Data Analysis for: Priming Power with 3 DVs

Erika Salomon

January 12, 2016

Summary

This file reports data manipulations and analyses performed on an experiment studying the effects of power primes. Participants recruited via Mechanical Turk completed a power priming manipulation in which they were asked recall and visualize a time when they were high or low in power. After writing a description of their visualization, they completed a series of tasks to assess three dependent variables.

Participants' **action/inaction intentions** were measured with 7 questions (e.g., "I am feeling energetic"; 1 = *not at all*, 7 = *extremely*; Albarracin, 2009, unpublished).

Participants' **trust** was measured with two measures. Six items from the World Values Survey asked how much participants trust people belonging to different groups (e.g., "People you meet for the first time"; 1 = *trust completely*, 4 = *do not trust at all*). Participants also completed the trust game (Berg et al., 1995), which asked them how many raffle tickets they would entrust to another person who may or may not return them.

Two measures assessed participants' **perspective taking**. Participants answered seven items from the perspective-taking subscale of the Interpersonal Reactivity Index (Davis, 1980). These items were slightly modified to assess state rather than trait perspective taking motivation (e.g., "Before criticizing somebody, I'd try to imagine how I would feel if I were in their place"; 1 = *does not describe me at all*, 5 = *describes me very well*). Participants also completed 10 items from the Adult Reading the Mind in the Eyes Test-Revised (Baron-Cohen, Wheelwright, Hill, Raste & Plumb, 2001), which asked them to look at images of people's eyes and choose which of four emotions was expressed.

Participants completed measures for each of the dependent variables in a random order. For trust and perspective-taking, the two measures were also randomly ordered.

Load the raw data

The data were collected using Qualtrics. The first row of the raw data file contains most of the variable names. The second row contains some further variable names and the question wording. Rows three and beyond contain participant data. So we will first load in the two sets of variable names and combine them. Then, we will load in the data.

```
filename <- "PR__Priming_Power__3_DVs__Action_PerspectiveTaking_Distrust.csv"


name1 <- names(read.csv(filename, nrow = 1))
name2 <- names(read.csv(filename, nrow = 1, skip = 1))
names <- c(name2[1:22], name1[23:(length(name1))])
raw <- read.csv(filename, header = FALSE, skip = 2, col.names = names,
                stringsAsFactors = FALSE)
```

Reduce data set to a final, usable set of cases

The final data set should only contain cases that represent:

- US participants

- Participants who passed the instructional manipulation check (IMC)

We will identify and count cases that do not meet these criteria.

```r
# how many cases total?
totalCases <- nrow(raw)

# identify surveys from outside the US
nonUScount <- nrow(raw[with(raw, Country != "United States"),])

# identify participants who failed the IMC
failedIMCcount <- nrow(raw[with(raw, IMC != 10),])

# identify participants who meet both exclusion criteria
bothExcludeCount <- nrow(raw[with(raw,IMC != 10 & Country != "United States"),])

# create final data set and count cases
final <- raw[with(raw, IMC == 10 & Country == "United States"),]
finalCases <- nrow(final)
```

From 626 raw cases, 19 had non-US IP addresses and 29 failed the IMC (1 failed the IMC *and* had a non-US IP), leaving 579 for analysis.

Create labels for categorical/ordinal demographic variables

The categorical and ordinal demographic variables (e.g., race, gender) are coded with numbers. We want to add category labels to these.

```r
# ethnicity/race
final$ethnic2 <- factor(final$ethnic2,
                        levels = c(1,2,3,4,5,6),
                        labels = c("African American",
                                   "Asian American",
                                   "European American",
                                   "Latino/Latina",
                                   "Native American",
                                   "Other"))

# gender
final$gender <- factor(final$gender,
                       levels = c(1,2),
                       labels = c("Male", "Female"))

# income
final$yyour <- factor(final$yyour,
                      levels = c(1,2,3,4,5,6,7,8),
                      labels = c("Under $15,000",
                                 "$15,001 - $25,000",
                                 "$25,000 - $35,000",
                                 "$35,001 - $50,000",
                                 "$50,001 - $75,000",
                                 "$75,001 - $100,000",
```

2

```
                                        "$100,001 - $150,000",
                                        "Over $150,000"))

# education
final$youred <- factor(final$youred,
                       levels = c(1,2,3,4,5),
                       labels = c("Less than high school",
                                  "High school graduation or equivalent",
                                  "Some college", "College graduation",
                                  "Professional / Post-graduate degree"))
```

Create condition variables for analyses

Conidition was recorded with a dummy code (0 = Low Power; 1 = High Power). We will convert these to effect codes with a 1 unit difference, weighted by the number of participants randomly assigned to the other condition. Weighted effect codes will produce regression coefficients that represent the mean difference between the groups.

```
# how many participants were assigned to each condition?
lowN <- nrow(final[final$Cond == 0,])
highN <- nrow(final[final$Cond == 1,])

# create weights
final$effect[final$Cond == 0] <- -highN / finalCases
final$effect[final$Cond == 1] <- lowN / finalCases
```

There were 299 and 280 participants in the Low and High Power conditions, respectively.

We will also turn the original dummy code into a factor, which will be useful for plotting and tables.

```
final$Cond <- factor(final$Cond,
                     levels = c(0,1),
                     labels = c("Low Power", "High Power"))
```

Exploratory data analysis

Before we run our preferred model, we should use exploratory data analysis to examine the distributions of variables and investigate the potential influence of order effects on the dependent variables.

