MR Replication Project

Vincent Caruso

November 11, 2015

VERY BRIEF SUMMARY of the study: The ability of men and women to perform mental rotation tasks was investigated. For each trial, a subject was shown pairs of images on a screen, one rotated relative to the other by a discrete angle (ranging from 0 to 120 degrees, in 20 degree increments). Some pairs of objects were identical to each other (these are referred to as "same" trials), some were similar but different ("different" trials). For each trial, the subject had to decide if the objects were identical or not by mentally rotating one of the objects. Accuracy and reaction time were measured for each trial, resulting in four dependent variables: accuracy for same trials, accuracy for different, reaction time for same trials, and reaction time for different.

Before we Begin!

Problems with Results:

Some of our Mann-Whitney tests give the same (or very similar) rank sum score as the original paper, while other values are different from those in the paper. We believe this is due to the data that were available to us: the investigators administered 448 trials per subject, but the data made public was only a summary, with one observation per degree of rotation (7) per dependent variable (4), for a total of 28 (summarized) observations per subject. Since we are looking at the summarized values of the original 448, we don't have access to the variability in the data that the paper was looking at, which likely changes some of the ranks that we calculate relative to those in the original study.

Give me libraries:

```
# provided to install packages that may be missing
#suppressWarnings(suppressMessages(install.packages("dplyr")))
#suppressWarnings(suppressMessages(install.packages("tidyr")))
#suppressWarnings(suppressMessages(install.packages("ggplot2")))
suppressWarnings(suppressMessages(library(dplyr)))
suppressWarnings(suppressMessages(library(tidyr)))
suppressWarnings(suppressMessages(library(ggplot2)))
```

Import the MR data from a tab-delimited text file, and convert sex variable to something sensical

```
MR_path <- "~/Downloads/"  #!!!!! Change this to the directory where you saved
                           #"Mental_Rotation_Data_Rfriend.txt" !!!!!
MR_file <- "Mental_Rotation_Data_Rfriend.txt"
MR_data <- read.table(paste(MR_path, MR_file, sep = ""), sep = "\t", header = TRUE)

#change the data in the table to F if the sex = 1, otherwise M
for (i in 1:length(MR_data$Sex)){
  MR_data$Sex[i] <- ifelse(MR_data$Sex[i] == 1, "F", "M")
}

# Table to verify that our number of participants matched the values in the paper
# This can be found on page 3 under heading "Method", sub-heading "Participants":
# 25 females, 26 males
table(MR_data$Sex)
```

```
 F  M 
25 26 
```

Before analysis, the dataset needs to be wrangled a bit. The independent variable 'degrees of rotation' is split over seven different columns, and we want those gathered into one column for linear regression analyses. But since there are several dependent variables, we first want to split the dataset into four subsets, one for each experiment/dependent variable.

```
# For each, keep the first two columns: participant, sex, digit ratio, and Intuitive Physics
# Test (IPT)
split_data <- function(x){
  MR_data %>% select(1:4,starts_with(x[1]))
}
MR_rt_same <- split_data("RT_same_")  #reaction time for "same" trials
MR_rt_diff <- split_data("RT_diff_")  #reaction time for "different" trials
MR_acc_same <- split_data("Acc_same_")  #accuracy for "same" trials
MR_acc_diff <- split_data("Acc_diff_")  #accuracy for "different" trials
MR_misc <- MR_data %>% select(1:4) #Columns 3 and 4 contain variables not related to
                                   #degrees of rotation. These are used in a separate
                                   #analysis at the end.
```

Take a look at the subset, and check for NA's

```
glimpse(MR_acc_diff)
```

```
Observations: 51
Variables: 11
$ Participant            (int) 3059, 10511, 10512, 10513, 10516, 14514...
$ Sex                    (chr) "F", "F", "F", "F", "F", "F", "F", "F",...
$ Digit.Ratio            (dbl) 0.99, 1.00, 0.96, 0.96, 0.89, 0.93, 0.9...
$ Intuitive.Physics.Test (int) 5, 2, 8, 13, 7, 9, 10, 11, 12, 6, 10, 1...
$ Acc_diff_0             (int) 100, 94, 88, 100, 63, 97, 100, 78, 78, ...
$ Acc_diff_20            (int) 100, 81, 84, 94, 63, 94, 100, 78, 81, 1...
$ Acc_diff_40            (int) 97, 84, 91, 94, 63, 94, 97, 56, 63, 91,...
$ Acc_diff_60            (int) 88, 88, 84, 94, 53, 97, 91, 44, 53, 81,...
$ Acc_diff_80            (int) 72, 84, 81, 88, 50, 94, 94, 44, 72, 97,...
$ Acc_diff_100           (int) 97, 75, 81, 84, 50, 88, 94, 41, 66, 81,...
$ Acc_diff_120           (int) 94, 84, 78, 78, 41, 94, 88, 53, 56, 81,...
```

```
table(is.na(MR_acc_diff))# tells us how many values are missing
```

```
FALSE 
  561 
```

Rename the columns of degrees in each subset to just # of degrees. Then gather all the degree columns into a single variable called `degrees`.

```
# function to limit repeated code that gathers all the values into the colnumn name in the parameter
# @param old_data_frame - A data frame that contains the information needed to be gathered
# @param col_name - String value that one would like to name the column
```

```r
# @return - returns the new data frame with the gathered data
gather_degrees <- function(old_data_frame,col_name) {
  colnames(old_data_frame) <- c("participant", "sex", "digit_ratio", "ipt", "0", "20", "40", "60", "80"
                   "100", "120") #rename columns of x
  new_data_frame <- old_data_frame %>% gather("degrees", col_name, 5:11, convert=TRUE) #gather renamed
  colnames(new_data_frame)[6] <- col_name # changes the last column to the specified column name in par
  return (new_data_frame)
}
```

