```r
#TIPI items, to construct OCEAN variables
personalityVariables = c('personality_extroverted', 'personality_critical', 'personality_dependable', 'p

#PANAS items, to construct pre, post, delta variables
preScaleItems = c('pre_Active', 'pre_Afraid', 'pre_Determined', 'pre_Nervous', 'pre_Attentive',
                  'pre_Upset', 'pre_Inspired', 'pre_Hostile', 'pre_Alert', 'pre_Ashamed')
postScaleItems = c('post_Active', 'post_Afraid', 'post_Determined', 'post_Nervous', 'post_Attentive', 'p
pre_post_scale_item_column_names = c('id', 'prePositiveAffect', 'preNegativeAffect', 'postPositiveAffec
```

```r
# Co-function. Remove non alphanumerics from string, save comma and minus
parse_string <- function(x) {
  return (gsub("[^[:alnum:][:blank:],/\\-]", "", x))
}

create_personality_vars <- function(df) {
  ###Apply parse_string###
  df$personalityResponses <- lapply(df$responses, parse_string)

  df[personalityVariables] <- as.numeric(str_split_fixed(df$personalityResponses, ',', 10))

  df$openness <- (df$personality_open + (4 - df$personality_conventional)) / 2

  df$conscientiousness <- (df$personality_dependable + (4 - df$personality_disorganized)) / 2

  df$extroversion <- (df$personality_extroverted + (4 - df$personality_reserved)) / 2

  df$agreeableness <- (df$personality_sympathetic + (4 - df$personality_critical)) / 2

  df$emot_stability <- (df$personality_calm + (4 - df$personality_anxious)) / 2

  return (df)

}
```

```r
create_pre_post_scale_items <- function(df) {
  #apply parse_string
  df$preSurveyResponses <- lapply(df$preSurveyResponses, parse_string)
  df$postSurveyResponses <- lapply(df$postSurveyResponses, parse_string)
  # Split SurveyResponses as numerics across new columns
  df[preScaleItems] <- as.numeric(str_split_fixed(df$preSurveyResponses, ',', 10))
  df[postScaleItems] <- as.numeric(str_split_fixed(df$postSurveyResponses, ',', 10))

  df$prePositiveAffect <- (((df$pre_Active) + (df$pre_Determined) + (df$pre_Attentive) + (df$pre_Inspir
  df$preNegativeAffect <- ((df$pre_Afraid + df$pre_Nervous + df$pre_Upset + df$pre_Hostile + df$pre_Asha

  df$postPositiveAffect <- ((df$post_Active + df$post_Determined + df$post_Attentive + df$post_Inspired
  df$postNegativeAffect <- ((df$post_Afraid + df$post_Nervous + df$post_Upset + df$post_Hostile + df$pos

  df$post_pre_PositiveAffect <- df$postPositiveAffect - df$prePositiveAffect
  df$post_pre_NegativeAffect <- df$postNegativeAffect - df$preNegativeAffect

  return (df)
}
```

```r
create_delta_scale_items <- function(df) {
  pre_items <- grep("^pre", names(df), value = TRUE)
  pre_items <- pre_items[! pre_items %in% 'preSurveyResponses']
  post_items <- grep("^post", names(df), value = TRUE)
  post_items <- post_items[! post_items %in% c('postSurveyResponses', 'post_pre_PositiveAffect', 'post_p
  # Iterating over each item pair and creating a new column for the difference
  for (i in seq_along(pre_items)) {
    item <- sub("^pre", "", pre_items[i])  # Extracting the item name
    # tmp[paste0(item, "_diff")] <- df_hostile_agent[post_items[i]] - df_hostile_agent[pre_items[i]]

    if (is.numeric(df[[pre_items[i]]]) && is.numeric(df[[post_items[i]]])) {
      df[paste0("post_pre", item)] <- df[[post_items[i]]] - df[[pre_items[i]]]
    } else {
      warning(paste("Columns", pre_items[i], "and", post_items[i], "contain non-numeric data. Skipping s
    }


  }
  return (df)
}
```

```r
# co-function
generate_histogram <- function(data, col, binwidth, y_limit) {
  ggplot(data, aes_string(x = col)) +
    geom_histogram(fill = "skyblue", color = "black", binwidth = binwidth) +
    xlab(col) +
    ylim(y_limit)
}
# co-function
apply_limits <- function(flag) {
  limits <- list(
    pre_post = list(y_limit = c(0, 25)),
    overall_pre_post = list(y_limit = c(0, 70)),
    overall_personality = list(y_limit = c(0, 65)),
    post_pre_differences = list(x_limit = c(-5, 5), y_limit = c(0, 30)),
    docile_post_pre_differences = list(x_limit = c(-5, 5), y_limit = c(0, 50)),
    overall_post_pre_differences = list(x_limit = c(-5, 5), y_limit = c(0, 85)),
    personality = list(y_limit = c(0, 25))
  )
  return(limits[[flag]])
}

gridOfHistograms <- function(data, plot_name, flag = "None", title_flag = FALSE) {
  list <- lapply(1:ncol(data), function(col) {
    if (flag != "None") {
      limits <- apply_limits(flag)
      generate_histogram(data, colnames(data)[col], 1, limits$y_limit)
    } else {
      generate_histogram(data, colnames(data)[col], 1, c(0, 10))
    }
  })

  if (title_flag) {
```

```r
    title <- ggdraw() + draw_label(plot_name, fontface = 'bold')
    grid_plot <- cowplot::plot_grid(plotlist = list)
    grid_plot <- ggdraw(grid_plot) +
      ggtitle(plot_name) +
      theme(plot.title = element_text(hjust = 0.5))
    print(grid_plot)
    ggsave(paste0(plot_name, ".png"), grid_plot)

  } else {
    grid_plot <- cowplot::plot_grid(plotlist = list)
    print(grid_plot)
    ggsave(paste0(plot_name, ".png"), grid_plot)
  }

  return(grid_plot)
}
```

```r
generate_boxplot <- function(data, col, x_var, num_rows, angle) {
  ggplot(data = data, aes_string(x = x_var, y = col)) +
    geom_boxplot(color = "black", fill = "blue", alpha = 0.2, outlier.shape = NA) +
    ylab(col) + xlab(NULL) +
    stat_summary(fun = mean, colour = "black", geom = "point", shape = 18, size = 3, show.legend = FALSE
    theme(axis.text.x = element_text(angle = angle, hjust = 1, size = 12), axis.title.y = element_text(
}

gridOfBoxplots <- function(data, plot_name, plot_name_flag = FALSE, behaviors_flag = FALSE, by_gender =

  if (!behaviors_flag) {
    if (!by_gender) {
      if (!by_familiarity) {
        num_columns <- 4
        num_rows <- 4
        x_var <- "combined_group"
        angle <- 45
      } else {
        x_var <- "GenderXFamiliarity"
        angle <- 25
      }
    } else {
      x_var <- "gender"
      angle <- 25
    }
  } else {
    if (!by_gender) {
      if (!by_familiarity) {
        num_columns <- 4
        num_rows <- 2
        x_var <- "combined_group"
        angle <- 45
      } else {
        x_var <- "GenderXFamiliarity"
        angle <- 25
      }
```

```
    } else {
      x_var <- "gender"
      angle <- 25
    }
  }

  if (!behaviors_flag) {
    list <- lapply(59:70, function(col) {
      generate_boxplot(data, colnames(data)[col], x_var, num_rows, angle)
    })
  } else {
    list <- lapply(9:15, function(col) {
      generate_boxplot(data, colnames(data)[col], x_var, num_rows, angle)
    })
  }

  if (plot_name_flag) {
    title <- ggdraw() + draw_label(plot_name, fontface = 'bold')
    grid_plot <- cowplot::plot_grid(plotlist = list)
    grid_plot <- ggdraw(grid_plot) +
      ggtitle(plot_name) +
      theme(plot.title = element_text(hjust = 0.5))
    print(grid_plot)
  } else {
    print("!plot_name_flag")
    grid_plot <- cowplot::plot_grid(plotlist = list)
    print(grid_plot)
  }

  return(grid_plot)
}
```

```
boxPlots_func = function(items_df, flag, plot_name){

  data.melt<-reshape2::melt(items_df, id="id")

  data.sum<-ddply(data.melt, .(variable), summarise,
                  mean = mean(value),
                  sd = sd(value),
                  min = min(value),
                  max = max(value))

  if (flag == 0){
    subscale_grouping <- c("PreSurvey","PreSurvey","PreSurvey","PreSurvey","PreSurvey","PreSurvey","PreS
  }
  if (flag == 1) {
    subscale_grouping <- c("PreSurvey","PreSurvey", "PostSurvey", "PostSurvey")

  }

  data.sum$Subscale <- subscale_grouping
  data.sum$Subscale <- as.factor(data.sum$Subscale)
  meanBoxPlots <- ggplot(data.sum, aes(x=variable)) +
```

```r
    geom_boxplot(aes(ymin =min, lower = mean-sd, middle = mean, upper = mean+sd, ymax =max, fill = Subs
    geom_text(aes(label = round(mean, 3), y = mean + 0.2), size = 3, position = position_dodge(width = 
    labs(title = plot_name) +
    labs(y = "item score") +
    theme(axis.text.x = element_text(angle=75, hjust=1))
  ggsave(paste0(plot_name,".png"), meanBoxPlots)
  plot(meanBoxPlots)
  return (meanBoxPlots)
}
```

```r
run_anova <- function(outcome_var, factor_1, factor_2=FALSE, interaction=FALSE, remove_outliers=FALSE){
  ##assumption 1 - normality of residuals
  # - check sample size per group, if all >= 30, continue
  #                               else, run Shapiro-Wilk Test on ANOVA residuals
  # run levene test for equality of variance
  if (factor_2) {

  model_df <- mobOutcomes[c(outcome_var, factor_1, factor_2, "id")]

  df_outliers <- model_df %>%
    group_by(factor_1, factor_2) %>%
    identify_outliers(outcome_var)

  print(sprintf("Running %s model with %d outliers", outcome_var, count_extreme_outliers))
  count_extreme_outliers <- sum(df_outliers.is.extreme)
  #Run model both with and without outlier removal, if outliers present
  aov_model <- aov(outcome_var ~ factor_1 + factor_2 + factor1:factor2,
                   data = model_df)
  #At least one value of the 'is.extreme' column is TRUE
  if (count_extreme_outliers > 0) {

    model_df <- model_df %>%
      anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")
    aov_model <- aov(outcome_var ~ factor_1 + factor_2 + factor1:factor2,
                     data = model_df)
    aov_model_no_extreme_outliers <- aov(outcome_var ~ factor_1 + factor_2 + factor1:factor2,
                     data = model_df)
  }

  model_df <- model_df %>%
    anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

  aov_model <- aov(outcome_var ~ factor_1 + factor_2 + factor1:factor2,
                     data = model_df)

  # human_agent                  1   1.26  1.2622   1.047  0.309
  # docile_hostile               1   0.76  0.7569   0.628  0.430
  # human_agent:docile_hostile   1   0.36  0.3567   0.296  0.588
  summary(aov_model)

  }
}
```

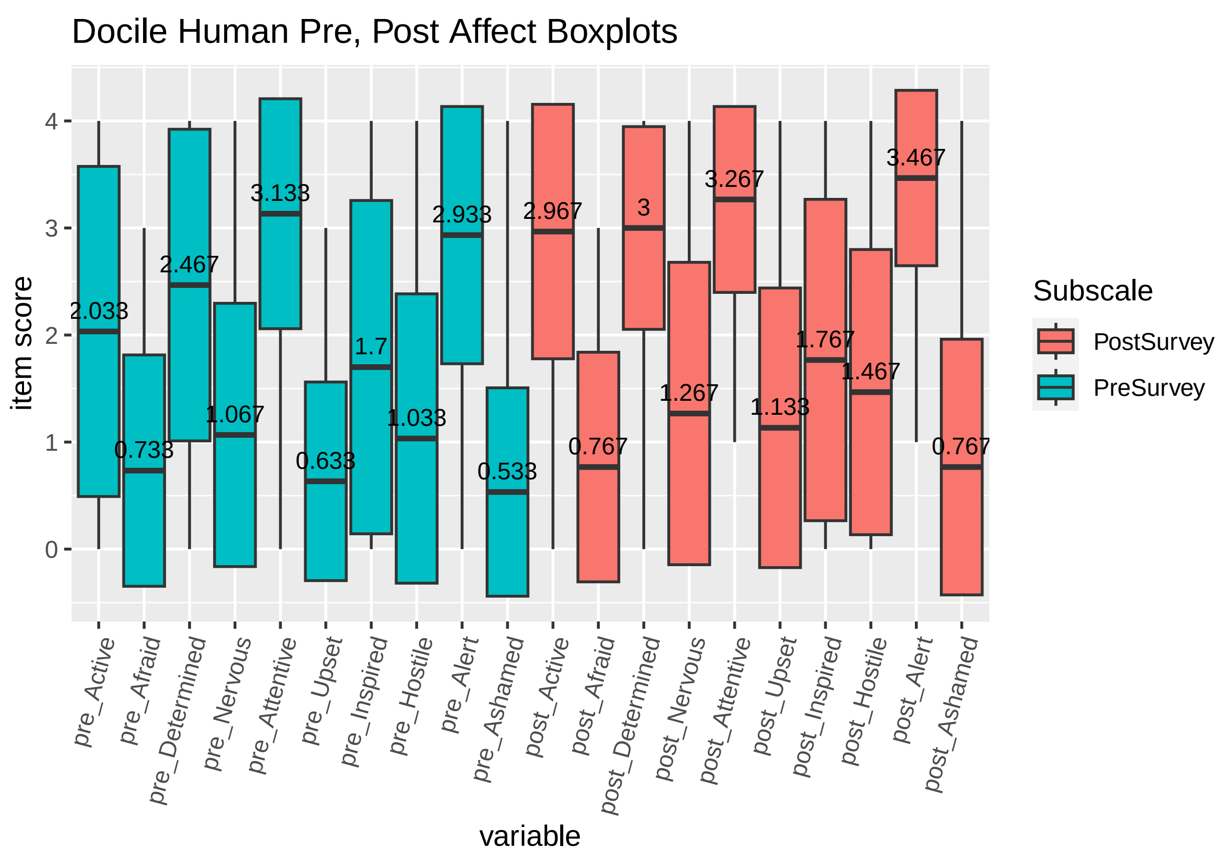
####dociileHuman#### ####################

```r
df_docile_human <-read.csv(file="docileHuman.csv", stringsAsFactors=FALSE, header=T)
df_docile_human <-as.data.frame(df_docile_human)
#30x16
dim(df_docile_human)
```

```
## [1] 30 16
```

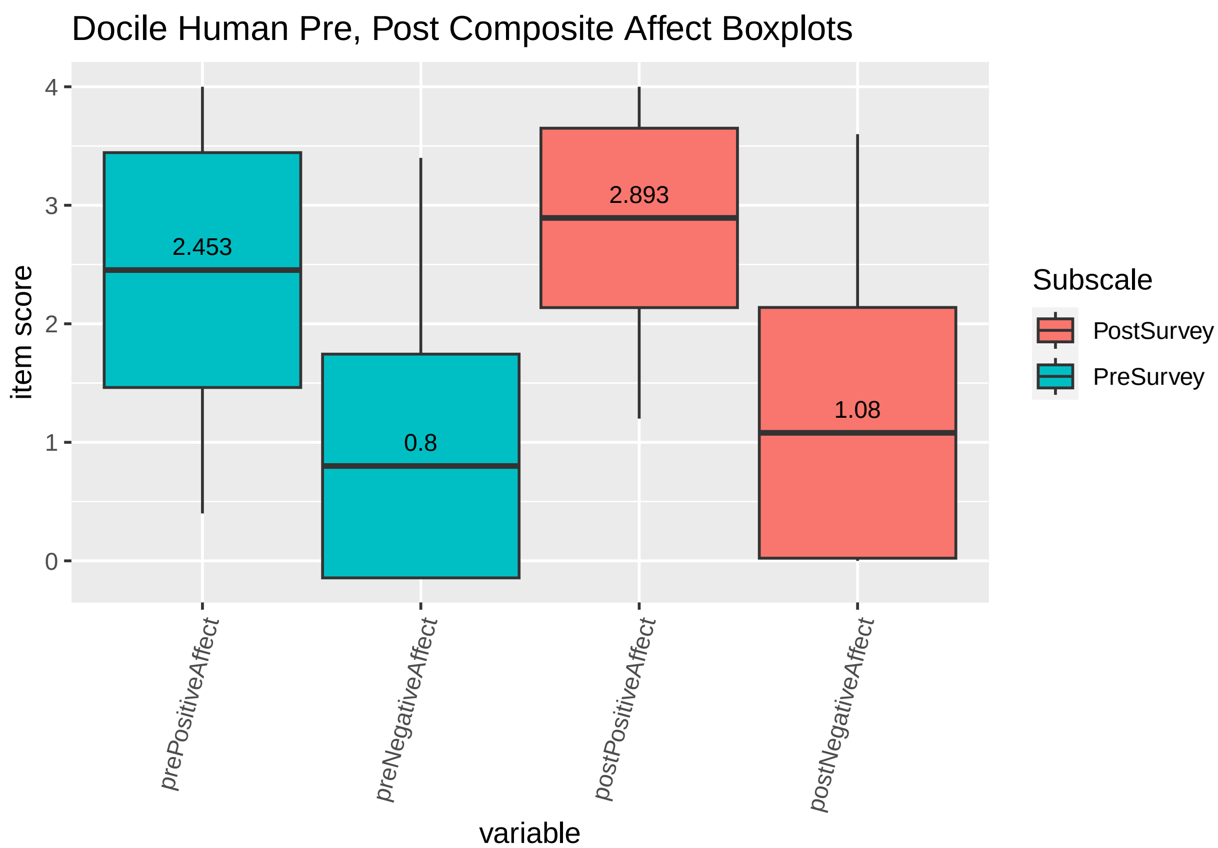
```r
df_docile_human <- create_personality_vars(df_docile_human)
df_docile_human <- create_pre_post_scale_items(df_docile_human)
df_docile_human <- create_delta_scale_items(df_docile_human)
#30x70

pre_post_scale_items <- df_docile_human[c('id', preScaleItems, postScaleItems)]
pre_post_box_plots <- boxPlots_func(pre_post_scale_items, 0, "Docile Human Pre, Post Affect Boxplots")
```

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




```r
names.use <- names(df_docile_human) %in% pre_post_scale_item_column_names
docile_human_pre_post <- df_docile_human[, names.use]
pre_post_box_plots <- boxPlots_func(docile_human_pre_post, 1, "Docile Human Pre, Post Composite Affect B
```

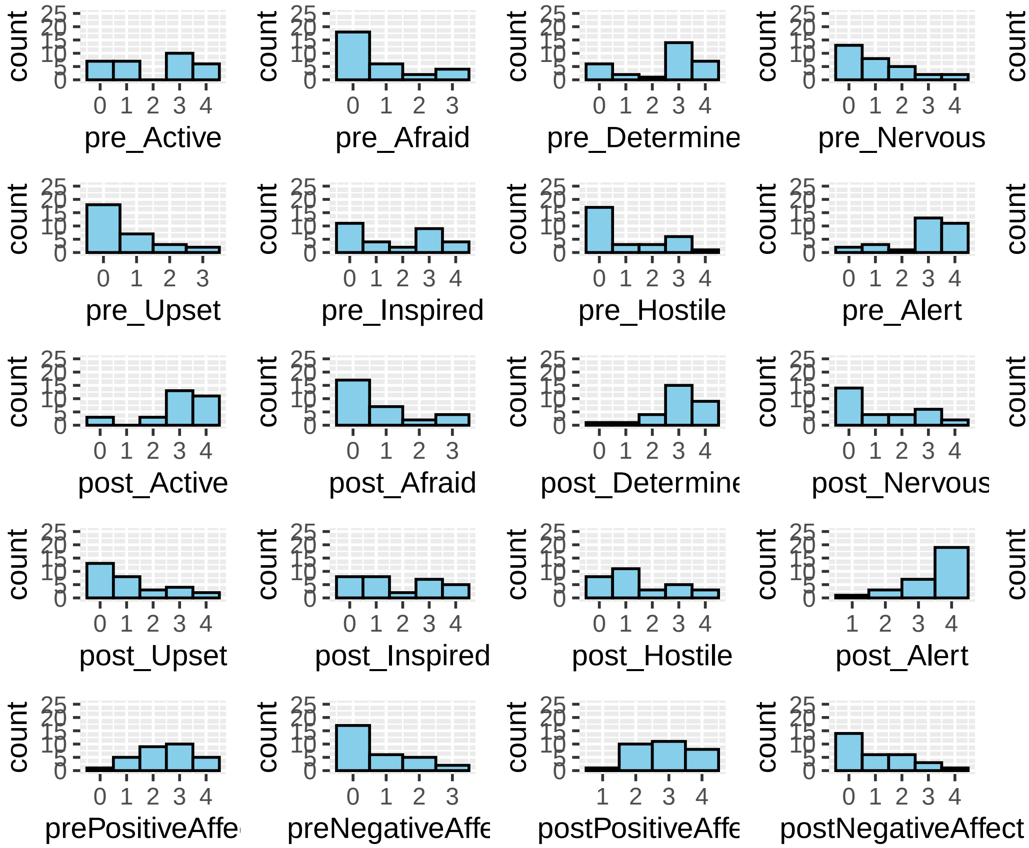
```
## Saving 6.5 x 4.5 in image
```




```
pre_post_scale_items <- df_docile_human[c(preScaleItems, postScaleItems, 'prePositiveAffect', 'preNegati
plot_docile_human_pre_post <- gridOfHistograms(pre_post_scale_items, "Docile Human Pre, Post Affect His
```

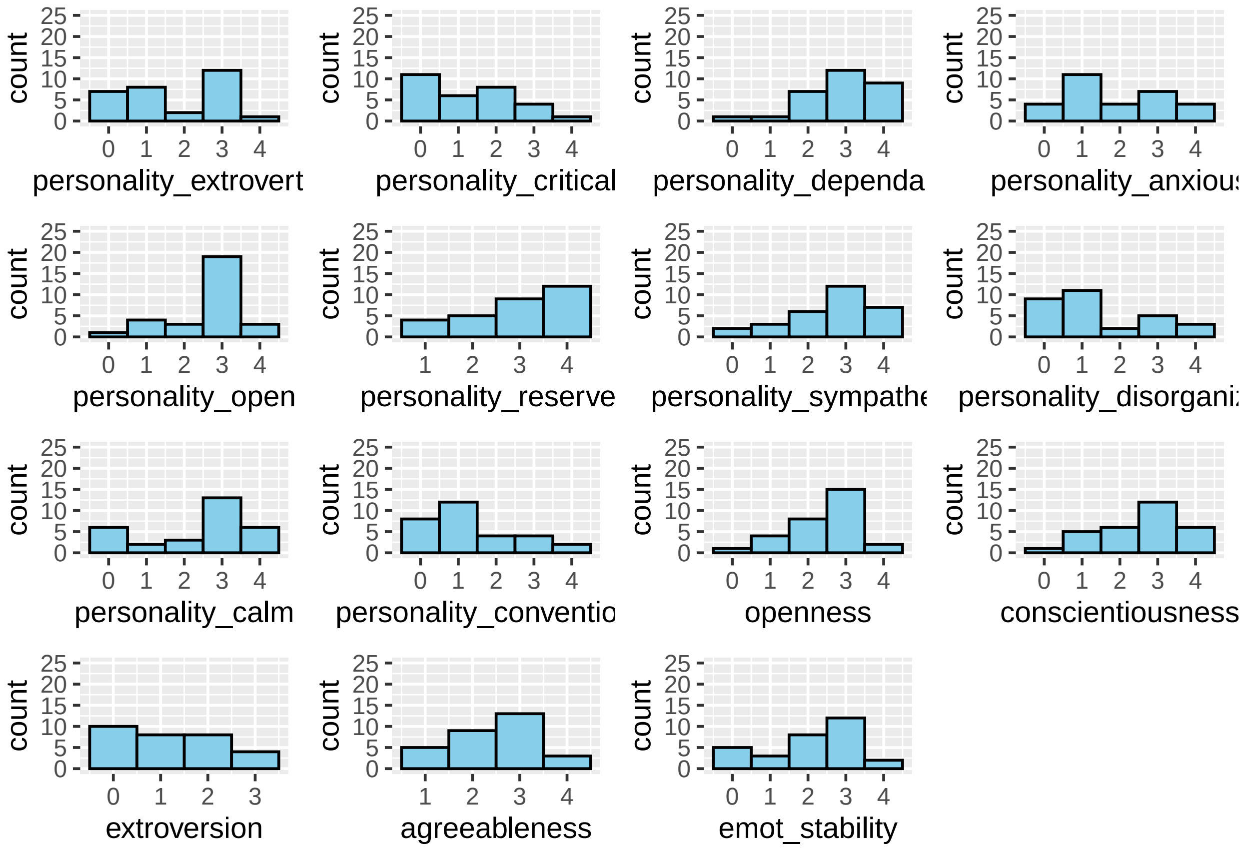
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation ideoms with `aes()`
```

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




