Analysis of B-endorphin/ChAT IHC Count Data

C-T Berezin

11/21/2021

```
library(knitr)
library(tidyverse)
library(dplyr)
library(WebPower)
library(ggpubr)
library(car)
library(ggthemes)
library(writexl)
library(DescTools)
```

Reading in and Tidying the Data

```
counts <- read_csv("../data/B-endo-totals.csv")
```

```
## Rows: 12 Columns: 11
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Light, Time
## dbl (9): Mouse, ChAT, Bendo, Percentage, GCLchat, GCLbendo, INLchat, INLbend...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
counts <- counts %>%
  dplyr::select(-ChAT, -Bendo, -Percentage) %>%
  #make columns lowercase
  rename(mouse = Mouse,
        light = Light,
        time = Time,
        GCL_chat = GCLchat,
        GCL_bendo = GCLbendo,
        INL_chat = INLchat,
        INL_bendo = INLbendo) %>%
  #update the classes of some columns
  mutate(mouse = as.factor(mouse),
         light = factor(as.factor(light), levels=c("light", "dark")),
         time = as.factor(time),
  #reintroduce total counts
         total_chat = GCL_chat+INL_chat,
         total_bendo = GCL_bendo+INL_bendo) %>%
  #calculate percentages
```

```
  mutate(GCL_perc=GCL_bendo/GCL_chat*100,
         INL_perc=INL_bendo/INL_chat*100,
         total_perc=total_bendo/total_chat*100)
counts <- counts %>% mutate(
  GCL_chat_mm = GCL_chat/Area,
  GCL_bendo_mm = GCL_bendo/Area,
  INL_chat_mm = INL_chat/Area,
  INL_bendo_mm = INL_bendo/Area
)
```

```
#manova for ChAT cells per mm2
counts %>% group_by(light, time) %>%  summarise(
  min_INL_chat_mm = min(INL_chat_mm),
  mean_INL_chat_mm = mean(INL_chat_mm),
  max_INL_chat_mm = max(INL_chat_mm),
  min_GCL_chat_mm = min(GCL_chat_mm),
  mean_GCL_chat_mm = mean(GCL_chat_mm),
  max_GCL_chat_mm = max(GCL_chat_mm)
)
```

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

```
## # A tibble: 4 x 8
## # Groups:   light [2]
##   light time  min_INL_chat_mm mean_INL_chat_mm max_INL_chat_mm min_GCL_chat_mm
##   <fct> <fct>           <dbl>            <dbl>           <dbl>           <dbl>
## 1 light day              825              954.           1071.            675.
## 2 light night            596.            1029.           1472.            509.
## 3 dark  day              945.            1008.           1068.            778.
## 4 dark  night            812.             929.           1044.            692.
## # ... with 2 more variables: mean_GCL_chat_mm <dbl>, max_GCL_chat_mm <dbl>
```

```
counts %>% summarise(
  mean_INL_chat_mm = mean(INL_chat_mm),
  sd_INL_chat_mm = sd(INL_chat_mm),
  mean_GCL_chat_mm = mean(GCL_chat_mm),
  sd_GCL_chat_mm = sd(GCL_chat_mm)
)
```

```
## # A tibble: 1 x 4
##   mean_INL_chat_mm sd_INL_chat_mm mean_GCL_chat_mm sd_GCL_chat_mm
##              <dbl>          <dbl>            <dbl>          <dbl>
## 1             980.           206.             842.           219.
```

```
shapiro.test(counts$GCL_chat_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$GCL_chat_mm
## W = 0.88757, p-value = 0.1096
```

```
shapiro.test(counts$INL_chat_mm)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  counts$INL_chat_mm
## W = 0.89834, p-value = 0.151
```

```
counts_long <- counts %>%
  #give one row for each GCL, INL, and total
  #retain only the percentages, not the raw counts
  dplyr::select(-c(GCL_chat:INL_bendo, total_chat:total_perc)) %>%
  pivot_longer(GCL_chat_mm:INL_bendo_mm,
               names_to="type",
               values_to="cells_mm") %>%
  separate(type, sep="_", into=c("layer", "protein", "size")) %>%
  dplyr::select(-size) %>%
  mutate(layer = as.factor(layer),
         protein = as.factor(protein))

#testing manova assumptions
#multivariate normality
#normal distribution for each combo of indep and dependent variables
counts_long %>% filter(layer =="GCL" & protein == "chat" & time == "day") %>% pull(cells_mm) %>% shapiro
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.99202, p-value = 0.9935
```

```
counts_long %>% filter(layer =="GCL" & protein == "chat" & time == "night") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.88836, p-value = 0.3097
```

```
counts_long %>% filter(layer =="INL" & protein == "chat" & time == "day") %>% pull(cells_mm) %>% shapiro
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.9109, p-value = 0.4424
```

```
counts_long %>% filter(layer =="INL" & protein == "chat" & time == "night") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.95001, p-value = 0.7404
```

```
counts_long %>% filter(layer =="GCL" & protein == "chat" & light == "light") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.85537, p-value = 0.1739
```

```r
counts_long %>% filter(layer =="GCL" & protein == "chat" & light == "dark") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.94687, p-value = 0.7149
```

```r