Descriptive summaries of demographic variables

First, we will look at the demographic composition of our sample.

```
# Gender
table(final[,"gender"])
```

```
##
##   Male Female
##    222    357
```

```
# Age
describe(final[,"age"])
```

```
##   vars   n  mean    sd median trimmed mad min max range skew kurtosis   se
## 1    1 579 32.06 13.17     29   30.26 8.9  18 227   209 6.24    81.73 0.55
```

```
# Race/ethnicity
prop.table(table(final$ethnic2))
```

```
##
##  African American    Asian American European American    Latino/Latina
##        0.07599309        0.05872193        0.68221071       0.05872193
##   Native American             Other
##        0.01208981        0.11226252
```

```
# Income
prop.table(table(final[,"yyour"]))
```

```
##
##      Under $15,000   $15,001 - $25,000   $25,000 - $35,000
##         0.20934256          0.15224913          0.14532872
##   $35,001 - $50,000   $50,001 - $75,000  $75,001 - $100,000
##         0.16782007          0.19377163          0.06747405
## $100,001 - $150,000      Over $150,000
##         0.05190311          0.01211073
```

```
# Education
prop.table(table(final$youred))
```

```
##
##          Less than high school High school graduation or equivalent
##                     0.01036269                           0.09844560
##                   Some college                   College graduation
##                     0.35233161                           0.42314335
##  Professional / Post-graduate degree
##                     0.11571675
```

Overall, the sample looks similar to other Mechanical Turk samples.

Examine distributions of dependent variables

Next, we will look at distributions of the dependent variables to assess whether there are issues with the way the data were recorded and whether there are obvious problems with the distributions.

Individual items First, we should examine the individual items. Are there issues with how the data were recorded (e.g., impossible values, reverse keyed).

Before we look at the items, we will need to score the items of the Reading the Mind in the Eyes (RME) test.

```
# the correct answers
answers <- c(1, 2, 3, 2, 3, 2, 3, 1, 4, 1)
answers <- answers + 18 # responses were recorded as 19 through 22

# for each item, initialize a score column with 0s, change to 1 if the
# participant got the right answer
for (i in 1:10) {
  colName <- names(final)[68 + i]
  newName <- paste(colName, "score", sep = "")
  final[, newName] <- 0
  final[final[, colName] == answers[i], newName] <- 1
}
```

Now we can look at the items.

```
# Which columns represent items?
actionCols <- c(44:50) # action goals
wvsCols <- c(51:56) # World Values Survey trust
gameCol <- 58 # trust game
iriCols <- c(61:67) # IRI
rmeCols <- c(117:126) # RME
dvCols <- c(actionCols, wvsCols, gameCol, iriCols, rmeCols)

# Create long versions of the data set for each scale, where each item rating
# for each participant is on a separate line
variable.names <- names(final)
actionDVs <- melt(final, id = "ResponseID",
                  measure.vars = variable.names[c(actionCols)])
wvsDVs <- melt(final, id = "ResponseID",
               measure.vars = variable.names[c(wvsCols)])
iriDVs <- melt(final, id = "ResponseID",
               measure.vars = variable.names[c(iriCols)])
rmeDVs <- melt(final, id = "ResponseID",
               measure.vars = variable.names[c(rmeCols)])
```

We'll look at plots for each set of variables.

Distributions of Raw Responses to Action/Inaction Goal Items
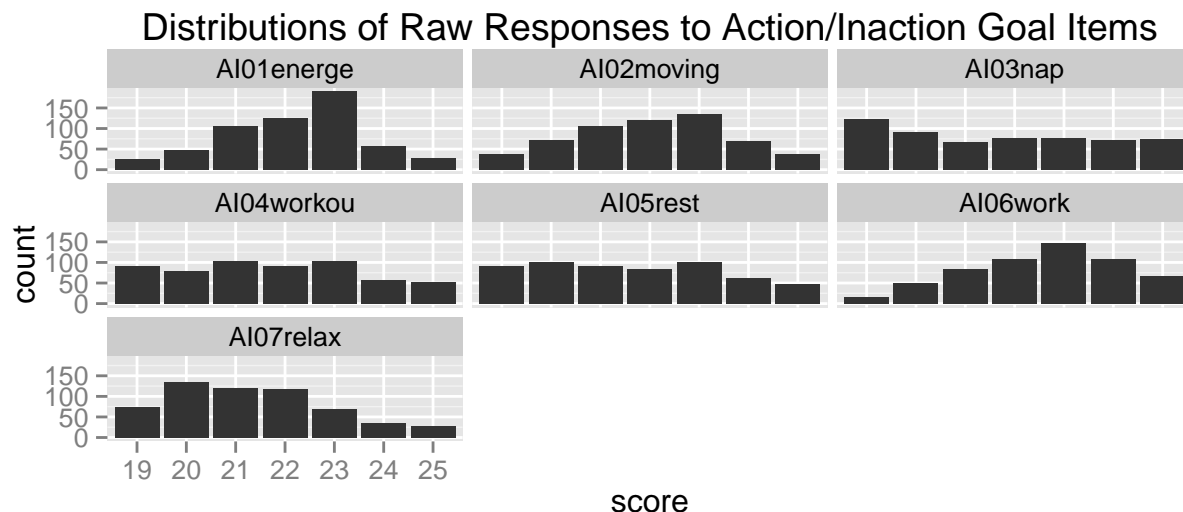
The action/inaction items were recorded as 19 to 26, instead of 1 to 7.

