R Code for Visualizations of EEG Data

Nicole Sullivan & Don Lyons

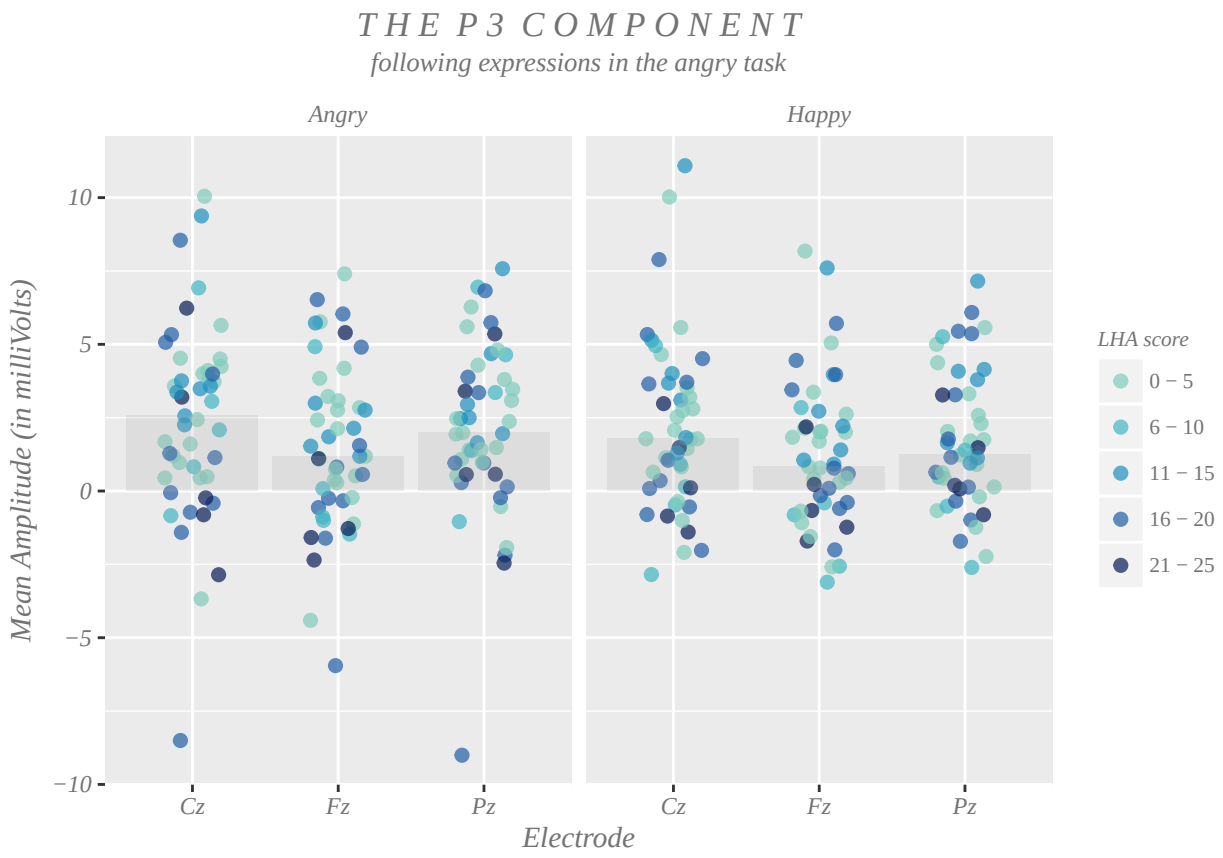
6/4/2017

```
library(tidyverse)
erp_erd <- read_csv("ERP_ERD.csv")
```

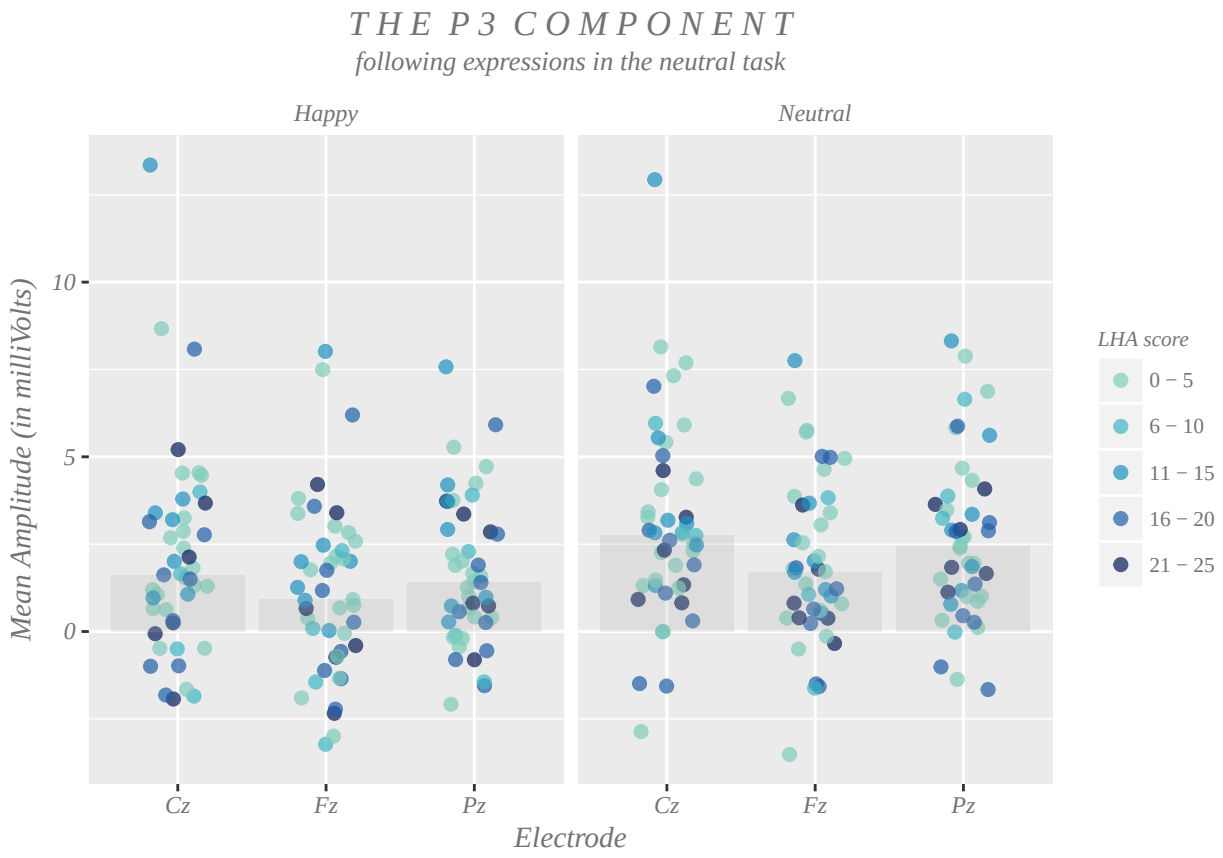
Stripcharts of Expressions in Angry and Neutral Tasks

```
strip_theme <- theme(plot.title = element_text(family = "Times", face = "italic", hjust = 0.5, color =
        plot.subtitle = element_text(family = "Times", face = "italic", hjust = 0.5, color = "gray45"),
        axis.title = element_text(family = "Times", face = "italic", color = "gray45"),
        axis.text = element_text(family = "Times", face = "italic", color = "gray45"),
        strip.text.x = element_text(family = "Times", face = "italic", color = "gray45"),
        strip.background = element_blank(),
        legend.title = element_text(family = "Times", face = "italic", size = 8, color = "gray45"),
        legend.text = element_text(family = "Times", color = "gray45", size = 8))

Cz_Fz_Pz <- erp_erd %>%
  filter(Electrode == "Cz" | Electrode == "Pz" | Electrode == "Fz") %>%
  mutate(LHA = ifelse(LHA == "NaN", 0, LHA)) %>%
  mutate(LHA_levels = ifelse(LHA == 0, "0 - 5",
                             ifelse(LHA == 1, "0 - 5",
                                    ifelse(LHA == 2, "0 - 5",
                                           ifelse(LHA == 3, "0 - 5",
                                                  ifelse(LHA == 4, "0 - 5",
                                                         ifelse(LHA == 5, "0 - 5",
                                                                ifelse(LHA == 6, "6 - 10",
                                                                       ifelse(LHA == 7, "6 - 10",
                                                                              ifelse(LHA == 8, "6 -10",
                                                                                     ifelse(LHA == 9, "