counts_long %>% filter(layer =="INL" & protein == "chat" & light == "light") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.95916, p-value = 0.8133
```

```r
counts_long %>% filter(layer =="INL" & protein == "chat" & light == "dark") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.93201, p-value = 0.5957
```

```r
#checking homogenous variance in each combo
car::leveneTest(counts$GCL_chat_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.6201 0.4493
##       10
```

```r
car::leveneTest(counts$INL_chat_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  2.2117 0.1678
##       10
```

```r
car::leveneTest(counts$GCL_chat_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  1.3156 0.2781
##       10
```

```r
car::leveneTest(counts$INL_chat_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  2.5293 0.1428
##       10
```

```r
#checking for multicollinearity
#low values are good
inllm <- lm(INL_chat_mm ~ light * time, counts)
car::vif(inllm)
```

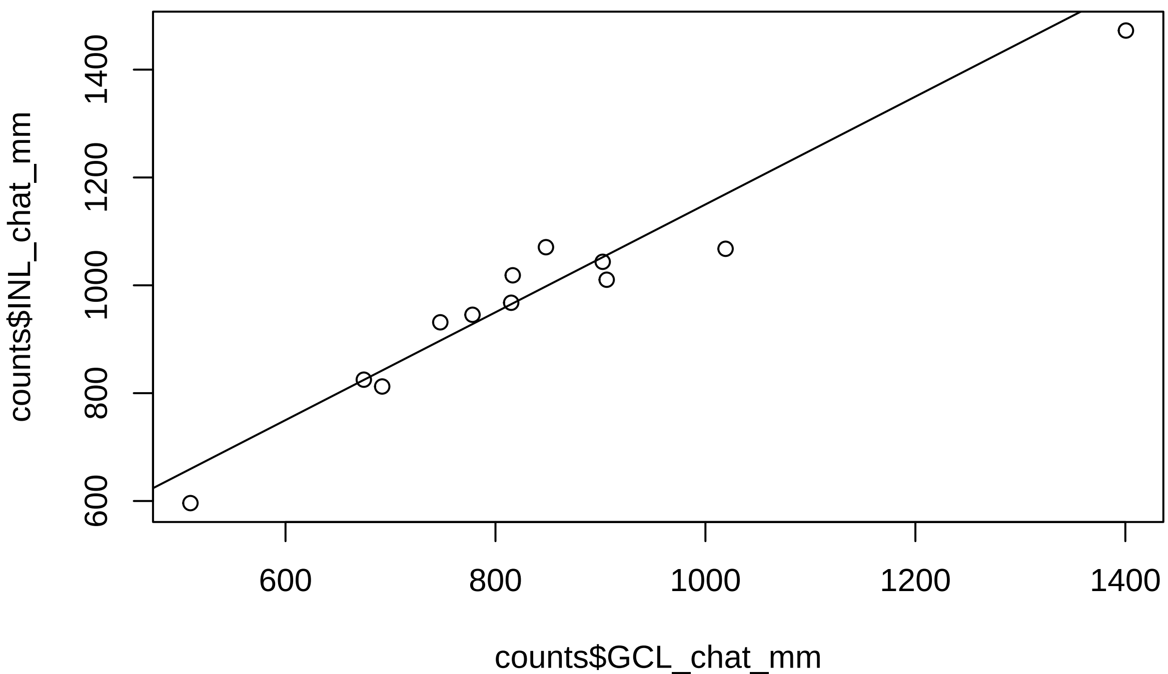
```
##      light       time light:time
```

```
##           2          2          3
```

```
gcllm <- lm(GCL_chat_mm ~ light * time, counts)
car::vif(gcllm)
```

```
##       light       time light:time
##           2          2          3
```

```
#linearity of dep variables
plot(counts$GCL_chat_mm, counts$INL_chat_mm)
abline(150,1)
```




```
#good to go on manova!

#manova
mantest <- manova(cbind(GCL_chat_mm, INL_chat_mm) ~ light * time, data=counts)
summary(mantest)
```

```
##             Df   Pillai approx F num Df den Df Pr(>F)
## light        1 0.054947  0.20350      2      7 0.8205
## time         1 0.005750  0.02024      2      7 0.9800
## light:time   1 0.254205  1.19298      2      7 0.3582
## Residuals    8
```

```
summary.aov(mantest)
```

```
##  Response GCL_chat_mm :
##              Df Sum Sq Mean Sq F value Pr(>F)
## light         1     31      31  0.0005 0.9825
```

```
## time          1     62      62  0.0010 0.9752
## light:time    1  46922   46922  0.7826 0.4021
## Residuals     8 479633   59954
##
##  Response INL_chat_mm :
##             Df Sum Sq Mean Sq F value Pr(>F)
## light        1   1626    1626  0.0290 0.8691
## time         1     13      13  0.0002 0.9884
## light:time   1  17622   17622  0.3141 0.5905
## Residuals    8 448865   56108
```

```
#light:time interaction is significant overall and for GCL and INL
#look at comparisons through univariate models for each dep variable
anova(gcllm)
```

```
## Analysis of Variance Table
##
## Response: GCL_chat_mm
##            Df Sum Sq Mean Sq F value Pr(>F)
## light       1     31      31  0.0005 0.9825
## time        1     62      62  0.0010 0.9752
## light:time  1  46922   46922  0.7826 0.4021
## Residuals   8 479633   59954
```

```
anova(inllm)
```

```
## Analysis of Variance Table
##
## Response: INL_chat_mm
##            Df Sum Sq Mean Sq F value Pr(>F)
## light       1   1626    1626  0.0290 0.8691
## time        1     13      13  0.0002 0.9884
## light:time  1  17622   17622  0.3141 0.5905
## Residuals   8 448865   56108
```

```
#manova for bendo cells per mm2
counts_sumstats <- counts %>% group_by(time, light) %>%
  summarise(mean_INL_bendo_mm = mean(INL_bendo_mm),
  sd_INL_bendo_mm = sd(INL_bendo_mm),
  mean_GCL_bendo_mm = mean(GCL_bendo_mm),
  sd_GCL_bendo_mm = sd(GCL_bendo_mm)
)
```

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

```
shapiro.test(counts$GCL_bendo_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$GCL_bendo_mm
## W = 0.90829, p-value = 0.2028
```

```
shapiro.test(counts$INL_bendo_mm)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  counts$INL_bendo_mm
## W = 0.92663, p-value = 0.3458
```

```
#testing manova assumptions
#multivariate normality
#normal distribution for each combo of indep and dependent variables
counts_long %>% filter(layer =="GCL" & protein == "bendo" & time == "day") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.92539, p-value = 0.545
```

```
counts_long %>% filter(layer =="GCL" & protein == "bendo" & time == "night") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.87924, p-value = 0.2656
```

```
counts_long %>% filter(layer =="INL" & protein == "bendo" & time == "day") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.85792, p-value = 0.1821
```

```
counts_long %>% filter(layer =="INL" & protein == "bendo" & time == "night") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.98211, p-value = 0.9616
```

```
counts_long %>% filter(layer =="GCL" & protein == "bendo" & light == "light") %>% pull(cells_mm) %>% sh
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.86634, p-value = 0.212
```

```
counts_long %>% filter(layer =="GCL" & protein == "bendo" & light == "dark") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.88645, p-value = 0.3
```

```
counts_long %>% filter(layer =="INL" & protein == "bendo" & light == "dark") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  .