## Distributions of Raw Responses to World Values Survey Trust Items
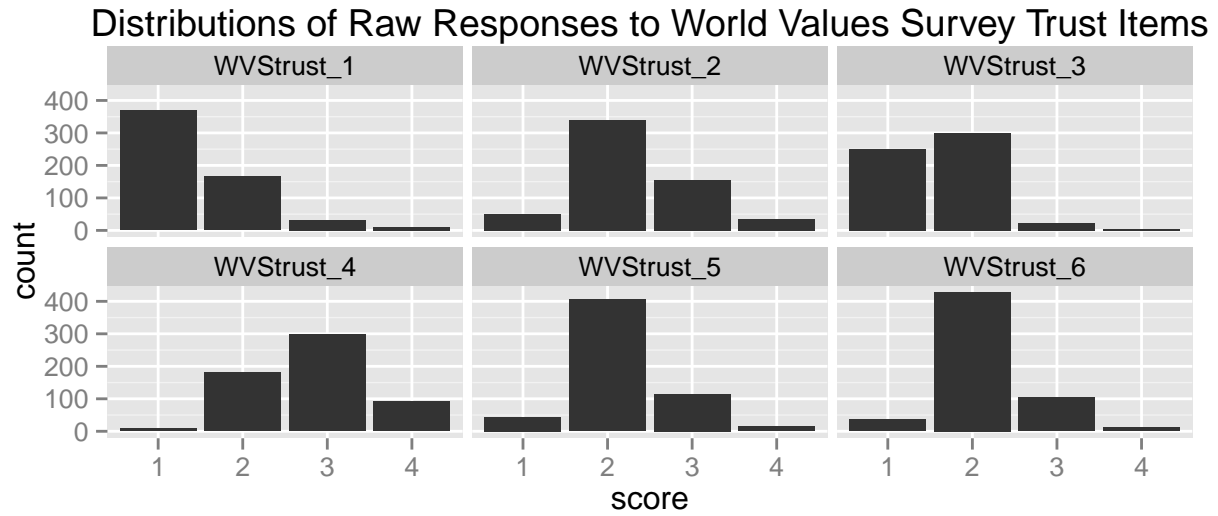
One of the World Values Survey trust items has a severely skewed distribution. This is unsurprising as it indicates that most people trust their families completely.

Historgram of Number of Tickets Given Away in Trust Game
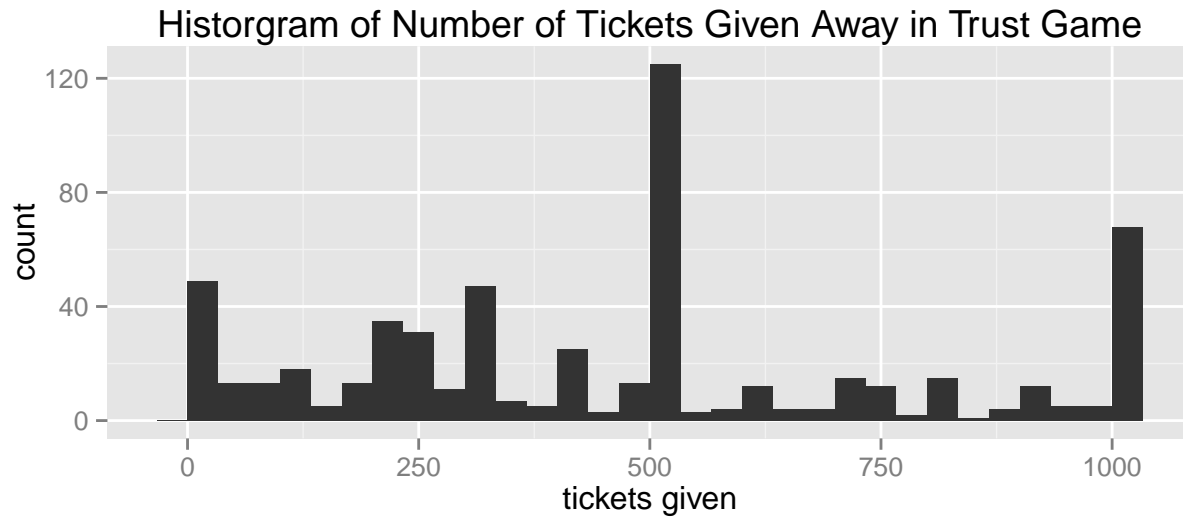
There is a good spread of responses across the Trust Game, with an obvious mode around 500.

Distributions of Raw Responses to IRI Items

Like the action/inaction items, the IRI items were recorded beginning at 19 instead of 1.

Distributions of Raw Responses to IRI Items
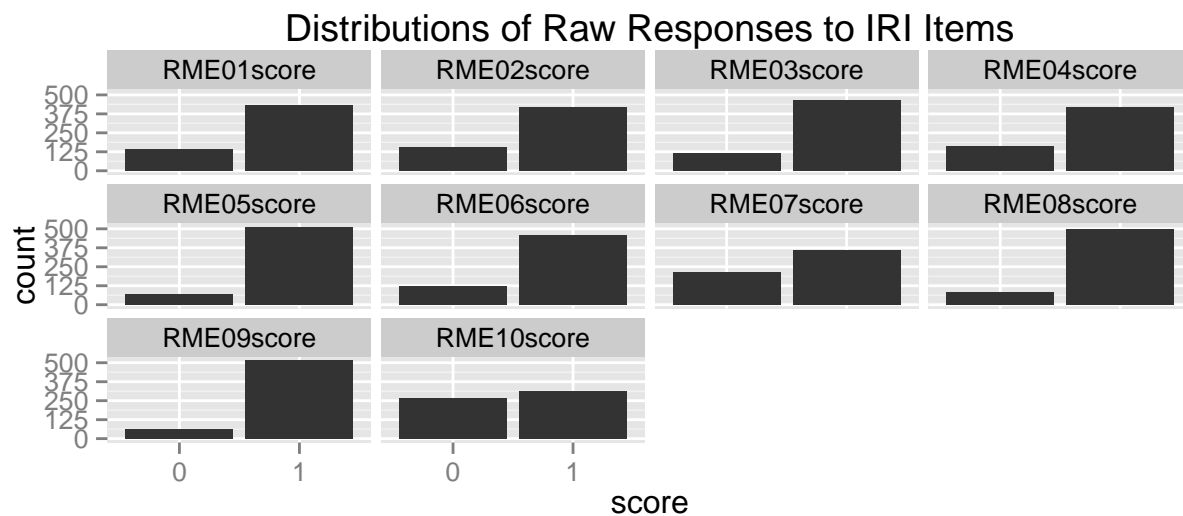
Overall, people were fairly accurate on the Reading the Mind in Eyes items, though some of items show substantial proportions of people who answered incorrectly.

