Physical Pragmatics (decider_1)

Preprocessing

```
# Read in each partition of the participant data.
# NOTE: VEOnwX4 had a duplicate entry. Deleted the latter one.
data_6 = tibble()
exclusions = tibble()
for (data_path in c(data_0_path, data_1_path, data_2_path)) {
  # Read in the participant data for this partition.
  data_0 = read_csv(file.path(data_path, "raw_data.csv"), quote="~")

  # Read in the MTurk results file for this partition.
  mturk_results = read_csv(file.path(data_path, "mturk_results.csv")) %>%
    mutate(Answer.surveycode=str_sub(Answer.surveycode, 3)) %>%
    rename(unique_id=Answer.surveycode)

  # Raise an error if there is a mismatch between the data and MTurk results
  # file.
  mismatch = length(setdiff(data_0$unique_id, mturk_results$unique_id)) != 0
  duplicates = length(unique(data_0$unique_id)) != length(data_0$unique_id)
  if (mismatch | duplicates) {
    stop(paste("There's a mismatch/duplicate between the data and the MTurk ",
               "results file.", sep=""))
  }

  # Convert the JSON string into JSON.
  data_1 = lapply(data_0$results, fromJSON)

  # Extract the trial information for each participant and stack them.
  data_3 = tibble()
  for (p in 1:length(data_1)) {
    # Trim the map and add the participant ID back in.
    data_2 = data_1[p][[1]]$trials %>%
      as.data.frame() %>%
      gather(type, num, trial_num, exclusion_num) %>%
      na.omit() %>%
      mutate(type=gsub("num", "", type)) %>%
      mutate(target=as.numeric(target), trial=paste(type, num, sep=""),
             object=data_1[p][[1]]$setup$object,
             unique_id=data_0$unique_id[p],
             age=as.numeric(data_1[p][[1]]$subject_information$age)) %>%
      select(-type, -num)

    # Stack the trial information for the current participant.
    data_3 = rbind(data_3, data_2)
  }
```

```
  # Exclude participants that said they couldn't walk through one of the doors.
  data_5 = tibble()
  for (participant in unique(data_3$unique_id)) {
    data_4 = data_3 %>%
      filter(unique_id==participant)

    if (filter(data_4, trial=="exclusion_2")[,"target"]==0) {
      data_5 = rbind(data_5, data_4)
    }
    else if (filter(data_4, trial=="exclusion_2")[,"target"]==1) {
      exclusions = rbind(exclusions, data_4)
    }
  }

  # Write the preprocessed data.
  write_csv(data_5, file.path(data_path, "data.csv"))

  # Append the preprocessed data.
  data_6 = rbind(data_6, data_5)
}
```