## W = 0.87501, p-value = 0.2469
```

```r
#did not pass, but transformations don't either
counts_long %>% filter(layer =="INL" & protein == "bendo" & light == "light") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.73385, p-value = 0.01378
```

```r
#checking homogenous variance in each combo
car::leveneTest(counts$GCL_bendo_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.0384 0.8485
##       10
```

```r
car::leveneTest(counts$INL_bendo_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2279 0.6434
##       10
```

```r
car::leveneTest(counts$GCL_bendo_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2692 0.6152
##       10
```

```r
car::leveneTest(counts$INL_bendo_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2504 0.6276
##       10
```

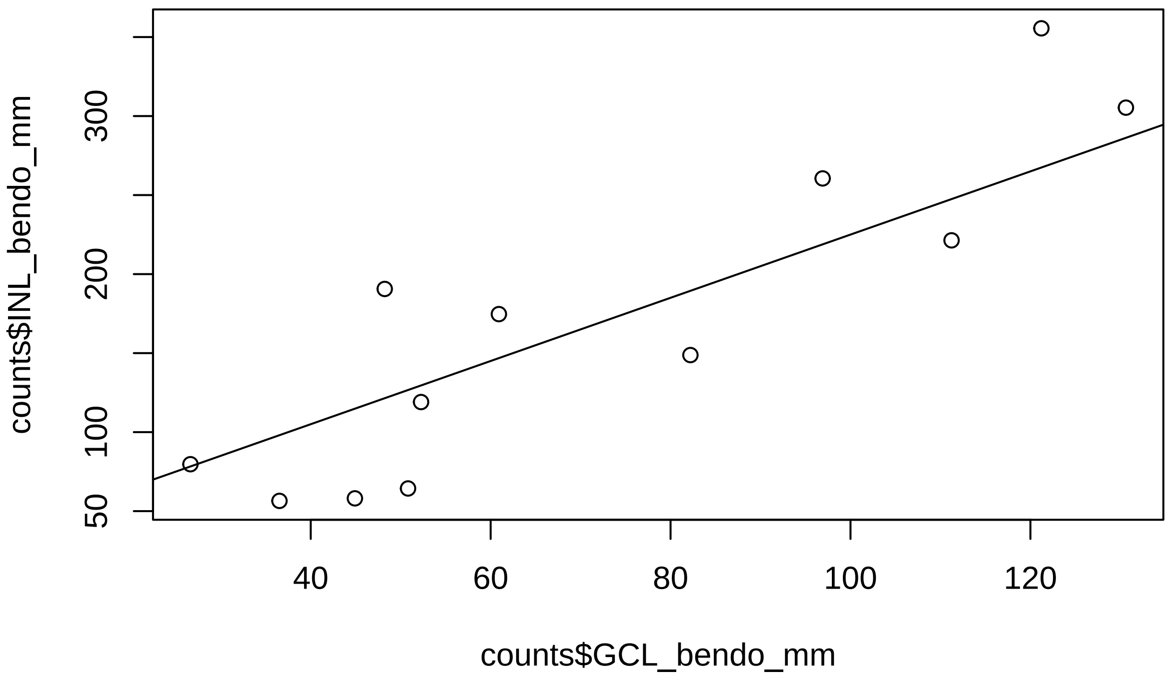
```r
#checking for multicollinearity
#low values are good
inl_bendo_lm <- lm(INL_bendo_mm ~ light * time, counts)
car::vif(inl_bendo_lm)
```

```
##      light       time light:time
##          2          2          3
```

```r
gcl_bendo_lm <- lm(GCL_bendo_mm ~ light * time, counts)
car::vif(gcl_bendo_lm)
```

```
##      light       time light:time
##          2          2          3
```

```r
#linearity of dep variables
plot(counts$GCL_bendo_mm, counts$INL_bendo_mm)
abline(25,2)
```




```r
#good to go on manova!

#manova
man_bendo_test <- manova(cbind(GCL_bendo_mm, INL_bendo_mm) ~ light * time, data=counts)
summary(man_bendo_test)
```

```
##             Df  Pillai approx F num Df den Df  Pr(>F)
## light        1 0.17650   0.7502      2      7 0.50678
## time         1 0.19087   0.8256      2      7 0.47651
## light:time   1 0.58085   4.8503      2      7 0.04767 *
## Residuals    8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(man_bendo_test, type="3")
```

```
##
## Type III MANOVA Tests: Pillai test statistic
##             Df test stat approx F num Df den Df  Pr(>F)
## (Intercept)  1   0.55789   4.4166      2      7 0.05746 .
## light        1   0.51678   3.7430      2      7 0.07844 .
## time         1   0.37018   2.0571      2      7 0.19827
## light:time   1   0.58085   4.8503      2      7 0.04767 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
pf(4.8503, 2, 7)
```

```
## [1] 0.9523261
```

```r
#light:time interaction is significant overall and for GCL, close in INL
#look at comparisons through univariate models for each dep variable
Anova(gcl_bendo_lm, type="3")
```

```
## Anova Table (Type III tests)
##
## Response: GCL_bendo_mm
##             Sum Sq Df F value  Pr(>F)
## (Intercept) 4990.5  1  6.9207 0.03014 *
## light       6163.3  1  8.5470 0.01919 *
## time        3346.1  1  4.6402 0.06337 .
## light:time  7249.1  1 10.0529 0.01318 *
## Residuals   5768.8  8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(inl_bendo_lm, type="3")
```