We will recode the items from the action/inaction goals scale and the IRI perspective-taking scale so that they are scored as 1 to 7 and 1 to 5, respectively.

```
# for each item, subtract 18 from the score
final[, c(actionCols, iriCols)] = final[, c(actionCols, iriCols)] - 18
```

Compute Scale Scores & Cronbach's alphas

Next, we will combine the items into scales and examine their intercorrelations. Some of the items need to be reverse scored before being combined with other items.

```
# reverse score inaction goal items from action/inaction goals scale
final[, c("AI03nap", "AI05rest", "AI07relax")] <- 8 - final[, c("AI03nap", "AI05rest", "AI07relax")]

# reverse score non-prespective-taking items from IRI
```

```r
final[, c("IRI01", "IRI04")] <- 6 - final[, c("IRI01", "IRI04")]

# reverse score items on the WVS trust measure so that 4 indicates high trust
final[, wvsCols] <- 5 - final[, wvsCols]


# create lists of key values identifying which items will be included in which
# scales
key.list <- list(action = c(1:7),
                 wvs    = c(8:13),
                 game   = c(14),
                 iri    = c(15:21),
                 rme    = c(22:31))

# link items names to key values
keys <- make.keys(31, key.list, item.labels=c(colnames(final)[dvCols]))

# create scores
scores <- scoreItems(keys,
                     final[, dvCols],
                     totals = FALSE,
                     missing = TRUE,
                     impute = "median")

# add the scores to the data set
final <- cbind(final, scores$scores)

### Create standardized scale scores
stdScores <- scale(scores$scores)
colnames(stdScores) <- paste(colnames(stdScores), ".z", sep = "")
final <- cbind(final, stdScores)

print(scores, short = FALSE)
```

```
## Call: scoreItems(keys = keys, items = final[, dvCols], totals = FALSE,
##     missing = TRUE, impute = "median")
##
## (Unstandardized) Alpha:
##        action  wvs game  iri rme
## alpha    0.84 0.74    1 0.83 0.2
##
## Standard errors of unstandardized Alpha:
##        action   wvs game  iri   rme
## ASE     0.019 0.027  NaN 0.02 0.051
##
## Average item correlation:
##             action  wvs game iri   rme
## average.r     0.43 0.32  NaN 0.4 0.025
##
##  Guttman 6* reliability:
##             action  wvs  game  iri  rme
## Lambda.6      0.87 0.74 0.079 0.82 0.25
##
## Signal/Noise based upon av.r :
```

```
##               action wvs game iri  rme
## Signal/Noise    5.3 2.8  NaN 4.7 0.26
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##        action   wvs    game    iri    rme
## action  0.841 0.108  0.0279 0.1869 -0.0676
## wvs     0.085 0.737  0.1095 0.0787  0.1896
## game    0.026 0.094  1.0000 0.1687 -0.0204
## iri     0.156 0.061  0.1533 0.8258  0.0099
## rme    -0.028 0.073 -0.0092 0.0041  0.2035
##
## Item by scale correlations:
##  corrected for item overlap and scale reliability
##             action   wvs  game   iri   rme
## AI01energe    0.71  0.08 -0.06  0.18 -0.08
## AI02moving    0.61  0.00  0.15  0.13 -0.13
## AI03nap       0.77  0.09  0.11  0.09  0.03
## AI04workou    0.46  0.10 -0.04  0.14 -0.20
## AI05rest      0.78  0.11  0.09  0.11  0.06
## AI06work      0.72  0.07  0.18  0.15 -0.03
## AI07relax     0.64  0.04  0.00  0.08  0.05
## WVStrust_1    0.08  0.43  0.18  0.09  0.13
## WVStrust_2    0.14  0.61  0.38 -0.04  0.01
## WVStrust_3    0.08  0.54 -0.01  0.08  0.20
## WVStrust_4   -0.02  0.50  0.23  0.01  0.11
## WVStrust_5    0.05  0.69  0.27  0.07  0.08
## WVStrust_6    0.03  0.69  0.25  0.09  0.07
## TrustGame_1   0.03  0.11  0.28  0.17 -0.02
## IRI01         0.10  0.10  0.31  0.61  0.17
## IRI02         0.10  0.03  0.32  0.68 -0.04
## IRI03         0.13  0.02  0.26  0.74 -0.15
## IRI04         0.07  0.03  0.51  0.49  0.05
## IRI05         0.15  0.06  0.50  0.69  0.01
## IRI06         0.15  0.08  0.35  0.67 -0.03
## IRI07         0.14  0.03  0.41  0.65 -0.02
## RME01score    0.01 -0.01  0.02  0.04 -0.09
## RME02score   -0.05  0.05 -0.21 -0.02  0.21
## RME03score   -0.01  0.04 -0.12 -0.02  0.24
## RME04score   -0.01  0.09 -0.03  0.00  0.37
## RME05score   -0.05  0.00  0.23 -0.02  0.20
## RME06score    0.00  0.09  0.27  0.08  0.19
## RME07score    0.05  0.03 -0.07 -0.03  0.11
## RME08score   -0.03 -0.01 -0.08  0.04  0.20
## RME09score   -0.02  0.07  0.04 -0.06  0.16
## RME10score   -0.01 -0.04 -0.08 -0.01  0.19
```

It looks like the two trust measures are weakly positively correlated, and the two perspective-taking measures are uncorrelated. For this reason, we will analyze these scales separately.

