A Template knitr + LaTeX Document

Julia Fisher
<julia@statlab.bio5.org>

November 6, 2016

Contents

1 Statement of Purpose

The purpose of this document is twofold. First, it is meant to serve as a template for statistical consulting reports. Second, it is meant to illustrate how to produce complex and beautifully-formatted tables using xtable and graphics using ggplot2. With those goals in mind, I have tried to show some of the capatilities of knitr+LaTeX, xtable, and ggplot2. However, I have not been exhaustive in that demonstration. For more information on knitr, I highly recommend Yihui Xie's website (`http://yihui.name/knitr/`). In particular, I regularly reference his page on chunk options (`http://yihui.name/knitr/options/`). For more information on xtable, I recommend the xtable gallery (`https://cran.r-project.org/web/packages/xtable/vignettes/xtableGallery.pdf`). For ggplot2, I recommend the ggplot2 webpages (`http://docs.ggplot2.org/current/index.html`).

This document is the result of poring over many people's code, especially that of Shannon Knapp and Isaac Jenkins. Thank you so much to them for numerous examples and ideas. Also, thank you both to everyone who took the time to post questions on internet forums and to everyone else who took the time to answer those questions.

To make this document fun, we simulate frauda (fake + data) for a hypothetical experiment. We use these frauda to illustrate knitr+LaTeX, xtable, and ggplot2. Enjoy!

2 The Hypothetical Experiment

A researcher wants to know if people will perform better on a particular class of puzzle-solving tasks if they are exposed to images of success and perserverence immediately before the tasks. She designs an experiment to test her hypothesis. Here are the relevant details:

- 50 people participate in the experiment.
- Each person is given six tasks from the target class. Before three of the tasks, they are briefly exposed to an image depicting success and perseverence (the "success" condition). Before the other three tasks, they are exposed to neutral images (the "neutral" condition).
- Participants receive two scores for each task. One is automatically calculated by the computer presenting the task (the "objective" score). The other is a subjective assesment of documents created during the task and a videotaping of the task (the "subjective" score). There is neither an upper nor lower bound on either task score. Thus, scores can exceed 100 and be less than 0. The objective and subjective scores are separate outcome variables.
- Note that the tasks are randomly-generated. No two people see exactly the same tasks. In fact, it is likely that no two people even see one identical task.
- In between the target six tasks, participants also complete unrelated (i.e. distractor) tasks. These data are not analyzed.
- The researcher also gathers some demographic information on her participants.

3 Tables

3.1 A Basic xtable Example

To begin, we present the collected demographic information.

Demographic Category	Count (%) / Mean ± SD
Gender	
Female	27 (54%)
Male	19 (38%)
Trans Female	2 (4%)
Trans Male	2 (4%)
Age	21.78 ± 2.79
Education	
Associate's Degree	5 (10%)
Bachelor's Degree	22 (44%)
Graduate Degree	4 (8%)
High School or GED	8 (16%)
Some College	11 (22%)

Table 1: Demographic information

3.2 A More Complicated xtable Example

Now, suppose we wanted a more complex table (where we have to merge some tables and columns and maybe rotate some text). We can do this. To motivate this table, suppose that our subjects are split into two groups based on some criterion. We might want to display the demographics of those two groups separately.

