

NV Analysis

Dani Cosme & Niko Verhoeven

2018-08-13

Contents

load libraries	1
set color palettes	2
EDE-QS	2
get survey data	2
score	3
describe	3
task	5
load data	5
tidy data	6
calculate reg. success	6
describe reg. success	6
plot reg. success	9
calculate mean effort	10
describe effort	11
correlations	12
reg. success	12
effort	13
plot correlations	14
reg. success	14
effort	16

load libraries

```
if (!require(tidyverse)) {  
  install.packages('tidyverse')  
}  
  
if (!require(knitr)) {  
  install.packages('knitr')  
}  
  
if (!require(wesanderson)) {  
  install.packages('wesanderson')  
}  
  
if (!require(devtools)) {  
  install.packages('devtools')  
}
```

```
if (!require(scorequaltrics)) {
  devtools::install_github('jflournoy/qualtrics')
}
```

set color palettes

```
palettegreen = "#93c47d"
palette3 = c("#93C47D", "#2E6171", "#333333")
```

EDE-QS

get survey data

```
# define variables
cred_file_location = '~/credentials.yaml.DEFAULT'
sid_column_name = '(subjectID)'
survey_name_filter = 'Freshman Project T1$'
sid_pattern = 'FP[0-9]{3}'
exclude_sid = c('FP999','999') # subject IDs to exclude
identifiableData = c('IPAddress') # exclude when printing duplicates

# load credential file
credentials = scorequaltrics::creds_from_file(cred_file_location)

# filter
surveysAvail = scorequaltrics::get_surveys(credentials)
surveysFiltered = filter(surveysAvail, grepl(survey_name_filter, SurveyName))

# get survey
surveys = scorequaltrics::get_survey_responses(credentials,
                                              surveyid = surveysFiltered$SurveyID[[1]])

# tidy surveys
surveys1 = surveys %>%
  # select responses matching subject ID pattern
  filter(grepl(sid_pattern, subjectID)) %>%
  # exclude test responses
  filter(!subjectID %in% exclude_sid)

# check number of observations
surveys1 %>%
  group_by(subjectID) %>%
  summarize(n = n()) %>%
  arrange(desc(n))

## # A tibble: 54 x 2
##   subjectID      n
##   <chr>      <int>
## 1 FP001          1
```

```
## 2 FP002      1
## 3 FP003      1
## 4 FP004      1
## 5 FP005      1
## 6 FP006      1
## 7 FP007      1
## 8 FP008      1
## 9 FP009      1
## 10 FP010     1
## # ... with 44 more rows

# select relevant columns
EDEQS = surveys1 %>%
  select(subjectID, SEX, starts_with("GENDER"), starts_with("EDE")) %>%
  # score EDE-QS
  mutate_at(vars(starts_with("EDE")), as.numeric) # convert to integer
```