```
df_docile_human_personality <- df_docile_human[, 18:32]
plot_docile_human_personality <- gridOfHistograms(df_docile_human_personality, "Docile Human Personality
```

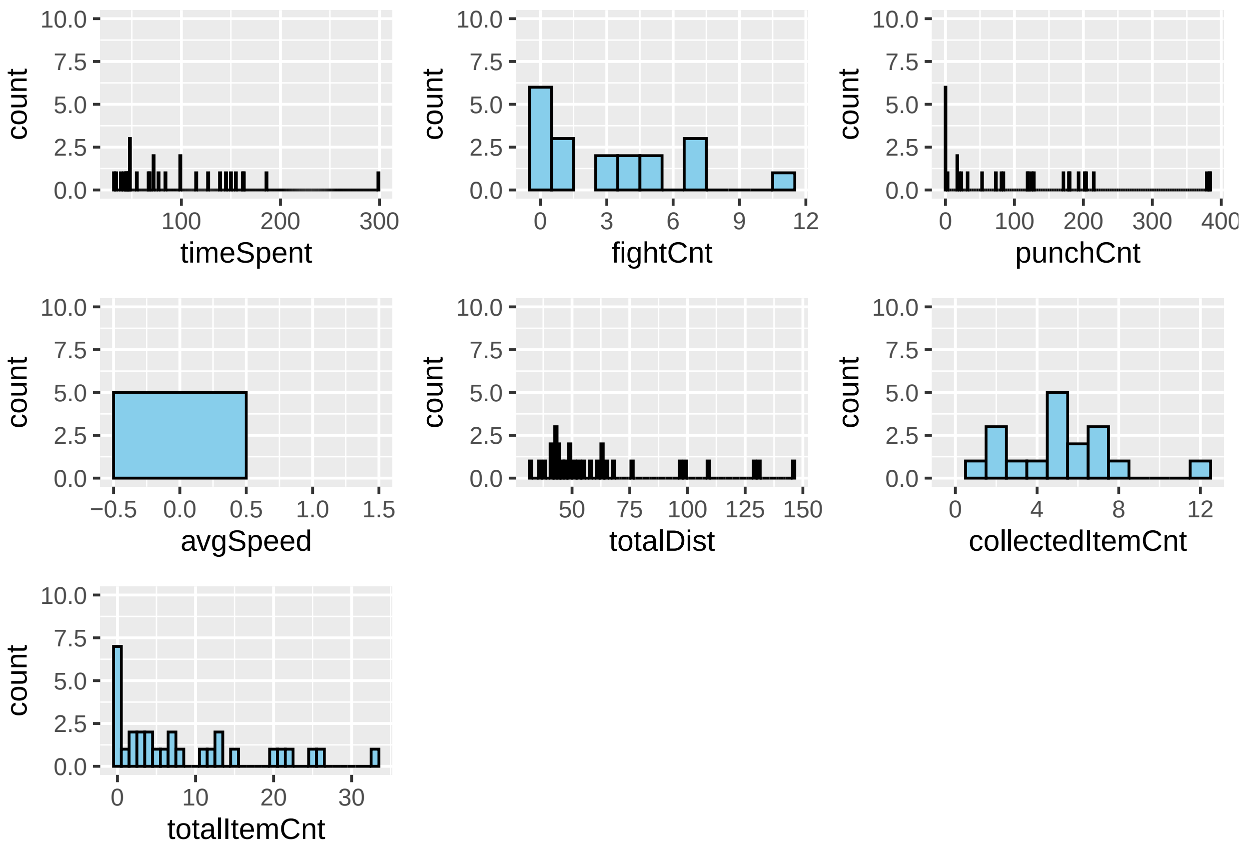
```
## Saving 6.5 x 4.5 in image
```




```r
df_docile_human_behaviors <- df_docile_human[, 9:15]
plot_docile_human_behaviors <- gridOfHistograms(df_docile_human_behaviors, "Docile Human Behaviors Hist
```

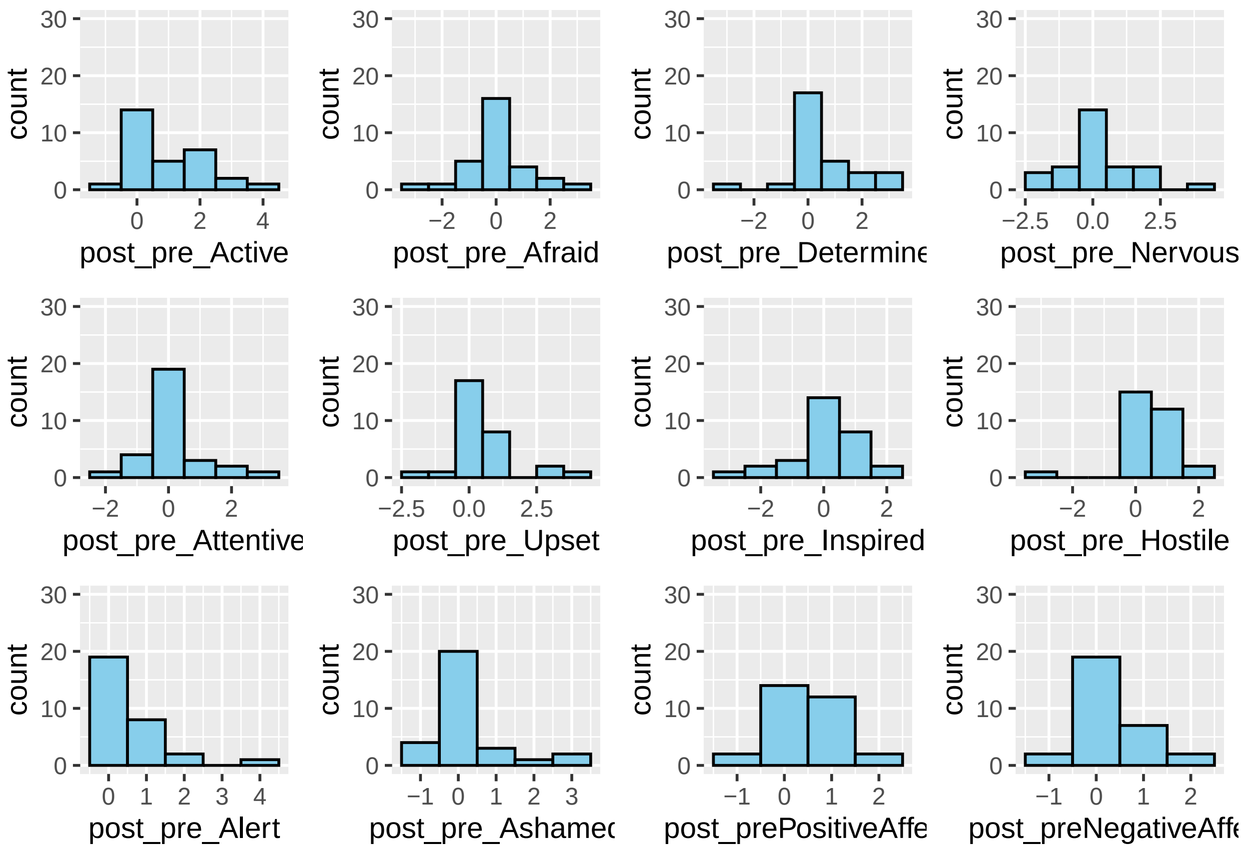
```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
```

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




```
df_docile_human_deltas <- df_docile_human[, 59:70]
plot_docile_human_deltas <- gridOfHistograms(df_docile_human_deltas, "Docile Human Post to Pre Differen
```

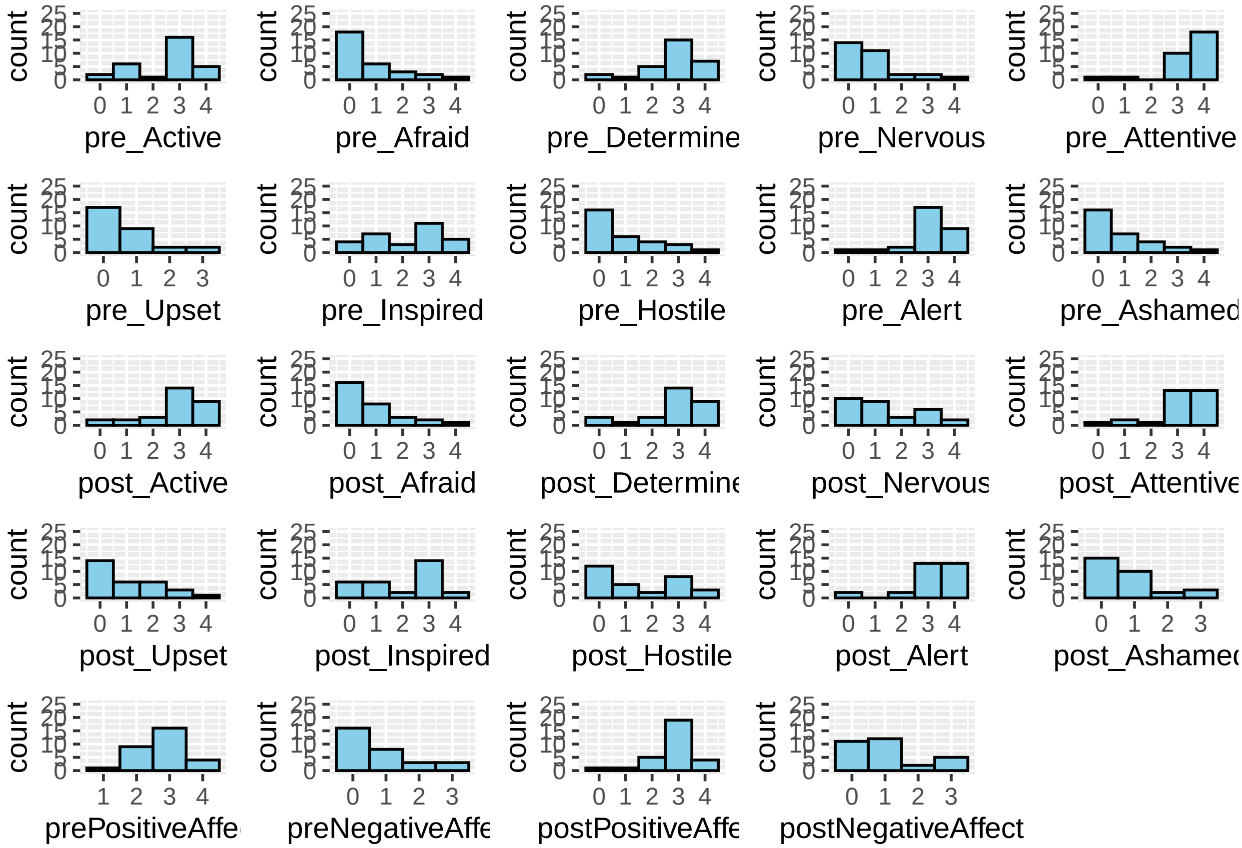
```
## Saving 6.5 x 4.5 in image
```



####docileAgent#### ##################

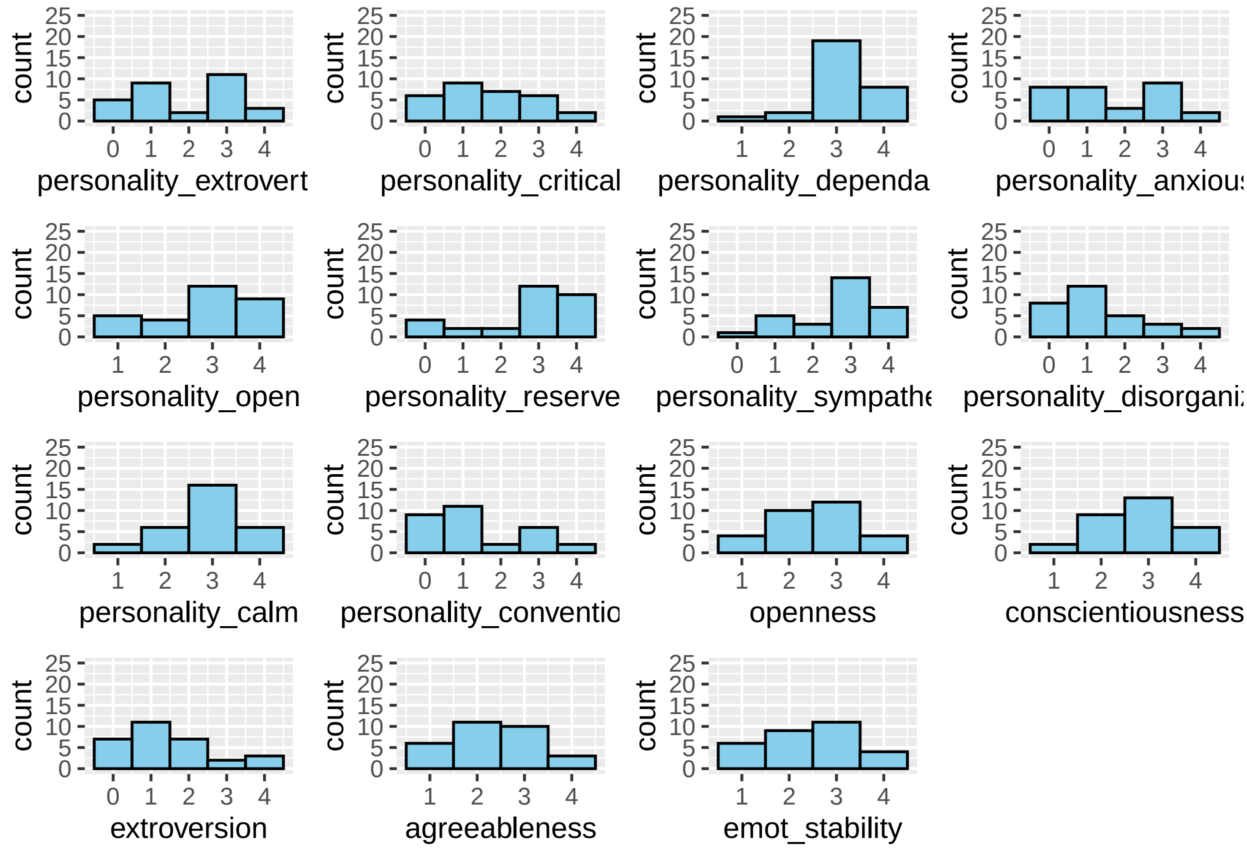
```
df_docile_agent <- read.csv(file="docileAgent.csv", stringsAsFactors=FALSE, header=T)
df_docile_agent <-as.data.frame(df_docile_agent)

df_docile_agent <- create_personality_vars(df_docile_agent)
df_docile_agent <- create_pre_post_scale_items(df_docile_agent)
df_docile_agent <- create_delta_scale_items(df_docile_agent)

pre_post_scale_items <- df_docile_agent[c(preScaleItems, postScaleItems, 'prePositiveAffect', 'preNegati
plot_docile_agent_pre_post <- gridOfHistograms(pre_post_scale_items, "Docile Agent Pre, Post Affect His
```

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




```
df_docile_agent_personality <- df_docile_agent[, 18:32]
plot_docile_human_personality <- gridOfHistograms(df_docile_agent_personality, "Docile Agent Personality
```

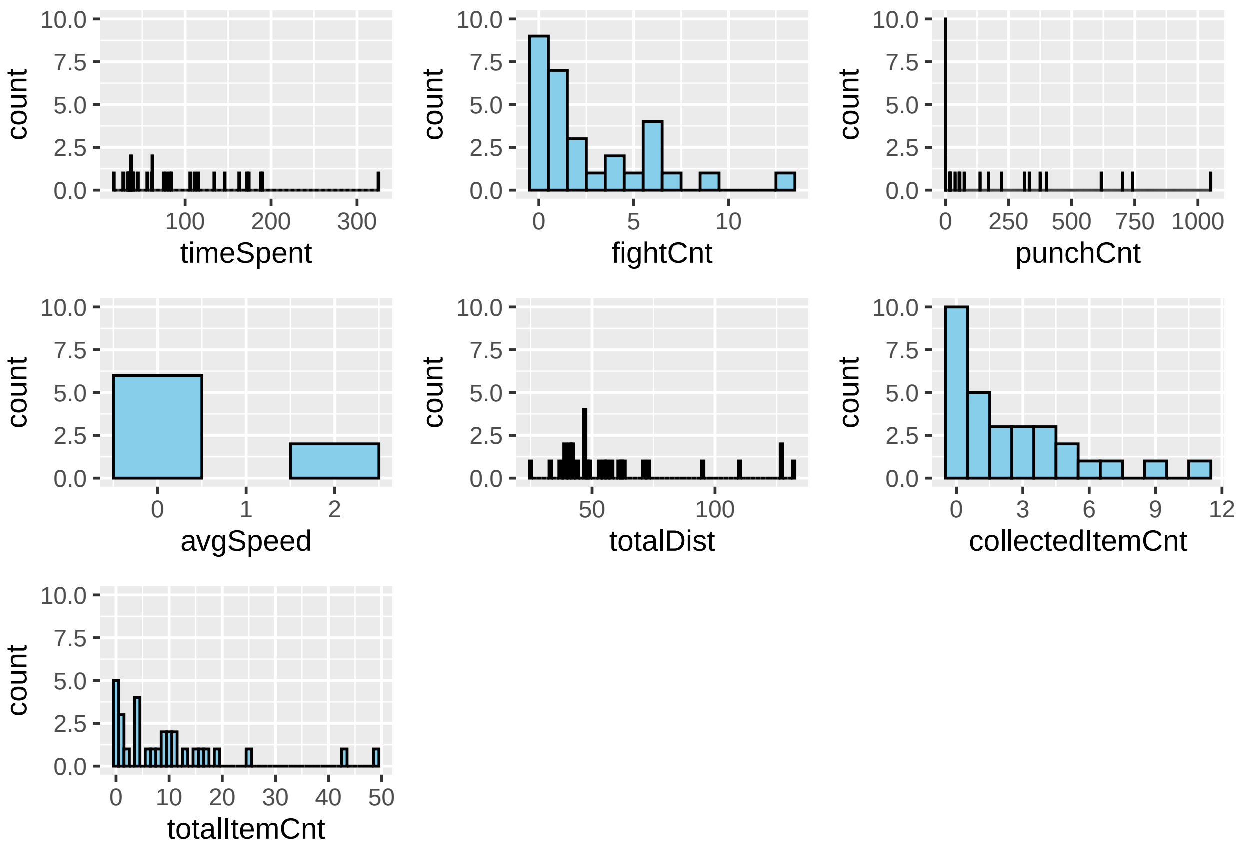
```
## Saving 6.5 x 4.5 in image
```




```
df_docile_agent_behaviors <- df_docile_agent[, 9:15]
plot_docile_agent_behaviors <- gridOfHistograms(df_docile_agent_behaviors, "Docile Agent Behaviors Hist
```

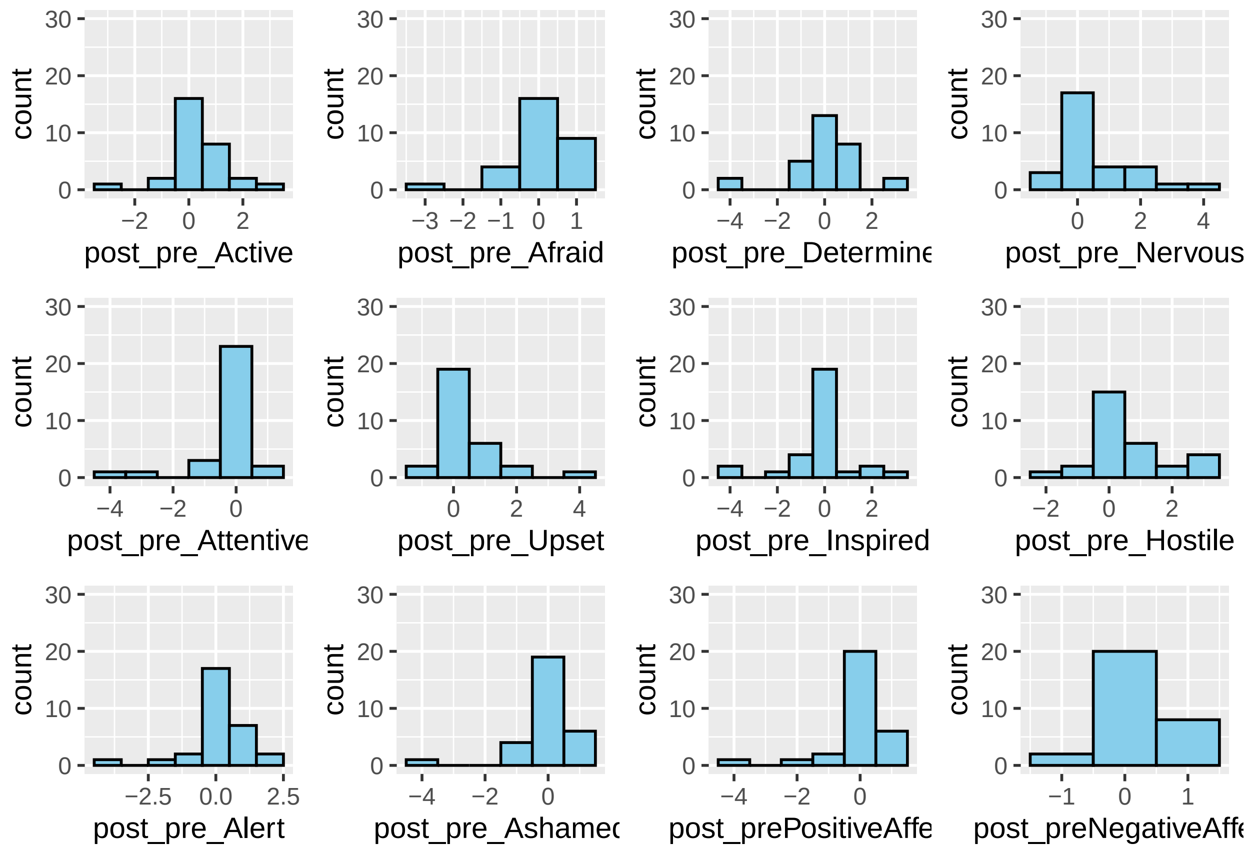
```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
```

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