allows us to superimpose the same colnames from the original data frame making it easier to gather information with parital substring matches.

```r
# A look at the structure of data is correct after function call
glimpse(MR_rt_same_tall <- gather_degrees(MR_rt_same,"rt_same"))
```

```
Observations: 357
Variables: 6
$ participant (int) 3059, 10511, 10512, 10513, 10516, 14514, 16509, 16...
$ sex         (chr) "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", ...
$ digit_ratio (dbl) 0.99, 1.00, 0.96, 0.96, 0.89, 0.93, 0.99, 0.97, 0....
$ ipt         (int) 5, 2, 8, 13, 7, 9, 10, 11, 12, 6, 10, 12, 7, 6, 9,...
$ degrees     (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
$ rt_same     (int) 1247, 2458, 3025, 1182, 1840, 1079, 792, 2050, 154...
```

```r
glimpse(MR_rt_diff_tall <- gather_degrees(MR_rt_diff,"rt_diff"))
```

```
Observations: 357
Variables: 6
$ participant (int) 3059, 10511, 10512, 10513, 10516, 14514, 16509, 16...
$ sex         (chr) "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", ...
$ digit_ratio (dbl) 0.99, 1.00, 0.96, 0.96, 0.89, 0.93, 0.99, 0.97, 0....
$ ipt         (int) 5, 2, 8, 13, 7, 9, 10, 11, 12, 6, 10, 12, 7, 6, 9,...
$ degrees     (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
$ rt_diff     (int) 1783, 1848, 3395, 2113, 1701, 1439, 1115, 1935, 19...
```

```r
glimpse(MR_acc_same_tall <- gather_degrees(MR_acc_same,"acc_same"))
```

```
Observations: 357
Variables: 6
$ participant (int) 3059, 10511, 10512, 10513, 10516, 14514, 16509, 16...
$ sex         (chr) "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", ...
$ digit_ratio (dbl) 0.99, 1.00, 0.96, 0.96, 0.89, 0.93, 0.99, 0.97, 0....
$ ipt         (int) 5, 2, 8, 13, 7, 9, 10, 11, 12, 6, 10, 12, 7, 6, 9,...
$ degrees     (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
$ acc_same    (int) 100, 100, 97, 100, 91, 100, 100, 91, 97, 97, 100, ...
```

```r
glimpse(MR_acc_diff_tall <- gather_degrees(MR_acc_diff,"acc_diff"))
```

```
Observations: 357
Variables: 6
```

```
$ participant (int) 3059, 10511, 10512, 10513, 10516, 14514, 16509, 16...
$ sex         (chr) "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", ...
$ digit_ratio (dbl) 0.99, 1.00, 0.96, 0.96, 0.89, 0.93, 0.99, 0.97, 0....
$ ipt         (int) 5, 2, 8, 13, 7, 9, 10, 11, 12, 6, 10, 12, 7, 6, 9,...
$ degrees     (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
$ acc_diff    (int) 100, 94, 88, 100, 63, 97, 100, 78, 78, 97, 94, 100...
```

Now merge the reshaped data back into one dataframe. When this is done, we will have a tall and narrow dataframe with seven rows for each participant, one for each discrete value of degrees of rotation.

```
# Function created to merge two data frames together where the values
# participant, sex, digit_ratio, ipt and degrees columns are the same
# @param data_frame1 the first dataframe that will be merged with data_frame2
# @param data_frame2 the second dataframe that will be merged with data_frame1
MR_data_tall <- data.frame()
merge_data <- function(data_frame1,data_frame2) {
  merge(data_frame1,data_frame2, by = c("participant", "sex",
                         "digit_ratio", "ipt",
                         "degrees"))
}

MR_data_tall <- merge_data(MR_rt_same_tall, MR_rt_diff_tall)
MR_data_tall <- merge_data(MR_data_tall, MR_acc_same_tall)
MR_data_tall <- merge_data(MR_data_tall, MR_acc_diff_tall)

# MR_data_tall$degrees <- as.integer(as.character((MR_data_tall$degrees)))
MR_data_tall$sex <- as.factor(MR_data_tall$sex)

# A look at the structure created from merging the data_frames
glimpse(MR_data_tall)
```

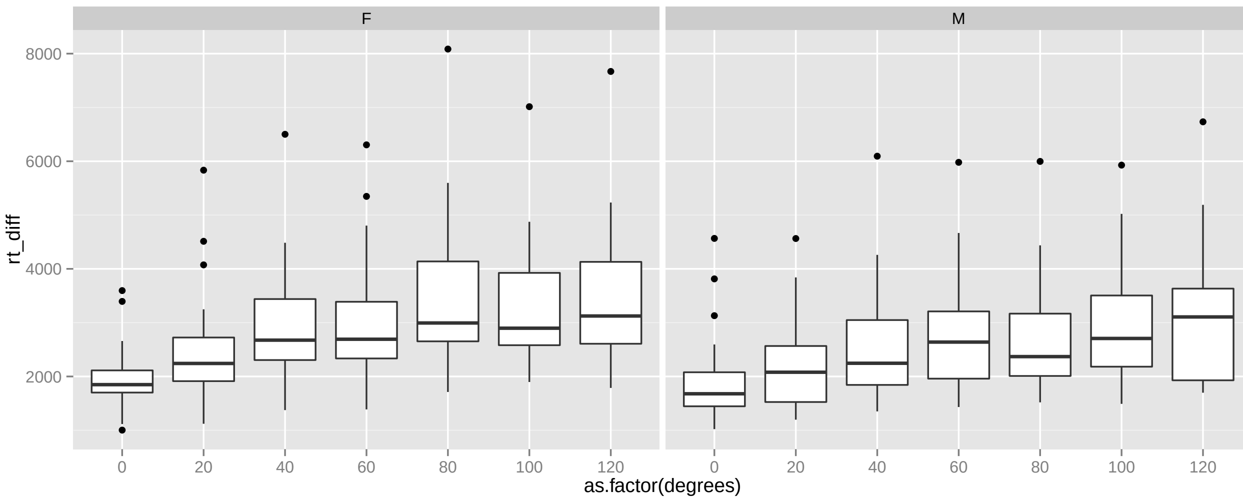
```
Observations: 357
Variables: 9
$ participant (int) 10511, 10511, 10511, 10511, 10511, 10511, 10511, 1...
$ sex         (fctr) F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F...
$ digit_ratio (dbl) 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 0.96, 0....
$ ipt         (int) 2, 2, 2, 2, 2, 2, 2, 8, 8, 8, 8, 8, 8, 8, 13, 13, ...
$ degrees     (int) 0, 100, 120, 20, 40, 60, 80, 0, 100, 120, 20, 40, ...
$ rt_same     (int) 2458, 6037, 4645, 2949, 3632, 4762, 4242, 3025, 33...
$ rt_diff     (int) 1848, 7014, 5233, 4074, 4485, 5347, 5408, 3395, 46...
$ acc_same    (int) 100, 56, 66, 91, 91, 88, 84, 97, 69, 72, 97, 97, 9...
$ acc_diff    (int) 94, 75, 84, 81, 84, 88, 84, 88, 81, 78, 84, 91, 84...
```