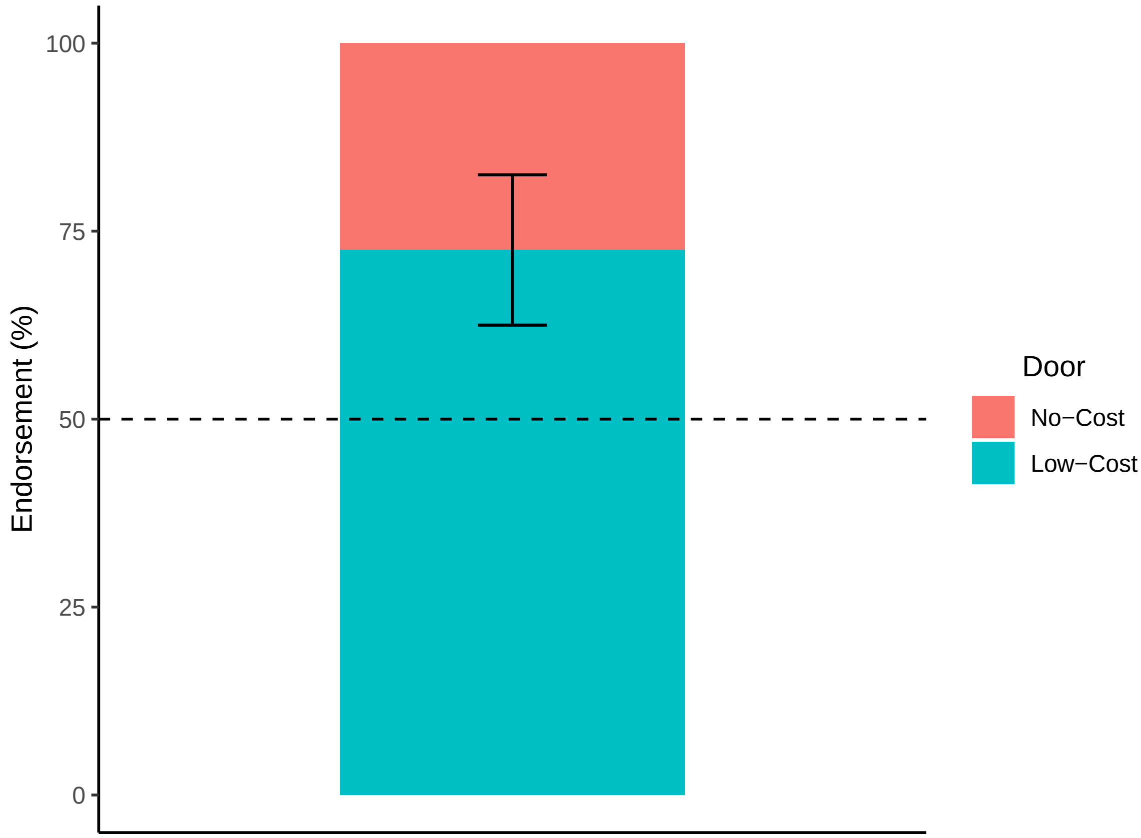
Compute Door Endorsement

a priori Labels

First, we'll analyze the participant endorsement (N=80; M=37.84 years, SD=12.22 years) for the low-cost door (i.e., if they believed the low-cost door to be more likely to have a communicative meaning) using *a priori* labels of door difficulty.

```
# Read in the preprocessed data.
data_6 = read_csv(file.path(data_0_path, "data.csv")) %>%
  rbind(read_csv(file.path(data_1_path, "data.csv"))) %>%
  rbind(read_csv(file.path(data_2_path, "data.csv")))

# Filter the trial of interest.
data_7 = data_6 %>%
  filter(trial=="trial_1")

# Set up the bootstrap functions.
compute_mean = function(data, indices) {
  return(mean(data[indices]))
}

compute_bootstrap = function(data) {
  simulations = boot(data=data$target,
                     statistic=compute_mean,
                     R=10000)

  return(boot.ci(simulations, type="perc")$perc)
}

# Set the seed and compute the 95% bootstrapped CI.
set.seed(seed)
```

```r
bootstrap_data = compute_bootstrap(data_7)
lower_ci = bootstrap_data[4] * 100
upper_ci = bootstrap_data[5] * 100

# Convert individual judgments to aggregate percentages.
data_8 = data_7 %>%
  group_by(target) %>%
  summarize(percentage=n()/nrow(data_7)*100) %>%
  mutate(target=as.character(target))

# Plot the data (using our labels of door difficulty).
plot_0 = data_8 %>%
  ggplot(aes(x="", y=percentage, fill=target)) +
  geom_bar(stat="identity", width=0.5) +
  geom_errorbar(aes(ymin=lower_ci, ymax=upper_ci), width=0.1) +
  geom_hline(yintercept=50, linetype="dashed", color="black") +
  theme_classic() +
  theme(aspect.ratio=1.0,
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        legend.title=element_text(hjust=0.5)) +
  ylab("Endorsement (%)") +
  scale_fill_discrete(name="Door",
                      limits=c("0", "1"),
                      labels=c("No-Cost", "Low-Cost"))
plot_0
```




```r
# Compute a binomial test with the alternative hypothesis that the participant
# endorsement of the low-cost door is above chance.
# NOTE: Computing a two-sided binomial test for "conservativeness".
binom.test(x=sum(data_7$target), n=length(data_7$target), p=0.5,
           alternative="two.sided")
```