Examine demographics by condition

Let's take a look at the demographic characteristics of each condition separately.

```
ddply(final, .(Cond), summarize,
      men = length(ResponseID[gender == "Male"]) / length(ResponseID),
      women = length(ResponseID[gender == "Female"]) / length(ResponseID),
      mean.age = mean(age),
      perc.white = length(ResponseID[ethnic2 == "European American"]) / length(ResponseID))
```

```
##          Cond       men      women mean.age perc.white
## 1  Low Power 0.3645485 0.6354515 33.14381  0.7190635
## 2 High Power 0.4035714 0.5964286 30.91071  0.6428571
```

Examine order effects

It is possible that the order in which the dependent measures were presented affected the sizes of the effects. For example, if the manipulation wears off quickly, participants may show larger differences on measures immediately following the manipulation than on later measures.

First we will need to convert the randomization variables to orders. We will focus on the order of presentation of action, trust, and perspective taking.

```
extractOrder <- function(block, var) {
  # given a block ID and a block order variable, return a vector containing
  # the order of that block for participant
  order <- rep(2, length(var))
  order[grepl(paste("^", block, sep=""), var)] <- 1
  order[grepl(paste(block, "$", sep=""), var)] <- 3

  return(order)
}

# extract orders
final$trust.order <- extractOrder("FL_89", final$DO.BR.FL_54)
final$action.order <- extractOrder("FL_90", final$DO.BR.FL_54)
final$perspective.order <- extractOrder("FL_91", final$DO.BR.FL_54)
```

Now, we will graph the responses to the dependent variables by condition and order.

Violin Plots of Action Goals Scores by Condition and Order of Task in Surv
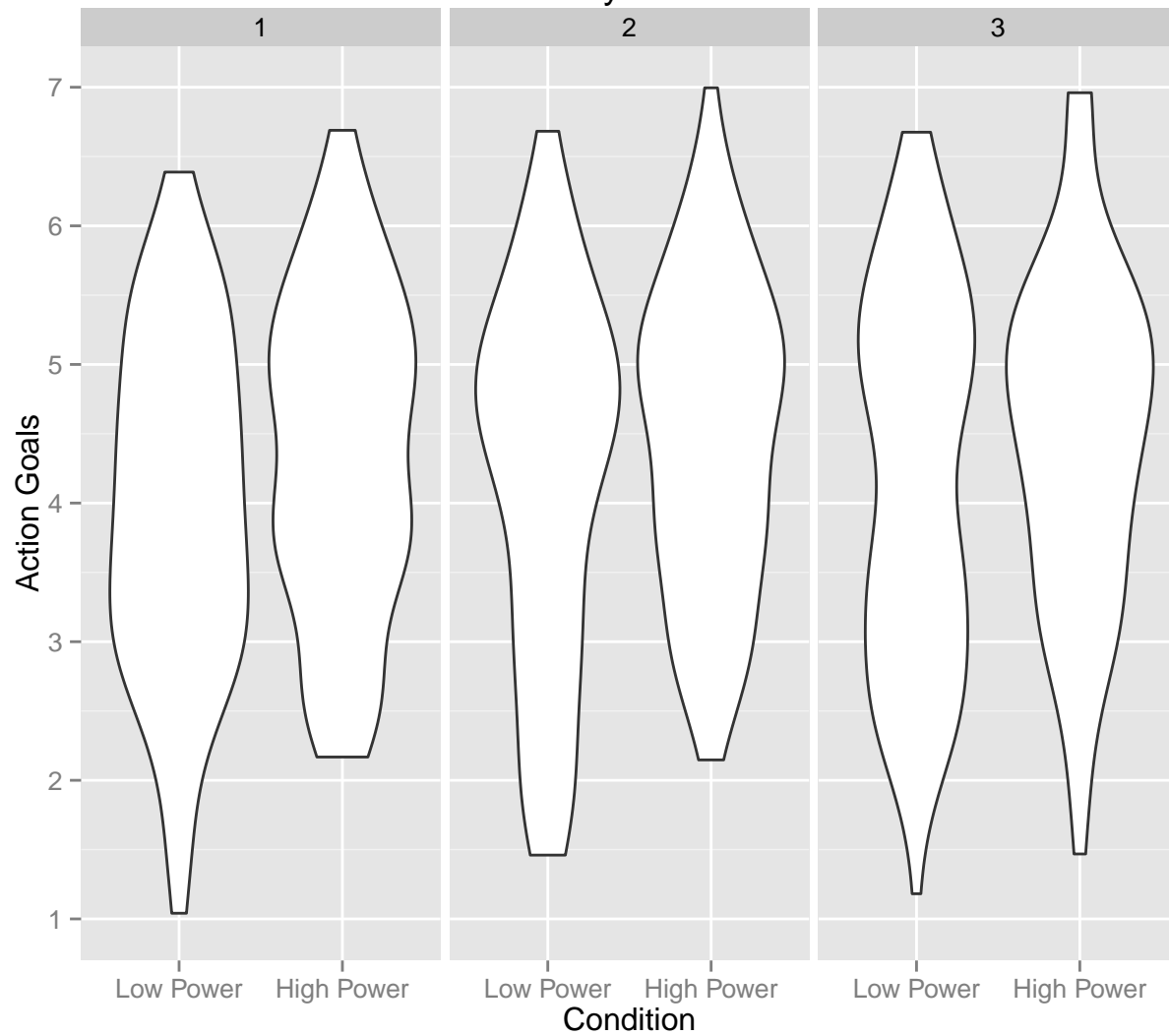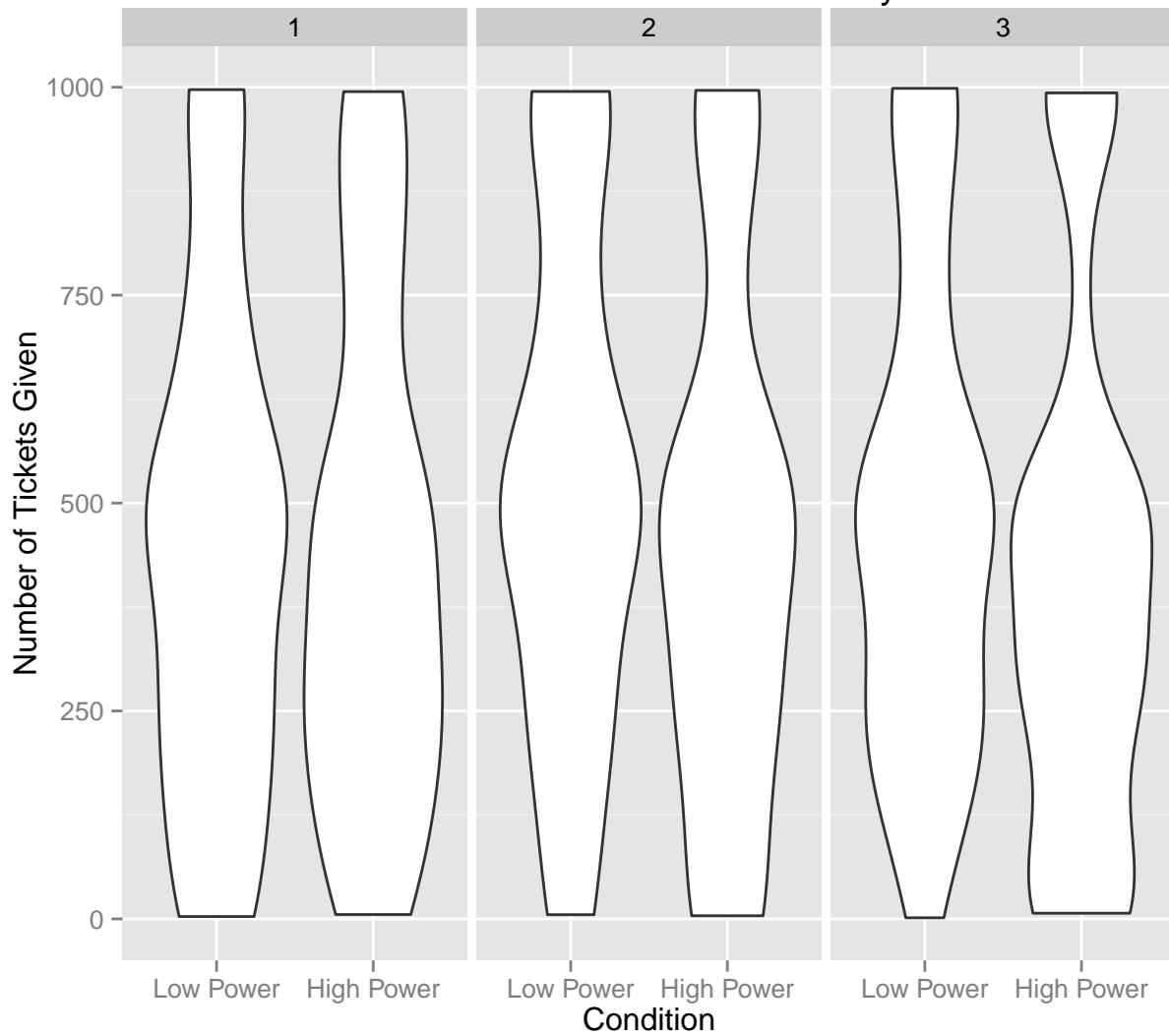