score

```
# calculate mean across all items
total = EDEQS %>%
  gather(EDEQS, value, starts_with("EDE")) %>%
  group_by(subjectID) %>%
  summarize(total = mean(value, na.rm = TRUE))

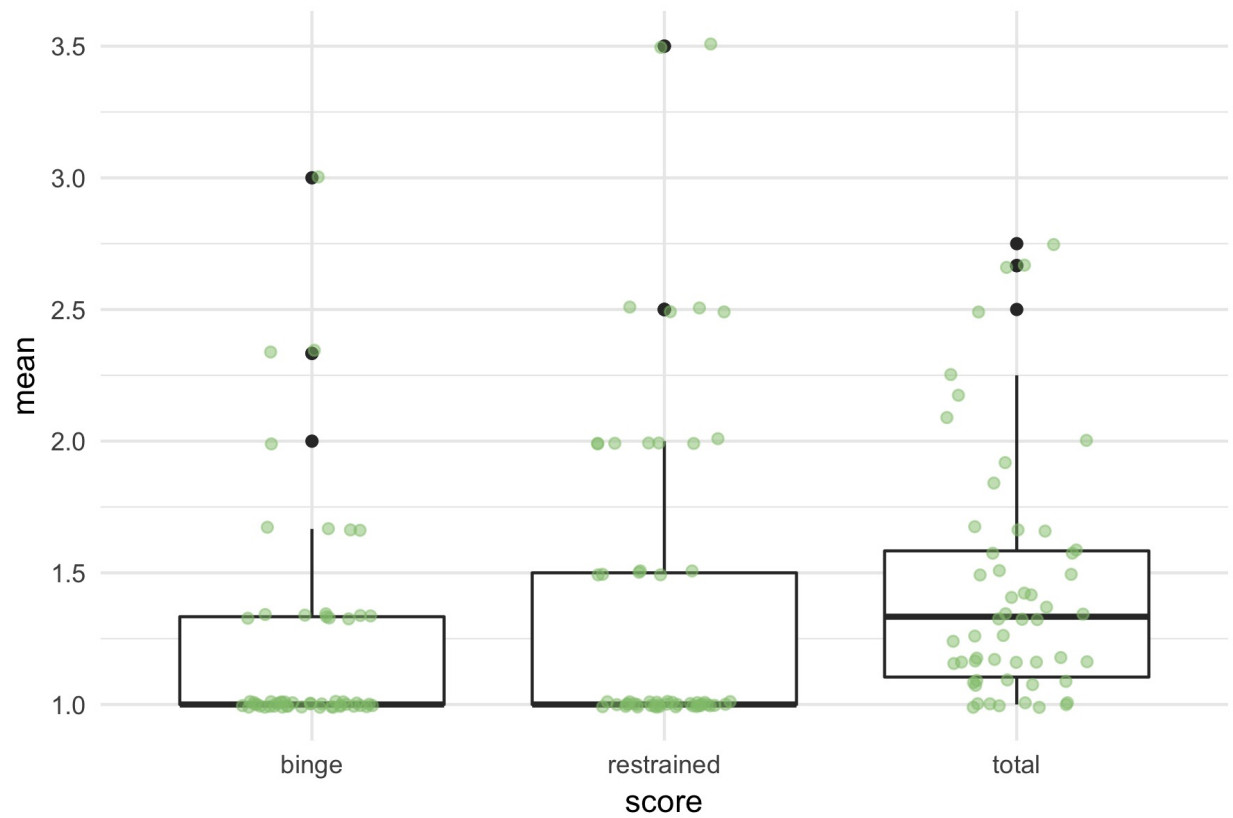
# calculate mean for restraint items
restrained = EDEQS %>%
  gather(EDEQS, value, starts_with("EDE")) %>%
  filter(EDEQS %in% c("EDE_QS_1", "EDE_QS_2")) %>%
  group_by(subjectID) %>%
  summarize(restrained = mean(value, na.rm = TRUE))

# calculate mean for binge items
binge = EDEQS %>%
  gather(EDEQS, value, starts_with("EDE")) %>%
  filter(EDEQS %in% c("EDE_QS_3", "EDE_QS_9", "EDE_QS_10")) %>%
  group_by(subjectID) %>%
  summarize(binge = mean(value, na.rm = TRUE))

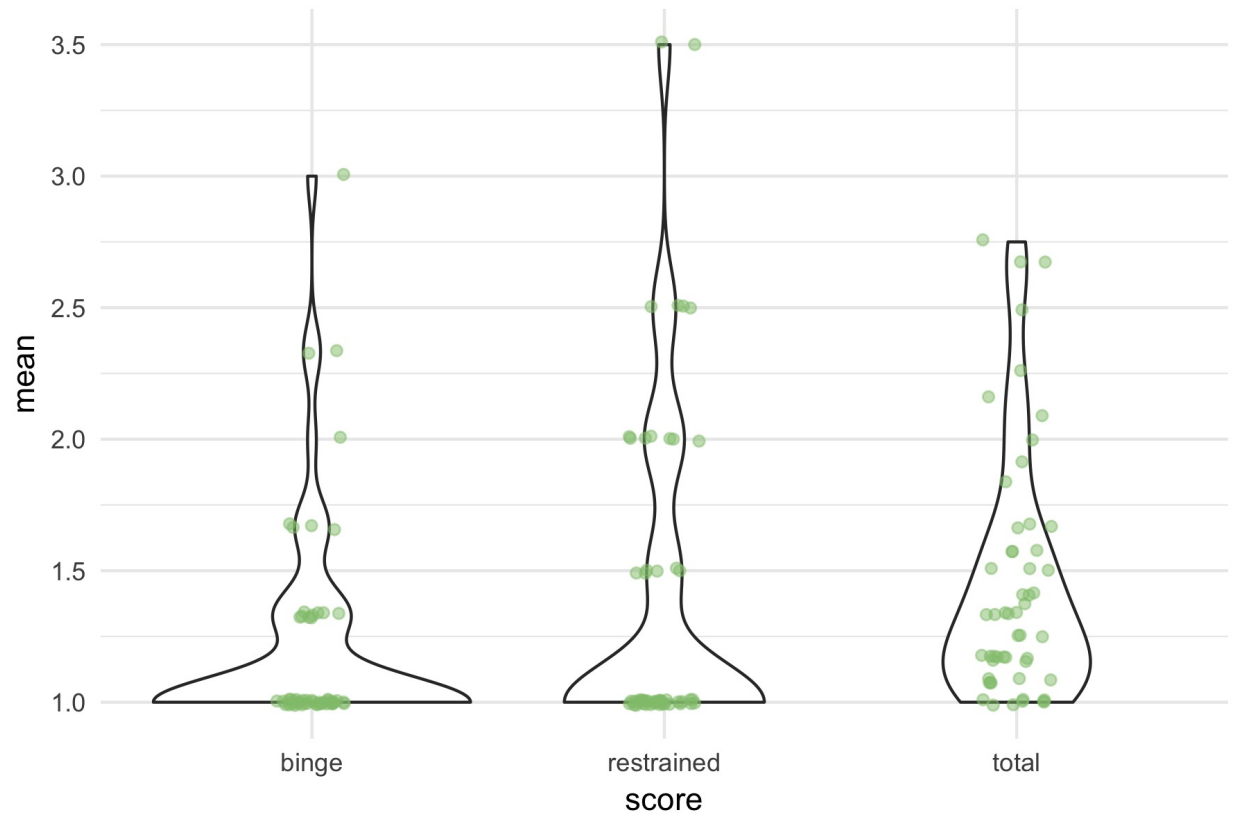
EDEQS = left_join(EDEQS, total, by = "subjectID") %>%
  left_join(., restrained, by = "subjectID") %>%
  left_join(., binge, by = "subjectID")
```

describe

```
EDEQS %>%
  gather(score, mean, total, restrained, binge) %>%
  ggplot(aes(score, mean)) +
  geom_boxplot() +
  geom_jitter(width = .2, alpha = .5, color = paletttegreen) +
  theme_minimal()
```



```
EDEQS %>%
  gather(score, mean, total, restrained, binge) %>%
  ggplot(aes(score, mean)) +
    geom_violin() +
    geom_jitter(width = .1, alpha = .5, color = palettegreen) +
    theme_minimal()
```



task

load data

```
# define variables and paths
sub_dir = "~/Dropbox (PfeiBer Lab)/FreshmanProject/Tasks/ROC-C/output/FP/"
sub_pattern = "FP[0-9]{3}"
subjects = list.files(sub_dir, pattern = sub_pattern)
runs = c("run1", "run2", "run3")

# initialize data frame
data = data.frame()

# loop through subjects and load data
for (sub in subjects) {
  for (run in runs) {
    file = paste0(sub_dir, sub, '/', sub, '_', run, '.csv')
    tmp = tryCatch(read.csv(file, stringsAsFactors = FALSE) %>%
      mutate(subjectID = sub,
              run = run,
              respCue = as.integer(as.character(respCue)),
              respRating = as.integer(as.character(respRating)),
              respEffort = as.integer(as.character(respEffort))), error = function(e) NULL)
    data = bind_rows(data, tmp)
  }
}
```

```

    rm(tmp)
  }
}