```
df_docile_agent_deltas <- df_docile_agent[, 59:70]
plot_docile_agent_deltas <- gridOfHistograms(df_docile_agent_deltas, "Docile Agent Post to Pre Differen
```

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

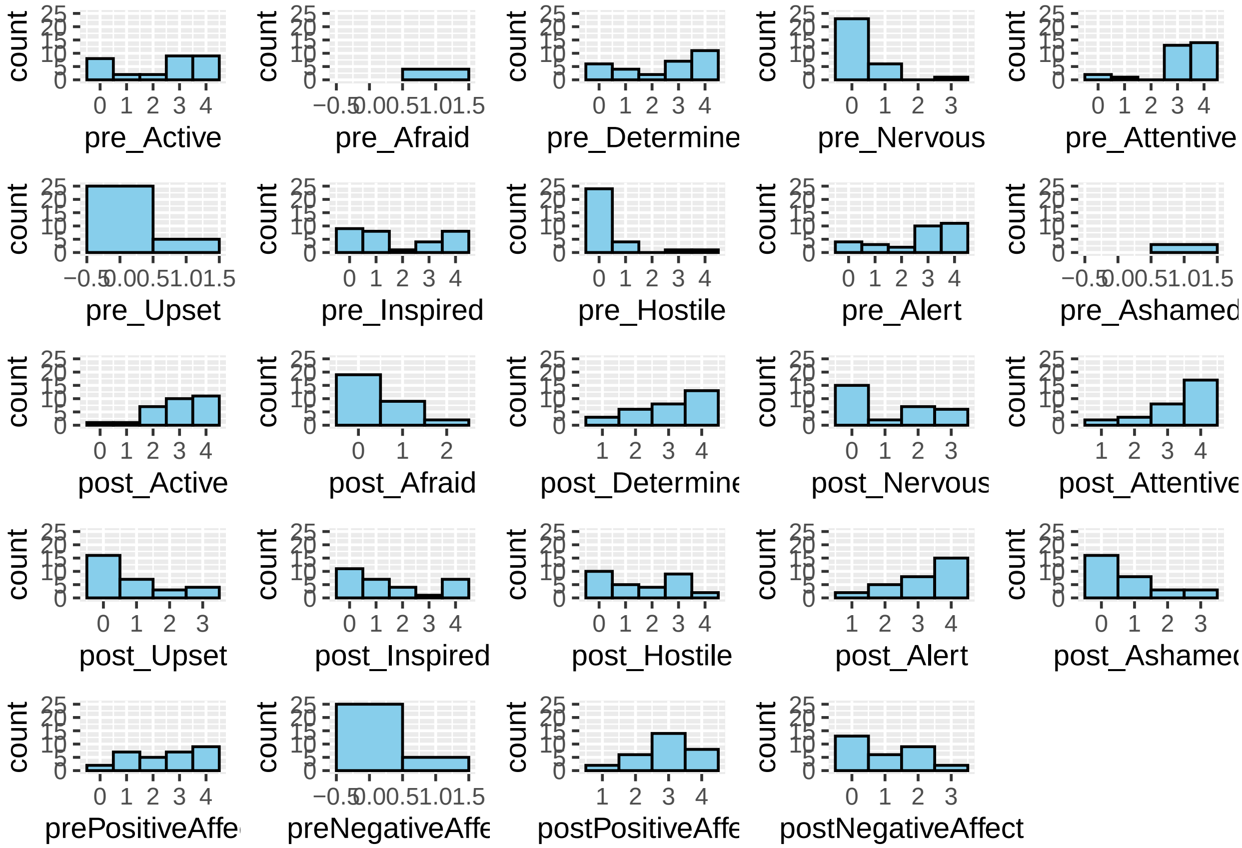


####hostileHuman### ##################

```
df_hostile_human <-read.csv(file="hostileHuman.csv", stringsAsFactors=FALSE, header=T)
df_hostile_human <-as.data.frame(df_hostile_human)
df_hostile_human <- select(df_hostile_human, -c(stolenItemCnt))

df_hostile_human <- create_personality_vars(df_hostile_human)
df_hostile_human <- create_pre_post_scale_items(df_hostile_human)
df_hostile_human <- create_delta_scale_items(df_hostile_human)

pre_post_scale_items <- df_hostile_human[c(preScaleItems, postScaleItems, 'prePositiveAffect', 'preNegat
plot_hostile_human_pre_post <- gridOfHistograms(pre_post_scale_items, "Hostile Human Pre, Post Affect Hi
```

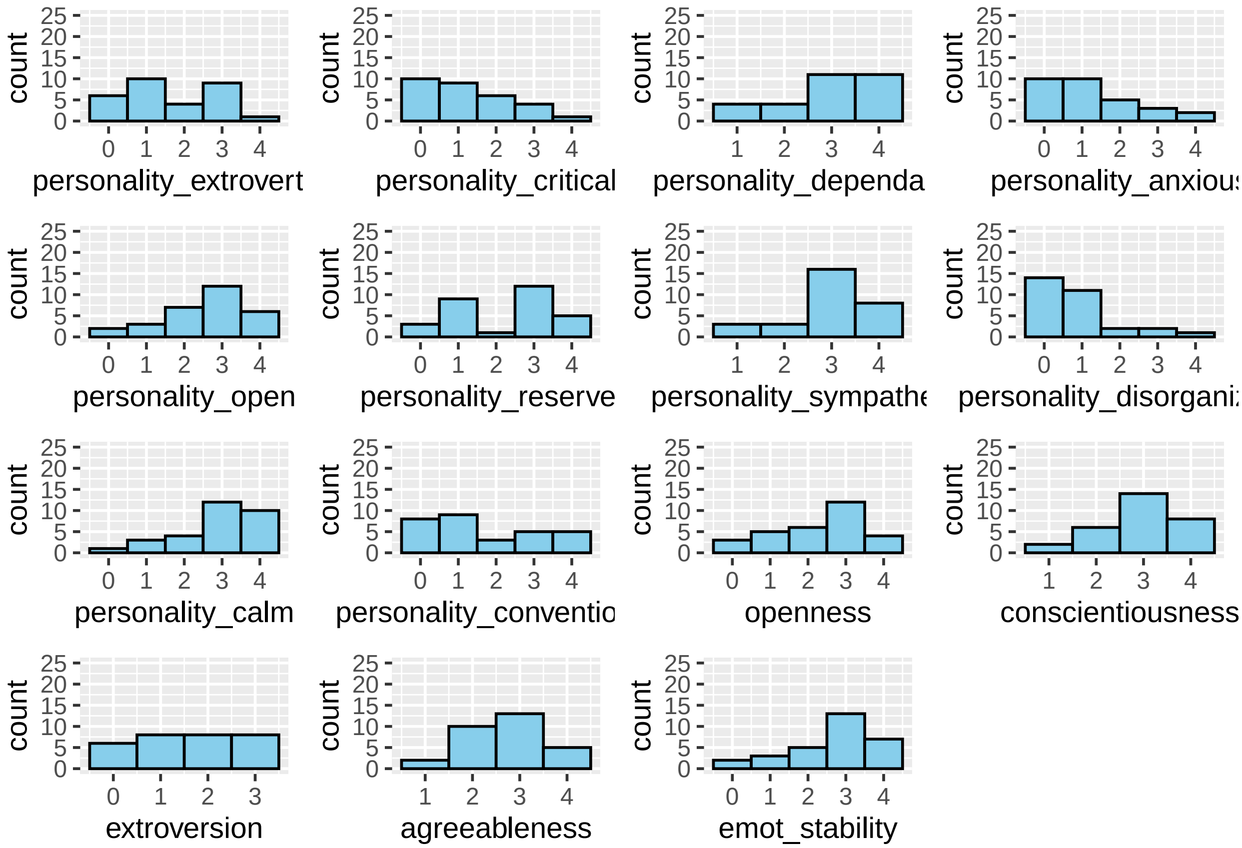
```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
```

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




```
df_hostile_human_personality <- df_hostile_human[, 18:32]
plot_hostile_human_personality <- gridOfHistograms(df_hostile_human_personality, "Hostile Human Personal
```

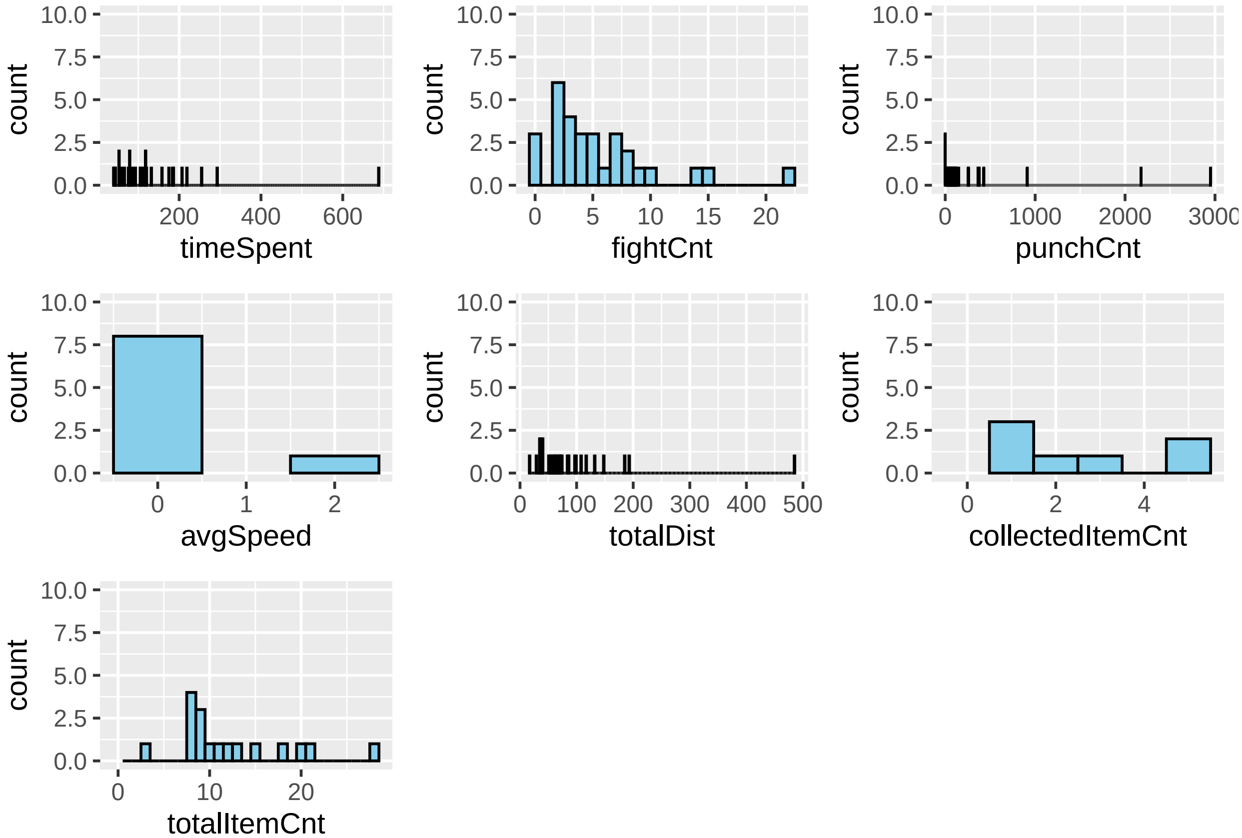
```
## Saving 6.5 x 4.5 in image
```




```
df_hostile_human_behaviors <- df_hostile_human[, 9:15]
plot_hostile_human_behaviors <- gridOfHistograms(df_hostile_human_behaviors, "Hostile Human Behaviors H
```

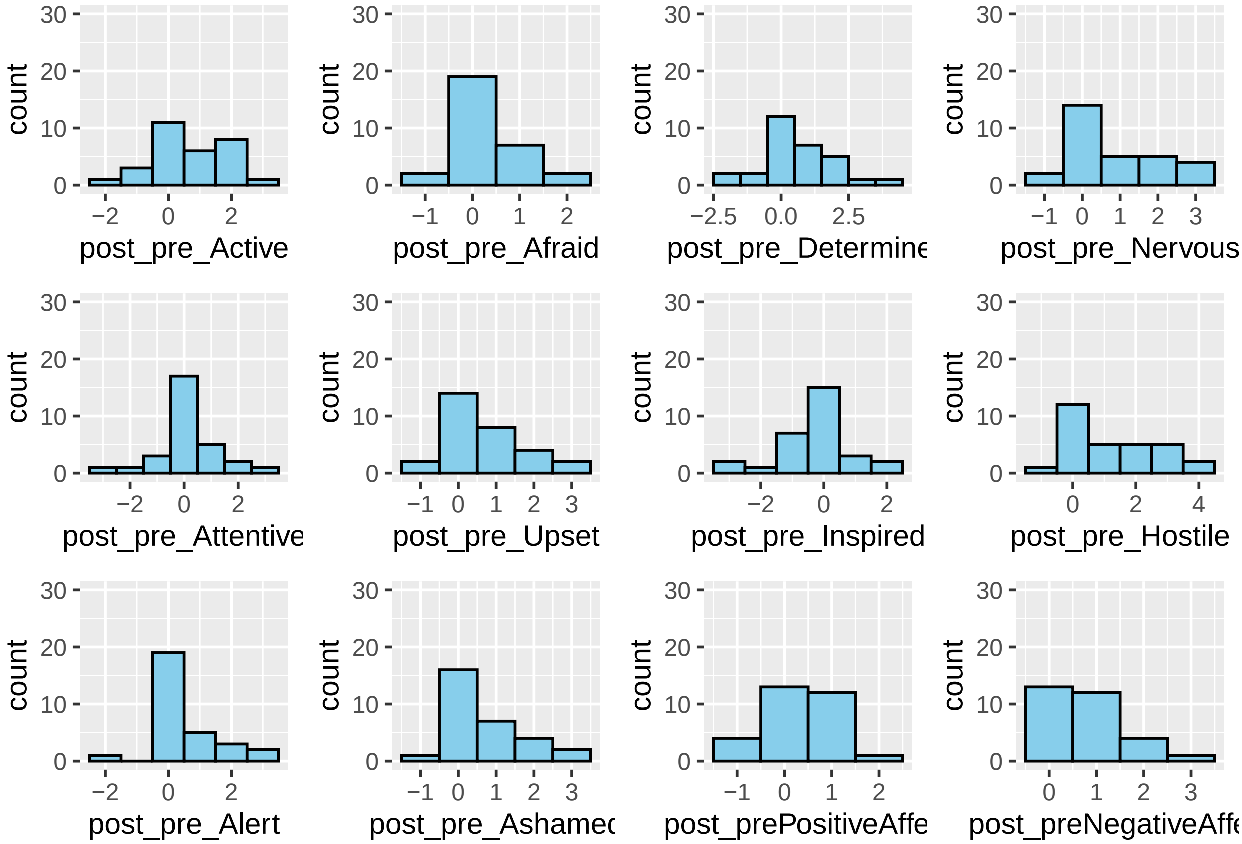
```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
```

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




```
df_hostile_human_deltas <- df_hostile_human[, 59:70]
plot_hostile_human_deltas <- gridOfHistograms(df_hostile_human_deltas, "Hostile Human Post to Pre Diffe
```

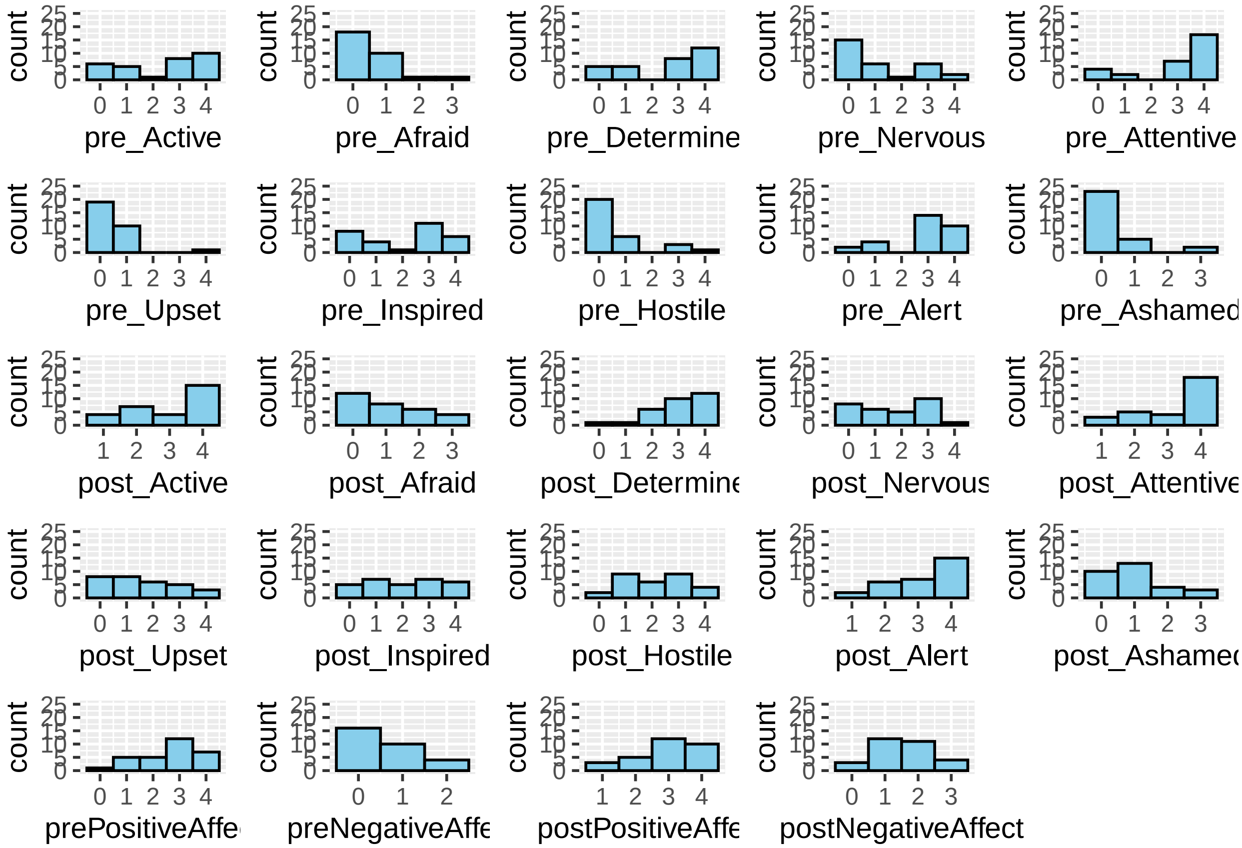
```
## Saving 6.5 x 4.5 in image
```



####hostileAgent### ##################

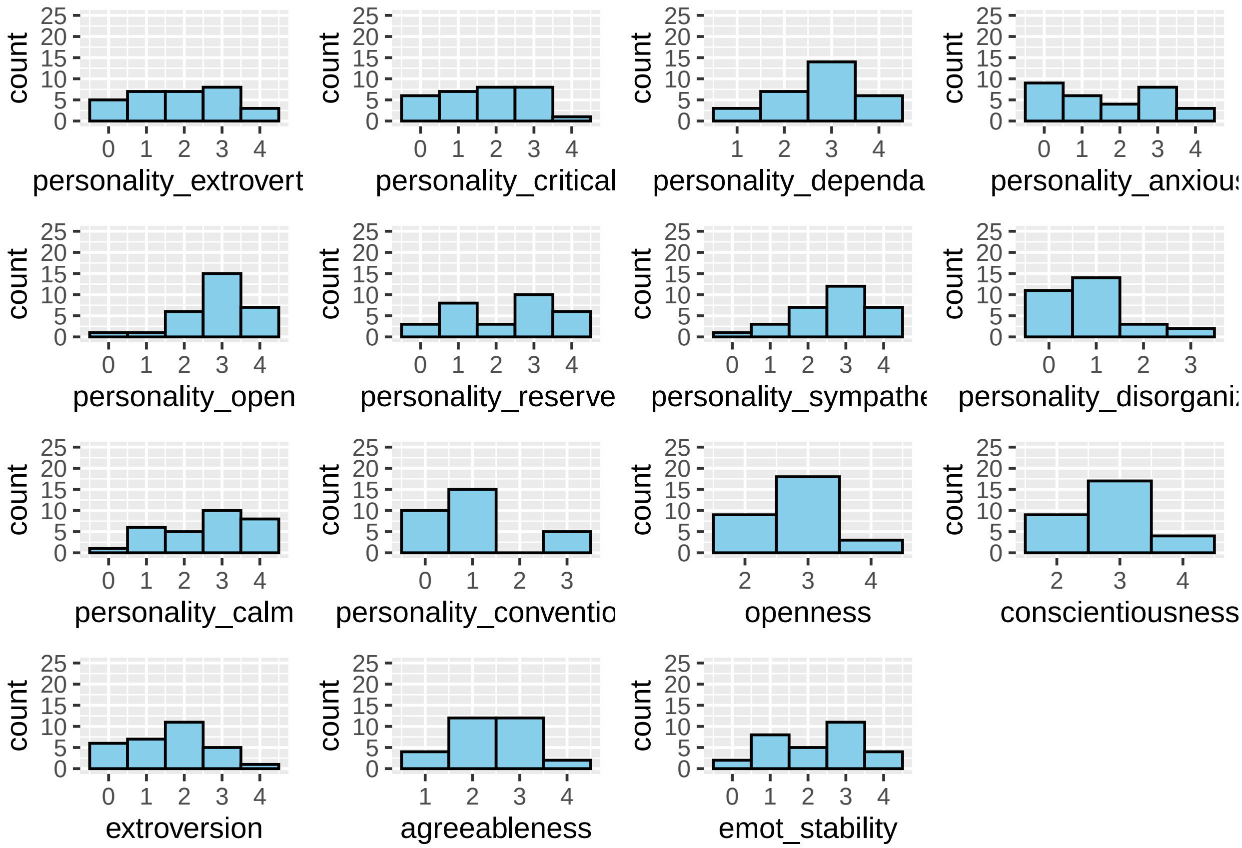
```
df_hostile_agent <-read.csv(file="hostileAgent.csv", stringsAsFactors=FALSE, header=T)
df_hostile_agent <-as.data.frame(df_hostile_agent)

df_hostile_agent <- create_personality_vars(df_hostile_agent)
df_hostile_agent <- create_pre_post_scale_items(df_hostile_agent)
df_hostile_agent <- create_delta_scale_items(df_hostile_agent)

pre_post_scale_items <- df_hostile_agent[c(preScaleItems, postScaleItems, 'prePositiveAffect', 'preNega
plot_hostile_agent_pre_post <- gridOfHistograms(pre_post_scale_items, "Hostile Agent Pre, Post Affect Hi
```

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




```
df_hostile_agent_personality <- df_hostile_agent[, 18:32]
plot_hostile_human_personality <- gridOfHistograms(df_hostile_agent_personality, "Hostile Agent Persona
```

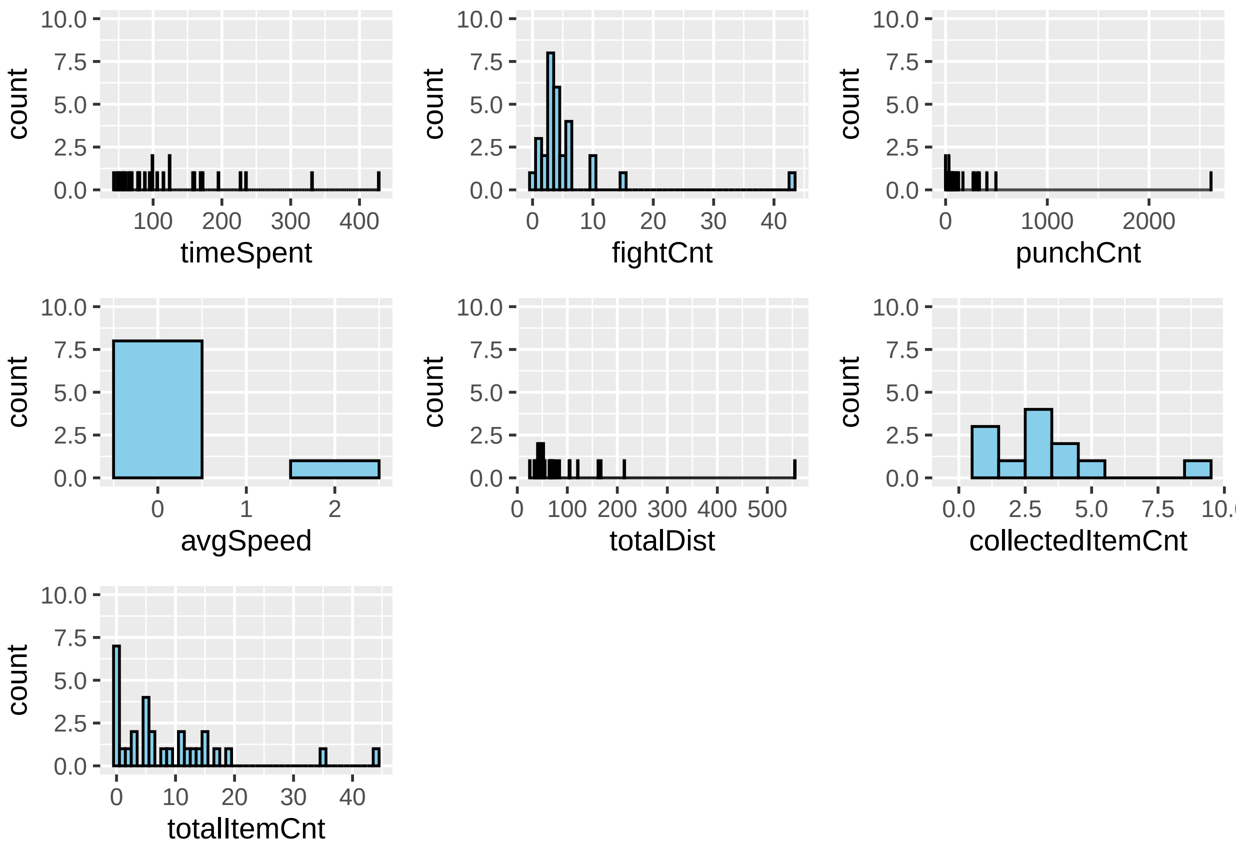
```
## Saving 6.5 x 4.5 in image
```