```
## Anova Table (Type III tests)
##
## Response: INL_bendo_mm
##             Sum Sq Df F value  Pr(>F)
## (Intercept)  13618  1  1.8076 0.21567
## light        45680  1  6.0635 0.03917 *
## time         29435  1  3.9071 0.08348 .
## light:time   36474  1  4.8415 0.05896 .
## Residuals    60270  8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
emmeans::emmeans(gcl_bendo_lm, pairwise ~ light | time)$contrasts
```

```
## time = day:
##  contrast     estimate   SE df t.ratio p.value
##  light - dark    -64.1 21.9  8  -2.924  0.0192
##
## time = night:
##  contrast     estimate   SE df t.ratio p.value
##  light - dark     34.2 21.9  8   1.560  0.1573
```

```r
emmeans::emmeans(gcl_bendo_lm, pairwise ~ time | light)$contrasts
```

```
## light = light:
##  contrast    estimate   SE df t.ratio p.value
##  day - night    -47.2 21.9  8  -2.154  0.0634
##
## light = dark:
##  contrast    estimate   SE df t.ratio p.value
##  day - night     51.1 21.9  8   2.330  0.0482
```

```r
emmeans::emmeans(inl_bendo_lm, pairwise ~ light | time)$contrasts
```

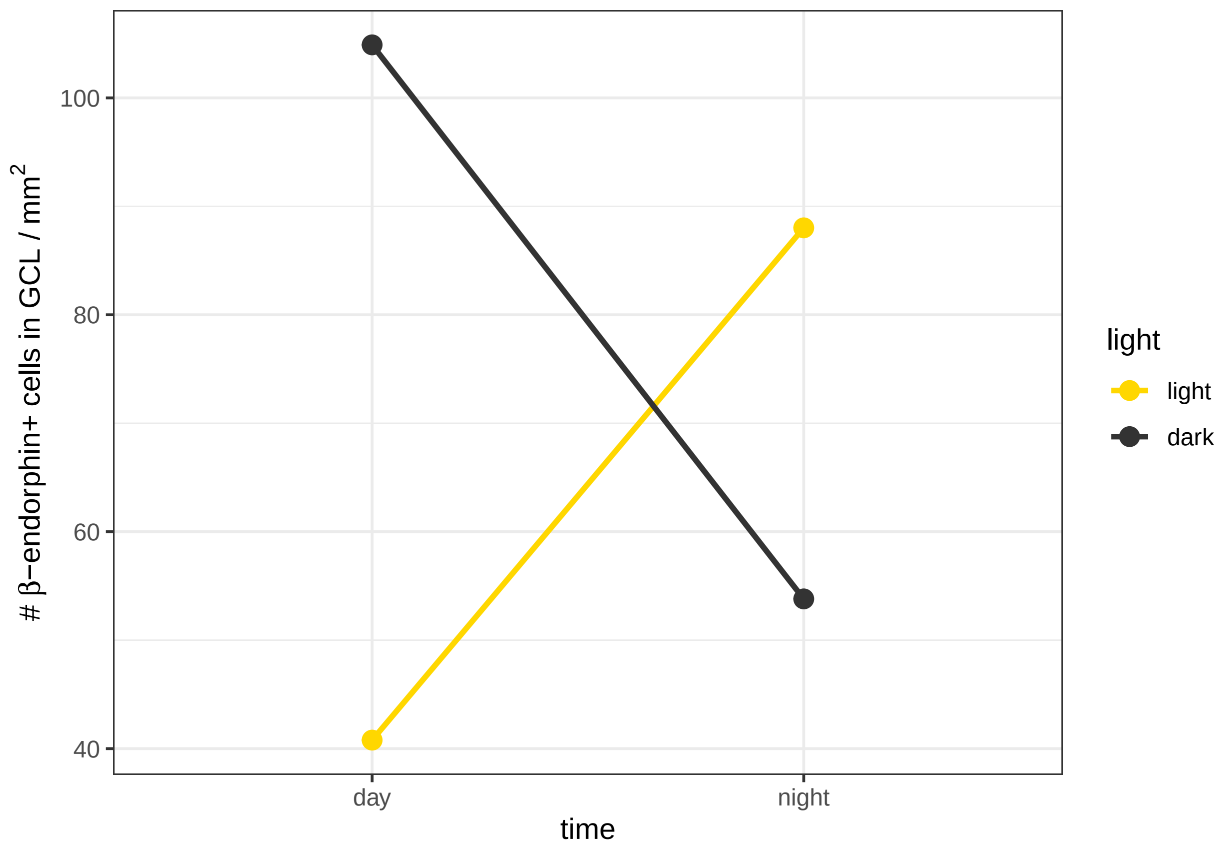
```
## time = day:
```

```
##   contrast     estimate   SE df t.ratio p.value
##   light - dark     -175 70.9  8  -2.462  0.0392
##
## time = night:
##   contrast     estimate   SE df t.ratio p.value
##   light - dark       46 70.9  8   0.649  0.5343
```

```
emmeans::emmeans(inl_bendo_lm, pairwise ~ time | light)$contrasts
```

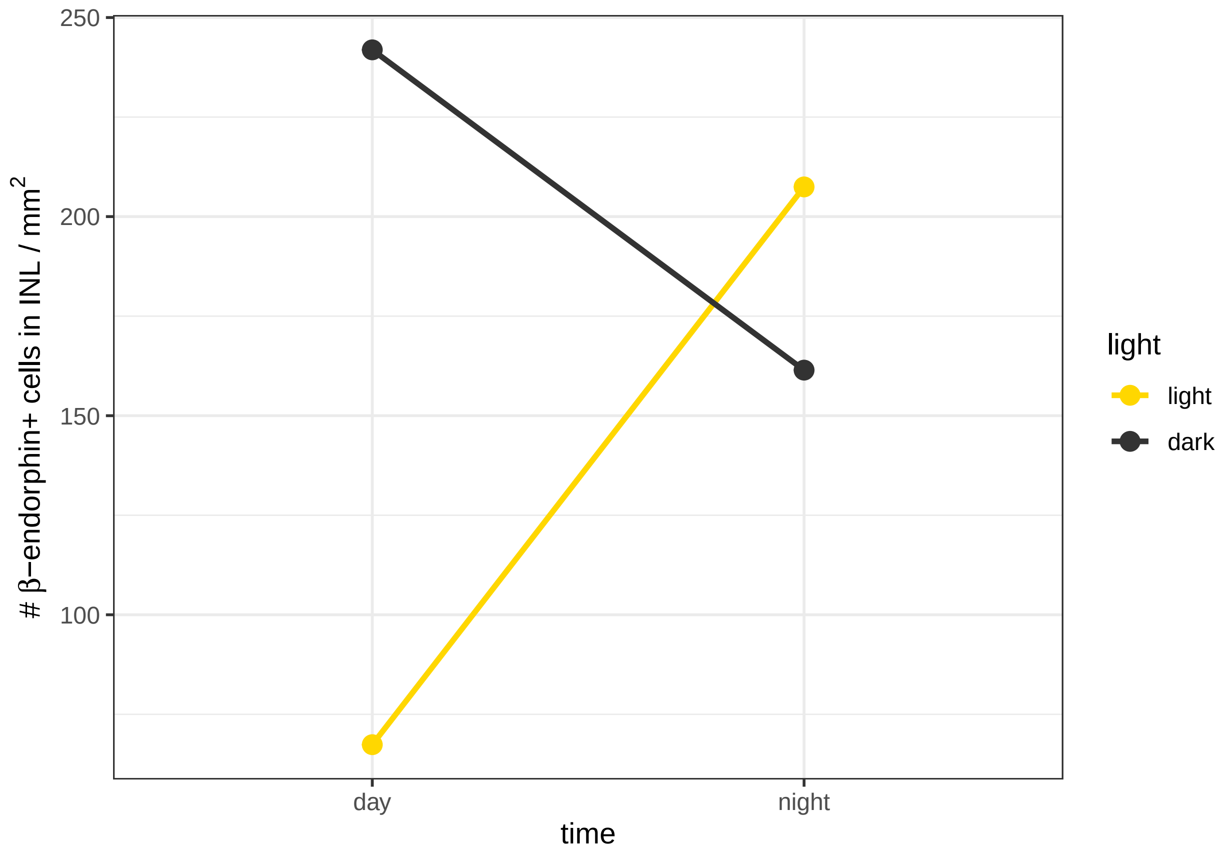
```
## light = light:
##   contrast    estimate   SE df t.ratio p.value
##   day - night   -140.1 70.9  8  -1.977  0.0835
##
## light = dark:
##   contrast    estimate   SE df t.ratio p.value
##   day - night     80.4 70.9  8   1.135  0.2892
```

```