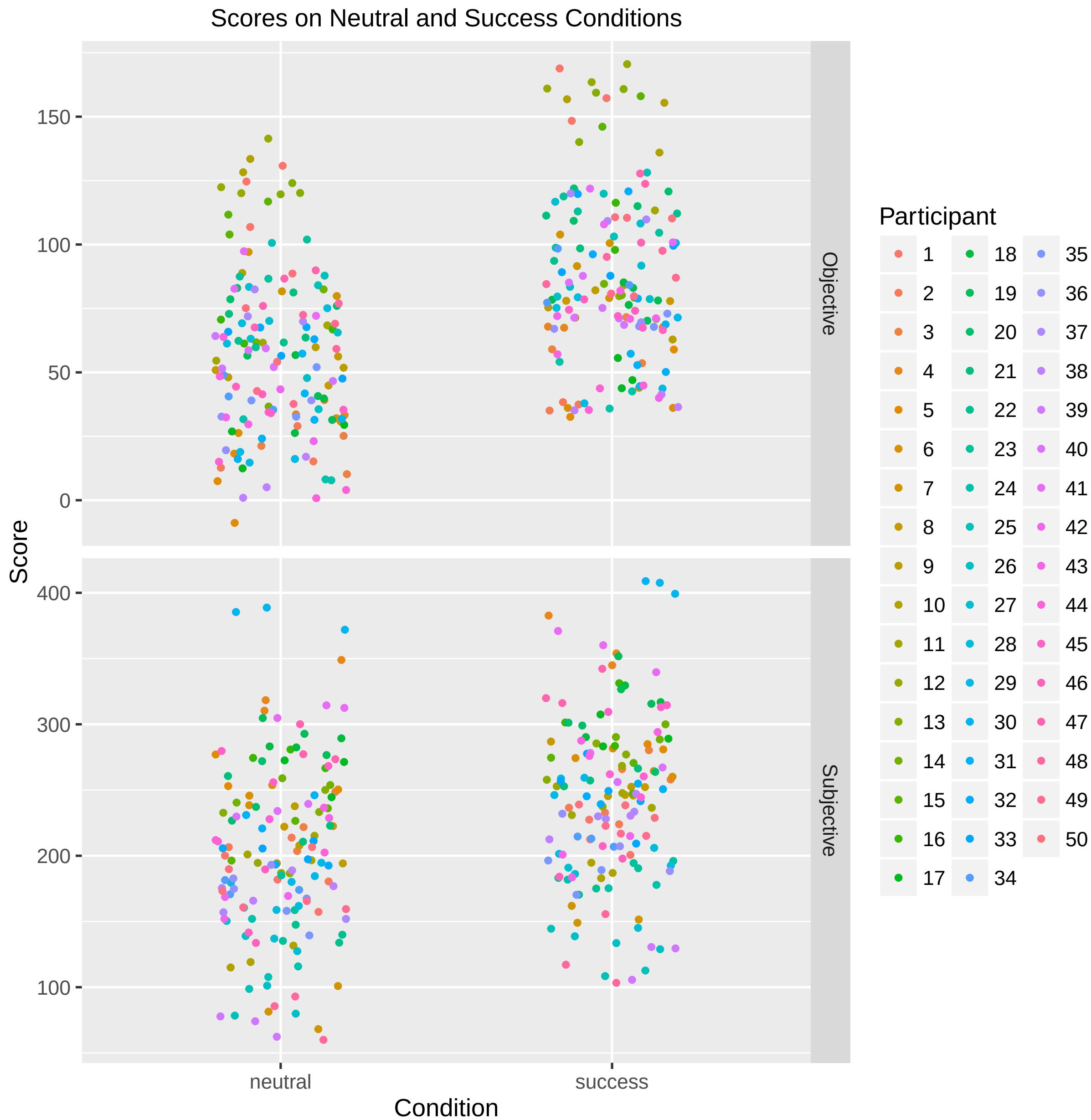
	Demographic Category	**Count (%) / Mean ± SD**		
		Group 1	**Group 2**	**Total**
	Number of children	24 (48%)	26 (52%)	50
Gender	Female	15 (62.5%)	12 (46.15%)	27 (54%)
	Male	8 (33.33%)	11 (42.31%)	19 (38%)
	Trans Female	1 (4.17%)	1 (3.85%)	2 (4%)
	Trans Male	0 (0%)	2 (7.69%)	2 (4%)
	Age	21.83 ± 2.32	21.73 ± 3.22	21.78 ± 2.79
Education	Associate's Degree	4 (16.67%)	1 (3.85%)	5 (10%)
	Bachelor's Degree	8 (33.33%)	14 (53.85%)	22 (44%)
	Graduate Degree	2 (8.33%)	2 (7.69%)	4 (8%)
	High School or GED	4 (16.67%)	4 (15.38%)	8 (16%)
	Some College	6 (25%)	5 (19.23%)	11 (22%)