                                                                                            ifelse(LHA =
                                                                                                   ifels
  na.omit()

Cz_Fz_Pz %>%
  filter(Task == "A") %>%
  mutate(Expression = ifelse(Expression == "A", "Angry", "Happy")) %>%
ggplot(aes(Electrode, Mean_Amplitude)) +
  geom_jitter(position=position_jitter(0.2), alpha = 0.7, aes(color = LHA_levels), size = 2) +
```

```
  stat_summary(fun.y=median, geom="bar", fill ="darkgray", size = 1, alpha = 0.2) +
  scale_color_manual(breaks = c("0 - 5", "6 - 10", "11 - 15", "16 - 20", "21 - 25"), values = c("0 - 5"
  labs(title = "T H E  P 3  C O M P O N E N T",
       subtitle = "following expressions in the angry task",
       color = "LHA score",
       y = "Mean Amplitude (in milliVolts)") +
  facet_wrap(~Expression) +
  strip_theme
```

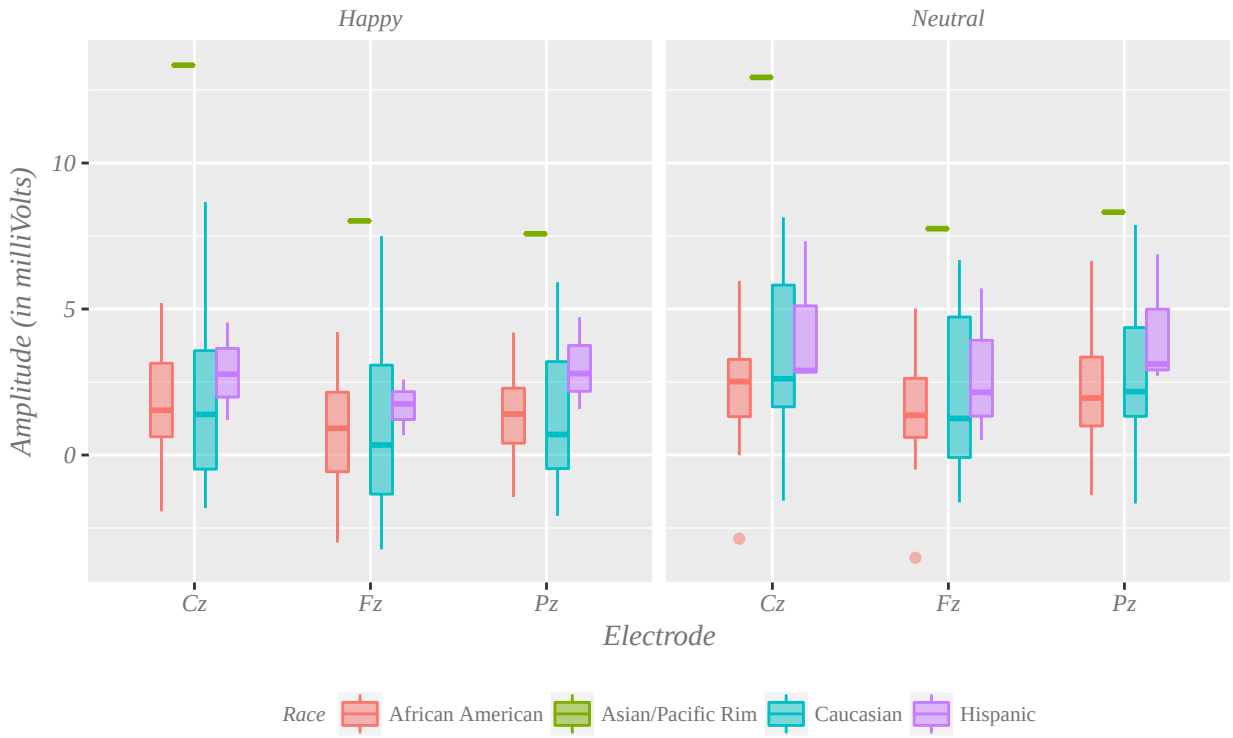



```
Cz_Fz_Pz %>%
  filter(Task == "N") %>%
  mutate(Expression = ifelse(Expression == "N", "Neutral", "Happy")) %>%
ggplot(aes(Electrode, Mean_Amplitude)) +
  geom_jitter(position=position_jitter(0.2), alpha = 0.7, aes(color = LHA_levels), size = 2) +
  stat_summary(fun.y=median, geom="bar", fill ="darkgray", size = 1, alpha = 0.2) +
  scale_color_manual(breaks = c("0 - 5", "6 - 10", "11 - 15", "16 - 20", "21 - 25"), values = c("0 - 5"
  labs(title = "T H E  P 3  C O M P O N E N T",
       subtitle = "following expressions in the neutral task",
       color = "LHA score",
       y = "Mean Amplitude (in milliVolts)") +
  facet_wrap(~Expression) +
  strip_theme
```



Race Boxplot for P3 in Angry Task

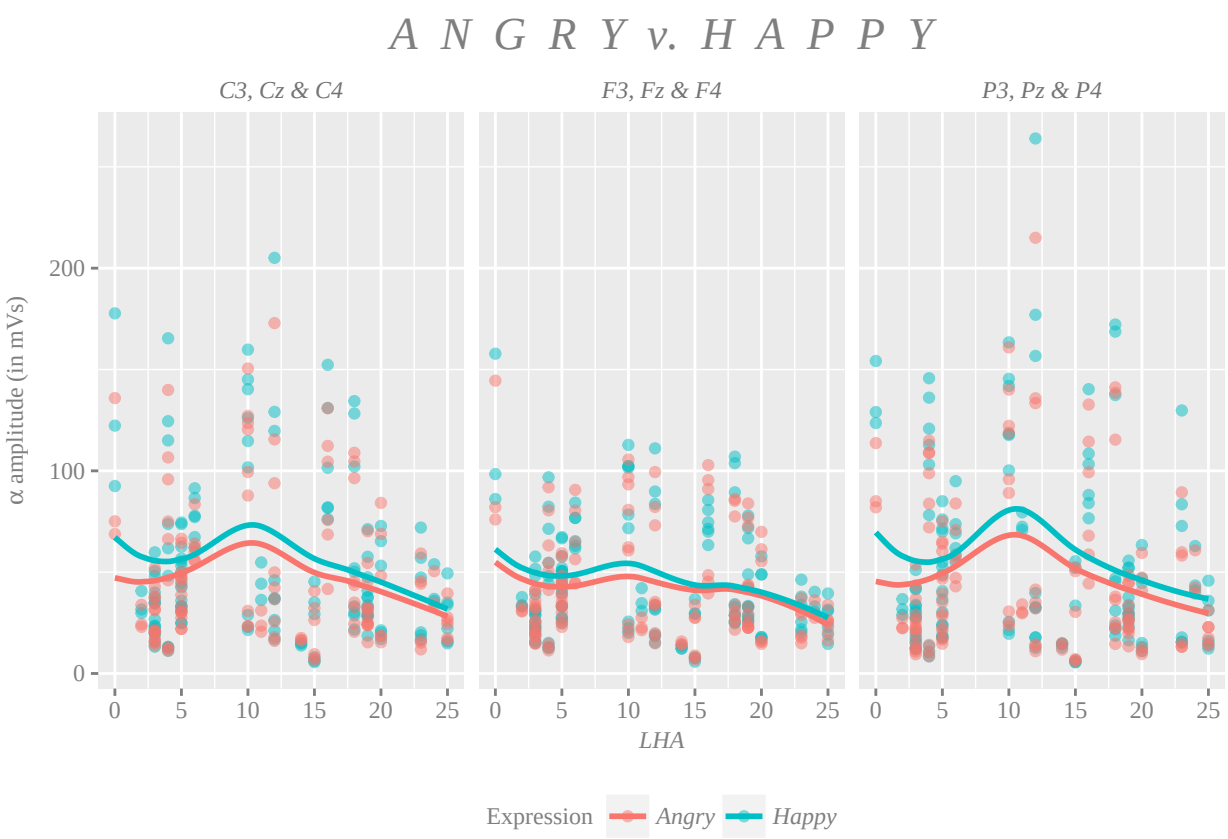
```
Cz_Fz_Pz %>%
  filter(Task == "N") %>%
  mutate(Expression = ifelse(Expression == "N", "Neutral", "Happy")) %>%
  mutate(Race = ifelse(Race == 1, "Caucasian",
                       ifelse(Race == 2, "African American",