```

tidy data

```

task = data %>%
  # exclude FP001 and FP999
  filter(!subjectID %in% c("FP001", "FP999")) %>%
  # recode values
  mutate(rtCue = ifelse(rtCue == "NaN", NA, rtCue), # NaN as NA
         rtRating = ifelse(rtRating == "NaN", NA, rtRating), # NaN as NA
         rtEffort = ifelse(rtEffort == "NaN", NA, rtEffort), # NaN as NA
         action = ifelse(respCue == 6, "look", # cue button presses
                        ifelse(respCue == 7, "regulate", NA)),
         action = ifelse(cond == "LOOK" & is.na(respCue), "look", # missing cue button presses
                        ifelse(cond == "REGULATE" & is.na(respCue), "regulate", action)),
         action = as.factor(action), # change to factor
         choice = ifelse(cond %in% c("LOOK", "REGULATE"), "no", # choice values
                        ifelse(cond == "CHOOSE", "yes", NA)),
         choice = as.factor(choice), # change to factor
         respRating = respRating - 5, # recode button box presses to 1-4 scale
         respRating = as.integer(respRating), # change to integer
         respEffort = respEffort - 5, # recode button box presses to 1-4 scale
         respEffort = as.integer(respEffort)) %>% # change to integer
  # add trial number
  group_by(subjectID) %>%
  mutate(trial = row_number()) %>%
  # reorder columns
  select(subjectID, run, trial, action, choice, cond, respCue, everything()) %>%
  ungroup()

```

calculate reg. success

```

reg.success = task %>%
  # remove missing data
  filter(!is.na(action)) %>%
  # group by subject and calculate mean
  group_by(subjectID, action) %>%
  summarize(mean = mean(respRating, na.rm = TRUE)) %>%
  # calculate regulation success
  spread(action, mean) %>%
  mutate(reg.success = look - regulate)

```

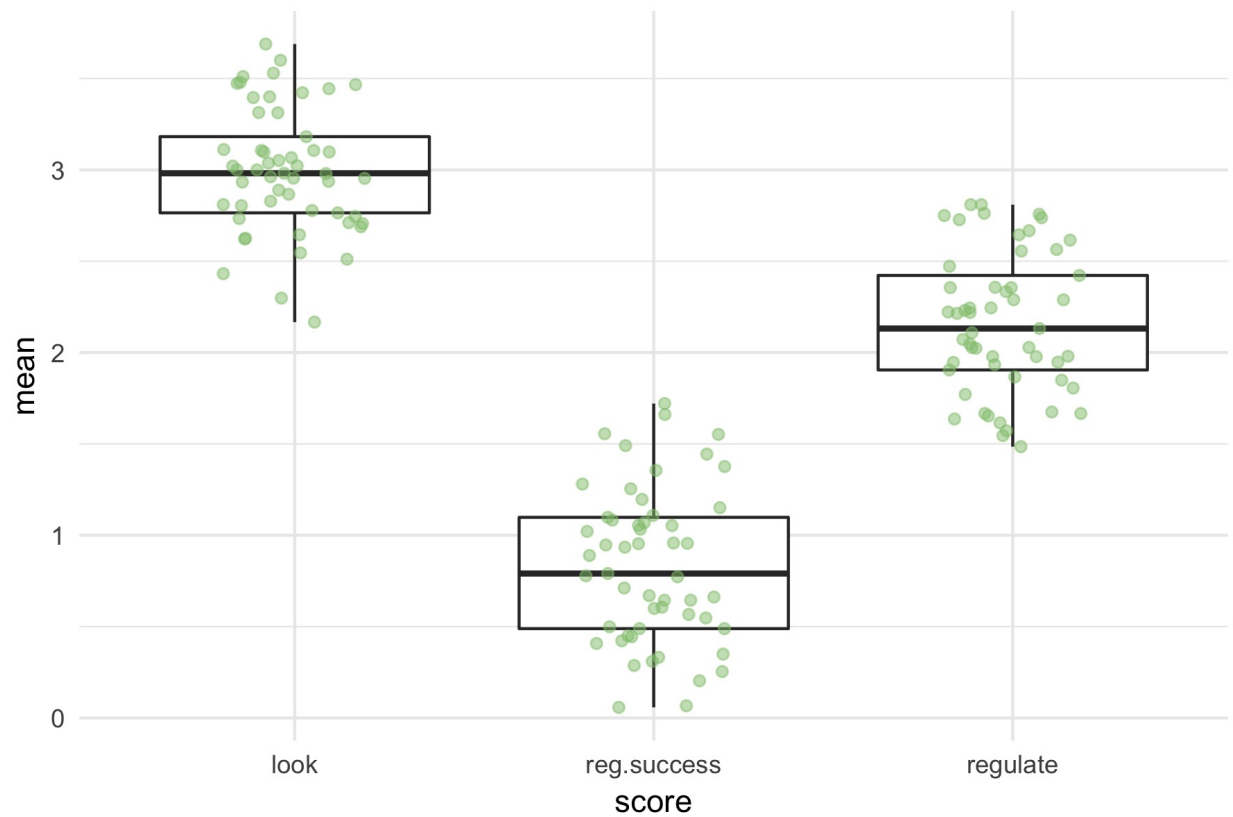
describe reg. success

```

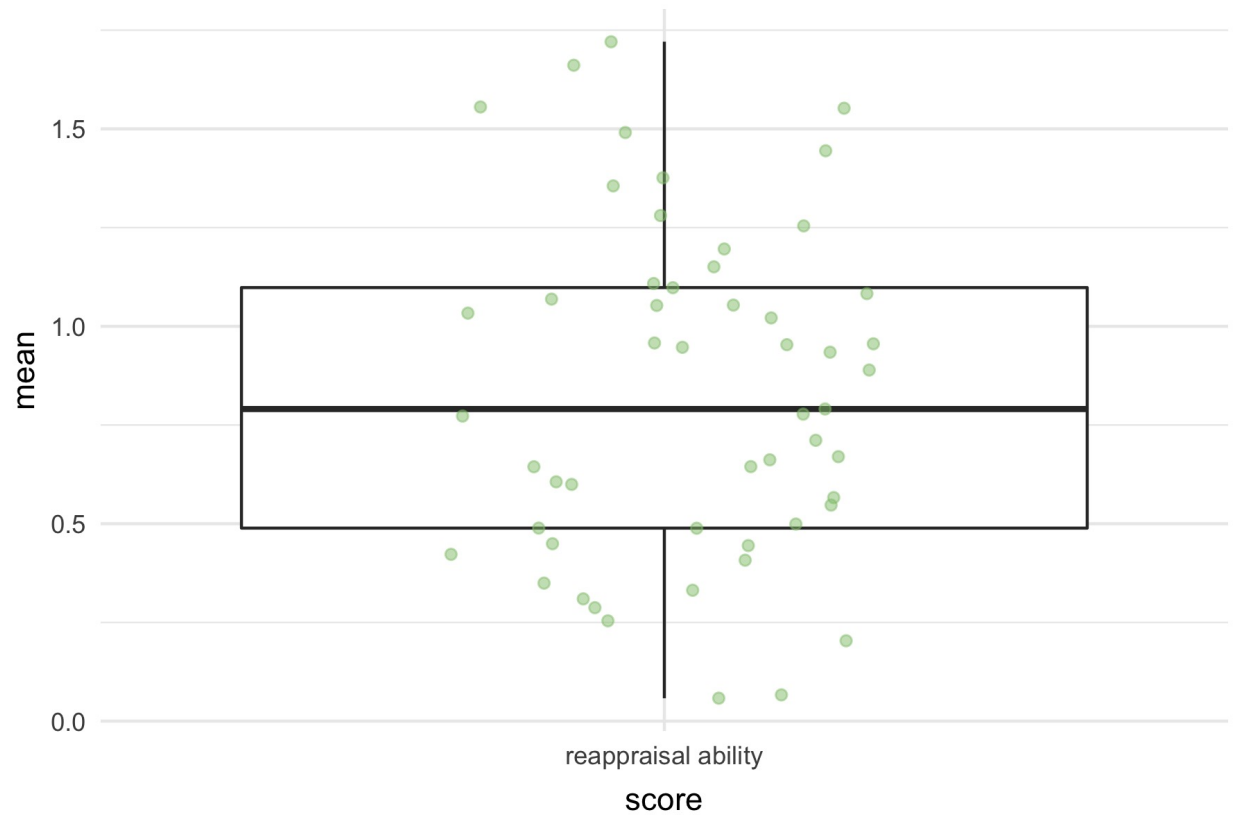
reg.success %>%
  gather(score, mean, look, regulate, reg.success) %>%
  ggplot(aes(score, mean)) +