```
df_hostile_agent_behaviors <- df_hostile_agent[, 9:15]
plot_hostile_agent_behaviors <- gridOfHistograms(df_hostile_agent_behaviors, "Hostile Agent Behaviors Hi
```

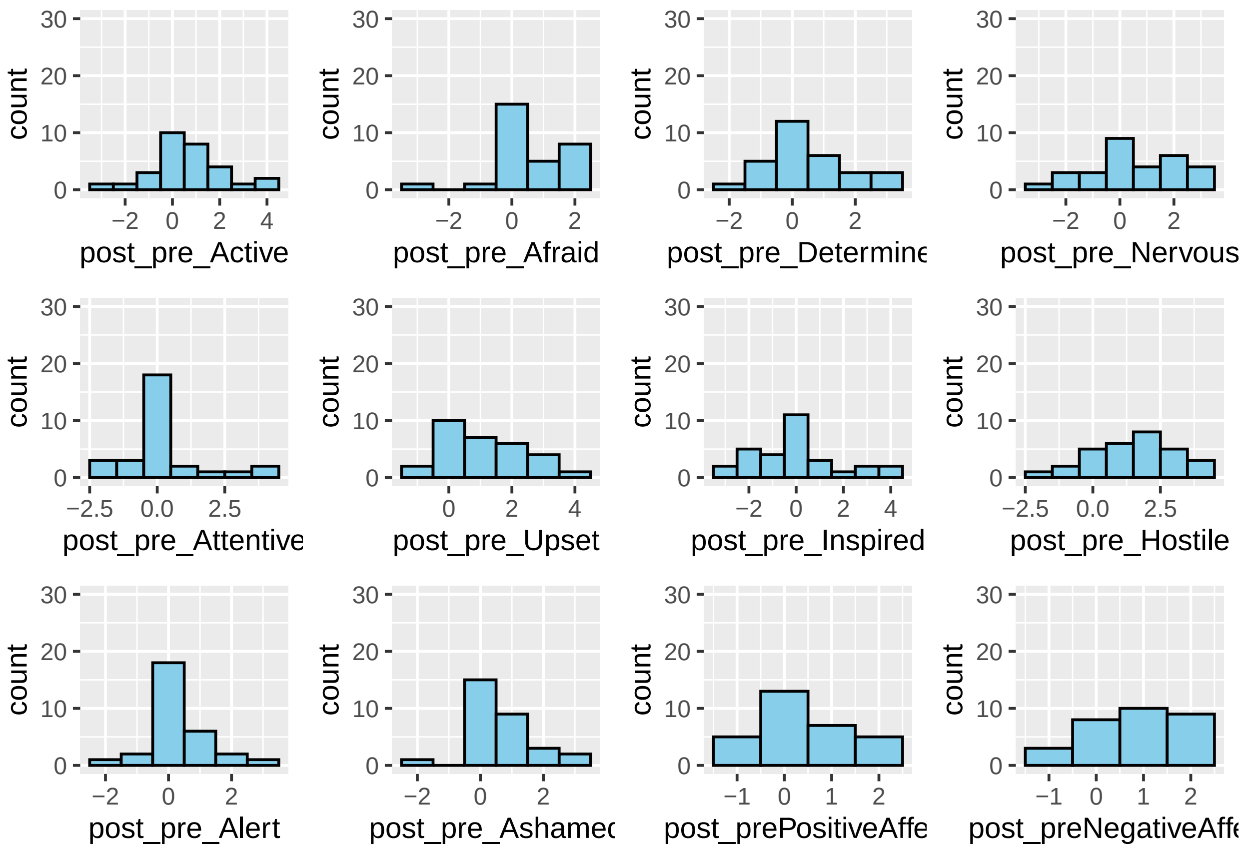
```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
```

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




```
df_hostile_agent_deltas <- df_hostile_agent[, 59:70]
plot_hostile_agent_deltas <- gridOfHistograms(df_hostile_agent_deltas, "Hostile Agent Post to Pre Diffe
```

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



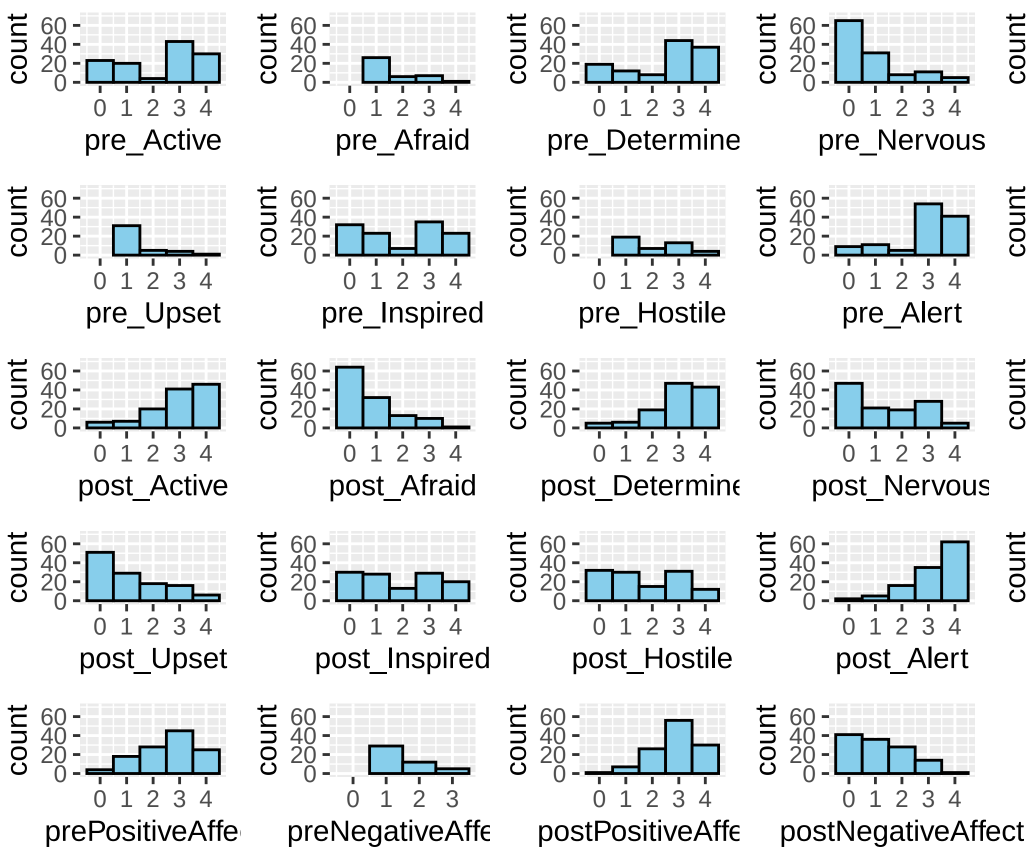
####combined####### ###################

```
df_docile_human$docile_hostile <- "Docile"
df_docile_human$human_agent <- "Human"
df_docile_human$combined_group <- "DocileHuman"
df_docile_agent$docile_hostile <- "Docile"
df_docile_agent$human_agent <- "Agent"
df_docile_agent$combined_group <- "DocileAgent"
df_hostile_human$docile_hostile <- "Hostile"
df_hostile_human$human_agent <- "Human"
df_hostile_human$combined_group <- "HostileHuman"
df_hostile_agent$docile_hostile <- "Hostile"
df_hostile_agent$human_agent <- "Agent"
df_hostile_agent$combined_group <- "HostileAgent"

mobOutcomes <- bind_rows(df_docile_human, df_docile_agent, df_hostile_human, df_hostile_agent)
mobOutcomes$docile_hostile<-as.factor(mobOutcomes$docile_hostile)
mobOutcomes$human_agent<-as.factor(mobOutcomes$human_agent)
mobOutcomes$combined_group<-as.factor(mobOutcomes$combined_group)
mobOutcomes$gender<-as.factor(mobOutcomes$gender)
mobOutcomes$nationality<-as.factor(mobOutcomes$nationality)

#create a punch_rate variable
mobOutcomes$punchRate <- mobOutcomes$punchCnt / mobOutcomes$timeSpent
#create a fight_rate variable
mobOutcomes$fightRate <- mobOutcomes$fightCnt / mobOutcomes$timeSpent
```

```
pre_post_scale_items <- mobOutcomes[c(preScaleItems, postScaleItems, 'prePositiveAffect', 'preNegativeA
plot_overall_pre_post <- gridOfHistograms(pre_post_scale_items, "Overall Pre, Post Affect Histograms",
```

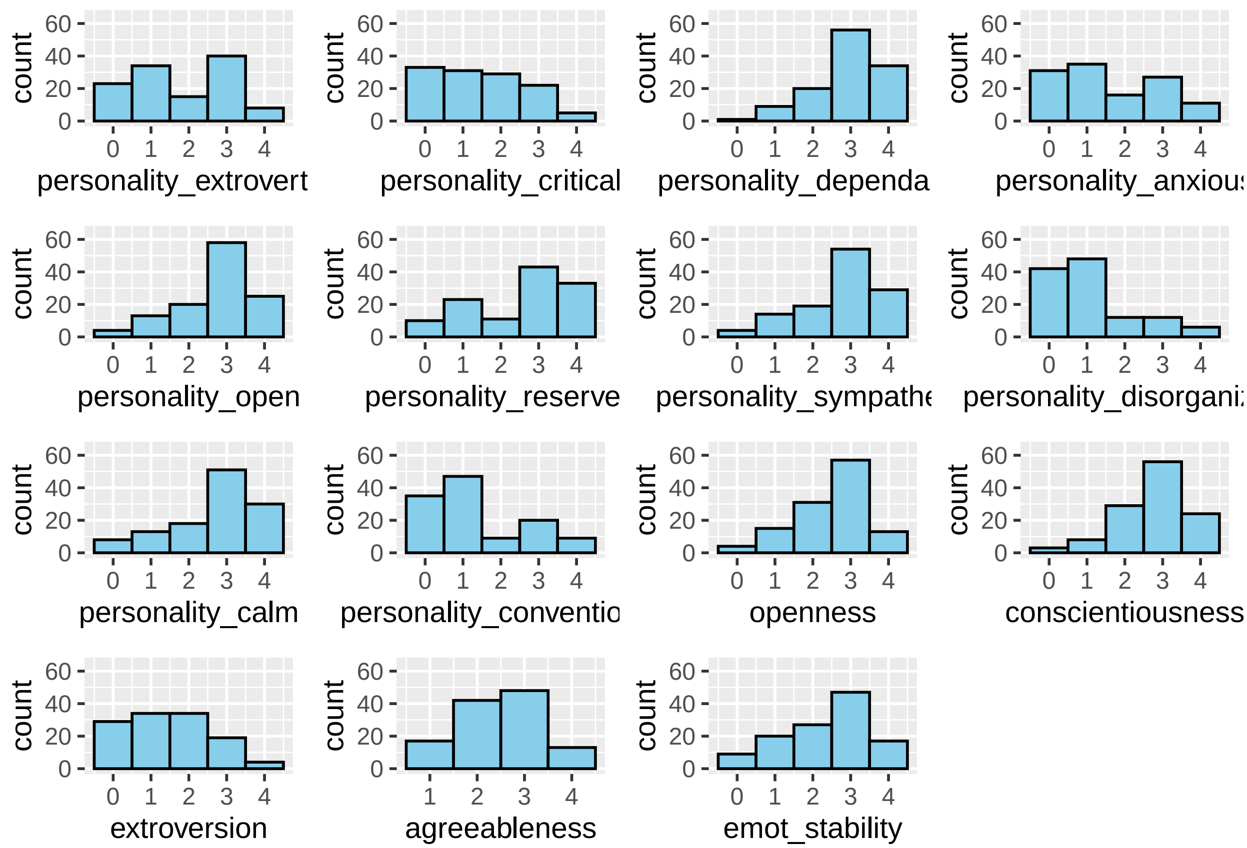
```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
## Removed 1 rows containing missing values (`geom_bar()`).
```

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




```
df_overall_personality <- mobOutcomes[, 18:32]
plot_overall_personality <- gridOfHistograms(df_overall_personality, "Overall Personality Histograms",
```

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




```r
df_overall_behaviors <- mobOutcomes[, 9:15]
plot_overall_behaviors <- gridOfHistograms(df_overall_behaviors, "Behavior Frequency Distributions")
```

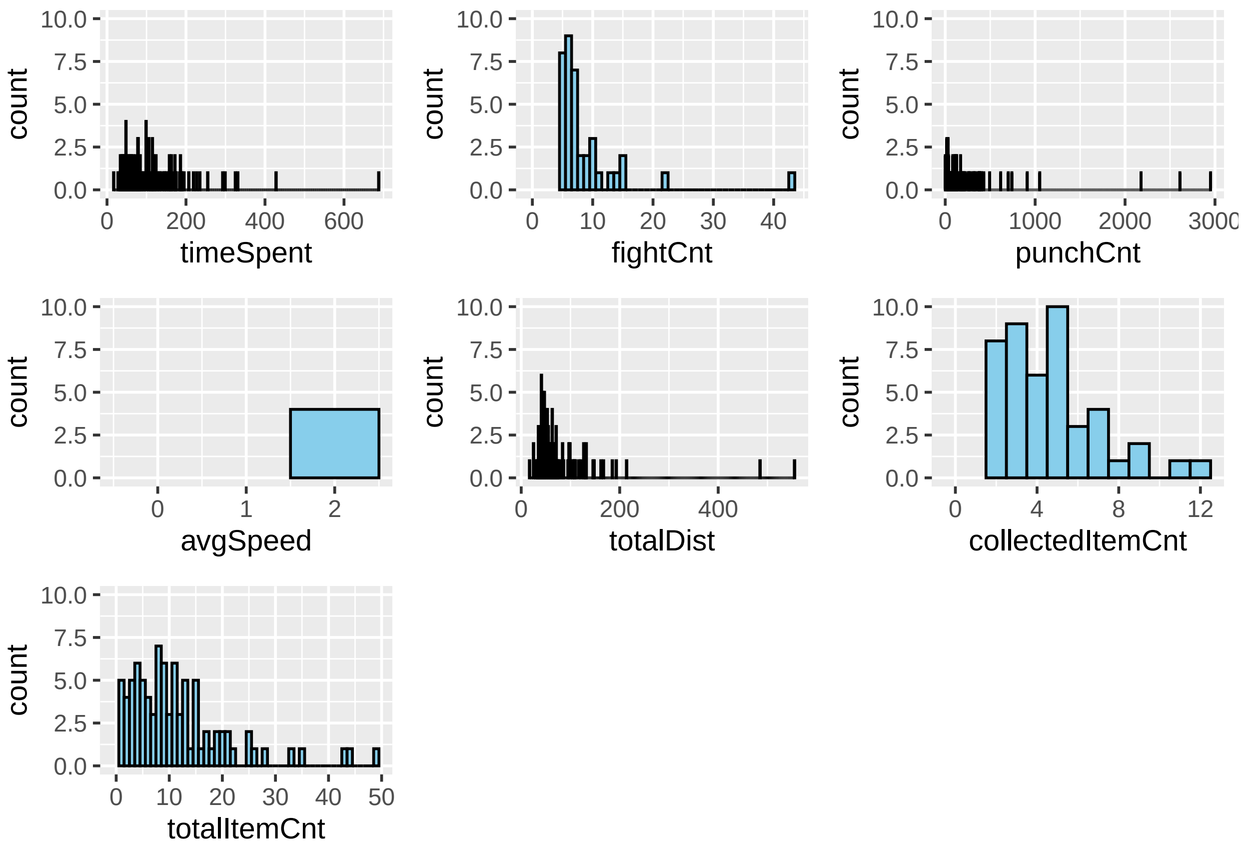
```
## Warning: Removed 5 rows containing missing values (`geom_bar()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
## Removed 2 rows containing missing values (`geom_bar()`).
```

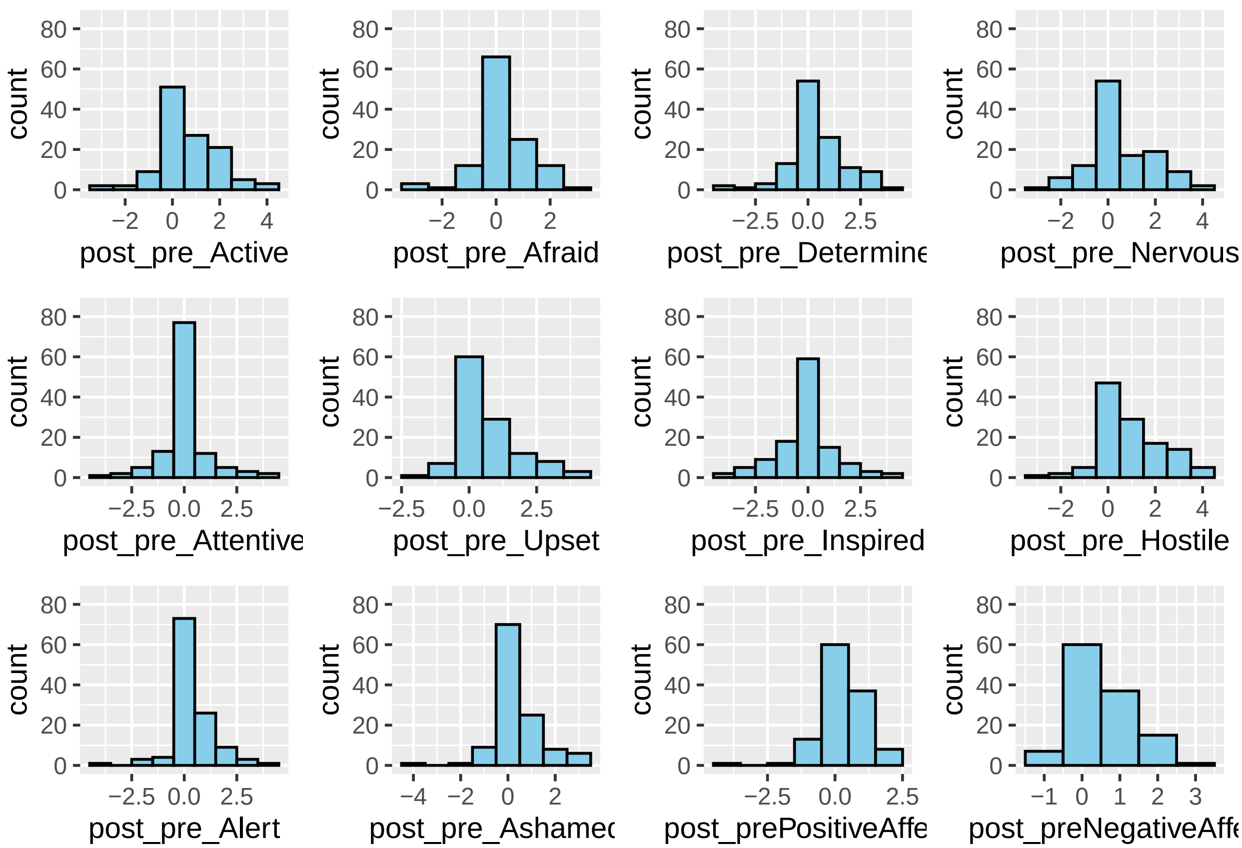
```
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
```

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



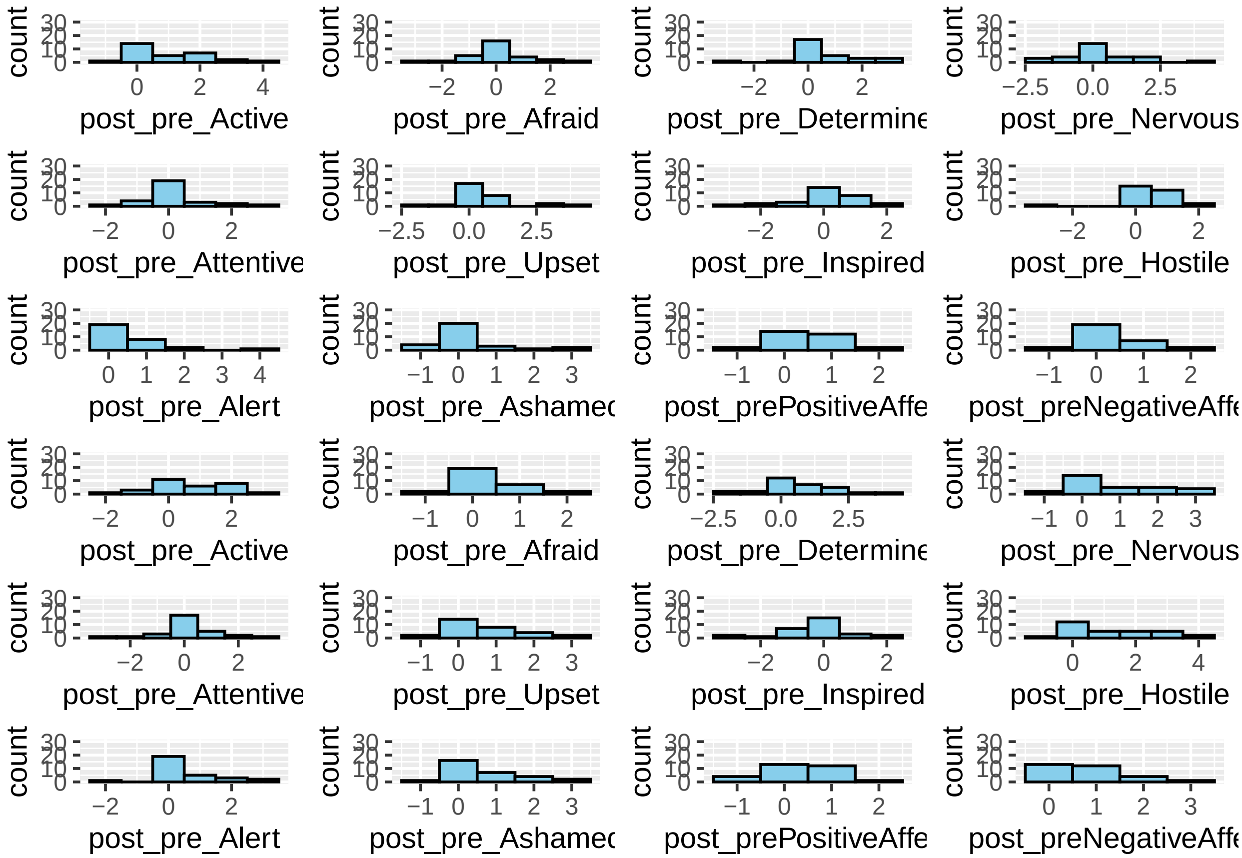

