Code for replication - 1

Marius Geantă

May 19, 2024

0. Read data files

```
# Read data files
df1 <- readRDS("df1.rds") # "overall" assessments
df10 <- readRDS("df10.rds") # "accuracy" assessments
df20 <- readRDS("df20.rds") # "timeliness" assessments
df30 <- readRDS("df30.rds") # "comprehensiveness" assessments
df40 <- readRDS("df40.rds") # "friendly" (or "easy to use") assessments
df1_long <- readRDS("df1_long.rds") # "overall" assessments - long format
df10_long <- readRDS("df10_long.rds") # "accuracy" assessments  - long format
df20_long <- readRDS("df20_long.rds") # "timeliness" assessments  - long format
df30_long <- readRDS("df30_long.rds") # "comprehensiveness" assessments  - long format
df40_long <- readRDS("df40_long.rds") # "friendly" (or "easy to use") assessments  - long format
```

```
# Create a list of data frames
data_frames <- list(
  df1 = df1,
  df10 = df10,
  df20 = df20,
  df30 = df30,
  df40 = df40,
  df1_long = df1_long,
  df10_long = df10_long,
  df20_long = df20_long,
  df30_long = df30_long,
  df40_long = df40_long
)
```

```
# Check the structure of the data frames
for (name in names(data_frames)) {
  cat("\nStructure of", name, ":\n")
  str(data_frames[[name]])
  cat("\nHead of", name, ":\n")
  print(head(data_frames[[name]]))
}
```

```
##
## Structure of df1 :
## Classes 'tbl_df', 'tbl' and 'data.frame':    8 obs. of  5 variables:
##  $ id     : chr  "1" "2" "3" "4" ...
##  $ chatgpt: num  432 408 377 434 411 456 451 363
##  $ guide  : num  409 249 343 376 372 359 435 349
##  $ gemini : num  373 275 341 394 388 363 406 354
```

```
##  $ copilot: num  432 335 377 410 394 416 448 366
##
## Head of df1 :
##   id chatgpt guide gemini copilot
## 1  1     432   409    373     432
## 2  2     408   249    275     335
## 3  3     377   343    341     377
## 4  4     434   376    394     410
## 5  5     411   372    388     394
## 6  6     456   359    363     416
##
## Structure of df10 :
## 'data.frame':    8 obs. of  5 variables:
##  $ id     : int  1 2 3 4 5 6 7 8
##  $ chatgpt: num  109 99 96 112 100 113 109 88
##  $ guide  : num  101 62 81 101 88 90 102 77
##  $ gemini : num  93 66 80 101 96 91 96 79
##  $ copilot: num  105 78 93 101 95 106 106 84
##
## Head of df10 :
##   id chatgpt guide gemini copilot
## 1  1     109   101     93     105
## 2  2      99    62     66      78
## 3  3      96    81     80      93
## 4  4     112   101    101     101
## 5  5     100    88     96      95
## 6  6     113    90     91     106
##
## Structure of df20 :
## 'data.frame':    8 obs. of  5 variables:
##  $ id     : int  1 2 3 4 5 6 7 8
##  $ chatgpt: num  108 100 98 112 99 109 112 92
##  $ guide  : num  104 66 98 104 94 95 114 98
##  $ gemini : num  93 66 80 101 96 91 96 79
##  $ copilot: num  110 77 101 103 93 100 115 97
##
## Head of df20 :
##   id chatgpt guide gemini copilot
## 1  1     108   104     93     110
## 2  2     100    66     66      77
## 3  3      98    98     80     101
## 4  4     112   104    101     103
## 5  5      99    94     96      93
## 6  6     109    95     91     100
##
## Structure of df30 :
## 'data.frame':    8 obs. of  5 variables:
##  $ id     : int  1 2 3 4 5 6 7 8
##  $ chatgpt: num  98 101 94 106 97 118 109 93
##  $ guide  : num  90 58 78 80 82 78 99 87
##  $ gemini : num  78 63 74 87 89 79 89 86
##  $ copilot: num  102 86 90 96 93 106 107 89
##
## Head of df30 :
```