Overall, it looks like there are similar differences by condition at each order, with possible attentuation when the scale is presented last.

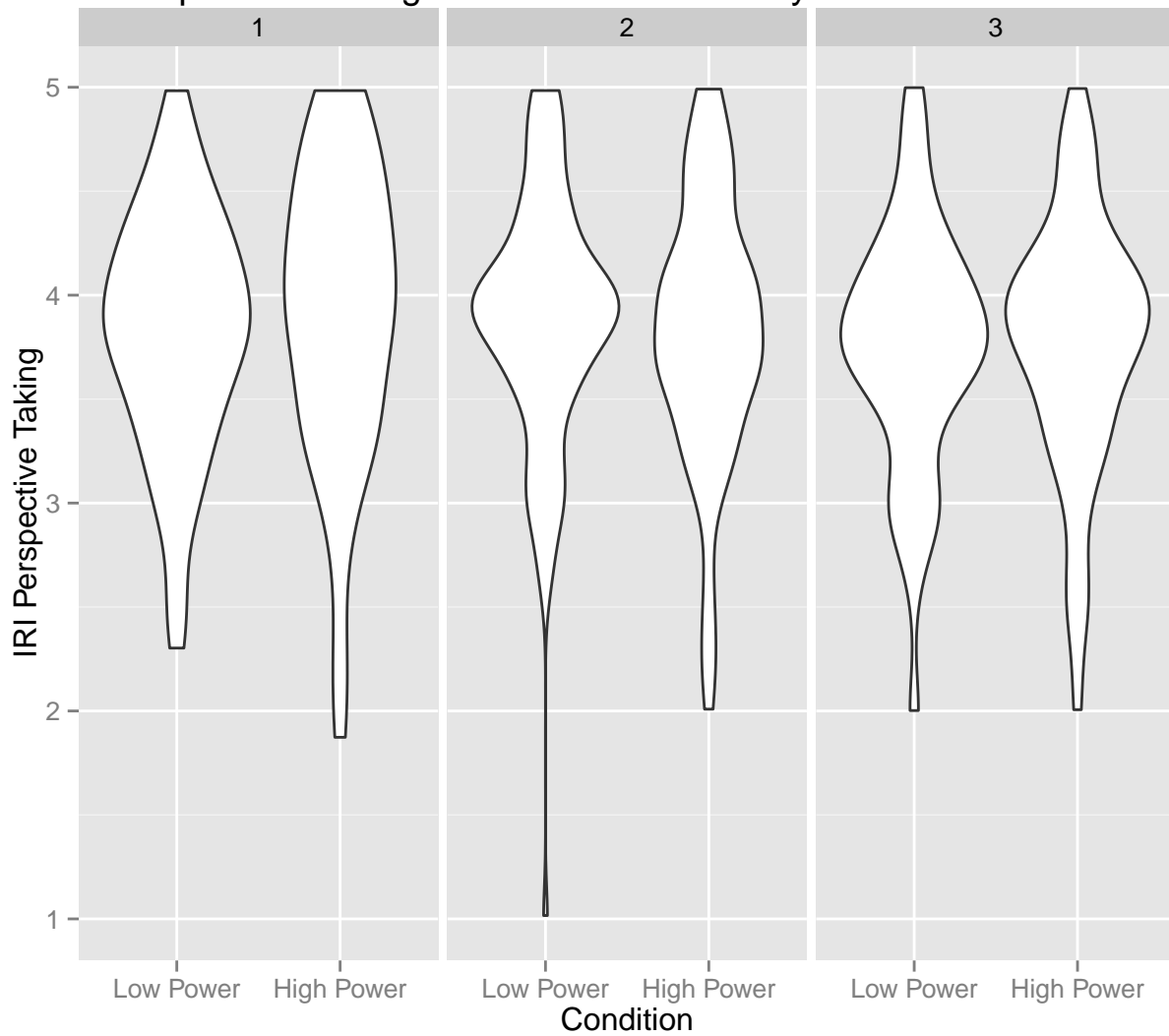## Plots of World Values Survey Trust Scores by Condition and Order of Task i
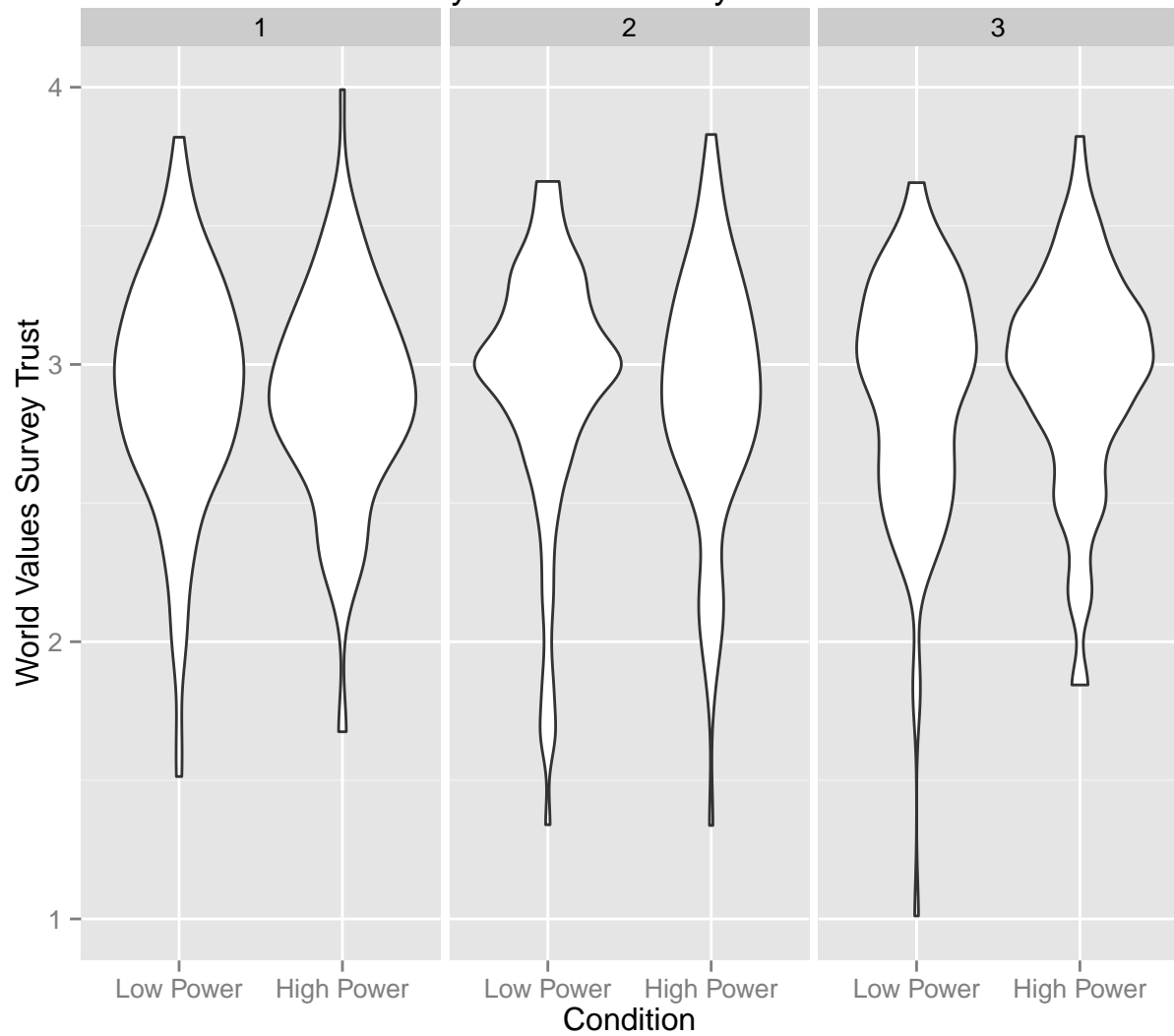
Despite some outlying respones, the overall pattern by condition looks similar regardless of presentation order.

Again, the overall pattern by condition looks similar regardless of presentation order.

Again, presentation order does not seem to influence the pattern of responses by condition.