                              ifelse(Race == 3, "Asian/Pacific Rim", "Hispanic")))) %>%
ggplot(aes(Electrode, Mean_Amplitude)) +
  geom_boxplot(aes(color = Race, fill = Race), alpha = 0.5, width = 0.5) +
  labs(title = "T H E  P 3  C O M P O N E N T",
       subtitle = "following expressions in the angry task",
       y = "Amplitude (in milliVolts)") +
  facet_wrap(~Expression) +
theme(plot.title = element_text(family = "Times", face = "italic", hjust = 0.5, color = "gray40"),
        plot.subtitle = element_text(family = "Times", face = "italic", hjust = 0.5, color = "gray35"),
        axis.title = element_text(family = "Times", face = "italic", color = "gray45"),
        axis.text = element_text(family = "Times", face = "italic", color = "gray45"),
        strip.text.x = element_text(family = "Times", face = "italic", color = "gray45"),
        strip.background = element_blank(),
      legend.position = "bottom",
        legend.title = element_text(family = "Times", face = "italic", size = 8, color = "gray45"),
        legend.text = element_text(family = "Times", color = "gray45", size = 8))
```



Comparing Responses as a Function of Aggression

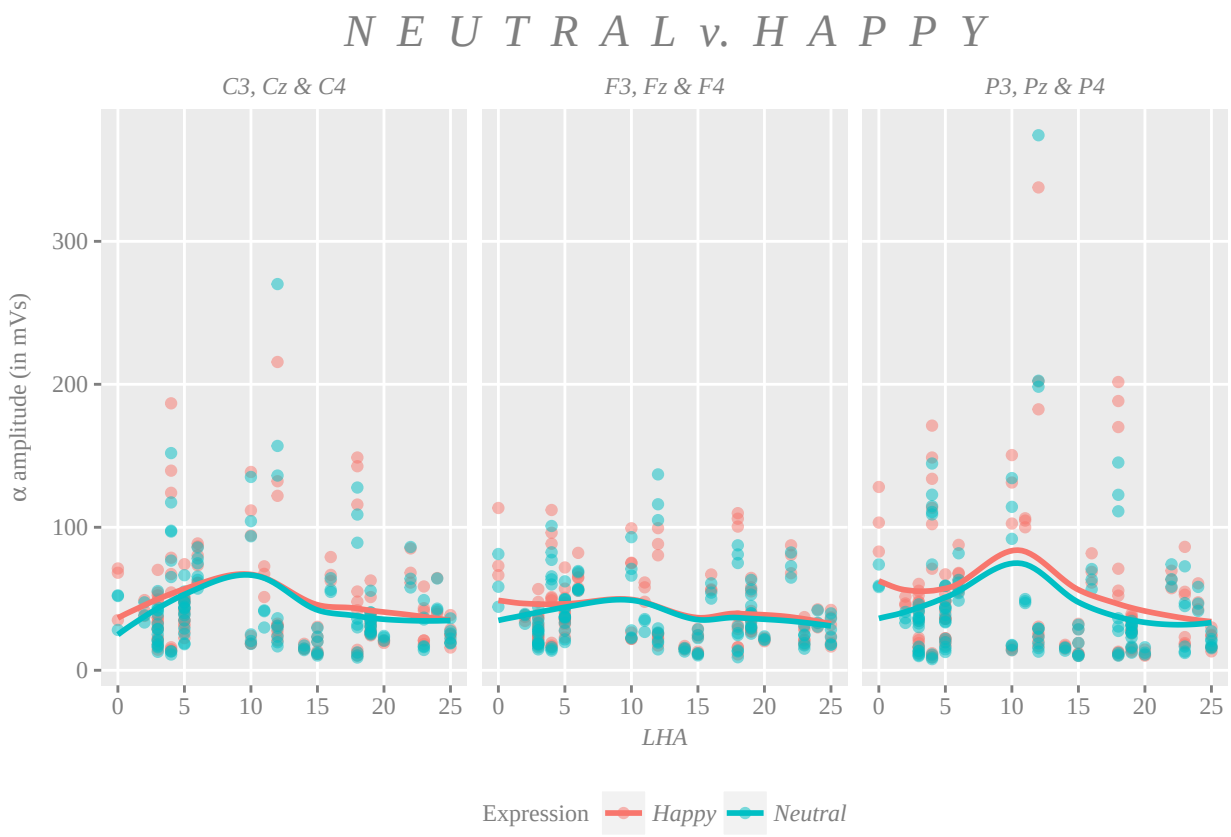
```
erp_erd_data <- erp_erd %>%
  mutate(Laterality = ifelse(Electrode == "C3" | Electrode == "P3" | Electrode == "F3", "L",
                             ifelse(Electrode == "Cz" | Electrode == "Pz" | Electrode == "Fz", "M",
                                    ifelse(Electrode == "C4" | Electrode == "P4" | Electrode == "F4", "
  mutate(Anterior_Posterior = ifelse(Electrode == "F3" | Electrode == "Fz" | Electrode == "F4", "F",
                                     ifelse(Electrode == "C3" | Electrode == "Cz" | Electrode == "C4",
                                            ifelse(Electrode == "P3" | Electrode == "Pz" | Electrode ==
  rename(Mean = Mean_Amplitude) %>%