Compute inverse efficiency scores for same and different trials, and add them as two new variables

```
MR_data_tall <- MR_data_tall %>% mutate("ie_same" = rt_same/acc_same,
                                        "ie_diff" = rt_diff/acc_diff)
```

Plot boxplots of reaction times for different tests, by gender (this is not part of the replication– no such figure exists in the paper)

```r
MR_data_tall %>%  ggplot(aes(x = as.factor(degrees), y = rt_diff)) +
  geom_boxplot() + facet_wrap(~sex)
```



Find medians at each degree value for each of the four tests. For some reason, these medians were used to generate the plots in Figure 1, page 6

```r
MR_med <- MR_data_tall %>% group_by(sex, degrees) %>%
  summarise(med_rt_same = median(rt_same), med_rt_diff = median(rt_diff),
            med_acc_same = median(acc_same), med_acc_diff = median(acc_diff),
            med_ie_same = median(ie_same), med_ie_diff = median(ie_diff))
# Table to show the median values of the females at each degree of rotation tested
MR_med %>%
  filter(sex == 'F')
```

```
Source: local data frame [7 x 8]
Groups: sex [1]

     sex degrees med_rt_same med_rt_diff med_acc_same med_acc_diff
  (fctr)   (int)       (dbl)       (dbl)        (dbl)        (dbl)
1      F       0        1359        1848          100           94
2      F      20        1768        2242           94           94
3      F      40        2087        2675           91           91
4      F      60        2080        2692           91           88
5      F      80        2494        2994           84           88
6      F     100        2774        2897           78           81
7      F     120        3091        3124           75           81
Variables not shown: med_ie_same (dbl), med_ie_diff (dbl)
```

```r
# Table to show the median values of the males at each degree of rotation tested
MR_med %>%
  filter(sex == 'M')
```

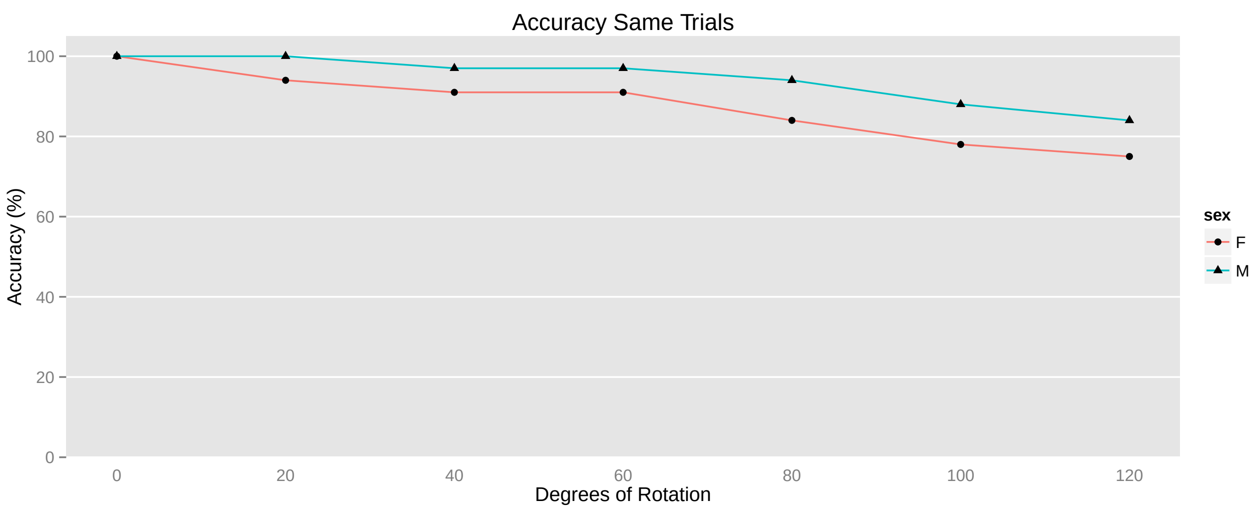
```
Source: local data frame [7 x 8]
Groups: sex [1]

     sex degrees med_rt_same med_rt_diff med_acc_same med_acc_diff
```

```
  (fctr)   (int)      (dbl)      (dbl)       (dbl)       (dbl)
1      M       0     1216.5     1678.5         100          97
2      M      20     1552.5     2079.0         100          97
3      M      40     1761.0     2245.5          97          91
4      M      60     1883.5     2639.0          97          91
5      M      80     2352.0     2369.0          94          91
6      M     100     2531.0     2706.5          88          91
7      M     120     2703.0     3106.5          84          88
Variables not shown: med_ie_same (dbl), med_ie_diff (dbl)
```

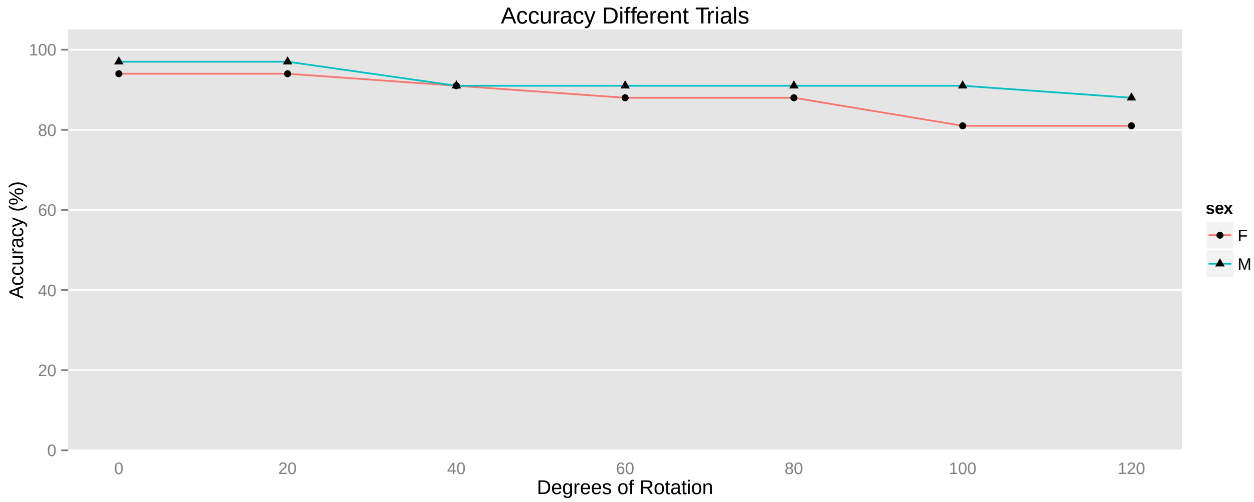
Plot accuracy score medians for same trials. This replicates plot 1 (left-to-right, top-to-bottom) of Figure 1, page 6

```
MR_med %>% ggplot(aes(x = degrees, y = med_acc_same)) +
  geom_path(aes(color = sex)) +
  geom_point(aes(shape = sex)) +
  coord_cartesian(ylim = c(0, 105)) +
  xlab("Degrees of Rotation") +
  ylab("Accuracy (%)") +
  ggtitle("Accuracy Same Trials") +
  scale_x_continuous(breaks=seq(0, 120, 20)) +
  scale_y_continuous(breaks=seq(0, 100, 20)) +
  theme(axis.ticks.x = element_blank(), panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank())
```