Table 2: Demographic information

4 Plots

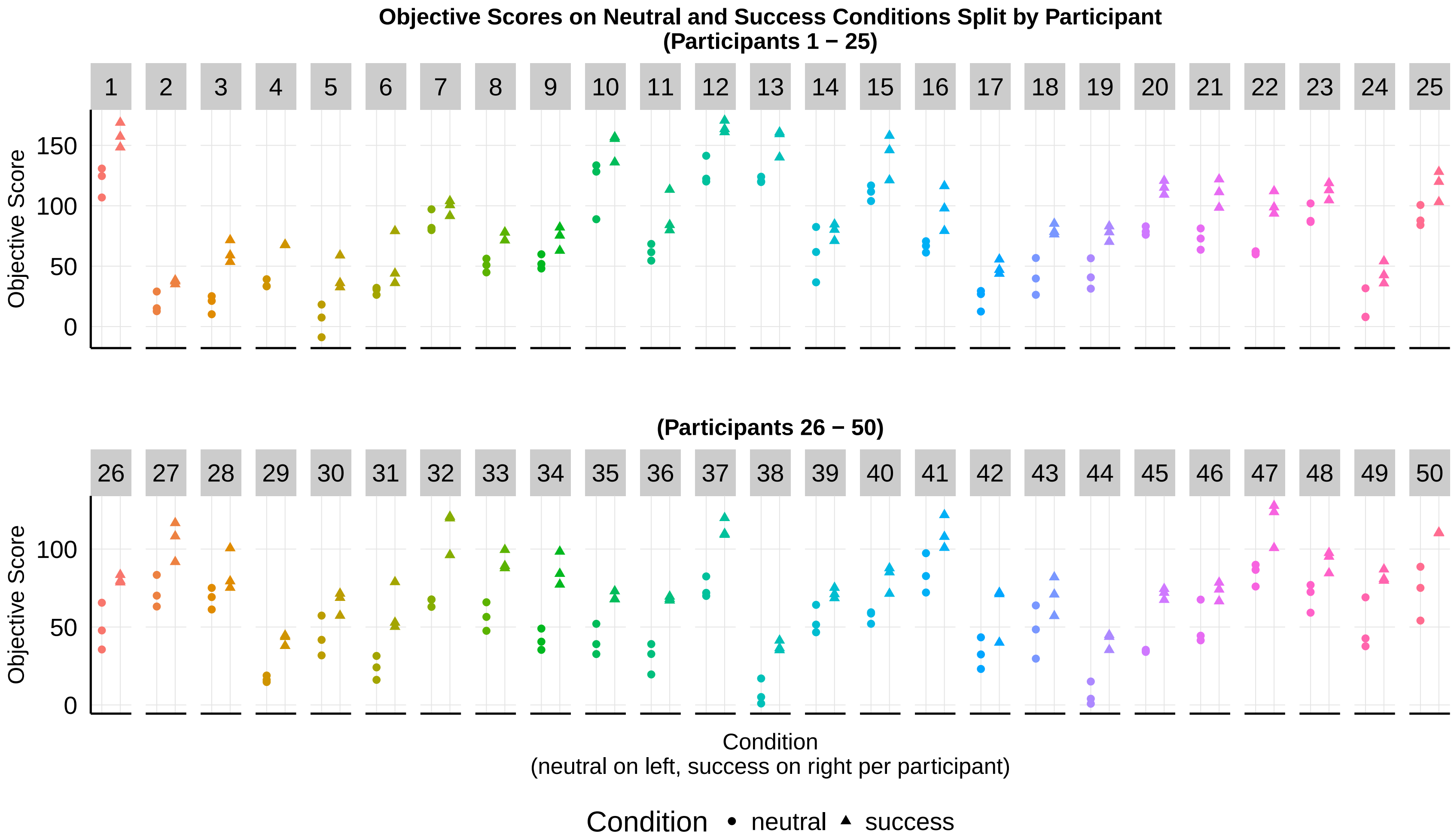
4.1 A Basic ggplot2 Example

We plot the distribution of both types of score. Each color is a different participant, and the points are jittered in the x-direction.



4.2 A More Complicated ggplot2 Example

It's a bit hard to see what is going on with each participant in the above plots. So, let's look at each person individually. This is a slightly more complicated plotting scenario, though, because we have too many participants to easily just use faceting in one direction. One way to get around this is to split our data in half and stack the two plots of half the data. (Also, I wanted to illustrate how to combine plots.)



5 Models

5.1 Model of Objective Score

Let's analyze the objective score. We fit a model with condition as a fixed effect and participant as a random intercept.