```
##   id chatgpt guide gemini copilot
## 1  1      98    90     78     102
## 2  2     101    58     63      86
## 3  3      94    78     74      90
## 4  4     106    80     87      96
## 5  5      97    82     89      93
## 6  6     118    78     79     106
##
## Structure of df40 :
## 'data.frame':    8 obs. of  5 variables:
##  $ id     : int  1 2 3 4 5 6 7 8
##  $ chatgpt: num  117 108 89 104 115 116 121 90
##  $ guide  : num  114 63 86 91 108 96 120 87
##  $ gemini : num  108 75 93 105 109 100 113 95
##  $ copilot: num  115 94 93 110 113 104 120 96
##
## Head of df40 :
##   id chatgpt guide gemini copilot
## 1  1     117   114    108     115
## 2  2     108    63     75      94
## 3  3      89    86     93      93
## 4  4     104    91    105     110
## 5  5     115   108    109     113
## 6  6     116    96    100     104
##
## Structure of df1_long :
## Classes 'tbl_df', 'tbl' and 'data.frame':    32 obs. of  3 variables:
##  $ id   : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ tool : Factor w/ 4 levels "guide","copilot",..: 4 1 3 2 4 1 3 2 4 1 ...
##  $ score: num  432 409 373 432 408 249 275 335 377 343 ...
##
## Head of df1_long :
##   id    tool score
## 1  1 chatgpt   432
## 2  1   guide   409
## 3  1  gemini   373
## 4  1 copilot   432
## 5  2 chatgpt   408
## 6  2   guide   249
##
## Structure of df10_long :
## Classes 'tbl_df', 'tbl' and 'data.frame':    32 obs. of  3 variables:
##  $ id   : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ tool : Factor w/ 4 levels "guide","copilot",..: 4 1 3 2 4 1 3 2 4 1 ...
##  $ score: num  109 101 93 105 99 62 66 78 96 81 ...
##
## Head of df10_long :
##   id    tool score
## 1  1 chatgpt   109
## 2  1   guide   101
## 3  1  gemini    93
## 4  1 copilot   105
## 5  2 chatgpt    99
## 6  2   guide    62
```

```
##
## Structure of df20_long :
## Classes 'tbl_df', 'tbl' and 'data.frame':    32 obs. of  3 variables:
##  $ id   : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ tool : Factor w/ 4 levels "guide","copilot",..: 4 1 3 2 4 1 3 2 4 1 ...
##  $ score: num  108 104 93 110 100 66 66 77 98 98 ...
##
## Head of df20_long :
##   id    tool score
## 1  1 chatgpt   108
## 2  1   guide   104
## 3  1  gemini    93
## 4  1 copilot   110
## 5  2 chatgpt   100
## 6  2   guide    66
##
## Structure of df30_long :
## Classes 'tbl_df', 'tbl' and 'data.frame':    32 obs. of  3 variables:
##  $ id   : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ tool : Factor w/ 4 levels "guide","copilot",..: 4 1 3 2 4 1 3 2 4 1 ...
##  $ score: num  98 90 78 102 101 58 63 86 94 78 ...
##
## Head of df30_long :
##   id    tool score
## 1  1 chatgpt    98
## 2  1   guide    90
## 3  1  gemini    78
## 4  1 copilot   102
## 5  2 chatgpt   101
## 6  2   guide    58
##
## Structure of df40_long :
## Classes 'tbl_df', 'tbl' and 'data.frame':    32 obs. of  3 variables:
##  $ id   : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ tool : Factor w/ 4 levels "guide","copilot",..: 4 1 3 2 4 1 3 2 4 1 ...
##  $ score: num  117 114 108 115 108 63 75 94 89 86 ...
##
## Head of df40_long :
##   id    tool score
## 1  1 chatgpt   117
## 2  1   guide   114
## 3  1  gemini   108
## 4  1 copilot   115
## 5  2 chatgpt   108
## 6  2   guide    63
```