```

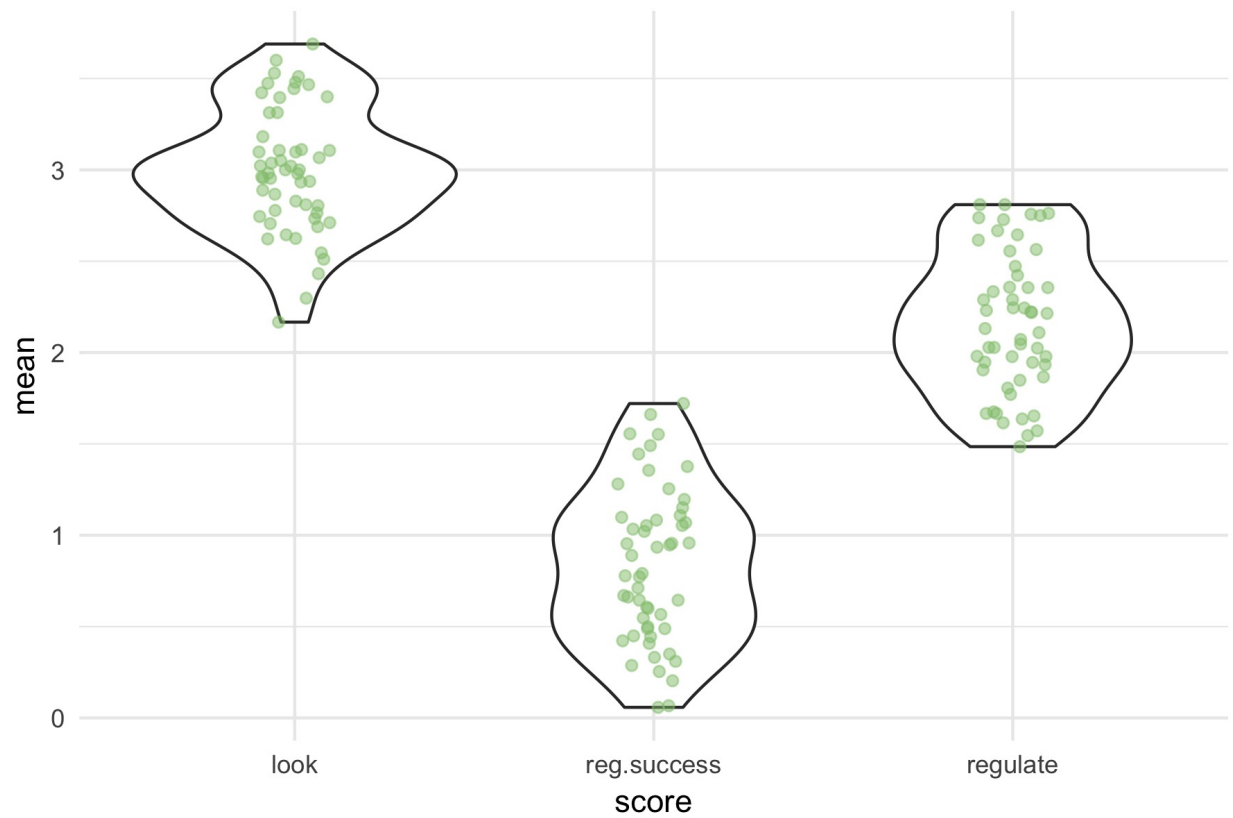
```
geom_boxplot() +
geom_jitter(width = .2, alpha = .5, color = palettegreen) +
theme_minimal()
```



```
reg.success %>%
gather(score, mean, look, regulate, reg.success) %>%
filter(score == "reg.success") %>%
mutate(score = ifelse(score == "reg.success", "reappraisal ability", score)) %>%
ggplot(aes(score, mean)) +
  geom_boxplot() +
  geom_jitter(width = .2, alpha = .5, color = palettegreen) +
  theme_minimal()
```