	Estimate	Std. Error	t value
Intercept	56.79	4.46	12.75
Condition: Success	30.30	1.19	25.56

Table 3: Estimated Coefficients

	F	Df	Df.res	Pr(>F)
Condition	653.09	1	249	0.0000

Table 4: Type II Wald F Tests with Kenward-Roger Degrees of Freedom

5.2 A Planned Comparison

Now, suppose the researcher wanted to know ahead of time whether or not the success condition produces scores that are more than 1.5 times higher than the neutral condition. In other words, our null and alternative hypotheses are as follows:

$$\begin{aligned} H_0 &: \ \mu_s \leq 1.5\mu_n \iff \mu_s - 1.5\mu_n \leq 0 \\ H_1 &: \ \mu_s > 1.5\mu_n \iff \mu_s - 1.5\mu_n > 0 \end{aligned}$$

where μ_s is the mean of the success condition, and μ_n is the mean of the neutral condition. We can test this easily using the lsmeans package. First, we print out the least-squares means of the two conditions.

condition	lsmean	SE	df	lower.CL	upper.CL
neutral	56.7906	4.4555	50.78	47.8448	65.7364
success	87.0861	4.4555	50.78	78.1403	96.0319

Degrees-of-freedom method: kenward-roger

Confidence level used: 0.95

Table 5: Least-Squares Means of Each Condition

Next, we test the null hypothesis. We find that we do not have enough evidence to reject the null hypothesis ($\hat{\mu}_s - 1.5\hat{\mu}_n = 1.9$; $t(99.12) = 0.71$, $p = 0.24$).

6 Appendix: R Version, Data Processing, Functions for Data Processing, Raw Output, and Additional Model Checking

This section of the document is specifically here for two reasons. One, it helps with reproducibility. The idea is to give someone who may want to reproduce this document all the information they need to be successful. Thus, we print out the versions of R and the various packages we used, any files we may have used, and other useful information. We also print out all code. Hopefully, someone who might want to reproduce this analysis would have the Rnw file that generated this document (and thus all the code would be in there), but if they don't, this is the back-up. Also, note that the code printed out here is in fact *just* printed out, not run. It was run in the sections above. Saving the printing of the code for this appendix, though, makes the main portion of the document easier to read.

The second reason we have this appendix is to print out any raw output or other information that may be useful. For example, we print out a summary of the model we ran above so that someone who wants additional information on the model has that available. Additionally, we provide some basic diagnostic plots.

6.1 Versions

The following is the session information for this analysis. This includes the version of R and versions of all packages used.

```
R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X Yosemite 10.10.5

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] cowplot_0.7.0      xtable_1.8-2       pbkrtest_0.4-6
 [4] car_2.1-3          lsmeans_2.24       estimability_1.1-1
 [7] lme4_1.1-12        Matrix_1.2-7.1     ggplot2_2.1.0
[10] dplyr_0.5.0        tidyr_0.6.0        knitr_1.14

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.7        formatR_1.4        nloptr_1.0.4
 [4] plyr_1.8.4         tools_3.3.2        digest_0.6.10
 [7] evaluate_0.10      tibble_1.2         nlme_3.1-128
[10] gtable_0.2.0       lattice_0.20-34    mgcv_1.8-15
[13] DBI_0.5-1          parallel_3.3.2     SparseM_1.72
[16] mvtnorm_1.0-5      coda_0.18-1        stringr_1.1.0
[19] MatrixModels_0.4-1 grid_3.3.2         nnet_7.3-12
[22] R6_2.2.0           survival_2.40-1    multcomp_1.4-6
[25] TH.data_1.0-7      minqa_1.2.4        reshape2_1.4.2
[28] magrittr_1.5       scales_0.4.0       codetools_0.2-15
[31] MASS_7.3-45        splines_3.3.2      assertthat_0.1
[34] colorspace_1.2-7   labeling_0.3       quantreg_5.29
[37] sandwich_2.3-4     stringi_1.1.2      lazyeval_0.2.0
[40] munsell_0.4.3      zoo_1.7-13
```

6.2 Datasets

If we had used a dataset from a separate file in this analysis, we would list that file here. See the code for how we would have done that.