  gather(Mean, Delta, Theta, Alpha, Beta, Gamma, key = "Amplitude_Type", value = "Amplitude")

alpha_theta <- erp_erd_data %>%
  filter(Electrode == "C3" | Electrode == "P3" | Electrode == "F3" | Electrode == "Cz" | Electrode == "
           Electrode == "C4" | Electrode == "P4" | Electrode == "F4") %>%
  filter(Amplitude_Type == "Alpha" | Amplitude_Type == "Theta")

alpha_theta %>%
  filter(Task == "A") %>%
  filter(Amplitude_Type == "Alpha") %>%
  mutate(Expression = ifelse(Expression == "A", "Angry", "Happy")) %>%
  mutate(Anterior_Posterior = ifelse(Anterior_Posterior == "C", "C3, Cz & C4",
                                     ifelse(Anterior_Posterior == "F", "F3, Fz & F4", "P3, Pz & P4"))) %
  na.omit %>%
ggplot(aes(LHA, Amplitude, color = Expression)) +
  geom_point(alpha = 0.5) +
```

```
  geom_smooth(se = FALSE) +
  facet_wrap(~Anterior_Posterior) +
  labs(title = "A  N  G  R  Y  v.  H  A  P  P  Y",
       y = expression(paste(alpha, " amplitude (in mVs)"))) +
  theme(plot.title = element_text(hjust = 0.5, size = 17, face = "italic",
                                  family = "Times", color = "gray50"),
        axis.title.x = element_text(hjust = 0.5, size = 9, face = "italic",
                                    family = "Times", color = "gray50"),
        axis.title.y = element_text(hjust = 0.5, size = 9, face = "italic",
                                    family = "Times", color = "gray50"),
        plot.caption = element_text(size = 8, color = "gray50", family = "Times"),
        strip.text.x = element_text(family = "Times", face = "italic", color = "gray45"),
        strip.background = element_blank(),
        axis.text.x = element_text(family = "Times", color = "gray50"),
        axis.text.y = element_text(family = "Times", color = "gray50"),
        axis.ticks = element_line(color = "gray50"),
        legend.position = "bottom",
        legend.title = element_text(family = "Times", color = "gray50", size = 9),