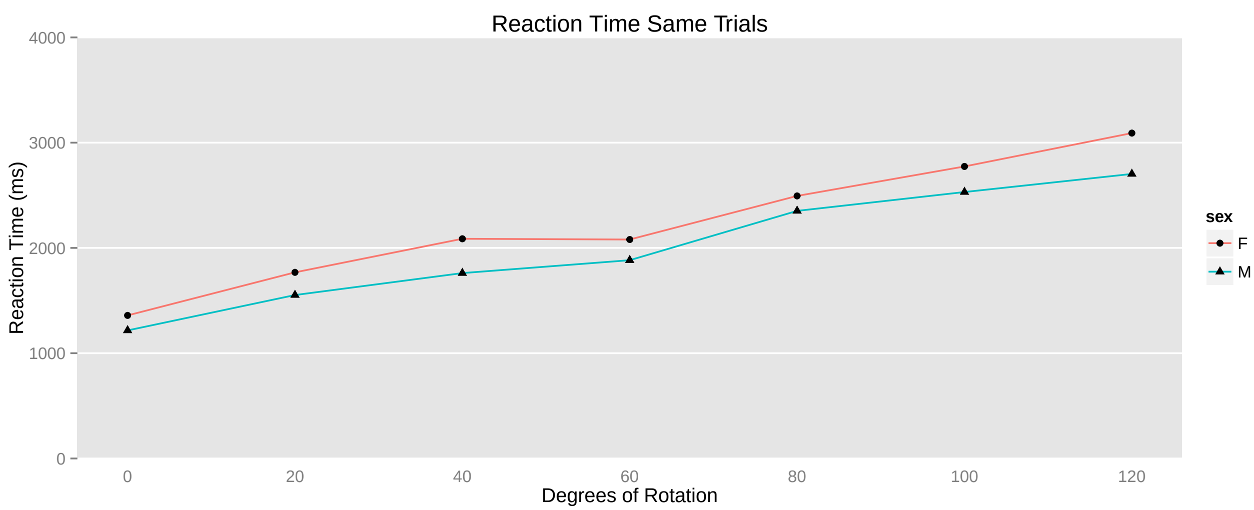
Plot accuracy score medians for different trials. This replicates plot 2 of Figure 1, page 6.

```
MR_med %>% ggplot(aes(x = degrees, y = med_acc_diff)) +
  geom_path(aes(color = sex)) +
  geom_point(aes(shape = sex)) +
  coord_cartesian(ylim = c(0, 105)) +
  xlab("Degrees of Rotation") +
  ylab("Accuracy (%)") +
  ggtitle("Accuracy Different Trials") +
  scale_x_continuous(breaks=seq(0, 120, 20)) +
  scale_y_continuous(breaks=seq(0, 100, 20)) +
  theme(axis.ticks.x = element_blank(), panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank())
```



Plot reaction time medians for same trials. This replicates plot 3 of Figure 1, page 6.

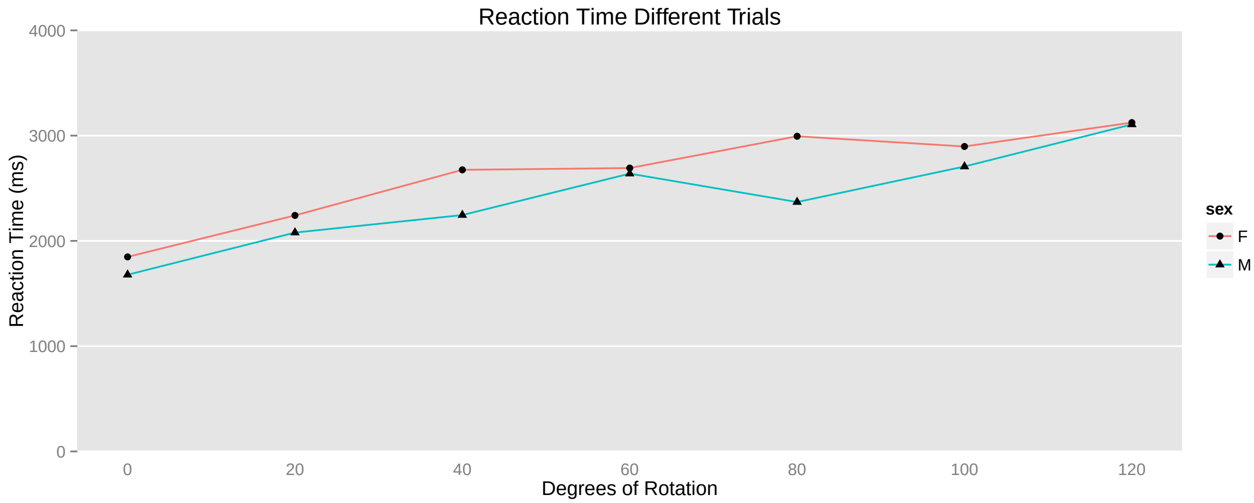
```
MR_med %>% ggplot(aes(x = degrees, y = med_rt_same)) +
  geom_path(aes(color = sex)) +
  geom_point(aes(shape = sex)) +
  coord_cartesian(ylim = c(0, 4000)) +
  xlab("Degrees of Rotation") +
  ylab("Reaction Time (ms)") +
  ggtitle("Reaction Time Same Trials") +
  scale_x_continuous(breaks=seq(0, 120, 20)) +
  scale_y_continuous(breaks=seq(0, 4000, 1000)) +
  theme(axis.ticks.x = element_blank(), panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank())
```



Plot reaction time medians for different trials. This replicates plot 4 of Figure 1, page 6.

