type",
  x = "Measurement Category",
  y = "Radius Size"
)
```




```
#check for outlier
Q1 <- quantile(ring_data$measurement, 0.25, na.rm = TRUE)
Q3 <- quantile(ring_data$measurement, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1

lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

ring_data <- ring_data %>%
  mutate(is_outlier = measurement < lower_bound | measurement > upper_bound)

table(ring_data$is_outlier)
```

```
##
## FALSE  TRUE
##  6880   109
```

```
#check boxplot for ring data
ggplot(mean_ring_data, aes(x = measurement, y = radiusSize, fill = measurement)) +
  geom_boxplot() +
  facet_wrap(~measurementType) +
  labs(
  title = "Distribution of Radius Size by Measurement Type",
  x = "Measurement Category",
  y = "Radius Size"
)
```




```
#check for outlier
Q1_m <- quantile(mean_ring_data$measurement, 0.25, na.rm = TRUE)
Q3_m <- quantile(mean_ring_data$measurement, 0.75, na.rm = TRUE)
IQR_m <- Q3_m - Q1_m

lower_bound_m <- Q1_m - 1.5 * IQR_m
upper_bound_m <- Q3_m + 1.5 * IQR_m

mean_ring_data <- mean_ring_data %>%
  mutate(is_outlier = measurement < lower_bound_m | measurement > upper_bound_m)

table(mean_ring_data$is_outlier)
```

```
##
## FALSE  TRUE
##   295    11
```

bootstrapping for control and experinemtal groups

```
# Undersampling - randomly select from control group to make the sample number as big as experimental g


set.seed(123)

balanced_data_under <- merged_data %>%
```

```
  group_by(Groups) %>%
  reframe(sample_n(cur_data(), size = min(table(merged_data$Groups)), replace = FALSE))
```

```
## Warning: There was 1 warning in `reframe()`.
## i In argument: `sample_n(cur_data(), size = min(table(merged_data$Groups)),
##   replace = FALSE)`.
## i In group 1: `Groups = control`.
## Caused by warning:
## ! `cur_data()` was deprecated in dplyr 1.1.0.
## i Please use `pick()` instead.
```

```
table(balanced_data_under$Groups)
```

```
##
##      control experimental
##        30231        30231
```

```
# Oversampling - randomly select from experimental group to make the sample number as big as control gr

set.seed(123)

balanced_data_over <- merged_data %>%
  group_by(Groups) %>%
  reframe(sample_n(cur_data(), size = max(table(merged_data$Groups)), replace = TRUE))


table(balanced_data_over$Groups)
```

```
##
##      control experimental
##       962130       962130
```