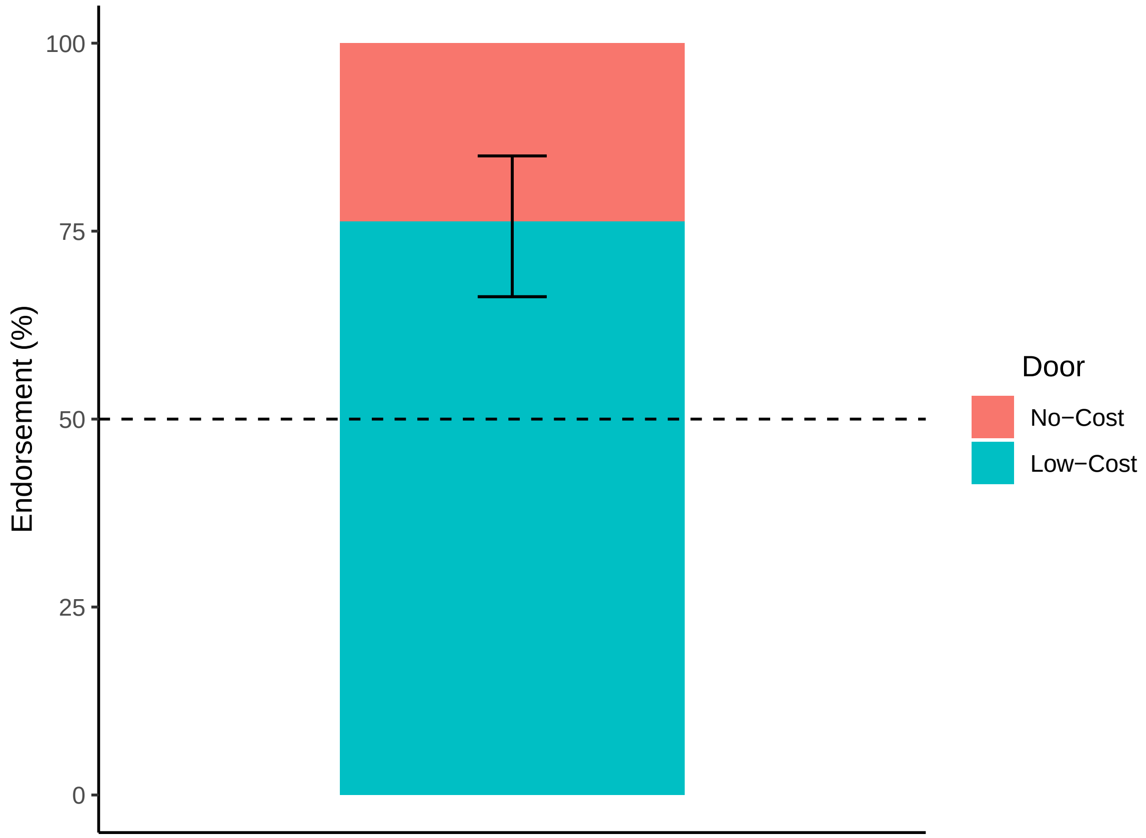
```
##
##  Exact binomial test
##
## data:  sum(data_7$target) and length(data_7$target)
## number of successes = 58, number of trials = 80, p-value = 7.011e-05
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6137566 0.8189624
## sample estimates:
## probability of success
##                  0.725
```

Participant Labels

Now, we'll analyze the participant endorsement (N=80; M=37.84 years, SD=12.22 years) for the low-cost door (i.e., if they believed the low-cost door to be more likely to have a communicative meaning) using participant labels of door difficulty.

```r
# Condition the target data based on which door the participants saw as more
# challenging to walk through (instead of using our cost labels).
data_9 = data_6 %>%
```

```r
  filter(trial %in% c("trial_1", "exclusion_1")) %>%
  spread(trial, target) %>%
  mutate(target=ifelse(trial_1==1 & exclusion_1==1, 1,
                       ifelse(trial_1==0 & exclusion_1==0, 1, 0)))

# Set the seed and compute the 95% bootstrapped CI.
set.seed(seed)
bootstrap_data = compute_bootstrap(data_9)
lower_ci = bootstrap_data[4] * 100
upper_ci = bootstrap_data[5] * 100

# Convert individual judgments to aggregate percentages.
data_10 = data_9 %>%
  group_by(target) %>%
  summarize(percentage=n()/nrow(data_9)*100) %>%
  mutate(target=as.character(target))

# Plot the data (using participant labels of door difficulty).
plot_1 = data_10 %>%
  ggplot(aes(x="", y=percentage, fill=target)) +
  geom_bar(stat="identity", width=0.5) +
  geom_errorbar(aes(ymin=lower_ci, ymax=upper_ci), width=0.1) +
  geom_hline(yintercept=50, linetype="dashed", color="black") +
  theme_classic() +
  theme(aspect.ratio=1.0,
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        legend.title=element_text(hjust=0.5)) +
  ylab("Endorsement (%)") +
  scale_fill_discrete(name="Door",
                      limits=c("0", "1"),
                      labels=c("No-Cost", "Low-Cost"))
plot_1
```




```r
# Compute a binomial test on the alternative hypothesis that the participant
# endorsement for the low-cost door is above chance.
# NOTE: Computing a two-sided binomial test for "conservativeness".
binom.test(x=sum(data_9$target), n=length(data_9$target), p=0.5,
           alternative="two.sided")
```