Plots of Reading the Mind in the Eyes Scores by Condition and Order of Tas
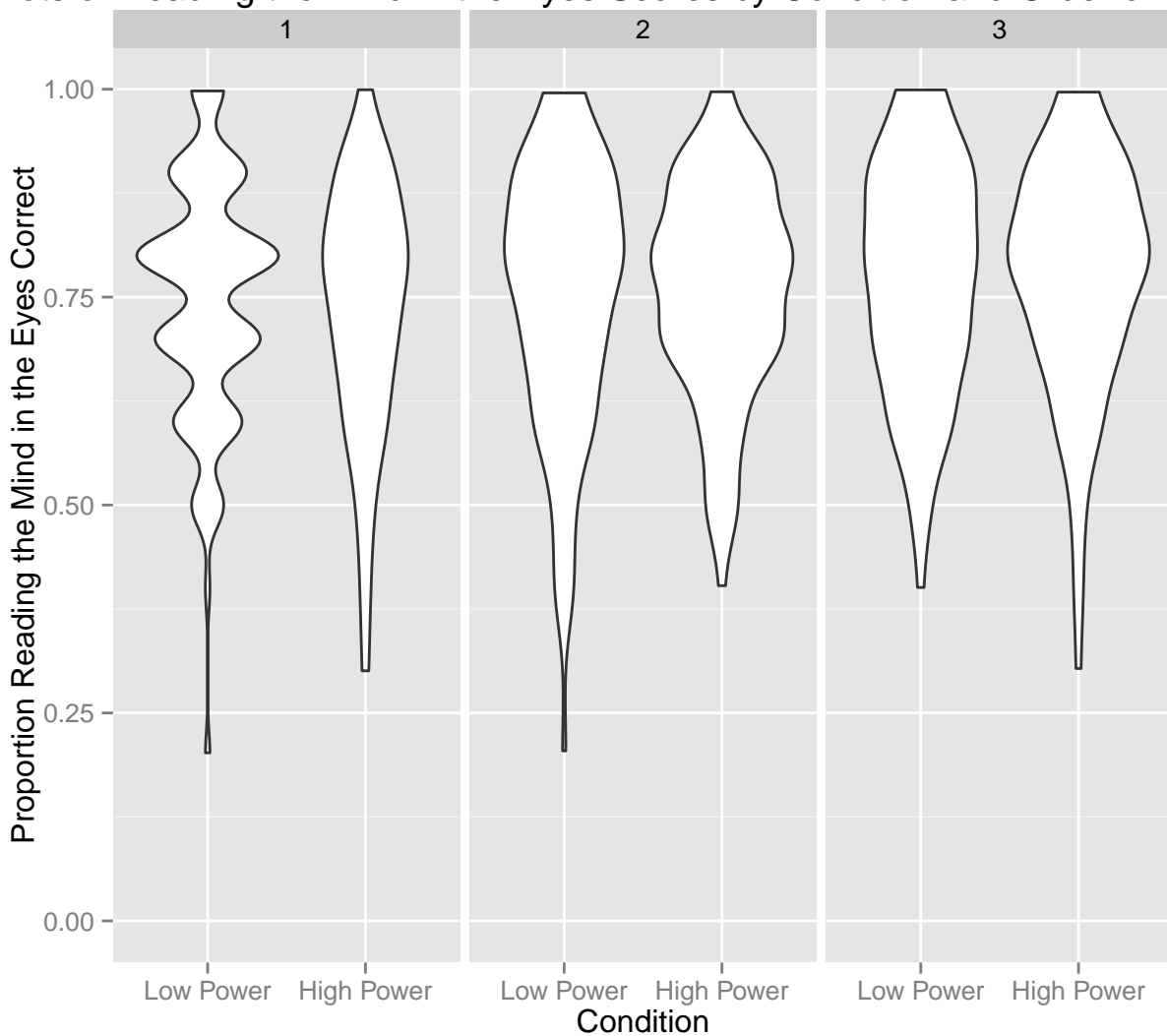
Once again, presentation order does not seem to influence the pattern of responses by condition.

Overall, it seems that order effects are not an important consideration in this study.

Confirmatory analyses

Finally, we will use t-tests to test for differences between conditions on our five dependent measures. Because we are running five tests, a Bonferroni correction would allow us to reject the null at p < .01.

```
# Action Goals
with(final, t.test(action ~ Cond))
```

```
##
##  Welch Two Sample t-test
##
## data:  action by Cond
## t = -3.128, df = 575.31, p-value = 0.001849
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.5190479 -0.1186390
## sample estimates:
##  mean in group Low Power mean in group High Power
##                 4.112279                 4.431122
```

```
# World Values Survey Trust
with(final, t.test(wvs ~ Cond))
```

```
##
##  Welch Two Sample t-test
##
## data:  wvs by Cond
## t = 0.14892, df = 576.82, p-value = 0.8817
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.06374644  0.07420590
## sample estimates:
##  mean in group Low Power mean in group High Power
##                 2.914158                 2.908929
```

```
# Trust Game
with(final, t.test(game ~ Cond))
```

```
##
##  Welch Two Sample t-test
##
## data:  game by Cond
## t = 2.1035, df = 571.76, p-value = 0.03585
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##    3.535029 103.137284
## sample estimates:
##  mean in group Low Power mean in group High Power
##                 487.2040                 433.8679
```

```
# IRI Perspective-Taking
with(final, t.test(iri ~ Cond))
```

```
##
##  Welch Two Sample t-test
##
## data:  iri by Cond
## t = -0.15316, df = 562.25, p-value = 0.8783
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.11013844  0.09420437
## sample estimates:
##  mean in group Low Power mean in group High Power
##                 3.824176                 3.832143
```

```
# Reading the Mind in the Eyes
with(final, t.test(rme ~ Cond))
```

```
##
##  Welch Two Sample t-test
##
## data:  rme by Cond
## t = 0.231, df = 577, p-value = 0.8174
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02094263  0.02652553
## sample estimates:
##  mean in group Low Power mean in group High Power
##                0.7588629                0.7560714
```

After Bonferroni correction, we find a signficant difference only for action.