```
df_overall_deltas <- mobOutcomes[, 59:70]
plot_overall_deltas <- gridOfHistograms(df_overall_deltas, "Overall Post to Pre Differences Histograms"
```

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

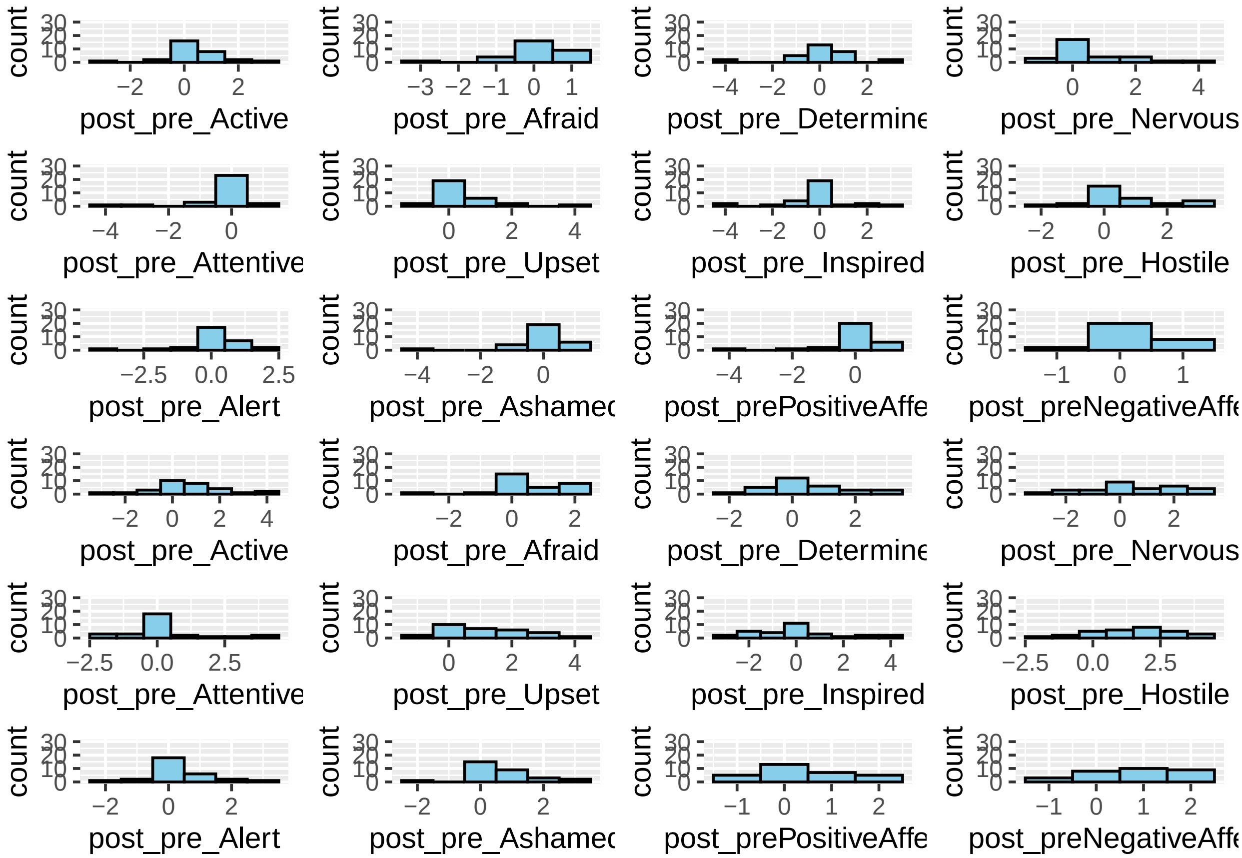


#GridBoxPlots ####################

```
docile <- subset(mobOutcomes, docile_hostile=="Docile")
hostile <- subset(mobOutcomes, docile_hostile=="Hostile")
human <- subset(mobOutcomes, human_agent=="Human")
agent <- subset(mobOutcomes, human_agent=="Agent")

deltas_docile_hostile_human <- ggarrange(plot_docile_human_deltas, plot_hostile_human_deltas,
                                         ncol=1, nrow=2, hjust = 0.01)
print(deltas_docile_hostile_human)
```




```
deltas_docile_hostile_agent <- ggarrange(plot_docile_agent_deltas, plot_hostile_agent_deltas,
                                          ncol=1, nrow=2, hjust = 0.01)
print(deltas_docile_hostile_agent)
```




```r
df_group_deltas <- subset(mobOutcomes, select = c(59:70))
docile_human <- bind_rows(docile, human)
docile_agent <- bind_rows(docile, agent)
hostile_human <- bind_rows(hostile, human)
hostile_agent <- bind_rows(hostile, agent)
```

#Corr Matrices ######################

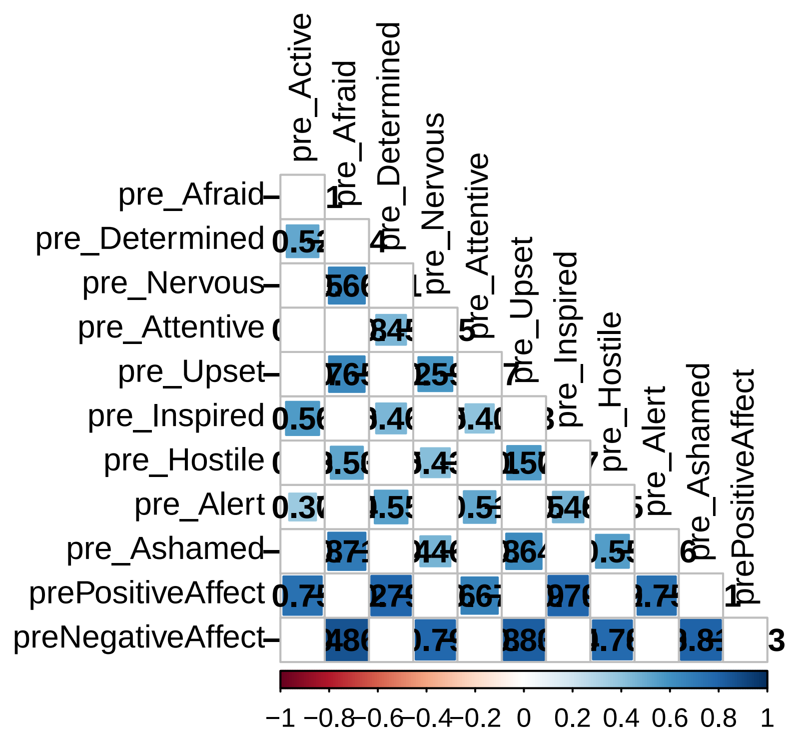
```r
subset_vars_pre <- mobOutcomes[, c(preScaleItems, 'prePositiveAffect', 'preNegativeAffect')]
# Calculate correlation and p-value matrices
p_mat_vars_pre <- cor_pmat(subset_vars_pre, method="pearson")
str(p_mat_vars_pre)
```

```
## pvalue [12 x 13] (S3: pvalue/tbl_df/tbl/data.frame)
##  $ rowname          : chr [1:12] "pre_Active" "pre_Afraid" "pre_Determined" "pre_Nervous" ...
##  $ pre_Active       : num [1:12] 0.00 9.23e-01 1.70e-09 1.12e-01 2.11e-03 ...
##  $ pre_Afraid       : num [1:12] 9.23e-01 0.00 6.33e-01 2.22e-16 1.82e-03 ...
##  $ pre_Determined   : num [1:12] 1.70e-09 6.33e-01 0.00 8.97e-01 1.94e-07 8.54e-01 1.52e-07 5.80e-01
##  $ pre_Nervous      : num [1:12] 1.12e-01 2.22e-16 8.97e-01 0.00 6.55e-03 ...
##  $ pre_Attentive    : num [1:12] 2.11e-03 1.82e-03 1.94e-07 6.55e-03 0.00 2.96e-03 3.79e-06 2.33e-01
##  $ pre_Upset        : num [1:12] 4.63e-01 1.77e-15 8.54e-01 1.71e-12 2.96e-03 ...
##  $ pre_Inspired     : num [1:12] 2.59e-11 8.05e-02 1.52e-07 6.05e-01 3.79e-06 ...
##  $ pre_Hostile      : num [1:12] 1.44e-01 7.18e-10 5.80e-01 1.20e-06 2.33e-01 ...
##  $ pre_Alert        : num [1:12] 3.37e-05 6.82e-01 1.19e-10 2.65e-01 2.27e-09 5.78e-01 3.85e-08 6.24
##  $ pre_Ashamed      : num [1:12] 4.09e-01 1.62e-19 6.57e-01 1.22e-07 1.95e-03 ...
```

```
##  $ prePositiveAffect: num [1:12] 6.66e-23 8.16e-01 3.45e-27 5.41e-01 4.12e-17 ...
##  $ preNegativeAffect: num [1:12] 7.01e-01 1.76e-36 9.65e-01 1.63e-26 1.63e-03 ...
```

```r
cor_matrix_vars_pre <- cor(subset_vars_pre, method = "pearson")
#convert p_mat to a matrix of doubles
#exclude first element
p_mat_vars_pre_test <- unlist(p_mat_vars_pre[-1])
#p_mat_vars_pre_4 <- matrix(unlist(p_mat_vars_pre), ncol = 10, byrow = FALSE)

# p adjust
p_adj_mat_vars_pre <- p.adjust(p_mat_vars_pre_test, method = "holm", n = length(p_mat_vars_pre_test))
matrix_p_adj_holm <- matrix(p_adj_mat_vars_pre, nrow = 12, ncol = 12)
rownames(matrix_p_adj_holm) <- c(preScaleItems, 'prePositiveAffect', 'preNegativeAffect')
colnames(matrix_p_adj_holm) <- c(preScaleItems, 'prePositiveAffect', 'preNegativeAffect')
pre_affect_corrs <- corrplot(cor_matrix_vars_pre, p.mat = matrix_p_adj_holm, method="square", type = 'l
```

Pre Affect Items Correlation Matrix

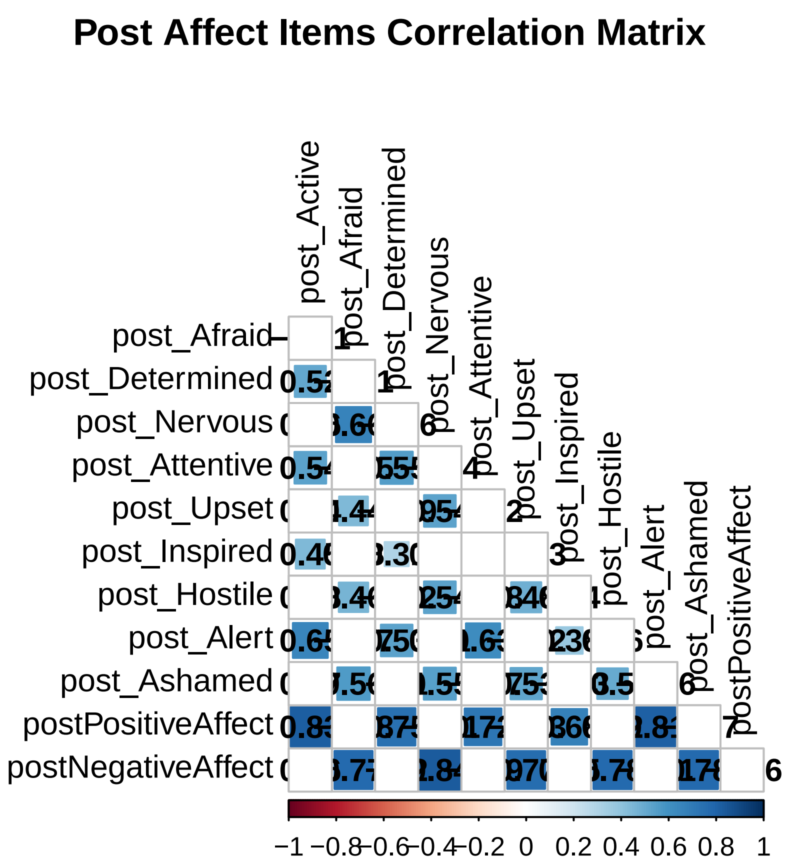



```r
subset_vars_pre <- mobOutcomes[, c(postScaleItems, 'postPositiveAffect', 'postNegativeAffect')]
# Calculate correlation and p-value matrices
p_mat_vars_pre <- cor_pmat(subset_vars_pre, method="pearson")
str(p_mat_vars_pre)
```

```
## pvalue [12 x 13] (S3: pvalue/tbl_df/tbl/data.frame)
##  $ rowname           : chr [1:12] "post_Active" "post_Afraid" "post_Determined" "post_Nervous" ...
##  $ post_Active       : num [1:12] 0.00 9.09e-01 1.53e-09 4.90e-01 2.76e-10 ...
##  $ post_Afraid       : num [1:12] 9.09e-01 0.00 1.92e-02 2.23e-16 1.07e-01 ...
```

```
##  $ post_Determined   : num [1:12] 1.53e-09 1.92e-02 0.00 7.47e-02 4.79e-11 ...
##  $ post_Nervous      : num [1:12] 4.90e-01 2.23e-16 7.47e-02 0.00 1.18e-01 ...
##  $ post_Attentive    : num [1:12] 2.76e-10 1.07e-01 4.79e-11 1.18e-01 0.00 ...
##  $ post_Upset        : num [1:12] 7.01e-01 3.73e-07 1.16e-03 2.83e-10 1.64e-02 0.00 7.14e-01 2.11e-08
##  $ post_Inspired     : num [1:12] 2.62e-07 3.64e-01 5.70e-04 1.31e-01 4.48e-02 7.14e-01 0.00 6.55e-01
##  $ post_Hostile      : num [1:12] 4.02e-01 9.66e-08 8.01e-01 2.47e-10 3.61e-01 2.11e-08 6.55e-01 0.00
##  $ post_Alert        : num [1:12] 7.57e-16 4.46e-01 3.11e-10 8.71e-01 2.19e-14 ...
##  $ post_Ashamed      : num [1:12] 4.75e-01 1.84e-11 2.88e-01 8.58e-11 6.69e-02 ...
##  $ postPositiveAffect: num [1:12] 2.19e-31 4.00e-01 8.04e-23 9.54e-01 1.48e-20 ...
##  $ postNegativeAffect: num [1:12] 5.02e-01 3.35e-25 3.11e-02 4.60e-33 3.68e-02 ...
```

```
cor_matrix_vars_post <- cor(subset_vars_pre, method = "pearson")
#convert p_mat to a matrix of doubles
#exclude first element
p_mat_vars_pre_test <- unlist(p_mat_vars_pre[-1])

# p adjust
p_adj_mat_vars_pre <- p.adjust(p_mat_vars_pre_test, method = "holm", n = length(p_mat_vars_pre_test))
matrix_p_adj_holm_2 <- matrix(p_adj_mat_vars_pre, nrow = 12, ncol = 12)
rownames(matrix_p_adj_holm_2) <- c(postScaleItems, 'postPositiveAffect', 'postNegativeAffect')
colnames(matrix_p_adj_holm_2) <- c(postScaleItems, 'postPositiveAffect', 'postNegativeAffect')
post_affect_cors <- corrplot(cor_matrix_vars_post, p.mat = matrix_p_adj_holm_2, method="square", type =
```



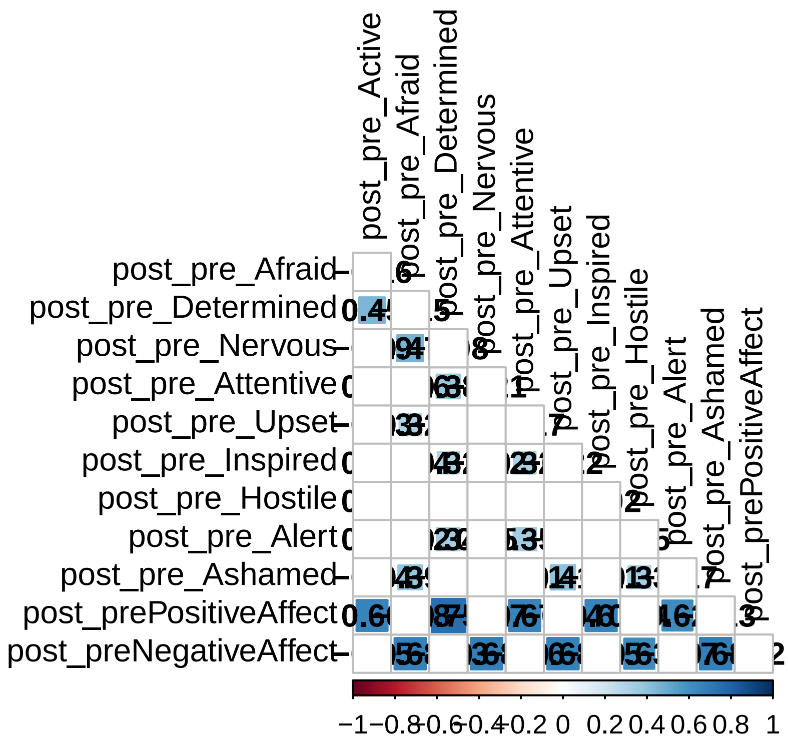

```
subset_vars_pre <- mobOutcomes[, 59:70]
# Calculate correlation and p-value matrices
```

```r
p_mat_vars_pre <- cor_pmat(subset_vars_pre, method="pearson")
str(p_mat_vars_pre)
```

```
## pvalue [12 x 13] (S3: pvalue/tbl_df/tbl/data.frame)
##  $ rowname               : chr [1:12] "post_pre_Active" "post_pre_Afraid" "post_pre_Determined" "pos
##  $ post_pre_Active       : num [1:12] 0.00 8.51e-02 2.41e-07 3.87e-02 5.98e-03 1.56e-01 8.22e-03 9.37
##  $ post_pre_Afraid       : num [1:12] 8.51e-02 0.00 1.01e-01 7.97e-08 7.25e-02 4.41e-04 6.53e-01 3.65
##  $ post_pre_Determined   : num [1:12] 2.41e-07 1.01e-01 0.00 3.72e-01 1.42e-05 4.92e-02 4.27e-04 9.66
##  $ post_pre_Nervous      : num [1:12] 3.87e-02 7.97e-08 3.72e-01 0.00 2.06e-02 8.87e-04 1.88e-01 4.15
##  $ post_pre_Attentive    : num [1:12] 5.98e-03 7.25e-02 1.42e-05 2.06e-02 0.00 5.90e-02 3.26e-04 2.33
##  $ post_pre_Upset        : num [1:12] 1.56e-01 4.41e-04 4.92e-02 8.87e-04 5.90e-02 0.00 1.41e-02 2.32
##  $ post_pre_Inspired     : num [1:12] 0.00822 0.653 0.000427 0.188 0.000326 0.0141 0 0.856 0.00211 0
##  $ post_pre_Hostile      : num [1:12] 0.937 0.0365 0.966 0.0415 0.233 0.00232 0.856 0 0.571 0.000216
##  $ post_pre_Alert        : num [1:12] 0.0042 0.191 0.000117 0.621 0.000111 0.356 0.00211 0.571 0 0.00
##  $ post_pre_Ashamed      : num [1:12] 6.59e-01 9.63e-06 6.90e-01 4.94e-03 2.37e-01 3.43e-06 2.17e-01
##  $ post_prePositiveAffect: num [1:12] 2.39e-16 4.35e-02 3.02e-23 6.11e-02 8.48e-17 ...
##  $ post_preNegativeAffect: num [1:12] 9.84e-02 2.45e-17 1.66e-01 8.93e-18 8.96e-02 ...
```

```r
cor_matrix_vars_post_pre <- cor(subset_vars_pre, method = "pearson")
#convert p_mat to a matrix of doubles
#exclude first element
p_mat_vars_pre_test <- unlist(p_mat_vars_pre[-1])

# p adjust
p_adj_mat_vars_pre <- p.adjust(p_mat_vars_pre_test, method = "holm", n = length(p_mat_vars_pre_test))
matrix_p_adj_holm_3 <- matrix(p_adj_mat_vars_pre, nrow = 12, ncol = 12)
rownames(matrix_p_adj_holm_3) <- colnames(subset_vars_pre)
colnames(matrix_p_adj_holm_3) <- colnames(subset_vars_pre)
post_pre_affect_cors <- corrplot(cor_matrix_vars_post_pre, p.mat = matrix_p_adj_holm_3, method="square"
```

Post−Pre Affect Items Correlation Matrix

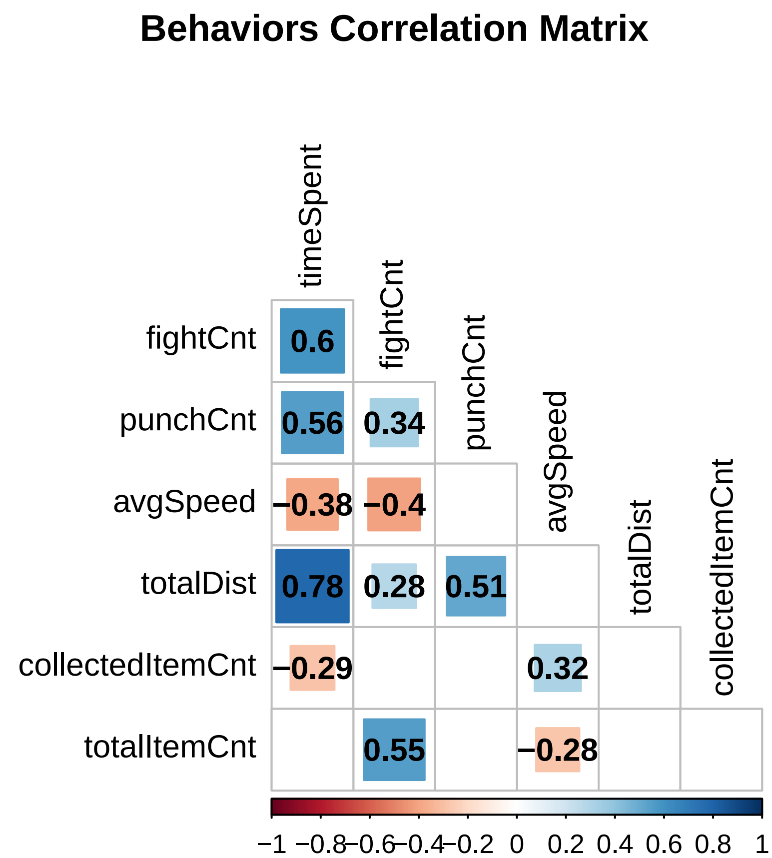



```r
subset_vars_pre <- mobOutcomes[, c(9,10,11,12,13,14,15)]
# Calculate correlation and p-value matrices
p_mat_vars_pre <- cor_pmat(subset_vars_pre, method="pearson")
str(p_mat_vars_pre)
```

```
## pvalue [7 x 8] (S3: pvalue/tbl_df/tbl/data.frame)
##  $ rowname         : chr [1:7] "timeSpent" "fightCnt" "punchCnt" "avgSpeed" ...
##  $ timeSpent       : num [1:7] 0.00 4.53e-13 3.08e-11 1.62e-05 4.63e-26 ...
##  $ fightCnt        : num [1:7] 4.53e-13 0.00 1.84e-04 5.09e-06 1.62e-03 ...
##  $ punchCnt        : num [1:7] 3.08e-11 1.84e-04 0.00 3.05e-01 2.14e-09 ...
##  $ avgSpeed        : num [1:7] 1.62e-05 5.09e-06 3.05e-01 0.00 1.26e-01 3.71e-04 2.05e-03
##  $ totalDist       : num [1:7] 4.63e-26 1.62e-03 2.14e-09 1.26e-01 0.00 ...
##  $ collectedItemCnt: num [1:7] 0.00138 0.00394 0.158 0.000371 0.11 0 0.462
##  $ totalItemCnt    : num [1:7] 7.45e-03 6.30e-11 1.05e-01 2.05e-03 8.49e-01 ...
```

```r
cor_matrix_vars_pre <- cor(subset_vars_pre, method = "pearson")
#convert p_mat to a matrix of doubles
#exclude first element
p_mat_vars_pre_test <- unlist(p_mat_vars_pre[-1])
#p_mat_vars_pre_4 <- matrix(unlist(p_mat_vars_pre), ncol = 10, byrow = FALSE)