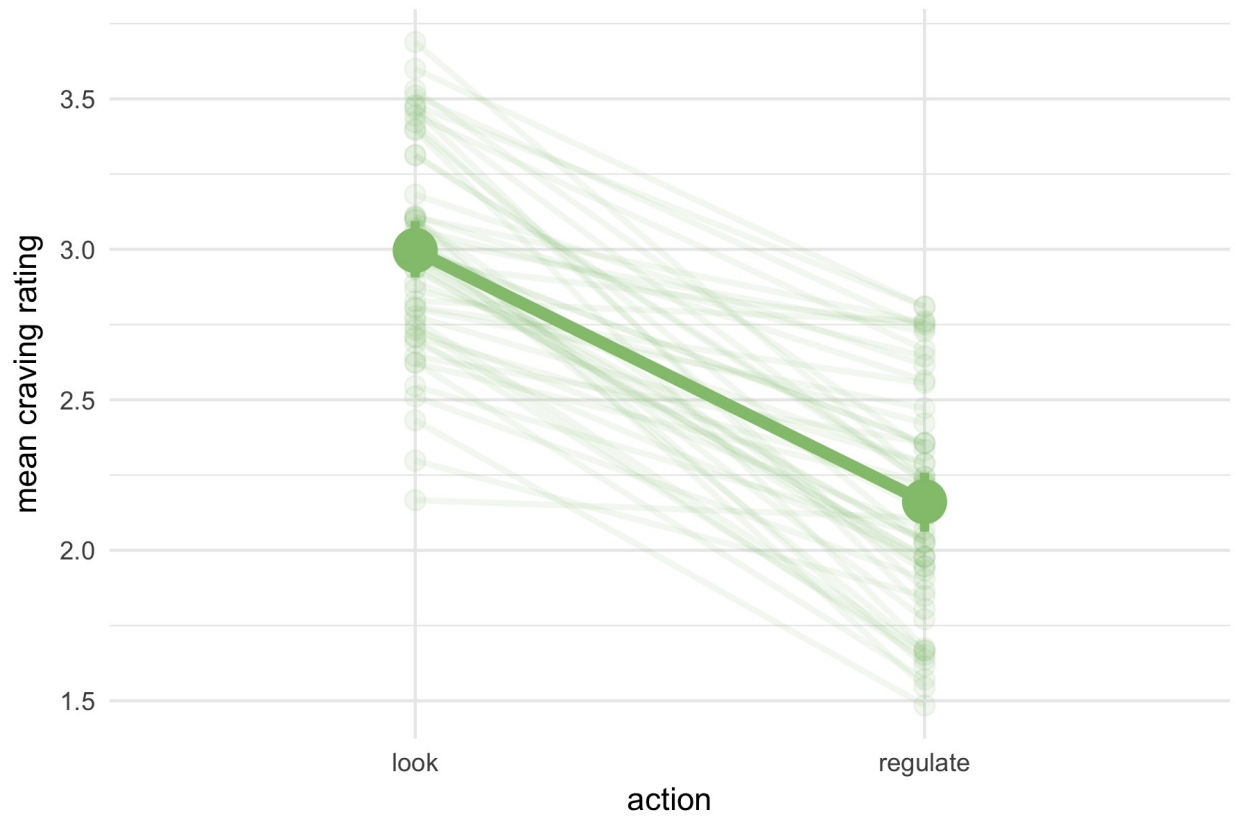


```
reg.success %>%
  gather(score, mean, look, regulate, reg.success) %>%
  ggplot(aes(score, mean)) +
    geom_violin() +
    geom_jitter(width = .1, alpha = .5, color = palettegreen) +
    theme_minimal()
```

plot reg. success

```
reg.success %>%
  gather(action, mean, look, regulate) %>%
  ggplot(aes(action, mean)) +
    geom_point(aes(group = subjectID), color = palettegreen, alpha = .1, size = 3) +
    geom_line(aes(group = subjectID), color = palettegreen, alpha = .1, size = 1) +
    stat_summary(aes(group = 1), color = palettegreen, fun.y = mean, geom = "line", size = 2) +
    stat_summary(color = palettegreen, fun.data = "mean_cl_boot", size = 1.5) +
    labs(y = "mean craving rating") +
    theme_minimal()
```



```
task %>%
  filter(!is.na(action)) %>%
  group_by(action) %>%
  summarize(mean = mean(respRating, na.rm = TRUE),
            sd = sd(respRating, na.rm = TRUE),
            n = n()) %>%
  kable(digits = 2, format = "pandoc", caption = "craving ratings")
```