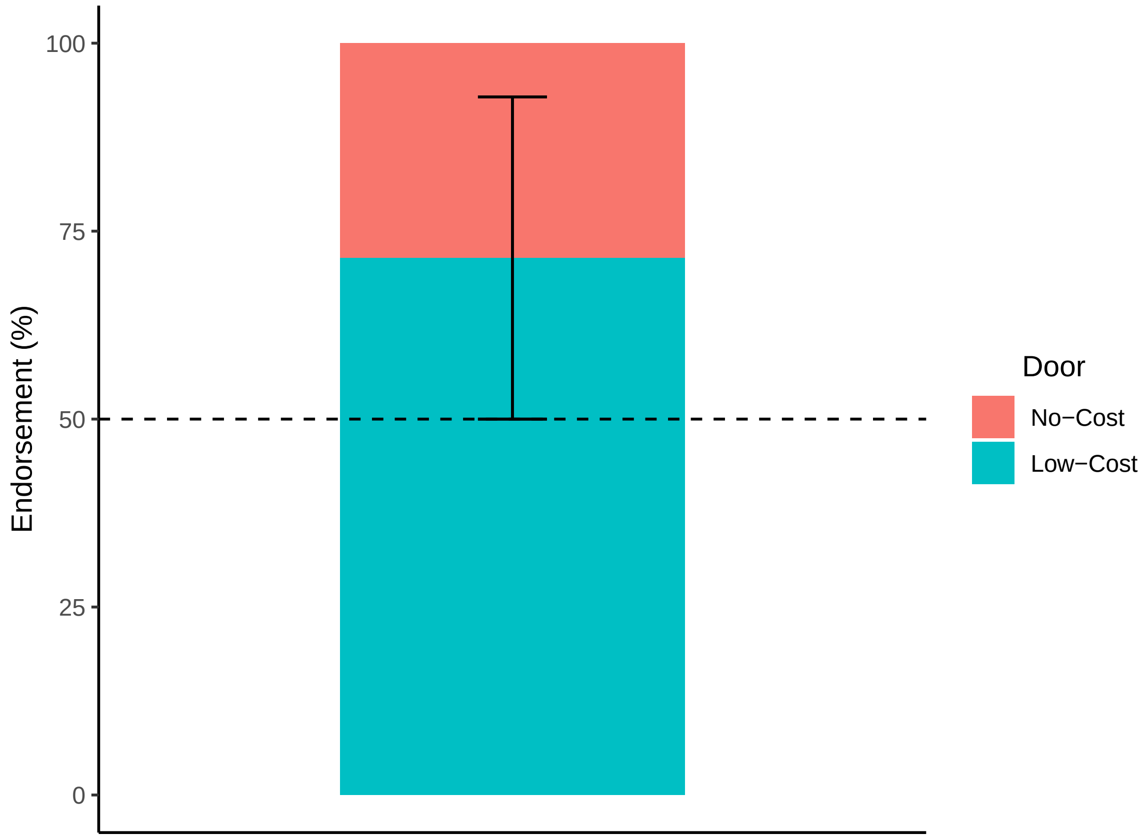
```
##
##  Exact binomial test
##
## data:  sum(data_9$target) and length(data_9$target)
## number of successes = 61, number of trials = 80, p-value = 2.732e-06
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6542230 0.8505462
## sample estimates:
## probability of success
##                 0.7625
```

Analyze Excluded Participants

Because our exclusion question was potentially ambiguous, here we analyze the data from excluded participants to see if their judgments align with our main results.

```r
# Filter the trial of interest.
data_11 = exclusions %>%
  filter(trial=="trial_1")
```

```r
# Set the seed and compute the 95% bootstrapped CI.
set.seed(seed)
bootstrap_data = compute_bootstrap(data_11)
lower_ci = bootstrap_data[4] * 100
upper_ci = bootstrap_data[5] * 100

# Convert individual judgments to aggregate percentages.
data_12 = data_11 %>%
  group_by(target) %>%
  summarize(percentage=n()/nrow(data_11)*100) %>%
  mutate(target=as.character(target))

# Plot the data (using our labels of door difficulty).
plot_2 = data_12 %>%
  ggplot(aes(x="", y=percentage, fill=target)) +
  geom_histogram(stat="identity", width=0.5) +
  geom_errorbar(aes(ymin=lower_ci, ymax=upper_ci), width=0.1) +
  geom_hline(yintercept=50, linetype="dashed", color="black") +
  theme_classic() +
  theme(aspect.ratio=1.0,
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        legend.title=element_text(hjust=0.5)) +
  ylab("Endorsement (%)") +
  scale_fill_discrete(name="Door",
                      limits=c("0", "1"),
                      labels=c("No-Cost", "Low-Cost"))
plot_2
```




```
# Compute a binomial test on the alternative hypothesis that the participant
# endorsement for the low-cost door is above chance.
# NOTE: Computing a two-sided binomial test for "conservativeness".
binom.test(x=sum(data_11$target), n=length(data_11$target), p=0.5,
           alternative="two.sided")
```

```
##
##  Exact binomial test
##
## data:  sum(data_11$target) and length(data_11$target)
## number of successes = 10, number of trials = 14, p-value = 0.1796
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4189647 0.9161107
## sample estimates:
## probability of success
##              0.7142857
```