# p adjust
p_adj_mat_vars_pre <- p.adjust(p_mat_vars_pre_test, method = "holm", n = length(p_mat_vars_pre_test))
matrix_p_adj_holm <- matrix(p_adj_mat_vars_pre, nrow = 7, ncol = 7)
rownames(matrix_p_adj_holm) <- colnames(subset_vars_pre)
```

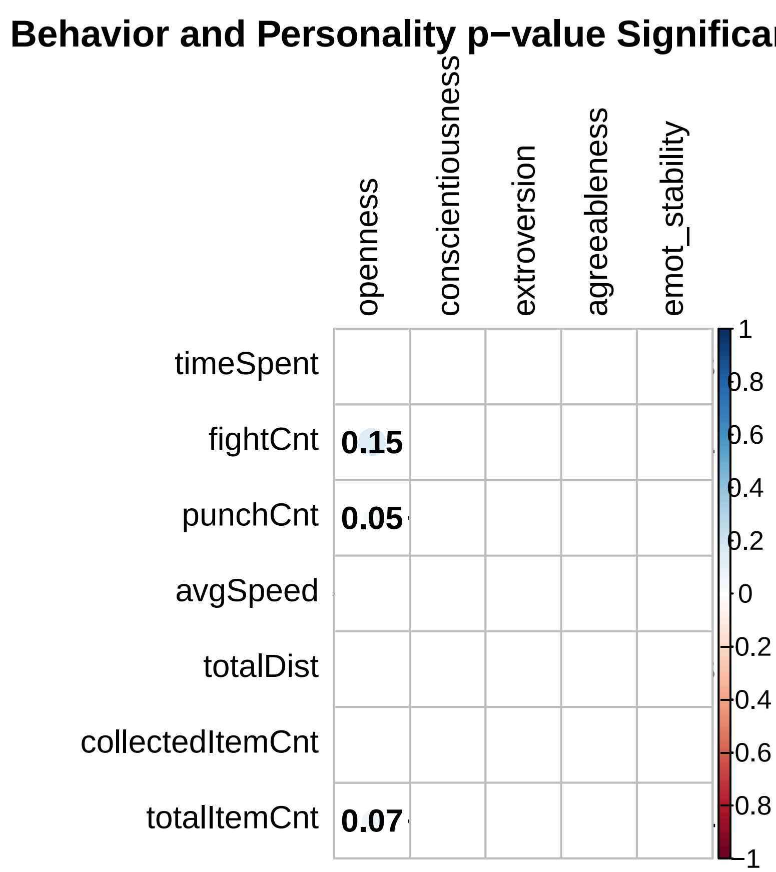
```r
colnames(matrix_p_adj_holm) <- colnames(subset_vars_pre)
corrplot(cor_matrix_vars_pre, p.mat = matrix_p_adj_holm, method="square", type = 'lower', diag = FALSE,
```




```r
# Select variables for correlation calculation
vars_X <- c("timeSpent", "fightCnt", "punchCnt", "avgSpeed", "totalDist", "collectedItemCnt", "totalIte
vars_Y <- c("openness", "conscientiousness", "extroversion", "agreeableness", "emot_stability")
selected_vars <- c(vars_X, vars_Y)

# Subset the data based on selected variables
subset_df <- mobOutcomes[, selected_vars]

# Calculate correlation matrix
p.mat <- cor_pmat(subset_df, method="spearman")
cor_matrix <- cor(subset_df, method = "spearman")

# Specify the quadrant you want to display (e.g., upper right)
quadrant_rows <- 1:length(vars_X)
quadrant_cols <- (length(vars_X) + 1):ncol(cor_matrix)
quadrant_matrix <- cor_matrix[quadrant_rows, quadrant_cols]
quadrant_p.mat <- p.mat[quadrant_rows, quadrant_cols]
dim(quadrant_p.mat)
```

```
## [1] 7 5
```

```
matrix_quadrant_p <- matrix(unlist(quadrant_p.mat), ncol = 5, byrow = FALSE)

#quadrant_p_adj.mat = num[1:35]
quadrant_p_adj.mat <- p.adjust(matrix_quadrant_p, method = "holm", n = length(matrix_quadrant_p))
View(quadrant_p_adj.mat)
matrix_quadrant_p_adj_holm <- matrix(quadrant_p_adj.mat, nrow = 7, ncol = 5)
rownames(matrix_quadrant_p_adj_holm) <- vars_X
colnames(matrix_quadrant_p_adj_holm) <- vars_Y

plot1 <- corrplot(quadrant_matrix, p.mat = matrix_quadrant_p_adj_holm, method="circle", tl.col="black",
```



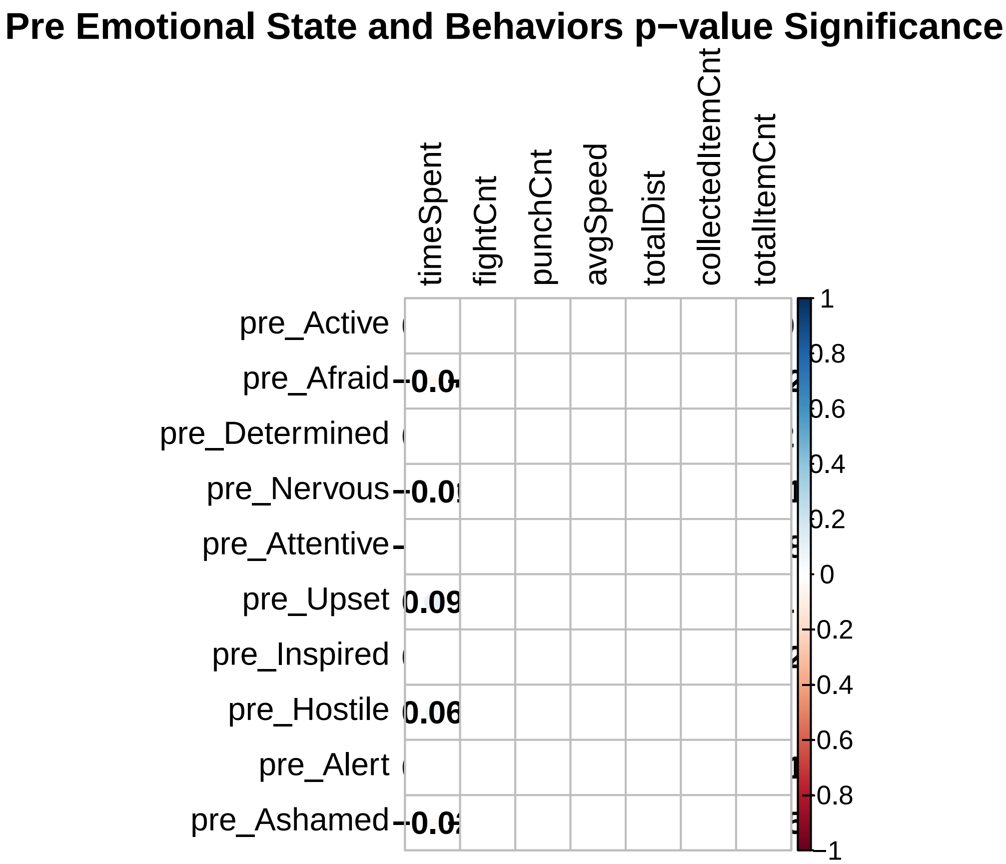

```
vars_X2 <- c("pre_Active", "pre_Afraid", "pre_Determined", "pre_Nervous", "pre_Attentive", "pre_Upset",
vars_X3 <- c("post_pre_Active", "post_pre_Afraid", "post_pre_Determined", "post_pre_Nervous", "post_pre

selected_vars_2 <- c(vars_X2, vars_X)
subset_df_2 <- mobOutcomes[, selected_vars_2]
p.mat_2 <- cor_pmat(subset_df_2, method="spearman")
cor_matrix_2 <- cor(subset_df_2, method="spearman")

quadrant_rows_2 <- 1:length(vars_X2)
quadrant_cols_2 <- (length(vars_X2) + 1):ncol(cor_matrix_2)
quadrant_matrix_2 <- cor_matrix_2[quadrant_rows_2, quadrant_cols_2]
quadrant_p.mat_2 <- p.mat_2[quadrant_rows_2, quadrant_cols_2]
dim(quadrant_p.mat_2)
```

```
## [1] 10  7
```

```r
matrix_quadrant_p_2 <- matrix(unlist(quadrant_p.mat_2), ncol = 7, byrow = FALSE)
typeof(matrix_quadrant_p_2)
```

```
## [1] "double"
```

```r
#quadrant_p_adj.mat_2 = num[1:70]
quadrant_p_adj.mat_2 <- p.adjust(matrix_quadrant_p_2, method = "holm", n = length(matrix_quadrant_p_2))
matrix_quadrant_p_adj_holm_2 <- matrix(quadrant_p_adj.mat_2, nrow = 10, ncol = 7)
rownames(matrix_quadrant_p_adj_holm_2) <- vars_X2
colnames(matrix_quadrant_p_adj_holm_2) <- vars_X


corrplot(quadrant_matrix_2, p.mat = matrix_quadrant_p_adj_holm_2, method="circle", tl.col="black", titl
```

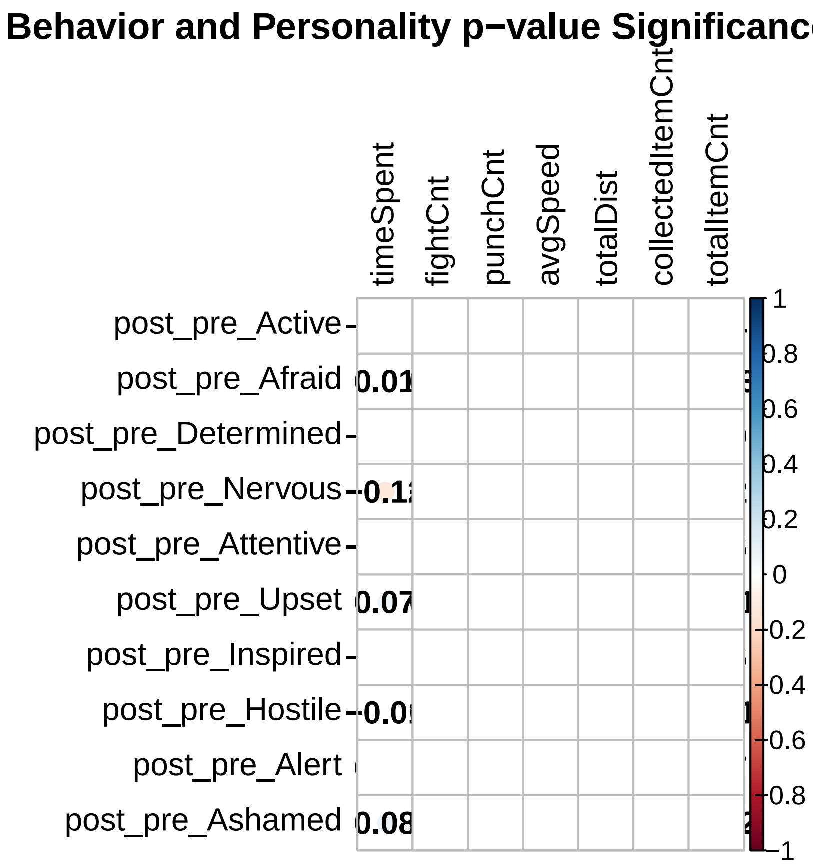



```r
selected_vars_3 <- c(vars_X3, vars_X)
subset_df_3 <- mobOutcomes[, selected_vars_3]
p.mat_3 <- cor_pmat(subset_df_3, method = "spearman")
cor_matrix_3 <- cor(subset_df_3, method = "spearman")

quadrant_rows_3 <- 1:length(vars_X3)
quadrant_cols_3 <- (length(vars_X3) + 1):ncol(cor_matrix_3)
quadrant_matrix_3 <- cor_matrix_3[quadrant_rows_3, quadrant_cols_3]
```

```
quadrant_p.mat_3 <- p.mat_3[quadrant_rows_3, quadrant_cols_3]
dim(quadrant_p.mat_3)
```

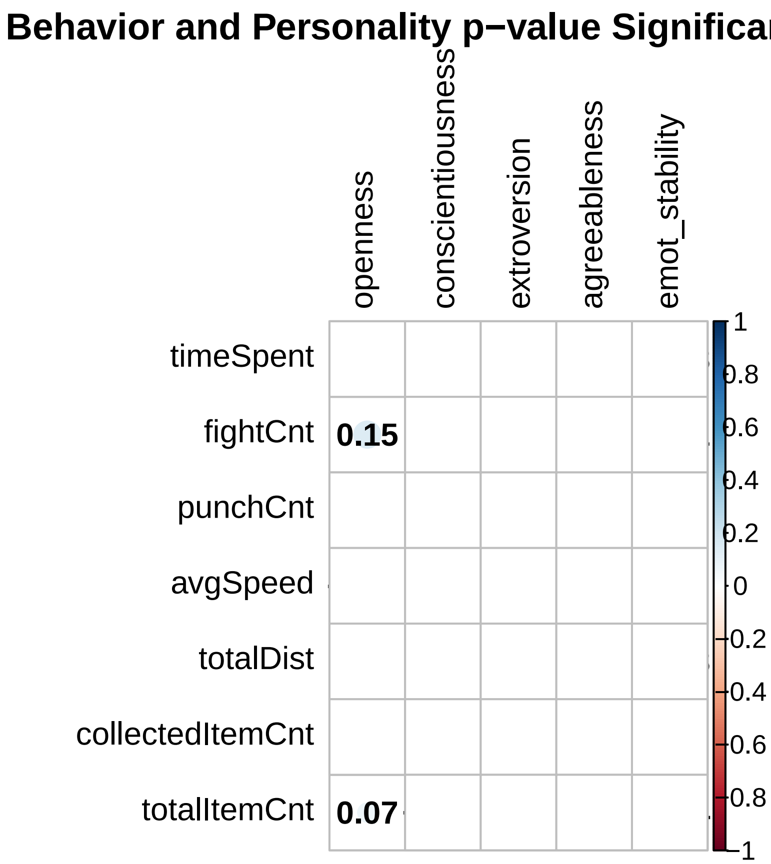
```
## [1] 10  7
```

```
matrix_quadrant_p_3 <- matrix(unlist(quadrant_p.mat_3), ncol = 7, byrow = FALSE)

#quadrant_p_adj.mat_3 = num[1:70]
quadrant_p_adj.mat_3 <- p.adjust(matrix_quadrant_p_3, method = "BH", n = length(matrix_quadrant_p_3))
matrix_quadrant_p_adj_holm_3 <- matrix(quadrant_p_adj.mat_3, nrow = 10, ncol = 7)
rownames(matrix_quadrant_p_adj_holm_3) <- vars_X3
colnames(matrix_quadrant_p_adj_holm_3) <- vars_X

plot_corr_3 <- corrplot(quadrant_matrix_3, p.mat = matrix_quadrant_p_adj_holm_3, method="circle", tl.co
```




```
#combine all 3 matrices by flattening
#7x5, 10x7, 10x7
#list of 5 (column wise elements)
vector_quadrant_p.mat=c(quadrant_p.mat)
#list of 7
vector_quadrant_p.mat_2=c(quadrant_p.mat_2)
#list of 7
vector_quadrant_p.mat_3=c(quadrant_p.mat_3)
#list of 19 (also column wise elements)
```

```
vector_concatenated_quadrant_p.mat <- c(vector_quadrant_p.mat,vector_quadrant_p.mat_2, vector_quadrant_p
double_concatenated_quadrant_p.mat <- as.numeric(unlist(vector_concatenated_quadrant_p.mat))
#num[1:175]
list_concatenated_adjusted_p_values <- p.adjust(double_concatenated_quadrant_p.mat, method = "holm", n =
#get first matrix set of adjusted p_values
list_1_adj_p_values_columns <- list(list_concatenated_adjusted_p_values[1:7], list_concatenated_adjusted
matrix_1_adj_p_values <- matrix(unlist(list_1_adj_p_values_columns), ncol = 5, byrow = FALSE)
rownames(matrix_1_adj_p_values) <- vars_X
colnames(matrix_1_adj_p_values) <- vars_Y

#get second matrix set of adjusted p_values
list_2_adj_p_values_columns <- list(list_concatenated_adjusted_p_values[36:45], list_concatenated_adjust
matrix_2_adj_p_values <- matrix(unlist(list_2_adj_p_values_columns), ncol = 7, byrow = FALSE)
rownames(matrix_2_adj_p_values) <- vars_X2
colnames(matrix_2_adj_p_values) <- vars_X
#get third matrix set of adjusted p_values
list_3_adj_p_values_columns <- list(list_concatenated_adjusted_p_values[106:115], list_concatenated_adju
matrix_3_adj_p_values <- matrix(unlist(list_3_adj_p_values_columns), ncol = 7, byrow = FALSE)
rownames(matrix_3_adj_p_values) <- vars_X3
colnames(matrix_3_adj_p_values) <- vars_X

###
corrplot(quadrant_matrix, p.mat = matrix_1_adj_p_values, method="circle", tl.col="black", title = "Beha
```



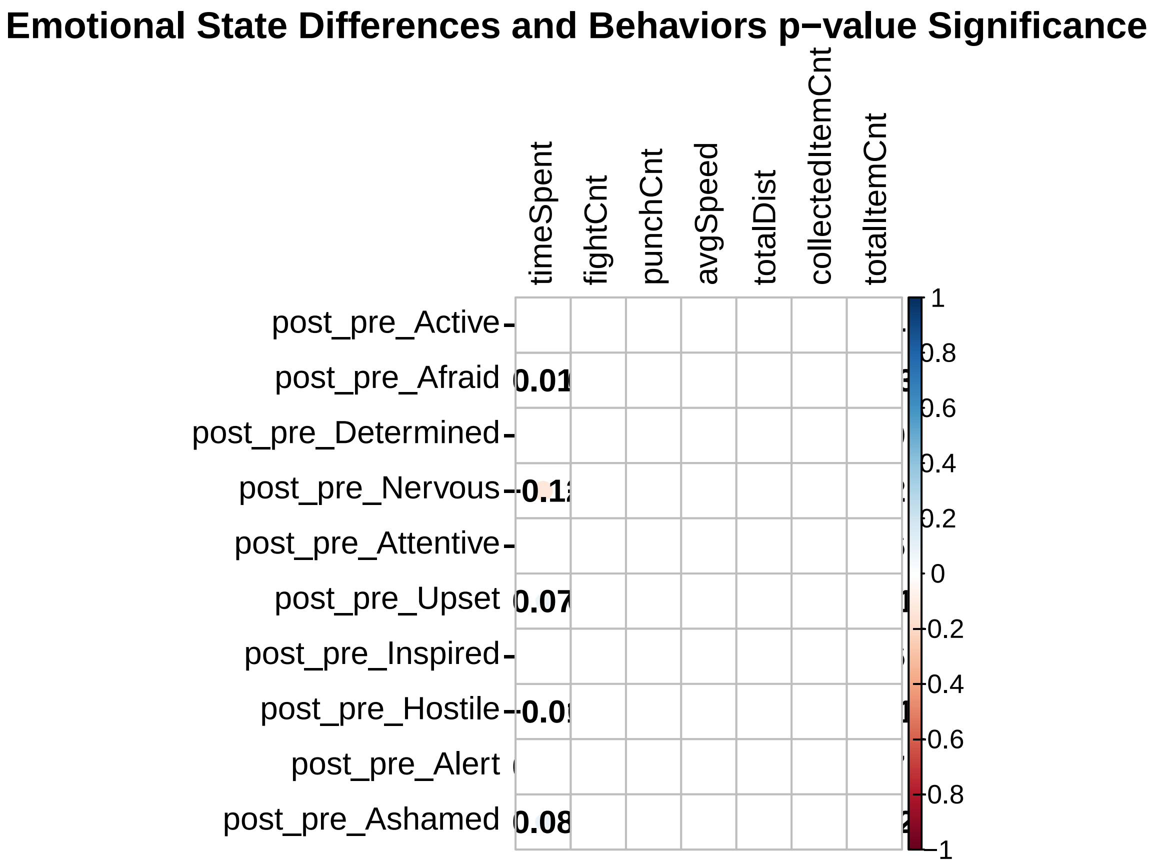
```r
plot.new();
```

```r
dev.off();
```

```
## null device
##           1
```

```r
par(mfrow=c(1,2), mai = c(1, 0.1, 0.1, 0.1))
corrplot(quadrant_matrix_2, p.mat = matrix_2_adj_p_values, method="circle", tl.col="black", title= "Pre
corrplot(quadrant_matrix_3, p.mat = matrix_3_adj_p_values, method="circle", tl.col="black", title= "Emo
```