Table 1: craving ratings

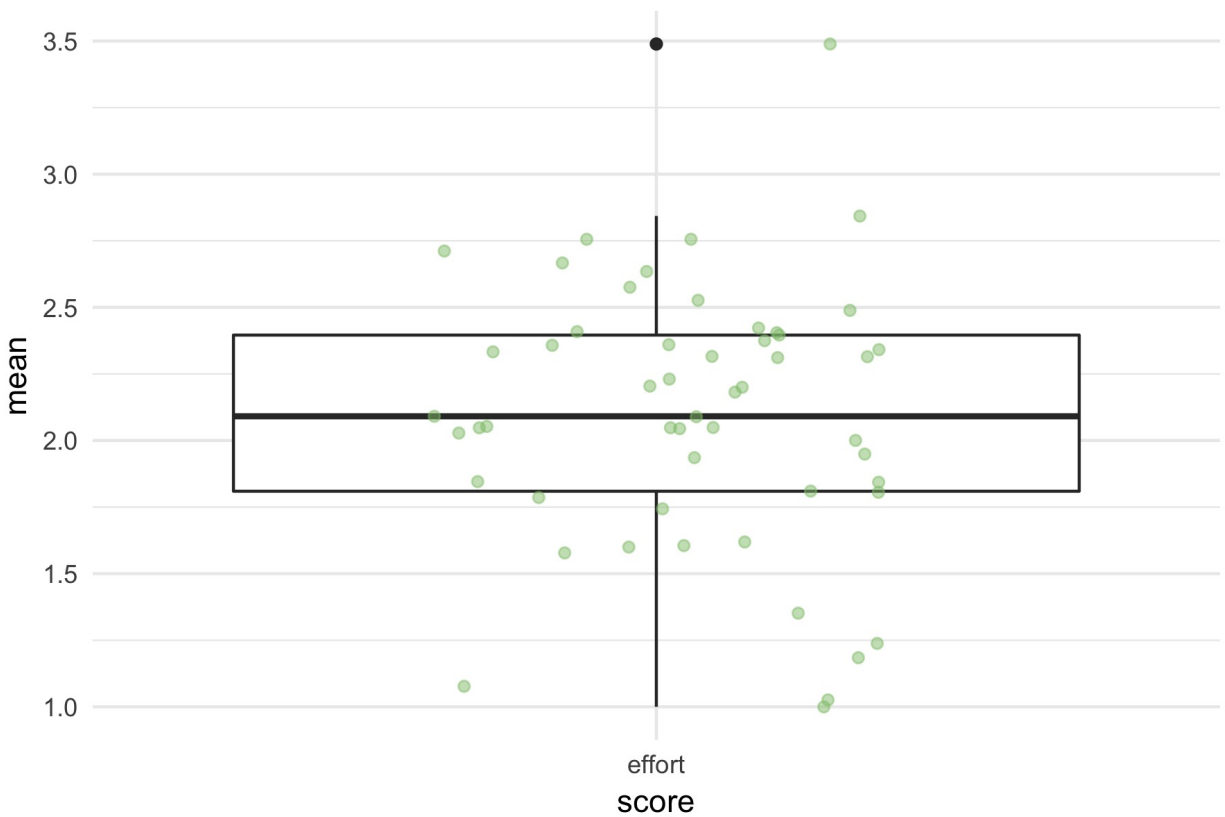
action	mean	sd	n
look	3.01	0.92	2430
regulate	2.16	0.86	2217

calculate mean effort

```
effort = task %>%
  filter(action == "regulate") %>%
  group_by(subjectID) %>%
  summarize(meanEffort = mean(respEffort, na.rm = TRUE))
```

describe effort

```
effort %>%  
  gather(score, mean, meanEffort) %>%  
  mutate(score = ifelse(score == "meanEffort", "effort", score)) %>%  
  ggplot(aes(score, mean)) +  
    geom_boxplot() +  
    geom_jitter(width = .2, alpha = .5, color = palettegreen) +  
    theme_minimal()
```



```
effort %>%  
  gather(score, mean, meanEffort) %>%  
  mutate(score = ifelse(score == "meanEffort", "effort", score)) %>%  
  ggplot(aes(score, mean)) +  
    geom_violin() +  
    geom_jitter(width = .1, alpha = .5, color = palettegreen) +  
    theme_minimal()
```



correlations

```
# merge data
merged = left_join(reg.success, EDEQS, by = "subjectID") %>%
  left_join(., effort, by = "subjectID")
```

reg. success

Example stats reporting format: $r = .19$, 95% CI = $[-.077, .44]$, $t(51) = 1.44$, $p = .156$

```
# total
cor.test(merged$reg.success, merged$total)
```

```
##
## Pearson's product-moment correlation
##
## data: merged$reg.success and merged$total
## t = 1.4399, df = 51, p-value = 0.156
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.07674486 0.44421149
## sample estimates:
## cor
## 0.1976489
```

```

# restrained
cor.test(merged$reg.success, merged$restrained)

##
## Pearson's product-moment correlation
##
## data: merged$reg.success and merged$restrained
## t = 1.2412, df = 51, p-value = 0.2202
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.1038586 0.4220034
## sample estimates:
## cor
## 0.1712424

# binge
cor.test(merged$reg.success, merged$binge)

##
## Pearson's product-moment correlation
##
## data: merged$reg.success and merged$binge
## t = 1.1514, df = 51, p-value = 0.255
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.1161222 0.4117471
## sample estimates:
## cor
## 0.1591674

```

effort

```

# total
cor.test(merged$meanEffort, merged$total)

##
## Pearson's product-moment correlation
##
## data: merged$meanEffort and merged$total
## t = -0.050249, df = 51, p-value = 0.9601
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.2768035 0.2637593
## sample estimates:
## cor
## -0.00703611

# restrained
cor.test(merged$meanEffort, merged$restrained)

##
## Pearson's product-moment correlation
##
## data: merged$meanEffort and merged$restrained