1. Statistical analysis

1.1 Distributions of aggregated scores given by panel experts

```
print(df1)
```

```
##   id chatgpt guide gemini copilot
## 1  1     432   409    373     432
```

```
## 2  2     408   249    275     335
## 3  3     377   343    341     377
## 4  4     434   376    394     410
## 5  5     411   372    388     394
## 6  6     456   359    363     416
## 7  7     451   435    406     448
## 8  8     363   349    354     366
```

```
print(df10)
```

```
##   id chatgpt guide gemini copilot
## 1  1     109   101     93     105
## 2  2      99    62     66      78
## 3  3      96    81     80      93
## 4  4     112   101    101     101
## 5  5     100    88     96      95
## 6  6     113    90     91     106
## 7  7     109   102     96     106
## 8  8      88    77     79      84
```

```
print(df20)
```

```
##   id chatgpt guide gemini copilot
## 1  1     108   104     93     110
## 2  2     100    66     66      77
## 3  3      98    98     80     101
## 4  4     112   104    101     103
## 5  5      99    94     96      93
## 6  6     109    95     91     100
## 7  7     112   114     96     115
## 8  8      92    98     79      97
```

```
print(df30)
```

```
##   id chatgpt guide gemini copilot
## 1  1      98    90     78     102
## 2  2     101    58     63      86
## 3  3      94    78     74      90
## 4  4     106    80     87      96
## 5  5      97    82     89      93
## 6  6     118    78     79     106
## 7  7     109    99     89     107
## 8  8      93    87     86      89
```

```
print(df40)
```

```
##   id chatgpt guide gemini copilot
## 1  1     117   114    108     115
## 2  2     108    63     75      94
## 3  3      89    86     93      93
## 4  4     104    91    105     110
## 5  5     115   108    109     113
## 6  6     116    96    100     104
## 7  7     121   120    113     120
## 8  8      90    87     95      96
```