6.3 Data Processing

This is the data processing that was done:

```
library(tidyr)
library(dplyr)
library(ggplot2)
library(lme4)
library(lsmeans)
library(car)
library(pbkrtest)
library(xtable)
```

```
# We're going to generate frauda for our hypothetical experiment.  Here's how:
# - We assume every person has a random base score for the neutral image tasks.
# - We generate that random base score by sampling a normal distribution with mean 50
#   and standard deviation 30.
# - We assume there is a fixed improvement to that base score that does not vary
#   across participant.

# Set your seed!
set.seed(7)

# Number of participants:
n.participants <- 50

# Error standard deviation:
error.sd.obj <- 10
error.sd.subj <- 15

# Base scores per participant:
participant.bases.obj <- rnorm(n.participants, mean = 50, sd = 30)
participant.bases.subj <- rnorm(n.participants, mean = 200, sd = 70)

# Fixed improvement amount:
fixed.improvement.obj <- 30
fixed.improvement.subj <- 40

# Gender of each participant:
gender.options <- c("Female",
                    "Male",
                    "Trans Female",
                    "Trans Male",
                    "Other")
gender.stats <- sample(x = gender.options,
                       size = n.participants,
                       replace = TRUE,
                       prob = c(0.45, 0.45, 0.04, 0.04, 0.02))

# Age of each participant:
ages <- floor(rnorm(n.participants, mean = 22, sd = 3))

# Education of each participant:
education.options <- c("High School or GED",
                       "Some College",
                       "Associate's Degree",
                       "Bachelor's Degree",
                       "Graduate Degree")
education.stats <- sample(x = education.options,
                          size = n.participants,
                          replace = TRUE,
                          prob = c(0.2, 0.2, 0.1, 0.4, 0.1))

# Create an empty data frame to fill.  Then fill it!
dta <- data.frame(participant = character(),
                  condition = character(),
                  score = numeric(),
```

```
                  gender = character(),
                  age = character(),
                  education = character(),
                  stringsAsFactors = FALSE)
for (i in 1:n.participants) {
  base.score.obj <- participant.bases.obj[i]
  base.score.subj <- participant.bases.subj[i]
  scores.obj <- rnorm(6, mean = base.score.obj, sd = error.sd.obj)
  scores.subj <- rnorm(6, mean = base.score.subj, sd = error.sd.subj)
  scores.obj[1:3] <- scores.obj[1:3] + fixed.improvement.obj
  scores.subj[1:3] <- scores.subj[1:3] + fixed.improvement.subj
  curr.dta <- data.frame(participant = as.character(i),
                         condition = c(rep("success", 3), rep("neutral", 3)),
                         obj_score = scores.obj,
                         subj_score = scores.subj,
                         gender = gender.stats[i],
                         age = ages[i],
                         education = education.stats[i],
                         stringsAsFactors = FALSE)
  dta <- rbind(dta, curr.dta)
}
```

6.4 Functions Used in Data Processing

If we had custom functions used in the data processing, we could put their code here.

/path/to/your_script.R

```
# If you had any functions you had written, you could just type function_name without parentheses here.
# This would print out the function so that you would know later on what it does.
```