```

```
## t = -1.9968, df = 51, p-value = 0.0512
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.502964856 0.001094393
## sample estimates:
##      cor
## -0.2692791

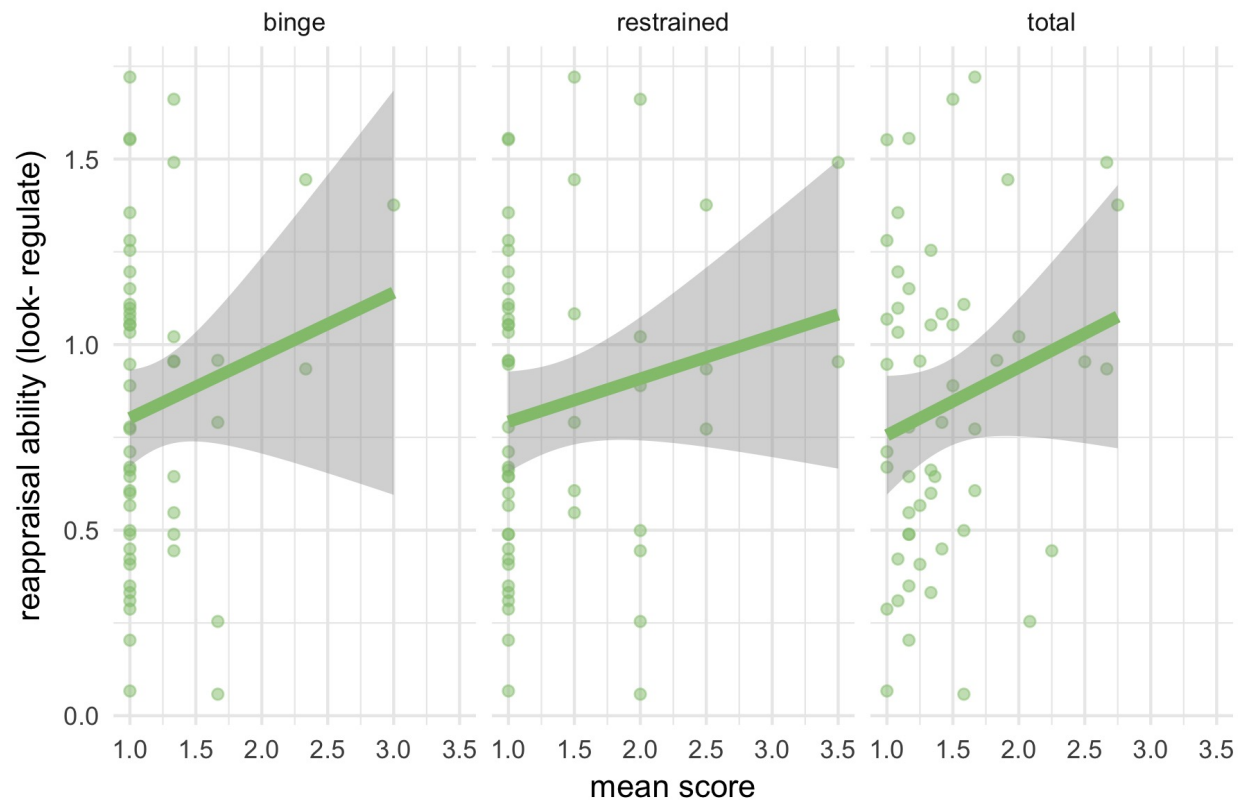
# binge
cor.test(merged$meanEffort, merged$binge)

##
## Pearson's product-moment correlation
##
## data: merged$meanEffort and merged$binge
## t = 1.6486, df = 51, p-value = 0.1054
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.04829991 0.46684152
## sample estimates:
##      cor
## 0.2249304
```