gcl_inter_plot <- ggplot() +
  stat_summary(data=counts, aes(x = time, y = GCL_bendo_mm, color = light, group = light),
               fun = mean, geom = "point", size=3) +
  stat_summary(data=counts, aes(x = time, y = GCL_bendo_mm, color = light, group = light),
               fun = mean, geom = "line", size=1) +
  #stat_summary(data=counts, aes(x = time, y = GCL_bendo_mm, group = light),
  #              fun.data = mean_se, geom="errorbar", width=0.2) +
  labs(y=expression(paste("# ", beta, "-endorphin+ cells in GCL /" ~mm^2))) +
  scale_color_manual(values=c("gold", "grey20")) +
  theme_bw()
gcl_inter_plot
```




```r
#ggsave("../figures/gcl_interaction_plot.svg", plot=gcl_inter_plot, width=5, height=4)

inl_inter_plot <- ggplot() +
  stat_summary(data=counts, aes(x = time, color = light, group = light, y = INL_bendo_mm),
               fun = mean, geom = "point", size=3) +
  stat_summary(data=counts, aes(x = time, color = light, group = light, y = INL_bendo_mm),
               fun = mean, geom = "line", size=1) +
  #stat_summary(data=counts, aes(x = time, y = INL_bendo_mm, group = light),
  #             fun.data = mean_se, geom="errorbar", width=0.2) +
  labs(y=expression(paste("# ", beta, "-endorphin+ cells in INL /" ~mm^2))) +
  scale_color_manual(values=c("gold", "grey20")) +
  theme_bw()
inl_inter_plot
```




```
#ggsave("../figures/inl_interaction_plot.svg", plot=inl_inter_plot, width=5, height=4)
```

```
#plotting cells/mm2
sumstats <- counts %>% group_by(light, time) %>%
            summarise(n = n(),
                      mean_inl = mean(INL_bendo_mm),
                      sd_inl = sd(INL_bendo_mm),
                      mean_gcl = mean(GCL_bendo_mm),
                      sd_gcl = sd(GCL_bendo_mm))
```