6.5 Demographic Information

Here is the code used to produce the demographic tables:

```
# Consolidate the information in order to present demographic information.
dta.for.table <- dta %>%
  select(participant, gender, age, education) %>%
  unique

# Get the demographic information into a format that we can easily put into a table.
table.info <- rbind(c("Gender", ""),
                    dta.for.table %>%
                      group_by(gender) %>%
                      tally %>%
                      as.data.frame %>%
                      mutate(category = paste0("~~~", gender),
                             n = paste0(n, " (", round(n/sum(.$n)*100, 0), "\\%)")) %>%
                      select(category, n)) %>%
  rbind(dta.for.table %>%
          summarise(mean.age = mean(age),
                    sd.age = sd(age)) %>%
          mutate(category = "Age",
                 n = paste0(round(mean.age, 2), " $\\pm$ ", round(sd.age, 2))) %>%
          select(category, n)) %>%
  rbind(c("Education", ""),
        dta.for.table %>%
          group_by(education) %>%
          tally %>%
          as.data.frame %>%
          mutate(category = paste0("~~~", education),
                 n = paste0(n, " (", round(n/sum(.$n)*100, 0), "\\%)")) %>%
```

```r
          select(category, n))

# Rename the columns.
colnames(table.info) <- c("Demographic Category", "Count (\\%) / Mean $\\pm$ SD")

# Create the xtable object.
xtable.table.info <- xtable(table.info,
                                caption = "Demographic information",
                                align = c("l", "l", "c"))

# Create extra information to add to some of the rows of the xtable object above.
addtorow <- list()
addtorow$pos <- list(5, 6, -1, 0, nrow(table.info))
addtorow$command <- c(rep(" \\specialrule{0.5pt}{3pt}{3pt} ", 2),
                       rep(" \\specialrule{1.5pt}{5pt}{5pt} ", 3))

# Print the xtable.
print(xtable.table.info,
      add.to.row = addtorow,
      sanitize.text.function = function(x) {x},
      sanitize.colnames.function = function(x) {x},
      include.rownames = FALSE,
      booktabs = FALSE,
      hline.after = c())
```

```r
# Set a seed.
set.seed(52315)

# Randomly assign people to a group
dta.for.table2 <- dta.for.table %>%
  mutate(group = sample(c("Group1", "Group2"), n.participants, replace = TRUE))

# Get the demographic information into a format that we can easily put into a table.
table.info2 <- dta.for.table2 %>%
  group_by(group) %>%
  tally %>%
  ungroup %>%
  as.data.frame %>%
  spread(group, n) %>%
  mutate(Group1 = as.character(Group1),
         Group2 = as.character(Group2)) %>%
  cbind(demo = "Number of children", .) %>%
  cbind(dta.for.table2 %>%
          tally %>%
          mutate(Total = as.character(n)) %>%
          select(-n)) %>%
  mutate(demo = as.character(demo))

# Get total enrollment per group for use in later calculations:
total.n <- as.numeric(table.info2[1, "Total"])
g1.n <- as.numeric(table.info2[1, "Group1"])
g2.n <- as.numeric(table.info2[1, "Group2"])

# Continue getting the demographic information into a format that we can easily put
# into a table.
table.info2 <- table.info2 %>%
  mutate(Group1 = paste0(Group1, " (",
                         round(as.numeric(Group1)/as.numeric(Total)*100, 2),
                         "\\%)"),
         Group2 = paste0(Group2, " (",
                          round(as.numeric(Group2)/as.numeric(Total)*100, 2),
                          "\\%)")) %>%
  rbind(dta.for.table2 %>%
          group_by(group, gender) %>%
          tally %>%
          ungroup %>%
          as.data.frame %>%
          spread(group, n, fill = 0) %>%
```

```r
        rename(demo = gender) %>%
        mutate(Group1 = paste0(Group1, " (", round(Group1/g1.n*100, 2), "\\%)"),
               Group2 = paste0(Group2, " (", round(Group2/g2.n*100, 2), "\\%)")) %>%
        cbind(dta.for.table2 %>%
                group_by(gender) %>%
                tally %>%
                ungroup %>%
                as.data.frame %>%
                rename(Total = n) %>%
                mutate(Total = paste0(Total, " (", round(Total/total.n*100, 2), "\\%)")) %>%
                select(-gender))) %>%
  rbind(dta.for.table2 %>%
        group_by(group) %>%
        summarise(mean_age = mean(age),
                  sd_age = sd(age)) %>%
        ungroup %>%
        as.data.frame %>%
        mutate(age = paste0(round(mean_age, 2), " $\\pm$ ", round(sd_age, 2))) %>%
        select(-mean_age, -sd_age) %>%
        spread(group, age) %>%
        cbind(demo = "Age", .) %>%
        cbind(dta.for.table2 %>%
                summarise(mean_age = mean(age),
                          sd_age = sd(age)) %>%
                mutate(Total = paste0(round(mean_age, 2), " $\\pm$ ", round(sd_age, 2))) %>%
                select(-mean_age, -sd_age))) %>%
  rbind(dta.for.table2 %>%
        group_by(group, education) %>%
        tally %>%
        ungroup %>%
        as.data.frame %>%
        spread(group, n, fill = 0) %>%
        rename(demo = education) %>%
        mutate(Group1 = paste0(Group1, " (", round(Group1/g1.n*100, 2), "\\%)"),
               Group2 = paste0(Group2, " (", round(Group2/g2.n*100, 2), "\\%)")) %>%
        cbind(dta.for.table2 %>%
                group_by(education) %>%
                tally %>%
                ungroup %>%
                as.data.frame %>%
                rename(Total = n) %>%
                mutate(Total = paste0(Total, " (", round(Total/total.n*100, 2), "\\%)")) %>%
                select(-education))) %>%
  cbind(blank = c("",
                "\\multirow{4}{*}{\\rotatebox[origin=c]{90}{\\parbox[c]{1cm}{\\centering Gender}}}",
                rep("", 4),
                "\\multirow{5}{*}{\\rotatebox[origin=c]{90}{\\parbox[c]{1.5cm}{\\centering Education}}}",
                rep("", 4)), .) %>%
  rbind(c("", "", "\\textbf{Group 1}", "\\textbf{Group 2}", "\\textbf{Total}"), .)

# Create extra information to add to some of the rows of the xtable object above.
addtorow <- list()
addtorow$pos <- list(0, 0, 2, 6, 7, -1, 1, nrow(table.info2))
addtorow$command <- c(paste0(" & ",
                             "\\multirow{2}{3cm}{\\parbox[c][9mm][c]{2.5cm}{\\textbf{Demographic Category}}}",
                             " & \\multicolumn{3}{c}{\\textbf{Count (\\%) / Mean $\\pm$ SD}} \\\\[1mm]\n "),
                      " \\cmidrule{3-5} ",
                      rep(" \\specialrule{0.5pt}{3pt}{3pt} ", 3),
                       rep(" \\specialrule{1.5pt}{5pt}{5pt} ", 3))

# Create the xtable objct.
xtable.info2 <- xtable(table.info2,
                       caption = "Demographic information")

# Set column alignment.
align(xtable.info2) <- "cclccc"
```

```r
# Print the xtable.
print(xtable.info2,
      add.to.row = addtorow,
      include.colnames = FALSE,
      include.rownames = FALSE,
      booktabs = FALSE,
      sanitize.text.function = function(x) {x},
      size = "footnotesize",
      hline.after = c())
```