        legend.text = element_text(family = "Times", color = "gray50", size = 9, face = "italic"))
```



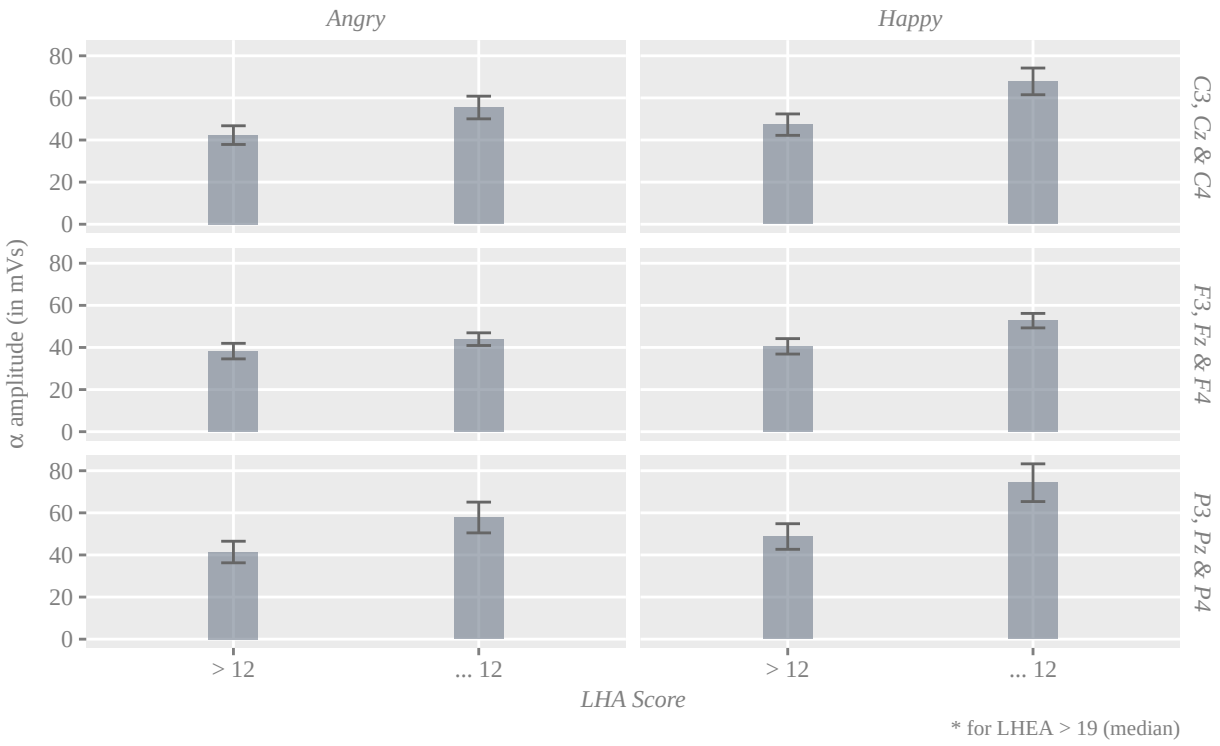

```
alpha_theta %>%
  filter(Task == "N" & Amplitude_Type == "Alpha") %>%
  mutate(Expression = ifelse(Expression == "N", "Neutral", "Happy")) %>%
  mutate(Anterior_Posterior = ifelse(Anterior_Posterior == "C", "C3, Cz & C4",
                                     ifelse(Anterior_Posterior == "F", "F3, Fz & F4", "P3, Pz & P4"))) %
  na.omit() %>%
ggplot(aes(LHA, Amplitude, color = Expression)) +
```

```
  geom_point(alpha = 0.5) +
  geom_smooth(se = FALSE) +
  facet_wrap(~Anterior_Posterior) +
  labs(title = "N  E  U  T  R  A  L  v.  H  A  P  P  Y",
       y = expression(paste(alpha, " amplitude (in mVs)"))) +
  theme(plot.title = element_text(hjust = 0.5, size = 17, face = "italic",
                                  family = "Times", color = "gray50"),
        axis.title.x = element_text(hjust = 0.5, size = 9, face = "italic",
                                    family = "Times", color = "gray50"),
        axis.title.y = element_text(hjust = 0.5, size = 9, face = "italic",
                                    family = "Times", color = "gray50"),
        plot.caption = element_text(size = 8, color = "gray50", family = "Times"),
        panel.grid.minor = element_blank(),
        strip.background = element_blank(),
        strip.text = element_text(family = "Times", color = "gray50", face = "italic"),
        axis.text.x = element_text(family = "Times", color = "gray50"),
        axis.text.y = element_text(family = "Times", color = "gray50"),
        axis.ticks = element_line(color = "gray50"),
        legend.position = "bottom",
        legend.title = element_text(family = "Times", color = "gray50", size = 9),
        legend.text = element_text(family = "Times", color = "gray50", size = 9, face = "italic"))