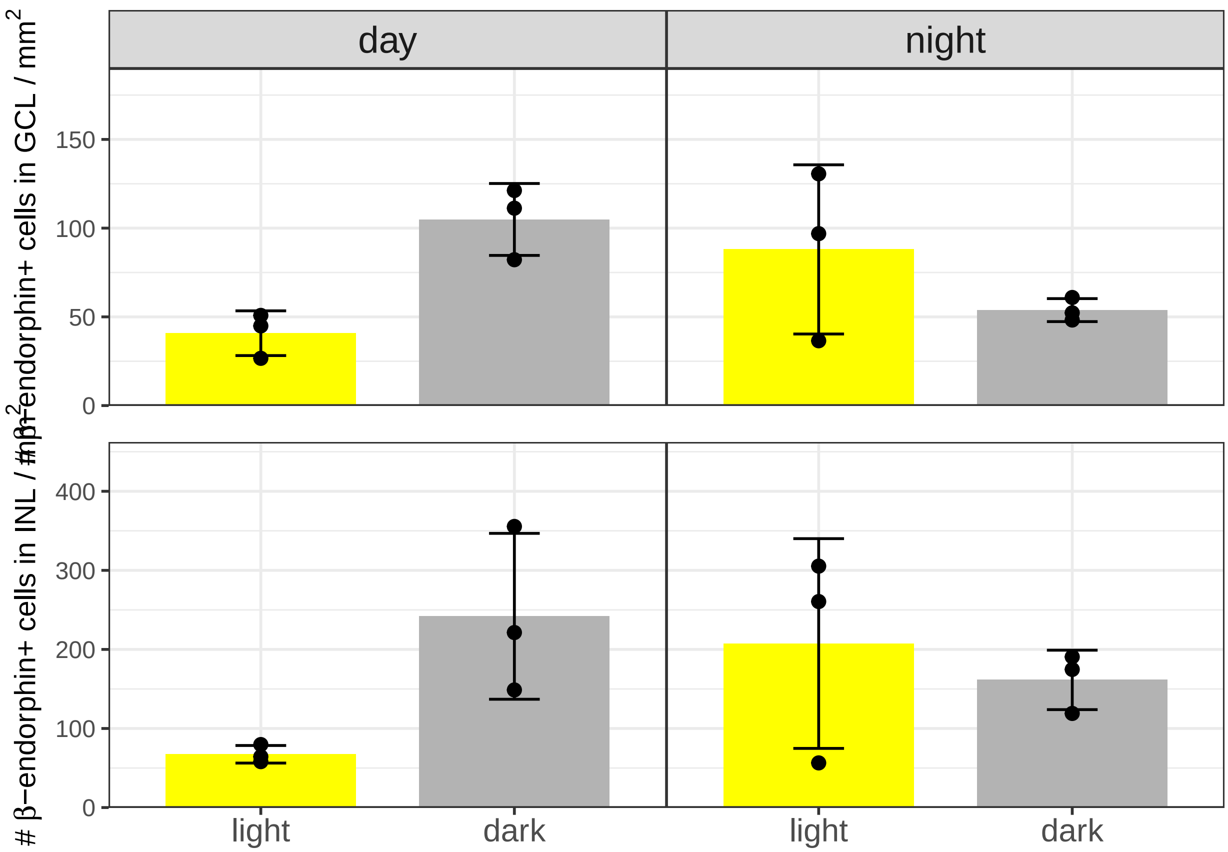
```
## `summarise()` has grouped output by 'light'. You can override using the `.groups` argument.
```

```
gcl_mm_plot <- ggplot() +
  geom_bar(data=sumstats, aes(x=light, y=mean_gcl, fill=light), stat="identity", width=0.75) +
  geom_errorbar(data=sumstats, aes(x=light,
                                   ymin=mean_gcl-sd_gcl,
                                   ymax=mean_gcl+sd_gcl), width=0.2) +
  geom_point(data=counts, aes(x=light, y=GCL_bendo_mm), size=2) +
  facet_wrap(~time) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        panel.spacing = unit(0, "pt"),
        strip.text.x = element_text(size = 14),
        legend.position="none",
        plot.background = element_rect(fill = "transparent",colour = NA),
```

```
        panel.background = element_rect(fill = "transparent",colour = NA)) +
  labs(y=expression(paste("# ", beta, "-endorphin+ cells in GCL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.4)))

inl_mm_plot <- ggplot() +
  geom_bar(data=sumstats, aes(x=light, y=mean_inl, fill=light), stat="identity", width=0.75) +
  geom_errorbar(data=sumstats, aes(x=light,
                            ymin=mean_inl-sd_inl,
                            ymax=mean_inl+sd_inl), width=0.2) +
  geom_point(data=counts, aes(x=light, y=INL_bendo_mm), size=2) +
  facet_wrap(~time) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.text.x = element_text(size = 12),
        panel.spacing = unit(0, "pt"),
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        legend.position="none",
        plot.background = element_rect(fill = "transparent",colour = NA),
        panel.background = element_rect(fill = "transparent",colour = NA)) +
  labs(y=expression(paste("# ", beta, "-endorphin+ cells in INL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.3)))

both_mm_plots <- ggarrange(gcl_mm_plot, inl_mm_plot, nrow=2)
both_mm_plots
```




```r
#ggsave("../figures/both_mm_plots.svg", plot=both_mm_plots, height=6, width=6, bg = "transparent")

ctrl_counts <- counts %>% filter(light == "light" & time == "day")

#between var
sum(3*(mean(counts$GCL_bendo_mm - mean(ctrl_counts$GCL_bendo_mm)))^2)
```

```
## [1] 2899.204
```

```r
#within.var
anova(gcl_bendo_lm)["Residuals", "Mean Sq"]
```

```
## [1] 721.0983
```

```r
power.anova.test(groups=4, n=3, between.var=2899.204,
                 within.var=721.0983, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 3
##     between.var = 2899.204
##      within.var = 721.0983
##       sig.level = 0.05
##           power = 0.9820665
##
## NOTE: n is number in each group
```

```r
#between var
sum(3*(mean(counts$INL_bendo_mm - mean(ctrl_counts$INL_bendo_mm)))^2)
```

```
## [1] 31312.94
```

```r
#within.var
anova(inl_bendo_lm)["Residuals", "Mean Sq"]
```

```
## [1] 7533.694
```

```r
power.anova.test(groups=4, n=3, between.var=31312.94,
                 within.var=7533.694, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 3
##     between.var = 31312.94
##      within.var = 7533.694
##       sig.level = 0.05
##           power = 0.9849714
##
## NOTE: n is number in each group
```

```r
#percentage / mm2
counts <- counts %>% mutate(inl_perc_mm = INL_bendo_mm/INL_chat_mm*100,
                            gcl_perc_mm = GCL_bendo_mm/GCL_chat_mm*100)

perc_sumstats <- counts %>% group_by(light, time) %>%
            summarise(n = n(),
                      mean_inl_perc = mean(inl_perc_mm),
                      sd_inl_perc = sd(inl_perc_mm),
                      mean_gcl_perc = mean(gcl_perc_mm),
                      sd_gcl_perc = sd(gcl_perc_mm))
```

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

```r
shapiro.test(counts$gcl_perc_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$gcl_perc_mm
## W = 0.94973, p-value = 0.633
```

```r
shapiro.test(counts$inl_perc_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$inl_perc_mm
## W = 0.94583, p-value = 0.5771
```

```r
counts_perc_long <- counts %>%
  #give one row for each GCL, INL, and total
  #retain only the percentages, not the raw counts
  dplyr::select(-c(GCL_chat:INL_bendo, total_chat:INL_bendo_mm)) %>%
```

```r
  pivot_longer(inl_perc_mm:gcl_perc_mm,
               names_to="type",
               values_to="perc_mm") %>%
  separate(type, sep="_", into=c("layer", "na", "size")) %>%
  dplyr::select(-c(na,size)) %>%
  mutate(layer = as.factor(layer))

#testing manova assumptions
#multivariate normality
#normal distribution for each combo of indep and dependent variables
counts_perc_long %>% filter(layer =="gcl" & time == "day") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.9171, p-value = 0.4847
```

```r
counts_perc_long %>% filter(layer =="gcl" & time == "night") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.96858, p-value = 0.8828
```

```r