6.6 Plotting Code

Here is the code used to produce the plots:

```
# Manipulate the data so that they are easier to plot.
dta.basic.plot <- dta %>%
  mutate(participant = factor(participant, levels = as.character(1:n.participants))) %>%
  gather(score_category, score, 3:4) %>%
  mutate(score_category = ifelse(score_category == "obj_score", "Objective", "Subjective")) %>%
  arrange(participant, score_category)

# Plot that data.
ggplot(dta.basic.plot, aes(x = condition, y = score, colour = participant)) +
  facet_grid(score_category ~ ., scales = "free") +
  geom_jitter(size = 1, height = 0, width = 0.5) +
  labs(title = "Scores on Neutral and Success Conditions",
       y = "Score",
       x = "Condition") +
  # guides(colour = FALSE) + # Uncomment this line if you want to get rid of the color legend.
  scale_colour_discrete(name = "Participant") +
  theme(axis.title.x = element_text(vjust = -0.6, size = 11),
        axis.title.y = element_text(vjust = 0.5, size = 11),
        plot.title = element_text(vjust = 0.5, size = 11))
```

```
# Load in the cowplot library to enable plot combining.
# Note that we didn't load this earlier because we only now want the white background.
# If we had loaded it earlier, both the simple and the complicated plots would have
# had white backgrounds.
library(cowplot)

# Split the data.
dta.basic.plot1 <- dta.basic.plot %>%
  filter(participant %in% 1:floor(n.participants/2))
dta.basic.plot2 <- dta.basic.plot %>%
  filter(participant %in% (floor(n.participants/2) + 1):n.participants)

# Create first plot.
plot1 <- ggplot(dta.basic.plot1[which(dta.basic.plot1$score_category == "Objective"), ],
                aes(x = condition, y = score, colour = participant, shape = condition)) +
  facet_grid(. ~ participant, scales = "free") +
  geom_point() +
  labs(title = paste0("Objective Scores on Neutral and Success Conditions Split by Participant\n",
                      "(Participants 1 - 25)"),
       y = "Objective Score",
       x = "") +
  guides(colour = FALSE,
         shape = FALSE) +
  scale_x_discrete(labels = NULL) +
  scale_shape_discrete(name = "Condition") +
  background_grid(major = "xy", minor = "none") +
  theme(axis.ticks = element_blank(),
        axis.title.x = element_text(vjust = -0.6, size = 11),
        axis.title.y = element_text(vjust = 0.5, size = 11),
        plot.title = element_text(vjust = 0.5, size = 11))

# Create second plot.
plot2 <- ggplot(dta.basic.plot2[which(dta.basic.plot2$score_category == "Objective"), ],
                aes(x = condition, y = score, colour = participant, shape = condition)) +
  facet_grid(. ~ participant, scales = "free") +
  geom_point() +
  labs(title = "(Participants 26 - 50)",
       y = "Objective Score",
       x = "Condition\n(neutral on left, success on right per participant)") +
  guides(colour = FALSE) +
  scale_x_discrete(labels = NULL) +
  scale_shape_discrete(name = "Condition") +
  background_grid(major = "xy", minor = "none") +
```

```r
  theme(axis.ticks = element_blank(),
        axis.title.x = element_text(vjust = -0.6, size = 11),
        axis.title.y = element_text(vjust = 0.5, size = 11),
        plot.title = element_text(vjust = 0.5, size = 11),
        legend.position = "bottom")

# Print the two plots in a grid.  I make the relative height of the second one just a
# tad larger because the legend at the bottom of that plot takes up a bit of room,
# making the bottom plot look a little smaller.  Making the relative height a bit bigger
# counteracts that effect.
plot_grid(plot1, plot2, ncol = 1, rel_heights = c(1, 1.15))
```