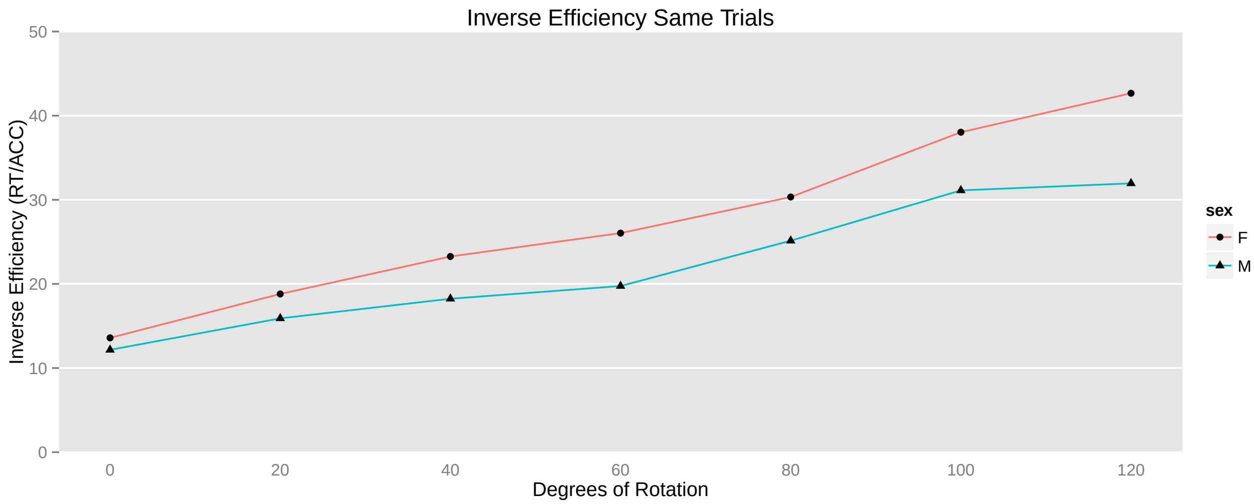
```
  MR_med %>% ggplot(aes(x = degrees, y = med_rt_diff)) +
  geom_path(aes(color = sex)) +
  geom_point(aes(shape = sex)) +
  coord_cartesian(ylim = c(0, 4000)) +
  xlab("Degrees of Rotation") +
```

```
  ylab("Reaction Time (ms)") +
  ggtitle("Reaction Time Different Trials") +
  scale_x_continuous(breaks=seq(0, 120, 20)) +
  scale_y_continuous(breaks=seq(0, 4000, 1000)) +
  theme(axis.ticks.x = element_blank(), panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank())
```



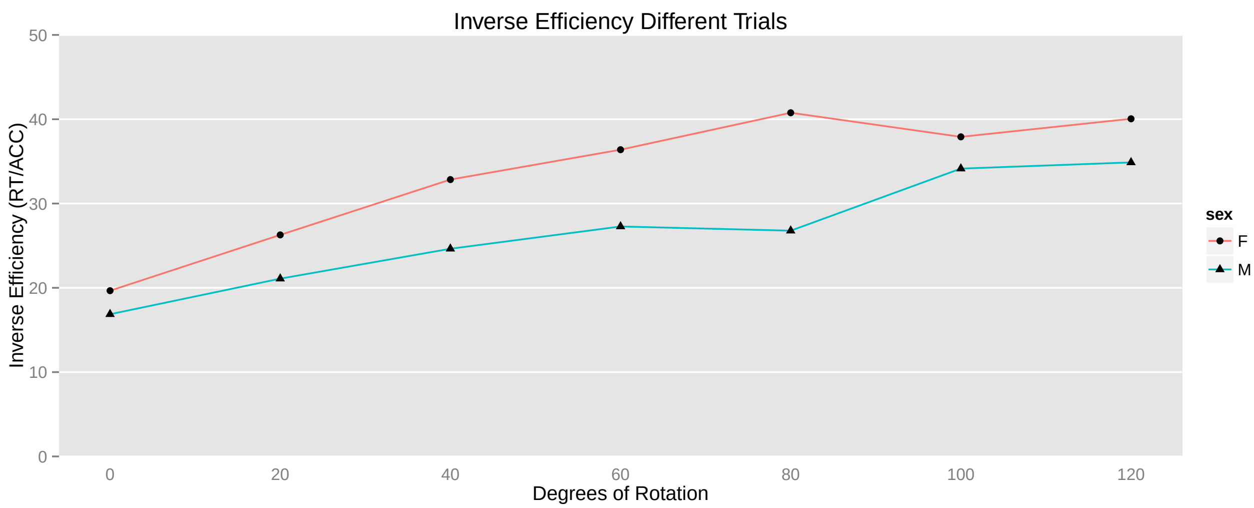
Plot medians of inverse efficiency for same trials. This replicates plot 5 of Figure 1, page 6.

```
MR_med %>% ggplot(aes(x = degrees, y = med_ie_same)) +
  geom_path(aes(color = sex)) +
  geom_point(aes(shape = sex)) +
  coord_cartesian(ylim = c(0, 50)) +
  xlab("Degrees of Rotation") +
  ylab("Inverse Efficiency (RT/ACC)") +
  ggtitle("Inverse Efficiency Same Trials") +
  scale_x_continuous(breaks=seq(0, 120, 20)) +
  scale_y_continuous(breaks=seq(0, 50, 10)) +
  theme(axis.ticks.x = element_blank(), panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank())
```