```



Comparing Group Responses Using Bar Charts

```
library(plyr)
bar_chart_ang <- alpha_theta %>%
  filter(Task == "A") %>%
  filter(Amplitude_Type == "Alpha") %>%
  mutate(Expression = ifelse(Expression == "A", "Angry", "Happy")) %>%
  mutate(Anterior_Posterior = ifelse(Anterior_Posterior == "C", "C3, Cz & C4",
                                     ifelse(Anterior_Posterior == "F", "F3, Fz & F4", "P3, Pz & P4"))) %
  mutate(high_low_LHA = ifelse(LHA <= 12, "\u2264 12", "> 12")) %>%
  filter(LHEA > 19) %>%
  na.omit %>%
  unite(Task_Exp_Axis_Group, Task, Expression, Anterior_Posterior, high_low_LHA)

avg_sd_ang <- plyr::ddply(bar_chart_ang, .(Task_Exp_Axis_Group), summarize, mean_amp = mean(Amplitude),

bars_ang <- avg_sd_ang %>%
  separate(Task_Exp_Axis_Group, into = c("Task", "Expression", "Anterior_Posterior", "high_low_LHA"), s
  mutate(se_amp = sd_amp/sqrt(49)) %>%
  mutate(ymax = mean_amp + se_amp,
         ymin = mean_amp - se_amp)


ggplot(bars_ang, aes(high_low_LHA, mean_amp)) +
  geom_bar(stat = "identity", fill = "lightsteelblue4", alpha = 0.6, width = 0.2) +
  geom_errorbar(aes(ymin = mean_amp - se_amp, ymax = mean_amp + se_amp), width = 0.1, color = "gray40")
  facet_grid(Anterior_Posterior ~ Expression) +
  scale_color_discrete(breaks = c("\u2264 12","> 12")) +
  labs(subtitle = "for individuals with high experienced aggression, but differing committed aggression*
       title = "Alpha-Amplitude following Expressions in the Angry Task",
       caption = "* for LHEA > 19 (median)",
       x = "LHA Score",
       y = expression(paste(alpha, " amplitude (in mVs)")),
       color = "LHA") +
  theme(plot.title = element_text(hjust = 0.5, size = 15, face = "italic",
                                  family = "Times", color = "gray50"),
        plot.subtitle = element_text(hjust = 0.5, size = 10, face = "italic",
                                  family = "Times", color = "gray50"),