counts_perc_long %>% filter(layer =="gcl" & time == "day") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.9171, p-value = 0.4847
```

```r
counts_perc_long %>% filter(layer =="gcl" & time == "night") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.96858, p-value = 0.8828
```

```r
counts_perc_long %>% filter(layer =="gcl" & light == "light") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.99193, p-value = 0.9934
```

```r
counts_perc_long %>% filter(layer =="gcl" & light == "dark") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.93389, p-value = 0.6104
```

```r
counts_perc_long %>% filter(layer =="inl" & light == "dark") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.91084, p-value = 0.442
```

```r
counts_perc_long %>% filter(layer =="inl" & light == "light") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.80407, p-value = 0.06392
```

```r
#checking homogenous variance in each combo
car::leveneTest(counts$gcl_perc_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.0151 0.9047
##       10
```

```r
car::leveneTest(counts$inl_perc_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.0504 0.8268
##       10
```

```r
car::leveneTest(counts$gcl_perc_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  2.5789 0.1394
##       10
```

```r
car::leveneTest(counts$inl_perc_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  1.3768 0.2678
##       10
```

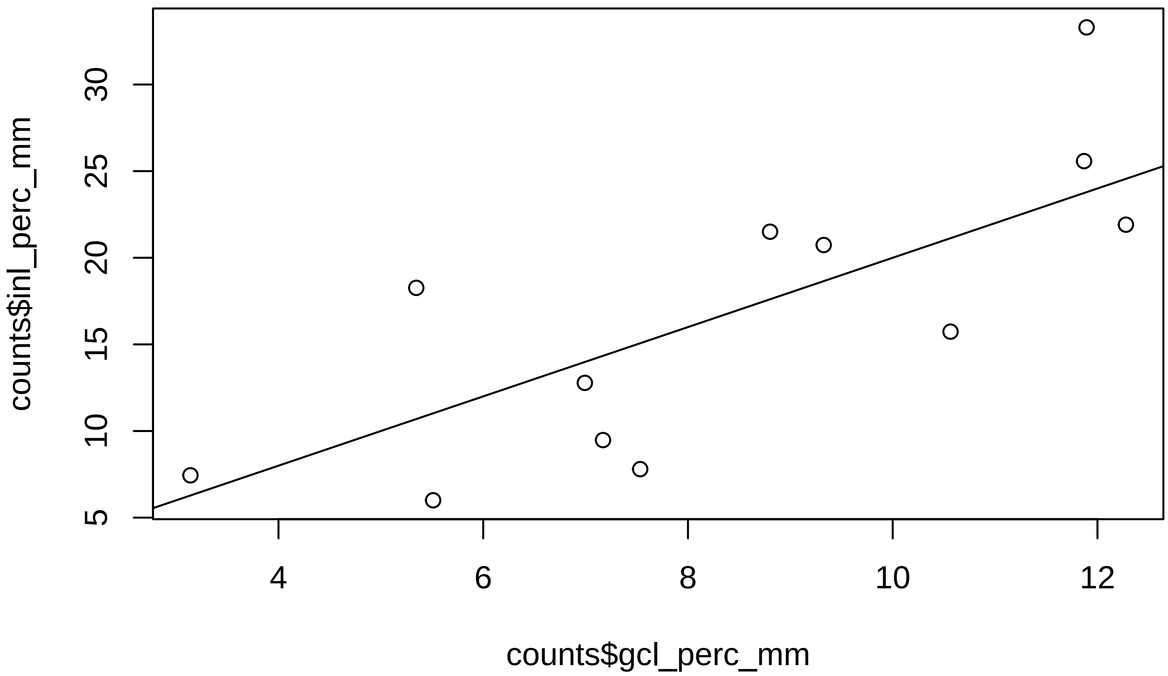
```r
#checking for multicollinearity
#low values are good
inl_perc_lm <- lm(inl_perc_mm ~ light * time, counts)
car::vif(inl_perc_lm)
```

```
##      light       time light:time
##          2          2          3
```

```r
gcl_perc_lm <- lm(gcl_perc_mm ~ light * time, counts)
car::vif(gcl_perc_lm)
```

```
##      light       time light:time
##          2          2          3
```

```r
#linearity of dep variables
plot(counts$gcl_perc_mm, counts$inl_perc_mm)
abline(0,2)
```




```r
#good to go on manova!

#manova
man_perc_test <- manova(cbind(gcl_perc_mm, inl_perc_mm) ~ light * time, data=counts)
summary(man_perc_test)
```

```
##            Df  Pillai approx F num Df den Df  Pr(>F)
## light       1 0.37461   2.0965      2      7 0.19343
## time        1 0.11699   0.4637      2      7 0.64697
## light:time  1 0.66218   6.8605      2      7 0.02241 *
## Residuals   8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary.aov(man_perc_test)
```

```
##  Response gcl_perc_mm :
##             Df Sum Sq Mean Sq F value   Pr(>F)
## light        1 10.693  10.693  3.0196 0.120464
## time         1  0.167   0.167  0.0471 0.833664
## light:time   1 55.374  55.374 15.6365 0.004211 **
## Residuals    8 28.331   3.541
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response inl_perc_mm :
##             Df Sum Sq Mean Sq F value  Pr(>F)
## light        1 179.82 179.818  4.2818 0.07231 .
## time         1  21.74  21.741  0.5177 0.49231
## light:time   1 233.58 233.584  5.5621 0.04607 *
## Residuals    8 335.96  41.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#light:time interaction is just about significant overall and for GCL and INL
#look at comparisons through univariate models for each dep variable
anova(gcl_perc_lm)
```

```
## Analysis of Variance Table
##
## Response: gcl_perc_mm
##            Df Sum Sq Mean Sq F value   Pr(>F)
## light       1 10.693  10.693  3.0196 0.120464
## time        1  0.167   0.167  0.0471 0.833664
## light:time  1 55.374  55.374 15.6365 0.004211 **
## Residuals   8 28.331   3.541
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(gcl_perc_lm)
```

```
##
## Call:
## lm(formula = gcl_perc_mm ~ light * time, data = counts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28500 -1.18579  0.03093  0.96322  2.41406
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.395      1.086   4.965  0.00110 **
## lightdark              6.184      1.537   4.025  0.00382 **
## timenight              4.061      1.537   2.643  0.02959 *
## lightdark:timenight   -8.593      2.173  -3.954  0.00421 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.882 on 8 degrees of freedom
## Multiple R-squared:  0.7004, Adjusted R-squared:  0.5881
## F-statistic: 6.234 on 3 and 8 DF,  p-value: 0.01728
```

```
anova(inl_perc_lm)
```

```
## Analysis of Variance Table
##
## Response: inl_perc_mm
##            Df Sum Sq Mean Sq F value  Pr(>F)
## light       1 179.82 179.818  4.2818 0.07231 .
```

```
## time        1  21.74  21.741  0.5177 0.49231
## light:time  1 233.58 233.584  5.5621 0.04607 *
## Residuals   8 335.96  41.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(inl_perc_lm)
```

```
##
## Call:
## lm(formula = inl_perc_mm ~ light * time, data = counts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1204 -2.4874  0.5396  2.6011  9.6505
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)             7.082      3.741   1.893   0.0950 .