```r
corrplot(quadrant_matrix_3, p.mat = matrix_3_adj_p_values, method="circle", tl.col="black", title= "Emo
```



#ANOVA for Affective Variables with respect to Agency and Hostility #Independence of the observations is assumed as measurements within and between the 4 groups are not related. # Normality. With a sample size 0f 30 per group, anf given that ANOVA is robust to the assumption of narmality, we do not need to check normality. # Nevertheless, from observing frequency histograms we find that the affective variables appear approximately normal across groups. # Equality of variances. We Perform Bartlett's test of the null that the variances in each of the groups (samples) are the same #Bartlett's test with multiple independent variables: the interaction() function must be used to collapse multiple factors into a single variable containing all combinations of the factors

Note: researchers should select only

variables for which there is a theoretical basis for inclusion. Then they should explore the data

with univariate and bivariate analyses, and only include variables that have potentially

informative results, or which are needed to serve as controls.

#Check for Outlier #two-way ANOVA tests #post_pre_Active #p = 0.21, we fail to reject the null of equal variances across factor levels bartlett.test(post_pre_Active ~ interaction(human_agent, docile_hostile), data = mobOutcomes) #Identify Outliers and run model mobOutcomesActive <- mobOutcomes[c(“human_agent”, “docile_hostile”, “combined_group”,“post_pre_Active”, “id”)] df_outliers <- mobOutcomesActive %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Active)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesActive <- mobOutcomesActive %>% anti_join(df_outliers[which(df_o %in% TRUE),], by = “id”)

aov_active <- aov(post_pre_Active ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesActive)

Df Sum Sq Mean Sq F value Pr(>F)

human_agent 1 3.01 3.0083 1.895 0.171

docile_hostile 1 0.01 0.0083 0.005 0.942

human_agent:docile_hostile 1 2.41 2.4083 1.517 0.221

Residuals 116 184.17 1.5876

summary(aov_active) #run model without removal of outliers #NA

#post_pre_Nervous #p = 0.2084, we fail to reject the null of equal variances across factor levels bartlett.test(post_pre_Nervous ~ interaction(human_agent, docile_hostile), data = mobOutcomes)

mobOutcomesNervous<- mobOutcomes[c(“human_agent”, “docile_hostile”, “combined_group”,“post_pre_Nervous”, “id”)] df_outliers <- mobOutcomesNervous %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Nervous)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesNervous <- mobOutcomesNervous %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = “id”)

aov_nervous <- aov(post_pre_Nervous ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesNervous)

Df Sum Sq Mean Sq F value Pr(>F)

human_agent 1 0.01 0.0083 0.005 0.946

docile_hostile 1 3.01 3.0083 1.647 0.202

human_agent:docile_hostile 1 3.01 3.0083 1.647 0.202

Residuals 116 211.90 1.8267

summary(aov_nervous)

#post_preNegativeAffect #p = 0.05365, we fail to reject the null of equal variances across factor levels bartlett.test(post_pre_NegativeAffect ~ interaction(human_agent, docile_hostile), data = mobOutcomes)

mobOutcomesNegative<- mobOutcomes[c("human_agent", "docile_hostile", "combined_group","post_pre_NegativeAffect "id")] df_outliers <- mobOutcomesNegative %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_NegativeAffect)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesNegative <- mobOutcomesNegative %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id") aov_negative <- aov(post_pre_NegativeAffect ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesNegative)

Df Sum Sq Mean Sq F value Pr(>F)

human_agent 1 0.18 0.176 0.315 0.575505

docile_hostile 1 7.80 7.803 13.954 0.000292 ***

human_agent:docile_hostile 1 0.08 0.075 0.134 0.714861

Residuals 116 64.87 0.559

summary(aov_negative) TukeyHSD(aov_negative)

#post_pre_ashamed #p = 0.962, we fail to reject the null of equal variances across factor levels bartlett.test(post_pre_Ashamed ~ interaction(human_agent, docile_hostile), data = mobOutcomes) mobOutcomesAshamed<- mobOutcomes[c("human_agent", "docile_hostile", "combined_group","post_pre_Ashamed", "id")] df_outliers <- mobOutcomesAshamed %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Ashamed)

#Remove 21 Extreme Outliers (n = 99) and run model mobOutcomesAshamed <- mobOutcomesAshamed %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aov_ashamed <- aov(post_pre_Ashamed ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesAshamed)

Df Sum Sq Mean Sq F value Pr(>F)

human_agent 1 0.00 0.004 0.006 0.93888

docile_hostile 1 9.99 9.993 15.919 0.00013 ***

human_agent:docile_hostile 1 0.01 0.007 0.010 0.91877

Residuals 95 59.63 0.628

summary(aov_ashamed) TukeyHSD(aov_ashamed)

#!!!Run model without removal of extreme outliers!!! aov_ashamed_2 <- aov(post_pre_Ashamed ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomes) summary(aov_ashamed_2) # Df Sum Sq Mean Sq F value Pr(>F)
human_agent 1 0.83 0.833 0.856 0.3567
docile_hostile 1 9.63 9.633 9.901 0.0021 ** # human_agent:docile_hostile 1 0.53 0.533 0.548 0.4606
Residuals 116 112.87 0.973

#post_pre_Hostile #!!!p = 0.0343, we reject the null of equal variances across factor levels and run a Welch ANOVA on a single extended factor instead!! bartlett.test(post_pre_Hostile ~ interaction(human_agent, docile_hostile), data = mobOutcomes)

mobOutcomesHostile<- mobOutcomes[c("combined_group","post_pre_Hostile", "id")] df_outliers <- mobOutcomesHostile %>% group_by(combined_group) %>% identify_outliers(post_pre_Hostile)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesHostile <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers
%in% TRUE),], by = "id")

#perform Welch's ANOVA # data: post_pre_Hostile and combined_group # F = 4.7989, num df = 3.000, denom df = 62.972, p-value = 0.004496 oneway.test(post_pre_Hostile ~ combined_group, data = mobOutcomesHostile, var.equal = FALSE)

> games__howell__test(mobOutcomesHostile, post__pre__Hostile ~ combined__group, conf.level = 0.95, detailed = FALSE)

A tibble: 6 × 8

.y. group1 group2 estimate conf.low conf.high p.adj p.adj.signif

*****

1 post__pre__Hostile DocileAgent DocileHuman -0.167 -0.911 0.578 0.934 ns

2 post__pre__Hostile DocileAgent HostileAgent 0.9 -0.0617 1.86 0.074 ns

3 post__pre__Hostile DocileAgent HostileHuman 0.633 -0.275 1.54 0.264 ns

4 post__pre__Hostile DocileHuman HostileAgent 1.07 0.196 1.94 0.011 *

5 post__pre__Hostile DocileHuman HostileHuman 0.8 -0.00998 1.61 0.054 ns

6 post__pre__Hostile HostileAgent HostileHuman -0.267 -1.28 0.744 0.897 ns

means__by__factor <- aggregate(post__pre__Hostile ~ combined__group, data = mobOutcomesHostile, FUN = mean) means__by__factor

combined_group post_pre_Hostile

1 DocileAgent 0.6000000

2 DocileHuman 0.4333333

3 HostileAgent 1.5000000

4 HostileHuman 1.2333333

games_howell_test(mobOutcomesHostile, post_pre_Hostile ~ combined_group, conf.level = 0.95, detailed = FALSE)

aov_hostile <- aov(post_pre_Hostile ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesHostile) TukeyHSD(aov_hostile) ##### #post_pre_Afraid mobOutcomesAfraid <- mobOutcomes[c("human_agent", "docile_hostile", "combined_group","post_pre_Afraid", "id")] df_outliers <- mobOutcomesAfraid %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Afraid)

#Remove 14 Extreme Outliers (n = 106) mobOutcomesAfraid <- mobOutcomes %>% anti_join(df_outliers[which(df_outlier
%in% TRUE),], by = "id")

aov_afraid <- aov(post_pre_Afraid ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesAfraid)

Df Sum Sq Mean Sq F value Pr(>F)

human_agent 1 0.38 0.381 0.529 0.4687

docile_hostile 1 4.44 4.443 6.163 0.0147 *

human_agent:docile_hostile 1 0.25 0.246 0.341 0.5603

Residuals 102 73.53 0.721

—

Signif. codes: 0 ‘’ *0.001* ‘’ *0.01* ‘’ 0.05 ‘.’ 0.1 ‘ ’ 1

summary(aov_afraid)

Repeat by Subset candidate variables to test group differences for (based on boxplot visualizations):

Active, Nervous, Negative, Ashamed, Hostile, Afraid

p_vector_subset <- c(0.171, 0.942, 0.221, 0.946, 0.202, 0.202, 0.575505, 0.000292, 0.714861, 0.93888, 0.00013, 0.91877, 0.362463, 0.000491, 0.833266 , 0.4687, 0.0147, 0.5603) #bonferroni # [1] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 0.005256 1.000000 1.000000 0.002340 1.000000 1.000000 0.008838 1.000000 1.000000 0.264600 1.000000 #holm # [1] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 0.004964 1.000000 1.000000 0.002340 1.000000 1.000000 0.007856 1.000000 1.000000 0.220500 1.000000 p.adjust(p_vector_subset, method = "bonferroni", n = length(p_vector_subset))

Repeat by Subset candidate variables to test group differences for (based on boxplot visualizations):

Active, Nervous, Negative, Ashamed, Hostile, Afraid, Determined, Attentive, Upset, Inspired, Alert, Positive

#post_pre_Determined mobOutcomesDetermined <- mobOutcomes[c("human_agent", "docile_hostile", "human_agent", "combined_group","post_pre_Determined", "id")] df_outliers <- mobOutcomesDetermined %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Determined)

#Remove 2 Extreme Outliers (n = 118) mobOutcomesDetermined <- mobOutcomes %>% anti_join(df_outliers[which(df_ou %in% TRUE),], by = "id")

aov_determined <- aov(post_pre_Determined ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesDetermined)

human_agent 1 0.11 0.1062 0.175 0.6764

docile_hostile 1 2.05 2.0495 3.381 0.0691 .

human_agent:docile_hostile 1 1.31 1.3058 2.154 0.1456

summary(aov_determined)

#post_pre_Attentive

mobOutcomesAttentive<- mobOutcomes[c("human_agent", "docile_hostile", "human_agent", "combined_group","post_pre_Attentive", "id")] df_outliers <- mobOutcomesAttentive %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Attentive)

#Remove 30 Extreme Outliers (n = 90) mobOutcomesAttentive <- mobOutcomes %>% anti_join(df_outliers[which(df_outli %in% TRUE),], by = "id")

aov_attentive <- aov(post_pre_Attentive ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesAttentive)

human_agent 1 1.26 1.2622 1.047 0.309

docile_hostile 1 0.76 0.7569 0.628 0.430

human_agent:docile_hostile 1 0.36 0.3567 0.296 0.588

summary(aov_attentive)

#post_pre_Upset

mobOutcomesUpset<- mobOutcomes[c(“human_agent”, “docile_hostile”, “human_agent”, “combined_group”,“post_pre_Upset”, “id”)] df_outliers <- mobOutcomesUpset %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Upset)

#Remove 12 Extreme Outliers (n = 108) mobOutcomesUpset <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers
%in% TRUE),], by = “id”)

aov_upset <- aov(post_pre_Upset ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesUpset)

human_agent 1 6.78 6.778 6.081 0.0153 *

docile_hostile 1 1.31 1.315 1.180 0.2800

human_agent:docile_hostile 1 0.71 0.714 0.641 0.4253

summary(aov_upset)

#post_pre_Inspired

mobOutcomesInspired<- mobOutcomes[c(“human_agent”, “docile_hostile”, “human_agent”, “combined_group”,“post_pre_Inspired”, “id”)] df_outliers <- mobOutcomesInspired %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Inspired)

#Remove 49 Extreme Outliers (n = 71) mobOutcomesInspired <- mobOutcomes %>% anti_join(df_outliers[which(df_outlie
%in% TRUE),], by = “id”)

aov_inspired <- aov(post_pre_Inspired ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesInspired)

human_agent 1 0 0 NaN NaN

docile_hostile 1 0 0 NaN NaN

human_agent:docile_hostile 1 0 0 NaN NaN

summary(aov_inspired)

#post_pre_Alert

mobOutcomesAlert<- mobOutcomes[c("human_agent", "docile_hostile", "human_agent", "combined_group","post_pre_Alert", "id")] df_outliers <- mobOutcomesAlert %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Alert)

#Remove 15 Extreme Outliers (n = 105) mobOutcomesAlert <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers %in% TRUE),], by = "id")

aov_alert <- aov(post_pre_Alert ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesAlert)

human_agent 1 0.31 0.3095 0.408 0.525

docile_hostile 1 0.24 0.2413 0.318 0.574

human_agent:docile_hostile 1 0.84 0.8397 1.106 0.295

summary(aov_alert)

#post_prePositiveAffect

mobOutcomesPositive<- mobOutcomes[c("human_agent", "docile_hostile", "combined_group","post_pre_PositiveAffect", "id")] df_outliers <- mobOutcomesPositive %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_PositiveAffect)

#Remove 3 Extreme Outliers (n = 117) mobOutcomesPositive <- mobOutcomesPositive %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aov_positive <- aov(post_pre_PositiveAffect ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesPositive)

#Compute Shapiro-Wilk test of normality #Significant p value shapiro_test(residuals(aov_positive))

human_agent 1 0.53 0.5308 1.592 0.20963

docile_hostile 1 2.59 2.5921 7.774 0.00622 **

human_agent:docile_hostile 1 0.01 0.0064 0.019 0.88980

summary(aov_positive)

Active, Nervous, Negative, Ashamed, Hostile, Afraid, Determined, Attentive, Upset, Inspired, Alert, Positive

p_vector <- c(.0777, .0016, 0.1390, 0.610, 0.155, 0.265, 0.6764, 0.0691, 0.1456, 0.309, 0.430, 0.588, 0.309, 0.430, 0.588, 0.0153, .280, .4253, NaN, NaN, NaN, .393, .141, .938, .525, .574, .295, .323, .435, .736)

#Bonferroni

 # [1] 1.000 0.048 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 0.459 1.000 1.000 NaN NaN NaN 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000

#holm

 # [1] 1.0000 0.0480 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.4437 1.0000 1.0000 NaN NaN NaN 1.00001.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000

p.adjust(p_vector, method = “holm”, n = length(p_vector))

p_vector_subset <- c(0.171, 0.942, 0.221, 0.946, 0.202, 0.202, 0.575505, 0.000292, 0.714861, 0.93888, 0.00013, 0.91877, 0.362463, 0.000491, 0.833266 , 0.4687, 0.0147, 0.5603, .454, .643, .863, .236, .295, .334, .4176, .0366, .1956, .748, .417, .551, .155, .943, .789, .345, .991, .389) p.adjust(p_vector_subset, “BH”, n = length(p_vector_subset))

[1] 0.7080000 0.9730286 0.7080000 0.9730286 0.7080000 0.7080000 0.8632575 0.0052560 0.9730286 0.9730286 0.0046800 0.9730286 0.7912421 0.0058920

[15] 0.9730286 0.8034857 0.1323000 0.8632575 0.8034857 0.9259200 0.9730286 0.7080000 0.7912421 0.7912421 0.7912421 0.2635200 0.7080000 0.9730286

[29] 0.7912421 0.8632575 0.7080000 0.9730286 0.9730286 0.7912421 0.9910000 0.7912421

p.adjust(p_vector_subset, "holm", n = length(p_vector_subset))

1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 0.010220 1.000000 1.000000 0.004680 1.000000 1.000000 0.016694 1.000000 1.000000 0.485100 1.000000

1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000

Perform ANOVA for Behavior Vars boxplots_behaviors_by_gender <- gridOfBoxplots(mobOutcomes, "Behavior by Gender Boxplots", behaviors_flag = TRUE, by_gender = TRUE)

#Identify Outliers mobOutcomesAvgSpeed <- mobOutcomes[c("human_agent", "docile_hostile", "avgSpeed", "id")] df_outliers <- mobOutcomesAvgSpeed %>% group_by(human_agent, docile_hostile) %>% identify_outliers(avgSpeed)

#Remove 0 Extreme Outlier (n = 120) mobOutcomesAvgSpeed <- mobOutcomes %>% anti_join(df_outliers[which(df_outli
%in% TRUE),], by = "id")

aovAvgSpeed <- aov(avgSpeed ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesAvgSpeed) # Df Sum Sq Mean Sq F value Pr(>F) # human_agent 1 0.032 0.03216 0.308 0.580 # docile_hostile 1 0.232 0.23163 2.215 0.139 # human_agent:docile_hostile 1 0.014 0.01435 0.137 0.712 # Residuals 116 12.131 0.10458
summary(aovAvgSpeed)

ggplot(data=mobOutcomes, aes(x=human_agent, y= mobOutcomes[, “avgSpeed”])) + geom_boxplot(color=“black”, fill=“blue”, alpha=0.2, outlier.shape = NA) +

stat_summary(fun=mean, colour=“black”, geom=“text”, aes(label = round(..y.., 2)))

#Identify Outliers mobOutcomesTimeSpent <- mobOutcomes[c(“human_agent”, “docile_hostile”, “timeSpent”, “id”)] df_outliers <- mobOutcomesTimeSpent %>% group_by(human_agent, docile_hostile) %>% identify_outliers(timeSpent)

#Remove 1 Extreme Outlier (n = 119) mobOutcomesTimeSpent <- mobOutcomes %>% anti_join(df_outliers[which(df_outl
%in% TRUE),], by = “id”)

aovTimeSpent <- aov(timeSpent ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesTimeSpent) # Df Sum Sq Mean Sq F value Pr(>F)
# human_agent 1 299 299 0.058 0.8108
# docile_hostile 1 22974 22974 4.425 0.0376 * # human_agent:docile_hostile 1 565 565 0.109 0.7421
# Residuals 115 597028 5192
summary(aovTimeSpent) TukeyHSD(aovTimeSpent)

#Identify Outliers mobOutcomesFightCnt <- mobOutcomes[c(“human_agent”, “docile_hostile”, “fightCnt”, “id”)] df_outliers <- mobOutcomesFightCnt %>% group_by(human_agent, docile_hostile) %>% identify_outliers(fightCnt)

#Remove 2 Extreme Outlier (n = 118) mobOutcomesFightCnt <- mobOutcomes %>% anti_join(df_outliers[which(df_outlie
%in% TRUE),], by = “id”)

aovFightCnt<- aov(fightCnt ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesFightCnt) # Df Sum Sq Mean Sq F value Pr(>F)
# human_agent 1 18.4 18.42 1.576 0.21193
# docile_hostile 1 115.3 115.26 9.862 0.00215 ** # human_agent:docile_hostile 1 19.0 19.03 1.628 0.20451
# Residuals 114 1332.3 11.69
summary(aovFightCnt) TukeyHSD(aovFightCnt)

#Identify Outliers mobOutcomesPunchCnt <- mobOutcomes[c(“human_agent”, “docile_hostile”, “punchCnt”, “id”)] df_outliers <- mobOutcomesPunchCnt %>% group_by(human_agent, docile_hostile) %>% identify_outliers(punchCnt)

#Remove 4 Extreme Outlier (n = 116) mobOutcomesPunchCnt <- mobOutcomes %>% anti_join(df_outliers[which(df_outli
%in% TRUE),], by = “id”)

aovPunchCnt<- aov(punchCnt ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesPunchCnt) # Df Sum Sq Mean Sq F value Pr(>F) # human_agent 1 33314 33314 1.083 0.300 # docile_hostile 1 26142 26142 0.850 0.359 # human_agent:docile_hostile 1 29493 29493 0.959 0.330
# Residuals 112 3445188 30761
summary(aovPunchCnt)

#Identify Outliers mobOutcomesTotalDist <- mobOutcomes[c(“human_agent”, “docile_hostile”, “totalDist”, “id”)] df_outliers <- mobOutcomesTotalDist %>% group_by(human_agent, docile_hostile) %>% identify_outliers(totalDist)

#Remove 7 Extreme Outlier (n = 113) mobOutcomesTotalDist <- mobOutcomes %>% anti_join(df_outliers[which(df_outlie
%in% TRUE),], by = “id”)

aovTotalDist <- aov(totalDist ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesTotalDist) # Df Sum Sq Mean Sq F value Pr(>F)
# human_agent 1 1956 1956 1.810 0.1813
# docile_hostile 1 6362 6362 5.888 0.0169 * # human_agent:docile_hostile 1 0 0 0.000 0.9857
# Residuals 109 117762 1080
summary(aovTotalDist)

#Identify Outliers mobOutcomesCollectedItem <- mobOutcomes[c("human_agent", "docile_hostile", "collectedItemCnt", "id")] df_outliers <- mobOutcomesCollectedItem %>% group_by(human_agent, docile_hostile) %>% identify_outliers(collectedItemCnt)

#Remove 7 Extreme Outlier (n = 113) mobOutcomesCollectedItem <- mobOutcomes %>% anti_join(df_outliers[which(df_c
%in% TRUE),], by = "id")

aovCollectedItem <- aov(collectedItemCnt ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesCollectedItem) # Df Sum Sq Mean Sq F value Pr(>F)
# human_agent 1 0.8 0.76 0.122 0.728
# docile_hostile 1 119.7 119.75 19.205 2.71e-05 *** # human_agent:docile_hostile 1 24.3 24.29 3.895 0.051
.
# Residuals 109 679.7 6.24
summary(aovCollectedItem) TukeyHSD(aovCollectedItem)

#Identify Outliers mobOutcomesTotalItem <- mobOutcomes[c("human_agent", "docile_hostile", "totalItemCnt", "id")] df_outliers <- mobOutcomesTotalItem %>% group_by(human_agent, docile_hostile) %>% identify_outliers(totalItemCnt)

#Remove 1 Extreme Outlier (n = 119) mobOutcomesTotalItem <- mobOutcomes %>% anti_join(df_outliers[which(df_outli
%in% TRUE),], by = "id")

aovTotalItem <- aov(totalItemCnt ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesTotalItem) # Df Sum Sq Mean Sq F value Pr(>F) # human_agent 1 20 20.42 0.241 0.624
# docile_hostile 1 19 19.13 0.226 0.635 # human_agent:docile_hostile 1 28 28.16 0.333 0.565
summary(aovTotalItem) TukeyHSD(aovTotalItem)

#avgSpeed, timeSpent, -fightCnt, punchCnt, totalDist, -collectedItemCnt, totalItemCnt p_vector <- c(.580, .139, .712, .8108, .0376, .7421, .21193, .00215, .20451, .300, .359, .330, .1813, .0169, .9857, .728, .0000271, .051, .624, .635, .565) p.adjust(p_vector, method = "holm", n = length(p_vector)) # [1] 1.0000000 1.0000000 1.0000000 1.0000000 0.6768000 1.0000000 1.0000000 0.0430000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.3211000 1.0000000 # [16] 1.0000000 0.0005691 0.8670000 1.0000000 1.0000000 1.0000000 #BH # 0.8202158 0.4865000 0.8202158 0.8513400 0.1974000 0.8202158 0.4945033 0.0225750 0.4945033 0.6282500 0.6282500 0.6282500 0.4945033 0.1183000 # 0.9857000 0.8202158 0.0005691 0.2142000 0.8202158 0.8202158 0.8202158

#avgSpeed, timeSpent, fightCnt, punchCnt, totalDist, collectedItemCnt, totalItemCnt

Perform ANOVA for Affective changes wrt gender X familiarity boxplots_behaviors_by_gender_familiarity <- gridOfBoxplots(mobOutcomes, "Behaviors by Familiarity Boxplots", behaviors_flag=TRUE, by_familiarity = TRUE)

mobOutcomes$familiarity < -as.factor(mobOutcomes$familiarity) str(mobOutcomes$familiarity)levels_{t}o_{b}in_{l}ower_{e}nd < -c("-3","-2","-1")mobOutcomes$familiarity_binned <- ifelse(mobOutcomes$familiarity$familiarity_binned <- factor(mobOutcomes$familiarity_binned, levels = c("low_familiarity", "high_familiarity"))

make an interaction between two factors

on x axis

mobOutcomes$GenderXFamiliarity < -interaction(mobOutcomes$gender, mobOutcomes$familiarity_binned)

#Identify Outliers Active mobOutcomesActive <- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity","post_pre_Active", "id")] df_outliers <- mobOutcomesActive %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_Active)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesActive <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers %in% TRUE),], by = "id")

aov_active <- aov(post_pre_Active ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesActive)

Df Sum Sq Mean Sq F value Pr(>F)

gender 1 0.00 0.0026 0.002 0.968

familiarity_binned 1 0.01 0.0066 0.004 0.950

gender:familiarity_binned 1 0.27 0.2682 0.164 0.686

Residuals 116 189.31 1.6320

summary(aov_active)

#Identify Outliers post_pre_Nervous

mobOutcomesNervous<- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity","post_pre_Nervous", "id")] df_outliers <- mobOutcomesNervous %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_Nervous)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesNervous <- mobOutcomes %>% anti_join(df_outliers[which(df_outlie %in% TRUE),], by = "id")

aov_nervous <- aov(post_pre_Nervous ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesNervous)

gender 1 0.23 0.2298 0.122 0.727

familiarity_binned 1 0.00 0.0009 0.000 0.983

gender:familiarity_binned 1 0.04 0.0419 0.022 0.881

summary(aov_nervous)

#Identify Outliers post_preNegativeAffect

mobOutcomesNegative<- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity","post_pre_NegativeAffect "id")] df_outliers <- mobOutcomesNegative %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_NegativeAffect)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesNegative <- mobOutcomesNegative %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id") aov_negative <- aov(post_pre_NegativeAffect ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesNegative)

gender 1 0.59 0.5908 0.981 0.324

familiarity_binned 1 1.37 1.3650 2.267 0.135

gender:familiarity_binned 1 1.12 1.1207 1.861 0.175

Residuals 116 64.87 0.559

summary(aov_negative) ##### #post_pre_ashamed

mobOutcomesAshamed<- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity","post_pre_Ashamed", "id")] df_outliers <- mobOutcomesAshamed %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_Ashamed)

#Remove 11 Extreme Outliers (n = 109) mobOutcomesAshamed <- mobOutcomes %>% anti_join(df_outliers[which(df_out %in% TRUE),], by = "id")

aov_ashamed <- aov(post_pre_Ashamed ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesAshamed)

gender 1 3.91 3.907 5.127 0.02560 *

familiarity_binned 1 6.44 6.437 8.448 0.00446 **

gender:familiarity_binned 1 2.97 2.971 3.900 0.05092 .

Residuals 105 80.01 0.762

summary(aov_ashamed) # $gender # diff lwr upr p adj # Male-Female 0.4385727 0.05453599 0.8226094 0.0256048 # # $familiarity_binned # diff lwr upr p adj # high_familiarity-low_familiarity -0.6907865 -1.172016 -0.2095573 0.0053209 TukeyHSD(aov_ashamed)

#post_pre_Hostile

mobOutcomesHostile<- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity","post_pre_Hostile", "id")] df_outliers <- mobOutcomesHostile %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_Hostile)

#Remove 0 Extreme Outliers (n = 120) mobOutcomesHostile <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers
%in% TRUE),], by = "id")

aov_hostile <- aov(post_pre_Hostile ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesHostile)

Df Sum Sq Mean Sq F value Pr(>F)

gender 1 0.32 0.3155 0.169 0.682

familiarity_binned 1 0.01 0.0056 0.003 0.957

gender:familiarity_binned 1 1.25 1.2468 0.666 0.416

summary(aov_hostile)

#post_pre_Afraid mobOutcomesAfraid <- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity","post_pre_Afraid", "id")] df_outliers <- mobOutcomesAfraid %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_Afraid)

#Remove 9 Extreme Outliers (n = 111) mobOutcomesAfraid <- mobOutcomesAfraid %>% anti_join(df_outliers[which(df_o
%in% TRUE),], by = "id")

aov_afraid <- aov(post_pre_Afraid ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesAfraid)

gender 1 0.60 0.6048 0.644 0.4241

familiarity_binned 1 2.71 2.7132 2.888 0.0921 .

gender:familiarity_binned 1 0.07 0.0702 0.075 0.7851

Residuals 107 100.52 0.9395

summary(aov_afraid)

Repeat by Subset candidate variables to test group differences for (based on boxplot visualizations):

Active, Nervous, Negative, Ashamed, Hostile, Afraid, Determined, Attentive, Upset, Inspired, Alert, Positive

#post_pre_Determined mobOutcomesDetermined <- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity", "post_pre_Determined", "id")] df_outliers <- mobOutcomesDetermined %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_Determined)

#Remove 2 Extreme Outliers (n = 118) mobOutcomesDetermined <- mobOutcomesDetermined %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aov_determined <- aov(post_pre_Determined ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesDetermined)

gender 1 0.01 0.0149 0.010 0.921

familiarity_binned 1 0.05 0.0482 0.032 0.859

gender:familiarity_binned 1 0.79 0.7925 0.523 0.471

Residuals 114 172.61 1.5141

summary(aov_determined)

#post_pre_Attentive

mobOutcomesAttentive<- mobOutcomes[c("gender", "familiarity_binned", "GenderXFamiliarity", "post_pre_Attentive", "id")] df_outliers <- mobOutcomesAttentive %>% group_by(gender, familiarity_binned) %>% identify_outliers(post_pre_Attentive)

#Remove 40 Extreme Outliers (n = 80) mobOutcomesAttentive <- mobOutcomes %>% anti_join(df_outliers[which(df_outli
%in% TRUE),], by = "id")

aov_attentive <- aov(post_pre_Attentive ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesAttentive)

human_agent 1 1.26 1.2622 1.047 0.309

docile_hostile 1 0.76 0.7569 0.628 0.430

human_agent:docile_hostile 1 0.36 0.3567 0.296 0.588

summary(aov_attentive)

#post_pre_Upset

mobOutcomesUpset<- mobOutcomes[c("human_agent", "docile_hostile", "human_agent", "combined_group","post_pre_Upset", "id")] df_outliers <- mobOutcomesUpset %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Upset)

#Remove 12 Extreme Outliers (n = 108) mobOutcomesUpset <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers
%in% TRUE),], by = "id")

aov_upset <- aov(post_pre_Upset ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesUpset)

human_agent 1 6.78 6.778 6.081 0.0153 *

docile_hostile 1 1.31 1.315 1.180 0.2800

human_agent:docile_hostile 1 0.71 0.714 0.641 0.4253

summary(aov_upset)

#post_pre_Inspired

mobOutcomesInspired<- mobOutcomes[c("human_agent", "docile_hostile", "human_agent", "combined_group","post_pre_Inspired", "id")] df_outliers <- mobOutcomesInspired %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Inspired)

#Remove 49 Extreme Outliers (n = 71) mobOutcomesInspired <- mobOutcomes %>% anti_join(df_outliers[which(df_outlie %in% TRUE),], by = "id")

aov_inspired <- aov(post_pre_Inspired ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesInspired)

human_agent 1 0 0 NaN NaN

docile_hostile 1 0 0 NaN NaN

human_agent:docile_hostile 1 0 0 NaN NaN

summary(aov_inspired)

#post_pre_Alert

mobOutcomesAlert<- mobOutcomes[c("human_agent", "docile_hostile", "human_agent", "combined_group","post_pre_Alert", "id")] df_outliers <- mobOutcomesAlert %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_Alert)

#Remove 15 Extreme Outliers (n = 105) mobOutcomesAlert <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers %in% TRUE),], by = "id")

aov_alert <- aov(post_pre_Alert ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesAlert)

human_agent 1 0.31 0.3095 0.408 0.525

docile_hostile 1 0.24 0.2413 0.318 0.574

human_agent:docile_hostile 1 0.84 0.8397 1.106 0.295

summary(aov_alert)

#post_prePositiveAffect

mobOutcomesPositive<- mobOutcomes[c("human_agent", "docile_hostile", "combined_group","post_pre_PositiveAffect", "id")] df_outliers <- mobOutcomesPositive %>% group_by(human_agent, docile_hostile) %>% identify_outliers(post_pre_PositiveAffect)

#Remove 3 Extreme Outliers (n = 117) mobOutcomesPositive <- mobOutcomesPositive %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aov_positive <- aov(post_pre_PositiveAffect ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomesPositive)

#Compute Shapiro-Wilk test of normality #Significant p value shapiro_test(residuals(aov_positive))

human_agent 1 0.53 0.5308 1.592 0.20963

docile_hostile 1 2.59 2.5921 7.774 0.00622 **

human_agent:docile_hostile 1 0.01 0.0064 0.019 0.88980

summary(aov_positive)

Active, Nervous, Negative, Ashamed, Hostile, Afraid, Determined, Attentive, Upset, Inspired, Alert, Positive

p_vector <- c(.0777, .0016, 0.1390, 0.610, 0.155, 0.265, 0.6764, 0.0691, 0.1456, 0.309, 0.430, 0.588, 0.309, 0.430, 0.588, 0.0153, .280, .4253, NaN, NaN, NaN, .393, .141, .938, .525, .574, .295, .323, .435, .736)

#Bonferroni

 # [1] 1.000 0.048 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 0.459 1.000 1.000 NaN NaN NaN 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000

#holm

 # [1] 1.0000 0.0480 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.4437 1.0000 1.0000 NaN NaN NaN 1.00001.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000

p.adjust(p_vector, method = “holm”, n = length(p_vector))

p_vector_subset <- c(0.171, 0.942, 0.221, 0.946, 0.202, 0.202, 0.575505, 0.000292, 0.714861, 0.93888, 0.00013, 0.91877, 0.362463, 0.000491, 0.833266 , 0.4687, 0.0147, 0.5603, .454, .643, .863, .236, .295, .334, .4176, .0366, .1956, .748, .417, .551, .155, .943, .789, .345, .991, .389) p.adjust(p_vector_subset, “BH”, n = length(p_vector_subset))

[1] 0.7080000 0.9730286 0.7080000 0.9730286 0.7080000 0.7080000 0.8632575 0.0052560 0.9730286 0.9730286 0.0046800 0.9730286 0.7912421 0.0058920

[15] 0.9730286 0.8034857 0.1323000 0.8632575 0.8034857 0.9259200 0.9730286 0.7080000 0.7912421 0.7912421 0.7912421 0.2635200 0.7080000 0.9730286

[29] 0.7912421 0.8632575 0.7080000 0.9730286 0.9730286 0.7912421 0.9910000 0.7912421

Perform ANOVA for Behaviors wrt gender X familiarity boxplots_behaviors_by_gender_familiarity <- gridOfBoxplots(mobOutcomes, "Behaviors by Familiarity Boxplots", behaviors_flag=TRUE, by_familiarity = TRUE)

#Identify Outliers fightCnt mobOutcomesFightFamiliarity <- mobOutcomes[c("familiarity_binned", "fightCnt", "id")] df_outliers <- mobOutcomesFightFamiliarity %>% group_by(familiarity_binned) %>% identify_outliers(fightCnt)

#Remove 1 Extreme Outlier (n = 119) mobOutcomesFightFamiliarity <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aovFightFamiliarity <- aov(fightCnt ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesFightFamiliarity) # gender 1 0.4 0.41 0.032 0.85864
# familiarity_binned 1 136.4 136.38 10.641 0.00146 ** # gender:familiarity_binned 1 1.0 1.03 0.081 0.77707
# Residuals 115 1473.9 12.82
summary(aovFightFamiliarity) # $familiarity_binned # diff lwr upr p adj # (1,2,3)-(-3,-2,-1) -3.171312 -5.129895 -1.212729 0.0017357 TukeyHSD(aovFightFamiliarity)

#Identify Outliers punchCnt mobOutcomesPunchFamiliarity <- mobOutcomes[c("familiarity_binned", "punchCnt", "id")] df_outliers <- mobOutcomesPunchFamiliarity %>% group_by(familiarity_binned) %>% identify_outliers(punchCnt)

#Remove 6 Extreme Outlier (n = 114) mobOutcomesPunchFamiliarity <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aovPunchFamiliarity <- aov(punchCnt ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesPunchFamiliarity) # Df Sum Sq Mean Sq F value Pr(>F) # gender 1 43547 43547 2.152 0.145 # familiarity_binned 1 13049 13049 0.645 0.424 # gender:familiarity_binned 1 12808 12808 0.633 0.428 # Residuals 110 2226450 20240
summary(aovPunchFamiliarity)

#Identify Outliers avgSpeed mobOutcomesSpeedFamiliarity <- mobOutcomes[c("familiarity_binned", "avgSpeed", "id")] df_outliers <- mobOutcomesSpeedFamiliarity %>% group_by(familiarity_binned) %>% identify_outliers(avgSpeed)

#Remove 0 Extreme Outlier (n = 120) mobOutcomesSpeedFamiliarity <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aovSpeedFamiliarity <- aov(avgSpeed ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesSpeedFamiliarity) # Df Sum Sq Mean Sq F value Pr(>F)
# gender 1 0.122 0.1223 1.254 0.26517
# familiarity_binned 1 0.690 0.6896 7.070 0.00895 ** # gender:familiarity_binned 1 0.282 0.2819 2.890 0.09179 . # Residuals 116 11.315 0.0975

summary(aovSpeedFamiliarity) # $familiarity_binned # diff lwr upr p adj # (1,2,3)-(-3,-2,-1) 0.2200336 0.05391286 0.3861544 0.0098752 TukeyHSD(aovSpeedFamiliarity)

#Identify Outliers timeSpent mobOutcomesTimeFamiliarity <- mobOutcomes[c("familiarity_binned", "timeSpent", "id")] df_outliers <- mobOutcomesTimeFamiliarity %>% group_by(familiarity_binned) %>% identify_outliers(timeSpent)

#Remove 0 Extreme Outlier (n = 120) mobOutcomesTimeFamiliarity <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aovTimeFamiliarity <- aov(timeSpent ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesTimeFamiliarity) # Df Sum Sq Mean Sq F value Pr(>F)
# gender 1 4979 4979 0.721 0.397
# familiarity_binned 1 135018 135018 19.560 2.21e-05 *** # gender:familiarity_binned 1 15464 15464 2.240 0.137
# Residuals 116 800702 6903
summary(aovTimeFamiliarity) # $familiarity_binned # diff lwr upr p adj # (1,2,3)-(-3,-2,-1) -97.35925 -141.5491 -53.16935 2.79e-05 TukeyHSD(aovTimeFamiliarity)

#Identify Outliers totalDist mobOutcomesDistFamiliarity <- mobOutcomes[c("familiarity_binned", "totalDist", "id")] df_outliers <- mobOutcomesDistFamiliarity %>% group_by(familiarity_binned) %>% identify_outliers(totalDist)

#Remove 7 Extreme Outlier (n = 113) mobOutcomesDistFamiliarity <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aovDistFamiliarity <- aov(totalDist ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesDistFamiliarity) # Df Sum Sq Mean Sq F value Pr(>F) # gender 1 789 789.3 0.930 0.337 # familiarity_binned 1 315 315.0 0.371 0.544 # gender:familiarity_binned 1 4 3.6 0.004 0.948 # Residuals 109 92551 849.1
summary(aovDistFamiliarity)

#Identify Outliers collectedItemCnt mobOutcomesCollectedItemFamiliarity <- mobOutcomes[c("familiarity_binned", "collectedItemCnt", "id")] df_outliers <- mobOutcomesCollectedItemFamiliarity %>% group_by(familiarity_binned) %>% identify_outliers(collectedItemCnt)

#Remove 3 Extreme Outlier (n = 117) mobOutcomesCollectedItemFamiliarity <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aovCollectedItemFamiliarity <- aov(collectedItemCnt ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesCollectedItemFamiliarity) # Df Sum Sq Mean Sq F value Pr(>F)
# gender 1 22.7 22.73 3.338 0.0703 . # familiarity_binned 1 42.4 42.43 6.229 0.0140 * # gender:familiarity_binned 1 2.4 2.37 0.347 0.5568
# Residuals 113 769.7 6.81
summary(aovCollectedItemFamiliarity) # $familiarity_binned # diff lwr upr p adj # (1,2,3)-(-3,-2,-1) 1.901042 0.3799676 3.422116 0.0147658 TukeyHSD(aovCollectedItemFamiliarity)

#Identify Outliers totalItemCnt mobOutcomesTotalItemFamiliarity <- mobOutcomes[c("familiarity_binned", "totalItemCnt", "id")] df_outliers <- mobOutcomesCollectedItemFamiliarity %>% group_by(familiarity_binned) %>% identify_outliers(totalItemCnt)

#Remove 3 Extreme Outlier (n = 117) mobOutcomesTotalItemFamiliarity <- mobOutcomes %>% anti_join(df_outliers[which(df_outliers$is.extreme %in% TRUE),], by = "id")

aovTotalItemFamiliarity <- aov(totalItemCnt ~ gender + familiarity_binned + gender:familiarity_binned, data = mobOutcomesTotalItemFamiliarity) # Df Sum Sq Mean Sq F value Pr(>F) # gender 1 25 25.38 0.291 0.590 # familiarity_binned 1 137 137.30 1.576 0.212 # gender:familiarity_binned 1 4 3.76 0.043 0.836
# Residuals 115 10020 87.13
summary(aovTotalItemFamiliarity) TukeyHSD(aovTotalItemFamiliarity)

#fight, punch, avgSPeed, timeSpent, totalDist, collectedItem, totalItemCnt p_vector <- c(0.85864, 0.00146, 0.7707, 0.145, 0.424, 0.428, 0.26517, 0.00895, 0.09179, 0.397, 0.0000221, 0.137, 0.337, 0.544, 0.948, 0.0703, 0.0140, 0.5568, 0.590, 0.212, 0.836) p.adjust(p_vector, “holm”, n = length(p_vector)) # 1.0000000 0.0292000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.1700500 1.0000000 1.0000000 0.0004641 1.0000000 1.0000000 1.0000000 # 1.0000000 1.0000000 0.2520000 1.0000000 1.0000000 1.0000000 1.0000000 p.adjust(p_vector, “BH”, n = length(p_vector)) # 0.9015720 0.0153300 0.8991500 0.3806250 0.6420000 0.6420000 0.5568570 0.0626500 0.3212650 0.6420000 0.0004641 0.3806250 0.6420000 0.7288235 # 0.9480000 0.2952600 0.0735000 0.7288235 0.7288235 0.4946667 0.9015720

#boxplots for sig. response vars, one versus multiple

Generate Grid of Boxplots

mobOutcomes %>% group_by(docile_hostile) %>% get_summary_stats(post_pre_Active, type = “mean_sd”)

data$human_agent <- factor(data$human_agent , levels=c(“Human”, “Agent”))

one_plot <- ggplot(data=mobOutcomes, aes(x=docile_hostile, y= mobOutcomes[, 59])) + geom_boxplot(color=“black”, fill=“blue”, alpha=0.2, outlier.shape = NA) + ylab(colnames(mobOutcomes)[59]) + xlab(NULL) + stat_summary(fun=mean, colour=“black”, geom=“point”, shape=18, size=3, show.legend=FALSE) + theme(axis.text.x = element_text(size = 15), legend.position=“none”) + ggplot2::ylim(-3, 3)

one_plot <- ggdraw(one_plot) + ggtitle(“Significant Emotion Group Differences Boxplots”) + theme(plot.title = element_text(hjust = 0.5)) print(one_plot)

#Non-parametric two way ANOVA alternative ####################

Perform the ART ANOVA

artTimeSpent <- art(timeSpent ~ human_agent * docile_hostile, data = mobOutcomes) # Print the test results # 1 human_agent 1 116 0.002763 0.9581697
# 2 docile_hostile 1 116 9.401034 0.0026985 ** # 3 human_agent:docile_hostile 1 116 0.547492 0.4608392
anova(artTimeSpent)

Perform the ART ANOVA

artPunchCnt <- art(punchCnt ~ human_agent * docile_hostile, data = mobOutcomes) # Print the test results # 1 human_agent 1 116 3.0420 0.083785 . # 2 docile_hostile 1 116 18.9281 2.9354e-05 *** # 3 human_agent:docile_hostile 1 116 1.3125 0.254292
anova(artPunchCnt)

Perform the ART ANOVA

artPunchRate <- art(punchRate ~ human_agent * docile_hostile, data = mobOutcomes)

Print the test results

1 human_agent 1 116 2.45459 0.11991

2 docile_hostile 1 116 0.12982 0.71928

3 human_agent:docile_hostile 1 116 0.13589 0.71307

anova(artPunchRate)

Perform the ART ANOVA

artTotalDist <- art(totalDist ~ human_agent * docile_hostile, data = mobOutcomes) # 1 human_agent 1 116 2.2367 0.137481
# 2 docile_hostile 1 116 6.5171 0.011982 * # 3 human_agent:docile_hostile 1 116 2.3118 0.131113
anova(artTotalDist)

Perform ANOVA #Identify Outliers mobOutcomesAvgSpeed <- mobOutcomes[c("human_agent", "docile_hostile", "avgSpeed", "id")] df_outliers <- mobOutcomesAvgSpeed %>% group_by(human_agent, docile_hostile) %>% identify_outliers(avgSpeed)

#Remove 1 Extreme Outlier (n = 120) mobOutcomesAvgSpeed <- mobOutcomes %>% anti_join(df_outliers[which(df_outli
%in% TRUE),], by = "id")

aovAvgSpeed <- aov(avgSpeed ~ human_agent + docile_hostile + human_agent:docile_hostile, data = mobOutcomes) # human_agent 1 0.522 0.5223 4.450 0.037 * # docile_hostile 1 0.393 0.3926 3.345 0.070 .
# human_agent:docile_hostile 1 0.009 0.0095 0.081 0.777
summary(aovAvgSpeed)

ggplot(data=mobOutcomes, aes(x=human_agent, y= mobOutcomes[, "avgSpeed"])) + geom_boxplot(color="black", fill="blue", alpha=0.2, outlier.shape = NA) + stat_summary(fun=mean, colour="black", geom="text", aes(label = round(..y.., 2)))

p_vector_art <- c(.9581697, .0026985, .4608392, .083785, .000029354, 0.254292, .11991, .71928, .71307, .137481, .131113, .037, .070, .777)

[1] 1.00000000 0.03777900 1.00000000 0.83785000 0.00044031 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 0.15576600 1.00000000

[13] 0.44400000 0.77000000 1.00000000

p.adjust(p_vector_art, method = "holm", n=length(p_vector_art))

list_multi_plot <- lapply(c(8,10), function(col) {