        axis.title.x = element_text(hjust = 0.5, size = 9, face = "italic",
                                    family = "Times", color = "gray50"),
        axis.title.y = element_text(hjust = 0.5, size = 9, face = "italic",
                                    family = "Times", color = "gray50"),
        plot.caption = element_text(size = 8, color = "gray50", family = "Times"),
        panel.grid.minor = element_blank(),
        axis.text.x = element_text(family = "Times", color = "gray50"),
        axis.text.y = element_text(family = "Times", color = "gray50"),
        axis.ticks = element_line(color = "gray50"),
        legend.position = "bottom",
        legend.title = element_text(family = "Times", color = "gray50", size = 9),
        legend.text = element_text(family = "Times", color = "gray50", size = 9, face = "italic"),
        strip.background = element_blank(),
        strip.text = element_text(family = "Times", color = "gray50", face = "italic"))
```




```
bar_chart_neu <- alpha_theta %>%
  filter(Task == "N") %>%
  filter(Amplitude_Type == "Alpha") %>%
  mutate(Expression = ifelse(Expression == "N", "Neutral", "Happy")) %>%
  mutate(Anterior_Posterior = ifelse(Anterior_Posterior == "C", "C3, Cz & C4",
                                     ifelse(Anterior_Posterior == "F", "F3, Fz & F4", "P3, Pz & P4"))) %
  mutate(high_low_LHA = ifelse(LHA <= 12, "\u2264 12", "> 12")) %>%
  filter(LHEA > 19) %>%
  na.omit %>%
  unite(Task_Exp_Axis_Group, Task, Expression, Anterior_Posterior, high_low_LHA)

avg_sd_neu <- plyr::ddply(bar_chart_neu, .(Task_Exp_Axis_Group), summarize, mean_amp = mean(Amplitude),

bars_neu <- avg_sd_neu %>%
  separate(Task_Exp_Axis_Group, into = c("Task", "Expression", "Anterior_Posterior", "high_low_LHA"), s
  mutate(se_amp = sd_amp/sqrt(49)) %>%
  mutate(ymax = mean_amp + se_amp,
         ymin = mean_amp - se_amp)