## lightdark              16.566      5.291   3.131   0.0140 *
## timenight              11.516      5.291   2.176   0.0612 .
## lightdark:timenight   -17.648      7.483  -2.358   0.0461 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.48 on 8 degrees of freedom
## Multiple R-squared:  0.5643, Adjusted R-squared:  0.4009
## F-statistic: 3.454 on 3 and 8 DF,  p-value: 0.07147
```

```
emmeans::emmeans(gcl_perc_lm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                 estimate   SE df t.ratio p.value
##  light day - dark day        -6.18 1.54  8  -4.025  0.0038
##  light day - light night     -4.06 1.54  8  -2.643  0.0296
##  light day - dark night      -1.65 1.54  8  -1.075  0.3136
##  dark day - light night       2.12 1.54  8   1.382  0.2043
##  dark day - dark night        4.53 1.54  8   2.950  0.0184
##  light night - dark night     2.41 1.54  8   1.567  0.1557
```

```
emmeans::emmeans(inl_perc_lm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                 estimate   SE df t.ratio p.value
##  light day - dark day       -16.57 5.29  8  -3.131  0.0140
##  light day - light night    -11.52 5.29  8  -2.176  0.0612
##  light day - dark night     -10.43 5.29  8  -1.972  0.0841
##  dark day - light night       5.05 5.29  8   0.954  0.3678
##  dark day - dark night        6.13 5.29  8   1.159  0.2799
##  light night - dark night     1.08 5.29  8   0.204  0.8431
```

```
gcl_perc_plot <- ggplot() +
  geom_bar(data=perc_sumstats, aes(x=light, y=mean_gcl_perc, fill=light), stat="identity") +
  geom_errorbar(data=perc_sumstats, aes(x=light,
                            ymin=mean_gcl_perc-sd_gcl_perc,
                            ymax=mean_gcl_perc+sd_gcl_perc), width=0.2) +
  geom_point(data=counts, aes(x=light, y=gcl_perc_mm)) +
  facet_wrap(~time) +
```

```
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        panel.spacing = unit(0, "pt"),
        legend.position="none") +
  labs(y=expression(paste("% ", beta, "-endorphin+/ChAT+ cells in GCL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.4)),
                     labels=function(x) paste0(x,"%"))

inl_perc_plot <- ggplot() +
  geom_bar(data=perc_sumstats, aes(x=light, y=mean_inl_perc, fill=light), stat="identity") +
  geom_errorbar(data=perc_sumstats, aes(x=light,
                              ymin=mean_inl_perc-sd_inl_perc,
                              ymax=mean_inl_perc+sd_inl_perc), width=0.2) +
  geom_point(data=counts, aes(x=light, y=inl_perc_mm)) +
  facet_wrap(~time) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.ticks.x = element_blank(),
        panel.spacing = unit(0, "pt"),
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        legend.position="none") +
  labs(y=expression(paste("% ", beta, "-endorphin+/ChAT+ cells in INL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.3)),
                     labels=function(x) paste0(x,"%"))

#,
#        plot.background = element_rect(fill = "transparent",colour = NA),
#        panel.background = element_rect(fill = "transparent",colour = NA)

both_perc_plots <- ggarrange(gcl_perc_plot, inl_perc_plot, nrow=2)

ggsave("../figures/both_perc_plots.png", plot=both_perc_plots, height=7, width=6, bg = "transparent")

ctrl_counts <- counts %>% filter(light == "light" & time == "day")

#between var
sum(3*(mean(counts$gcl_perc_mm - mean(ctrl_counts$gcl_perc_mm)))^2)
```

```
## [1] 26.5387
```

```
#within.var
anova(gcl_perc_lm)["Residuals", "Mean Sq"]
```

```
## [1] 3.541329
```

```
power.anova.test(groups=4, n=3, between.var=26.5387,
                 within.var=3.541329, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
```

```
##
##          groups = 4
##               n = 3
##     between.var = 26.5387
##      within.var = 3.541329
##       sig.level = 0.05
##           power = 0.9998598
##
## NOTE: n is number in each group
```

```r
#between var
sum(3*(mean(counts$inl_perc_mm - mean(ctrl_counts$inl_perc_mm)))^2)
```

```
## [1] 278.1516
```

```r
#within.var
anova(inl_perc_lm)["Residuals", "Mean Sq"]
```

```
## [1] 41.99559
```

```r
power.anova.test(groups=4, n=3, between.var=278.1516,
                 within.var=41.99559, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 3
##     between.var = 278.1516
##      within.var = 41.99559
##       sig.level = 0.05
##           power = 0.9995018
##
## NOTE: n is number in each group
```

```r
summary_table <- counts %>% group_by(light, time) %>%
  summarise(n = n(),
            "Total # ChAT+ counted in INL" = sum(INL_chat),
            "Total # B-endo+ counted in INL" = sum(INL_bendo),
            "Mean ChAT+ cells in INL/mm2" = paste0(round(mean(INL_chat_mm), 0),
                                                   "±", round(sd(INL_chat_mm),0)),
            "Mean B-endo+ cells in INL/mm2" = paste0(round(mean(INL_bendo_mm), 0),
                                                    "±", round(sd(INL_bendo_mm),0)),
            "Total # ChAT+ counted in GCL" = sum(GCL_chat),
            "Total # B-endo+ counted in GCL" = sum(GCL_bendo),
            "Mean ChAT+ cells in GCL/mm2" = paste0(round(mean(GCL_chat_mm), 0),
                                                   "±", round(sd(GCL_chat_mm),0)),
            "Mean B-endo+ cells in GCL/mm2" = paste0(round(mean(GCL_bendo_mm), 0),
                                                    "±", round(sd(GCL_bendo_mm),0)))
```

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

```r
summary_table
```

```
## # A tibble: 4 x 11
## # Groups:   light [2]
##   light time      n `Total # ChAT+ coun~ `Total # B-endo+ c~ `Mean ChAT+ cells ~
##   <fct> <fct> <int>                <dbl>               <dbl> <chr>
```

```
## 1 light day       3                10834                  767 954±123
## 2 light night     3                14180                 2782 1029±438
## 3 dark  day       3                14630                 3416 1008±61
## 4 dark  night     3                14103                 2573 929±116
## # ... with 5 more variables: Mean B-endo+ cells in INL/mm2 <chr>,
## #   Total # ChAT+ counted in GCL <dbl>, Total # B-endo+ counted in GCL <dbl>,
## #   Mean ChAT+ cells in GCL/mm2 <chr>, Mean B-endo+ cells in GCL/mm2 <chr>
```

```
#write_xlsx(summary_table, "../data/bendo_summary_table.xlsx")
```