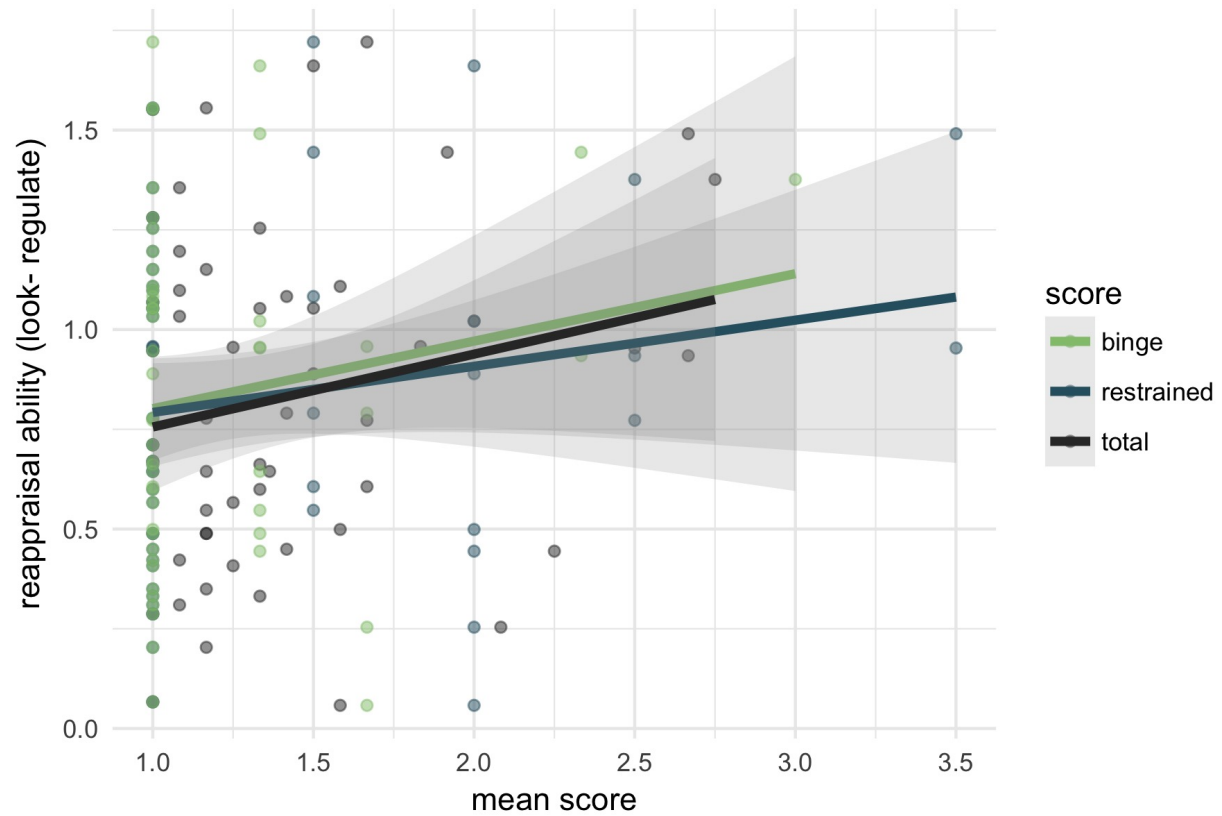
plot correlations

reg. success

```
merged %>%
  gather(score, mean, total, restrained, binge) %>%
  ggplot(aes(mean, reg.success)) +
    geom_point(color = palettegreen, alpha = .5) +
    geom_smooth(method = "lm", color = palettegreen, size = 2) +
    facet_grid(~score) +
    labs(x = "mean score", y = "reappraisal ability (look- regulate)") +
    theme_minimal()
```

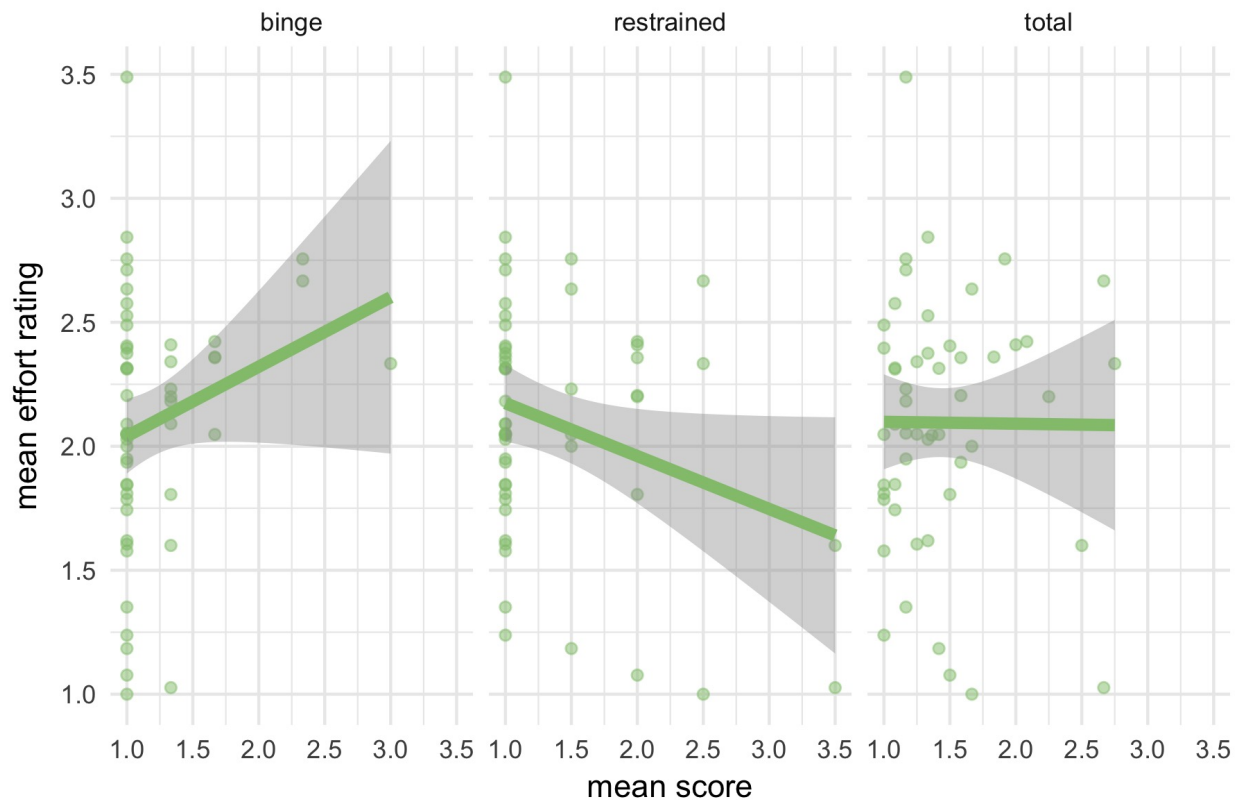


```
merged %>%
  gather(score, mean, total, restrained, binge) %>%
  ggplot(aes(mean, reg.success, color = score)) +
    geom_point(alpha = .5) +
    geom_smooth(method = "lm", alpha = .2, size = 1.5) +
    scale_color_manual(values = palette3) +
    labs(x = "mean score", y = "reappraisal ability (look-regulate)") +
    theme_minimal()
```



effort

```
merged %>%
  gather(score, mean, total, restrained, binge) %>%
  ggplot(aes(mean, meanEffort)) +
    geom_point(color = palettegreen, alpha = .5) +
    geom_smooth(method = "lm", color = palettegreen, size = 2) +
    facet_grid(~score) +
    labs(x = "mean score", y = "mean effort rating") +
    theme_minimal()
```

```
merged %>%
  gather(score, mean, total, restrained, binge) %>%
  ggplot(aes(mean, meanEffort, color = score)) +
    geom_point(alpha = .5) +
    geom_smooth(method = "lm", alpha = .2, size = 1.5) +
    scale_color_manual(values = palette3) +
    labs(x = "mean score", y = "mean effort rating") +
    theme_minimal()
```