Plot medians of inverse efficiency for different trials. This replicates plot 6 of Figure 1, page 6.

```
MR_med %>% ggplot(aes(x = degrees, y = med_ie_diff)) +
  geom_path(aes(color = sex)) +
  geom_point(aes(shape = sex)) +
  coord_cartesian(ylim = c(0, 50)) +
  xlab("Degrees of Rotation") +
  ylab("Inverse Efficiency (RT/ACC)") +
  ggtitle("Inverse Efficiency Different Trials") +
  scale_x_continuous(breaks=seq(0, 120, 20)) +
  scale_y_continuous(breaks=seq(0, 50, 10)) +
  theme(axis.ticks.x = element_blank(), panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank())
```



Generate linear models of degrees of rotation and accuracy, as well as inverse efficiency scores. The coefficients of these models will be used for the Mann-Whitney U-tests.

```
#linear_model <- function(col_values){
# result <- MR_data_tall %>% group_by(participant) %>%
#    do(model = lm(col_value ~ degrees, data = .))
#  return (result)
#}

#rt_same_model <- linear_model(MR_data_tall$rt_same)
#rt_diff_model <- linear_model(MR_data_tall$rt_diff)
#acc_same_model <- linear_model(MR_data_tall$acc_same_model)
#acc_diff_model <- linear_model(MR_data_tall$acc_diff_model)
#ie_same_model <- linear_model(MR_data_tall$ie_same_model)
#ie_diff_model <- linear_model(MR_data_tall$ie_diff_model)

rt_same_model <- MR_data_tall %>% group_by(participant) %>%
  do(model = lm(rt_same ~ degrees, data = .))

rt_diff_model <- MR_data_tall %>% group_by(participant) %>%
  do(model = lm(rt_diff ~ degrees, data = .))

acc_same_model <- MR_data_tall %>% group_by(participant) %>%
```

```r
  do(model = lm(acc_same ~ degrees, data = .))

acc_diff_model <- MR_data_tall %>% group_by(participant) %>%
  do(model = lm(acc_diff ~ degrees, data = .))

ie_same_model <- MR_data_tall %>% group_by(participant) %>%
  do(model = lm(ie_same ~ degrees, data = .))

ie_diff_model <- MR_data_tall %>% group_by(participant) %>%
  do(model = lm(ie_diff ~ degrees, data = .))
```

Extract the coefficients from the linear models

```r
# Function created to extract the coefficients created from the linear models
# created above
# @param old_data_frame - dataframe created from calling lm function
# @param col_name - vector containg the new column names
coefs_creation <- function(old_data_frame,col_name){
  new_data_frame <- data_frame()
  for (i in 1:length(old_data_frame$model)){
    new_data_frame[i,1] <- old_data_frame[i,"participant"]
    new_data_frame[i,2] <- coef(old_data_frame$model[[i]])[1] #get the intercepts
    new_data_frame[i,3] <- coef(old_data_frame$model[[i]])[2] #get the slopes
  }
  colnames(new_data_frame)[2:3] <- col_name
  return (new_data_frame)
}

rt_same_coefs <- coefs_creation(rt_same_model,c("rt_same_itcpt", "rt_same_slope"))

rt_diff_coefs <- coefs_creation(rt_diff_model,c("rt_diff_itcpt", "rt_diff_slope"))

acc_same_coefs <- coefs_creation(acc_same_model,c("acc_same_itcpt", "acc_same_slope"))

acc_diff_coefs <- coefs_creation(acc_diff_model,c("acc_diff_itcpt", "acc_diff_slope"))

ie_same_coefs <- coefs_creation(ie_same_model,c("ie_same_itcpt", "ie_same_slope"))

ie_diff_coefs <- coefs_creation(ie_diff_model,c("ie_diff_itcpt", "ie_diff_slope"))
```

Add rank columns to the coefficient data frames, for use in double-checking the Mann-Whitney results. Because the variable names are getting long, the following abbreviations are used: 'si' for 'same intercept', 'ss' for 'same slope', 'di' for 'different intercept', and 'ds' for 'different slope'.

```r
# tried to create a function to simplify but may be too difficult to make as concise
#*add_rank <- function(x,itcpt,slope) {
#  x %>%
#    mutate(y[1] = rank(itcpt, ties.method = "average"),
#           z[1] = rank(slope, ties.method = "average"))
#}

acc_same_coefs <- acc_same_coefs %>%
  mutate(acc_si_rank = rank(acc_same_itcpt, ties.method = "average"),
```

```
        acc_ss_rank = rank(acc_same_slope, ties.method = "average"))

acc_diff_coefs <- acc_diff_coefs %>%
  mutate(acc_di_rank = rank(acc_diff_itcpt, ties.method = "average"),
         acc_ds_rank = rank(acc_diff_slope, ties.method = "average"))

rt_same_coefs <- rt_same_coefs %>%
  mutate(rt_si_rank = rank(rt_same_itcpt, ties.method = "average"),
         rt_ss_rank = rank(rt_same_slope, ties.method = "average"))

rt_diff_coefs <- rt_diff_coefs %>%
  mutate(rt_di_rank = rank(rt_diff_itcpt, ties.method = "average"),
         rt_ds_rank = rank(rt_diff_slope, ties.method = "average"))

ie_same_coefs <- ie_same_coefs %>%
  mutate(ie_si_rank = rank(ie_same_itcpt, ties.method = "average"),
         ie_ss_rank = rank(ie_same_slope, ties.method = "average"))

ie_diff_coefs <- ie_diff_coefs %>%
  mutate(ie_di_rank = rank(ie_diff_itcpt, ties.method = "average"),
         ie_ds_rank = rank(ie_diff_slope, ties.method = "average"))
```

Merge the coefficient data frames.

```
# First, add the sex variable back in by merging with a subset of `MR_data_tall`, with just
# one entry per participant
MR_coefs <- merge(MR_data_tall[MR_data_tall$degrees == 0, c("participant", "sex")],
                  rt_same_coefs, by = "participant")
MR_coefs <- merge(MR_coefs, rt_diff_coefs, by = "participant")
MR_coefs <- merge(MR_coefs, acc_same_coefs, by = "participant")
MR_coefs <- merge(MR_coefs, acc_diff_coefs, by = "participant")
MR_coefs <- merge(MR_coefs, ie_same_coefs, by = "participant")
MR_coefs <- merge(MR_coefs, ie_diff_coefs, by = "participant")
# MR_coefs$sex <- as.factor(MR_coefs$sex)
```