ggplot(bars_neu, aes(high_low_LHA, mean_amp)) +
  geom_bar(stat = "identity", fill = "lightsteelblue4", alpha = 0.7, width = 0.2) +
  facet_grid(Anterior_Posterior ~ Expression) +
  scale_color_discrete(breaks = c("\u2264 12","> 12")) +
   geom_errorbar(aes(ymin = mean_amp - se_amp, ymax = mean_amp + se_amp), width = 0.1, color = "gray40"
  labs(subtitle = "for individuals with high experienced aggression, but differing committed aggression
       title = "Alpha-Amplitude following Expressions in the Neutral Task",
       caption = "* for LHEA > 19 (median)",
```

```
      x = "LHA score",
      y = expression(paste(alpha, " amplitude (in mVs)")),
      color = "LHA") +
 theme(plot.title = element_text(hjust = 0.5, size = 15, face = "italic",
                                 family = "Times", color = "gray50"),
       plot.subtitle = element_text(hjust = 0.5, size = 10, face = "italic",
                                 family = "Times", color = "gray50"),
       axis.title.x = element_text(hjust = 0.5, size = 9, face = "italic",
                                   family = "Times", color = "gray50"),
       axis.title.y = element_text(hjust = 0.5, size = 9, face = "italic",
                                   family = "Times", color = "gray50"),
       plot.caption = element_text(size = 8, color = "gray50", family = "Times"),
       panel.grid.minor = element_blank(),
       axis.text.x = element_text(family = "Times", color = "gray50"),
       axis.text.y = element_text(family = "Times", color = "gray50"),
       axis.ticks = element_line(color = "gray50"),
       legend.position = "bottom",
       legend.title = element_text(family = "Times", color = "gray50", size = 9),
       legend.text = element_text(family = "Times", color = "gray50", size = 9, face = "italic"),
       strip.background = element_blank(),
       strip.text = element_text(family = "Times", color = "gray50", face = "italic"))
```