```
ggplot(data=mobOutcomes, aes(x=docile_hostile, y= mobOutcomes[, col])) + geom_boxplot(colo
  stat_summary(fun=mean, colour="black", geom="point",
```

```
                    shape=18, size=3, show.legend=FALSE) +
            theme(axis.text.x = element_text(angle = 25, hjust = 1), legend.position="none") + ggplo
        })
```

multi_plot <- cowplot::plot_grid(plotlist = list_multi_plot) multi_plot <- ggdraw(multi_plot) + ggtitle("Significant Behavior Group Differences Boxplots") + theme(plot.title = element_text(hjust = 0.5)) print(multi_plot)

##Visualization## #################

#Get means and std deviations by group, post_pre_PositiveAffect means_pos_affect <- aggregate(post_pre_PositiveAffect ~ human_agent, mobOutcomesCombined, mean) h_a_pos_affect <- ggplot(data=mobOutcomesCombined, aes(x=human_agent, y=post_pre_PositiveAffect, fill=post_pre_PositiveAffect)) + geom_boxplot() + stat_summary(fun=mean, colour="gray", geom="point", shape=18, size=3, show.legend=FALSE) + geom_text(data = means_pos_affect, aes(label =sprintf("%0.2f", round(post_pre_PositiveAffect, digits = 3)), y = post_pre_PositiveAffect+ .14)) + theme(legend.position="none")

plot(h_a_pos_affect)

means_pos_affect_2 <- aggregate(post_pre_PositiveAffect ~ docile_hostile, mobOutcomesCombined, mean) d_h_pos_affect <- ggplot(data=mobOutcomesCombined, aes(x=docile_hostile, y=post_pre_PositiveAffect, fill=post_pre_PositiveAffect)) + geom_boxplot() + stat_summary(fun=mean, colour="gray", geom="point", shape=18, size=3, show.legend=FALSE) + geom_text(data = means_pos_affect_2, aes(label =sprintf("%0.2f", round(post_pre_PositiveAffect, digits = 3)), y = post_pre_PositiveAffect+ .14)) + theme(legend.position="none")

means_pos_affect_3 <- aggregate(post_pre_PositiveAffect ~ group, mobOutcomesCombined, mean) g_pos_affect <- ggplot(data=mobOutcomesCombined, aes(x=group, y=post_pre_PositiveAffect, fill=post_pre_PositiveAffect)) + geom_boxplot() + stat_summary(fun=mean, colour="gray", geom="point", shape=18, size=3, show.legend=FALSE) + geom_text(data = means_pos_affect_3, aes(label =sprintf("%0.2f", round(post_pre_PositiveAffect, digits = 3)), y = post_pre_PositiveAffect+ .14)) + theme(legend.position="none")

ggarrange(h_a_pos_affect, d_h_pos_affect, g_pos_affect, ncol = 2, nrow = 2)

mobOutcomesCombined %>% group_by(group) %>% get_summary_stats(totalItemCnt, type = "mean_sd") #Plot them totalItemCnt = ggboxplot(mobOutcomesCombined, x = "group", y = "post_pre_PositiveAffect") ggsave(paste0("post_pre_PositiveAffect",".png"), totalItemCnt)

#Get means and std deviations by group, post_pre_NegativeAffect mobOutcomesCombined %>% group_by(group) %>% get_summary_stats(totalItemCnt, type = "mean_sd") #Plot them totalItemCnt = ggboxplot(mobOutcomesCombined, x = "group", y = "post_pre_NegativeAffect") ggsave(paste0("post_pre_NegativeAffect",".png"), totalItemCnt)

#Get means and std deviations by group, fightcount mobOutcomesCombined %>% group_by(group) %>% get_summary_stats(fightCnt, type = "mean_sd") #Plot them fightCount = ggboxplot(mobOutcomes, x = "group", y = "fightCnt") ggsave(paste0("Fight_Count_By_Group",".png"), fightCount)

#Get means and std deviations by group, punchcount mobOutcomesCombined %>% group_by(group) %>% get_summary_stats(fightCnt, type = "mean_sd") #Plot them punchCount = ggboxplot(mobOutcomesCombined, x = "group", y = "punchCnt") ggsave(paste0("Punch_Count_By_Group",".png"), punchCount)

#Get means and std deviations by group, avgSpeed mobOutcomesCombined %>% group_by(group) %>% get_summary_stats(fightCnt, type = "mean_sd") #Plot them punchCount = ggboxplot(mobOutcomesCombined, x = "group", y = "avgSpeed") ggsave(paste0("Avg_Speed_By_Group",".png"), punchCount)

#Get means and std deviations by group, totalDist mobOutcomesCombined %>% group_by(group) %>% get_summary_stats(fightCnt, type = "mean_sd") #Plot them punchCount = ggboxplot(mobOutcomesCombined, x = "group", y = "totalDist") ggsave(paste0("Total_Dist_By_Group",".png"), punchCount)

#Get means and std deviations by group, collectedItemCnt mobOutcomes %>% group_by(group) %>% get_summary_stats(collectedItemCnt, type = “mean_sd”) #Plot them collectedItemCnt = ggboxplot(mobOutcomesCombined, x = “group”, y = “collectedItemCnt”) ggsave(paste0(“Collected_Item_Count_By_Group”,“.p collectedItemCnt)

#Get means and std deviations by group, totalItemCnt mobOutcomesCombined %>% group_by(group) %>% get_summary_stats(totalItemCnt, type = “mean_sd”) #Plot them totalItemCnt = ggboxplot(mobOutcomesCombined, x = “group”, y = “totalItemCnt”) ggsave(paste0(“totalItemCnt”,“.png”), totalItemCnt)