Find the medians of the slopes and intercepts. This replicates the data in Table 2, page 5. You will find that our results for reaction time differ significantly from theirs. This is probably because of the compressed nature of the data set we have access to.

```
# For males and females, find the medians of accuracy coefficients
MR_coefs %>% group_by(sex) %>% summarise("med_acc_slope_same" = median(acc_same_slope))
```

```
Source: local data frame [2 x 2]

     sex med_acc_slope_same
  (fctr)              (dbl)
1      F         -0.1892857
2      M         -0.1160714
```

```
MR_coefs %>% group_by(sex) %>% summarise("med_acc_slope_diff" = median(acc_diff_slope))
```

```
Source: local data frame [2 x 2]
```

```
     sex med_acc_slope_diff
  (fctr)              (dbl)
1      F        -0.09642857
2      M        -0.05892857
```

```r
MR_coefs %>% group_by(sex) %>% summarise("med_acc_itcpt_same" = median(acc_same_itcpt))
```

```
Source: local data frame [2 x 2]

     sex med_acc_itcpt_same
  (fctr)              (dbl)
1      F           99.28571
2      M          100.73214
```

```r
MR_coefs %>% group_by(sex) %>% summarise("med_acc_itcpt_diff" = median(acc_diff_itcpt))
```

```
Source: local data frame [2 x 2]

     sex med_acc_itcpt_diff
  (fctr)              (dbl)
1      F           92.32143
2      M           97.23214
```

```r
# For males and females, find the medians of reaction time coefficients
MR_coefs %>% group_by(sex) %>% summarise("med_rt_slope_same" = median(rt_same_slope))
```

```
Source: local data frame [2 x 2]

     sex med_rt_slope_same
  (fctr)             (dbl)
1      F          14.23036
2      M          11.99286
```

```r
MR_coefs %>% group_by(sex) %>% summarise("med_rt_slope_diff" = median(rt_diff_slope))
```

```
Source: local data frame [2 x 2]

     sex med_rt_slope_diff
  (fctr)             (dbl)
1      F          9.817857
2      M          8.033929
```

```r
MR_coefs %>% group_by(sex) %>% summarise("med_rt_itcpt_same" = median(rt_same_itcpt))
```

```
Source: local data frame [2 x 2]

     sex med_rt_itcpt_same
  (fctr)             (dbl)
1      F          1446.714
2      M          1287.946
```

```r
MR_coefs %>% group_by(sex) %>% summarise("med_rt_itcpt_diff" = median(rt_diff_itcpt))
```

```
Source: local data frame [2 x 2]

     sex med_rt_itcpt_diff
  (fctr)             (dbl)
1      F          2010.107
2      M          1876.143
```

Do Mann-Whitney U-tests on the slopes and intercepts for each of the four dependent variables, as well as the two composite variables. In the paper, these values are not tabulated, but rather are only given in the text on pages 4-5, under the "Results" heading. Here again, some of our results are pretty close to the originals, and some deviate notably. While the paper only cites the smaller U-statistic, R does not automatically do this. If the larger U (or W) is given in R's output, you have to manually subtract it from the total corrected rank sum (650 for this study).

The r statistic given in this section of the paper is not explained. It is probably some kind of correlation, possibly Spearman's rho, but a different symbol is used for Spearman's rho under the "Systemising" subheading of the "Results" section.

```r
# Find the total rank sum
N <- 51
total_rank_sum <- N * (N + 1) / 2
# Find the minimum rank sums for each group
n_F = 25
min_rank_sum_F <- n_F * (n_F + 1) / 2
n_M = 26
min_rank_sum_M <- n_M * (n_M + 1) / 2
# Find the corrected total rank sum
total_rs_corr <- total_rank_sum - (min_rank_sum_F + min_rank_sum_M)
total_rs_corr  #650
```

```
[1] 650
```

```r
# Next, the reaction time analyses. None of these values are cited explicitly in the paper.
# Only the "largest U = 258.5" is given (although the smallest U would be more relevant, so
# this may be a typo)

# Function created to reduce line of repeated values and prints the new structure
# @param col_name - name of the column to use for the Mann Whitney test
# @param col_values - values of the column name
# @ return
mann_whitney <- function(col_values,col_name){
  test_result <- wilcox.test(col_values ~ sex, data = MR_coefs,correct = FALSE, exact = TRUE)
  test_result[7][1] <- paste(col_name,"by sex", sep = " ") # changes the name of the column to col_name
  print(test_result)
  return (test_result)
}
```

The accuracy analyses are referenced first

```r
# Rank sum for slope of the accuracy value from same rotation test
MW_acc_same_slope <- mann_whitney(MR_coefs$acc_same_slope,"acc_same_slope")
```

```
    Wilcoxon rank sum test

data:  acc_same_slope by sex
W = 192, p-value = 0.01221
alternative hypothesis: true location shift is not equal to 0
```

```r
# U = 191.5, p = .012 in the paper
```

```r
# Rank sum for slope of the accuracy value from different rotation test
MW_acc_diff_slope <- mann_whitney(MR_coefs$acc_diff_slope,"acc_diff_slope")
```

```
    Wilcoxon rank sum test

data:  acc_diff_slope by sex
W = 230, p-value = 0.07483
alternative hypothesis: true location shift is not equal to 0
```

```r
# U = 224, p = .056 in the paper
```

```r
# Rank sum for itcpt of the of the accuracy value from same rotation test
MW_acc_same_itcpt <- mann_whitney(MR_coefs$acc_same_itcpt,"acc_same_itcpt")
```

```
    Wilcoxon rank sum test

data:  acc_same_itcpt by sex
W = 266.5, p-value = 0.2703
alternative hypothesis: true location shift is not equal to 0
```

```r
# No value given in the paper
```

```r
# Rank sum for itcpt of the of the accuracy value from different rotation test
MW_acc_diff_itcpt <- mann_whitney(MR_coefs$acc_diff_itcpt,"acc_diff_itcpt")
```

```
    Wilcoxon rank sum test

data:  acc_diff_itcpt by sex
W = 230, p-value = 0.07483
alternative hypothesis: true location shift is not equal to 0
```

```r
# U = 229.5, p = .072 in the paper
```

```r
# Rank sum for slope of the rotational value from same rotation test
MW_rt_same_slope <- mann_whitney(MR_coefs$rt_same_slope,"rt_same_slope")
```

```
    Wilcoxon rank sum test

data:  rt_same_slope by sex
W = 401, p-value = 0.1561
alternative hypothesis: true location shift is not equal to 0
```

```r
# Rank sum for slope of the rotational values from different rotation test
MW_rt_diff_slope <- mann_whitney(MR_coefs$rt_diff_slope,"rt_diff_slope")
```

```
    Wilcoxon rank sum test

data:  rt_diff_slope by sex
W = 405, p-value = 0.135
alternative hypothesis: true location shift is not equal to 0
```

```r
# Rank sum for itcpt of the results from same rotation test
MW_rt_same_itcpt <- mann_whitney(MR_coefs$rt_same_itcpt,"rt_same_itcpt")
```

```
    Wilcoxon rank sum test

data:  rt_same_itcpt by sex
W = 403, p-value = 0.1453
alternative hypothesis: true location shift is not equal to 0
```

```r
# Rank sum for itcpt of the results from different rotation test
MW_rt_diff_itcpt <- mann_whitney(MR_coefs$rt_diff_itcpt,"rt_diff_itcpt")
```

```
    Wilcoxon rank sum test

data:  rt_diff_itcpt by sex
W = 392, p-value = 0.2122
alternative hypothesis: true location shift is not equal to 0
```