6.7 Modeling Code

Here is the modeling code:

```r
# Make "participant" and "condition" factors.
dta.mdl <- dta %>%
  mutate(participant = factor(participant),
         condition = factor(condition))

# Run the model.
mdl <- lmer(obj_score ~ condition + (1 | participant), data = dta.mdl)

# Pull out the estimated slopes:
estimates.mdl <- summary(mdl)$coefficients

# Change the names for easier viewing:
rownames(estimates.mdl) <- c("Intercept", "Condition:  Success")

# Print the estimated slopes:
print(xtable(estimates.mdl, caption = "Estimated Coefficients"))

# Pull out the anova(ish) table and conduct Type II Wald F tests of the factor "condition":
Ftest.mdl <- Anova(mdl, test = "F")
rownames(Ftest.mdl) <- c("Condition")

# Print the anova(ish) table:
print(xtable(Ftest.mdl, caption = "Type II Wald F Tests with Kenward-Roger Degrees of Freedom"))
```

```r
# Let's use the Kenward-Roger degrees of freedom and covariance adjustment.
lsm.options(lmer.df = "kenward-roger")

# Get the least-squares means and print them out:
lsmeans.mdl <- lsmeans(mdl, ~ condition, mode = "kenward-roger")
print(xtable(lsmeans.mdl, caption = "Least-Squares Means of Each Condition",
             sanitize.text.function = function(str) gsub("_", "\\_", str, fixed = TRUE)))
```

```r
# Test whether or not the success scores are 1.5 or more times the neutral scores.
est <- test(contrast(lsmeans.mdl,
                     list("Test that " = c(-1.5, 1)), by = NULL), side = ">")

# Extract key information from the above.
contrast.est <- round(est$estimate, 2)
t.val <- round(est$t.ratio, 2)
df <- round(est$df, 2)
p.val <- round(est$p.value, 2)
```

6.8 Raw Output

Here is the raw output from the model run above.

```
Linear mixed model fit by REML ['lmerMod']
Formula: obj_score ~ condition + (1 | participant)
   Data: dta.mdl

REML criterion at convergence: 2440.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.7290 -0.6009 -0.0296  0.5165  2.7395

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 957.5    30.94
 Residual                105.4    10.27
Number of obs: 300, groups:  participant, 50

Fixed effects:
                 Estimate Std. Error t value
(Intercept)        56.791      4.456   12.75
conditionsuccess   30.295      1.185   25.56

Correlation of Fixed Effects:
            (Intr)
condtnsccss -0.133
```

6.9 Model Diagnostics

Here are some basic residual plots from the model above.