Lastly, the composite analyses are given

```r
# Rank sum for slope of the inverse effeciency of same rotation test
MW_ie_same_slope <- mann_whitney(MR_coefs$ie_same_slope,"ie_same_slope")
```

```
    Wilcoxon rank sum test

data:  ie_same_slope by sex
W = 469, p-value = 0.006114
alternative hypothesis: true location shift is not equal to 0
```

```r
# U = 212, p = .024 in the paper
```

```r
# Rank sum for slope of the inverse effeciency of different rotation test
MW_ie_diff_slope <- mann_whitney(MR_coefs$ie_diff_slope,"ie_diff_slope")
```

```
    Wilcoxon rank sum test

data:  ie_diff_slope by sex
W = 455, p-value = 0.01378
alternative hypothesis: true location shift is not equal to 0
```

```r
# U = 214.5, p = .027 in the paper
```

```r
# Rank sum for itcpt of the inverse effeciency of same rotation test
MW_ie_same_itcpt <- mann_whitney(MR_coefs$ie_same_itcpt,"ie_same_itcpt")
```

```
    Wilcoxon rank sum test

data:  ie_same_itcpt by sex
W = 386, p-value = 0.2567
alternative hypothesis: true location shift is not equal to 0
```

```r
# No value given in the paper
```

```r
# rank sum for itcpt of the inverse effeciency of different rotation test
MW_ie_diff_itcpt <- mann_whitney(MR_coefs$ie_diff_itcpt,"ie_same_itcpt")
```

```
    Wilcoxon rank sum test

data:  ie_same_itcpt by sex
W = 426, p-value = 0.05782
alternative hypothesis: true location shift is not equal to 0
```

```r
# U = 223, p = .055 in the paper
```

Do a Student's t-test on the IPT test scores for men and women. Also, calculate Spearman's rho correlation between IPT and reaction time scores. This is an attempt to replicate the results given under the "Systemising" subheading in the "Results" section, page 5.

```r
# subset the variables of interest
MR_data_misc <- MR_data_tall %>% filter(degrees == 0) %>%
  select(participant, sex, digit_ratio, ipt)

# Merge them with the reaction time slope data
MR_data_misc <-
  merge(MR_data_misc, MR_coefs[, c("participant", "rt_same_slope", "rt_diff_slope")],
        by = "participant")

ipt_t_test <- t.test(ipt ~ sex, data = MR_data_misc, var.equal = TRUE)
ipt_t_test
```

```
	Two Sample t-test

data:  ipt by sex
t = -4.2539, df = 49, p-value = 9.427e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.655089 -1.667988
sample estimates:
mean in group F mean in group M
        8.80000        11.96154
```

```
# t(49) = -4.25, p < .001 in the paper
```

```
ipt_rho_test_same <-
  cor.test(MR_data_misc$ipt, MR_data_misc$rt_same_slope, method = "spearman")
ipt_rho_test_same
```

```
	Spearman's rank correlation rho

data:  MR_data_misc$ipt and MR_data_misc$rt_same_slope
S = 29844, p-value = 0.01171
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.3503958
```

```
# rho = -.48, p = .014 in the paper
```

```
ipt_rho_test_diff <-
  cor.test(MR_data_misc$ipt, MR_data_misc$rt_diff_slope, method = "spearman")
ipt_rho_test_diff
```

```
	Spearman's rank correlation rho

data:  MR_data_misc$ipt and MR_data_misc$rt_diff_slope
S = 29074, p-value = 0.02408
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.3155838
```

```
# rho = -.34, p = .094 in the paper
```

Do a Student's t-test on handedness for men and women. This attempts to replicate the results under "Handedness" in the "Results" section, page 5. Since the handedness data were not included in the dataset, I'm creating a data.frame manually from information given in the paper under "Participants" in the "Method" section, page 3. Here, it says that 21 of 25 females were right-handed, and 24 of 26 males were right-handed.

```r
sex <- c(rep("F", times = 25), rep("M", times = 26))
hand <- c(rep("R", times = 21), rep("L", times = 4), rep("R", times = 24),
          rep("L", times = 2))
handedness <- data.frame(cbind(sex, hand))
handedness$hand <- as.integer(handedness$hand)

hand_t_test <- t.test(hand ~ sex, data = handedness, var.equal = TRUE)
hand_t_test
```

```

    Two Sample t-test

data:  hand by sex
t = -0.9099, df = 49, p-value = 0.3673
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2665588  0.1004049
sample estimates:
mean in group F mean in group M
       1.840000        1.923077
```

```r
# t(48) = -1.14, p = .260 in the paper
# Why there are only 48 degrees of freedom here is baffling
```

Do a Student's t-test on the 2D:4D digit ratios for men and women. This attempts to replicate the results under "2d:4d ratios" in the "Results" section, page 5.

```r
d_ratio_t_test <- t.test(digit_ratio ~ sex, data = MR_data_misc, var.equal = TRUE)
d_ratio_t_test
```

```

    Two Sample t-test

data:  digit_ratio by sex
t = 0.53526, df = 49, p-value = 0.5949
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.01533988  0.02647835
sample estimates:
mean in group F mean in group M
      0.9848000       0.9792308
```

```r
# t(49) = .54, p = .595
```