1.2 Tests for normality and homogeneity of variances

```
# check normality for scores within each tool
by(df1_long$score, df1_long$tool, shapiro.test)
```

```
## df1_long$tool: guide
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.91237, p-value = 0.3711
##
## ------------------------------------------------------------
## df1_long$tool: copilot
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.98284, p-value = 0.9756
##
## ------------------------------------------------------------
## df1_long$tool: gemini
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.88878, p-value = 0.228
##
## ------------------------------------------------------------
## df1_long$tool: chatgpt
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.93274, p-value = 0.5414
```

```
by(df10_long$score, df10_long$tool, shapiro.test)
```

```
## df10_long$tool: guide
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.90624, p-value = 0.3284
##
## ------------------------------------------------------------
## df10_long$tool: copilot
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.88498, p-value = 0.21
##
## ------------------------------------------------------------
## df10_long$tool: gemini
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.90508, p-value = 0.3207
##
## ------------------------------------------------------------
## df10_long$tool: chatgpt
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.91362, p-value = 0.3802
```

```
by(df20_long$score, df20_long$tool, shapiro.test)
```

```
## df20_long$tool: guide
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.83798, p-value = 0.0718
##
## ------------------------------------------------------------
## df20_long$tool: copilot
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.94442, p-value = 0.6551
##
## ------------------------------------------------------------
## df20_long$tool: gemini
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.90508, p-value = 0.3207
##
## ------------------------------------------------------------
## df20_long$tool: chatgpt
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.90091, p-value = 0.2944
```

```
by(df30_long$score, df30_long$tool, shapiro.test)
```

```
## df30_long$tool: guide
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.92876, p-value = 0.5048
```

```
##
## ------------------------------------------------------------
## df30_long$tool: copilot
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.91887, p-value = 0.4207
##
## ------------------------------------------------------------
## df30_long$tool: gemini
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.87805, p-value = 0.1804
##
## ------------------------------------------------------------
## df30_long$tool: chatgpt
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.91983, p-value = 0.4285
```

```
by(df40_long$score, df40_long$tool, shapiro.test)
```

```
## df40_long$tool: guide
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.95716, p-value = 0.7827
##
## ------------------------------------------------------------
## df40_long$tool: copilot
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.91231, p-value = 0.3706
##
## ------------------------------------------------------------
## df40_long$tool: gemini
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.90322, p-value = 0.3088
##
## ------------------------------------------------------------
## df40_long$tool: chatgpt
##
##  Shapiro-Wilk normality test
##
```

```
## data:  dd[x, ]
## W = 0.87014, p-value = 0.1512
```

```
# check homogeneity of variances
# Install and load the car package if not already installed
if (!require(car)) {
  install.packages("car", dependencies = TRUE)
  library(car)
}
```

```
# Perform Levene's Test for homogeneity of variances
leveneTest(score ~ tool, data = df1_long)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  0.1927 0.9005
##       28
```

```
leveneTest(score ~ tool, data = df10_long)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  0.3158 0.8138
##       28
```

```
leveneTest(score ~ tool, data = df20_long)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  0.1206 0.9472
##       28
```

```
leveneTest(score ~ tool, data = df30_long)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  0.1023  0.958
##       28
```

```
leveneTest(score ~ tool, data = df40_long)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3   0.675 0.5746
##       28
```

```
# Perform Bartlett's Test for homogeneity of variances
bartlett.test(score ~ tool, data = df1_long)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  score by tool
## Bartlett's K-squared = 1.945, df = 3, p-value = 0.5839
```

```
bartlett.test(score ~ tool, data = df10_long)
```

```
##
##  Bartlett test of homogeneity of variances
##
```

```
## data:  score by tool
## Bartlett's K-squared = 1.4537, df = 3, p-value = 0.693
```

```
bartlett.test(score ~ tool, data = df20_long)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  score by tool
## Bartlett's K-squared = 2.4542, df = 3, p-value = 0.4836
```

```
bartlett.test(score ~ tool, data = df30_long)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  score by tool
## Bartlett's K-squared = 1.2837, df = 3, p-value = 0.733
```

```
bartlett.test(score ~ tool, data = df40_long)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  score by tool
## Bartlett's K-squared = 2.4896, df = 3, p-value = 0.4772
```

```
# Perform Fligner-Killeen Test for homogeneity of variances
fligner.test(score ~ tool, data = df1_long)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  score by tool
## Fligner-Killeen:med chi-squared = 0.28729, df = 3, p-value = 0.9624
```

```
fligner.test(score ~ tool, data = df10_long)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  score by tool
## Fligner-Killeen:med chi-squared = 0.97968, df = 3, p-value = 0.8062
```

```
fligner.test(score ~ tool, data = df20_long)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  score by tool
## Fligner-Killeen:med chi-squared = 0.19041, df = 3, p-value = 0.9791
```

```
fligner.test(score ~ tool, data = df30_long)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  score by tool
## Fligner-Killeen:med chi-squared = 0.090494, df = 3, p-value = 0.993
```

```r
fligner.test(score ~ tool, data = df40_long)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  score by tool
## Fligner-Killeen:med chi-squared = 1.4162, df = 3, p-value = 0.7017
```

1.3 Post hoc tests for comparing sources of information

```r
# Install and load lme4, if not already loaded
if (!require(lme4)) {
  install.packages("lme4")
  library(lme4)
}
```

```
## Loading required package: lme4
```

```
## Warning: package 'lme4' was built under R version 4.2.3
```

```
## Loading required package: Matrix
```

```
## Warning: package 'Matrix' was built under R version 4.2.3
```

```r
# Fit a linear mixed-effects models
model1 <- lmer(score ~ tool + (1|id), data = df1_long)
model10 <- lmer(score ~ tool + (1|id), data = df10_long)
model20 <- lmer(score ~ tool + (1|id), data = df20_long)
model30 <- lmer(score ~ tool + (1|id), data = df30_long)
model40 <- lmer(score ~ tool + (1|id), data = df40_long)
```

```r
# Summary of the model to see results
summary(model1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ tool + (1 | id)
##    Data: df1_long
##
## REML criterion at convergence: 280
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.22932 -0.49796 -0.02348  0.45012  2.28036
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 1266.8   35.59
##  Residual              531.8   23.06
## Number of obs: 32, groups:  id, 8
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   361.50      14.99  24.109
## toolcopilot    35.75      11.53   3.100
## toolgemini      0.25      11.53   0.022
## toolchatgpt    55.00      11.53   4.770
```

```
##
## Correlation of Fixed Effects:
##             (Intr) tlcplt tolgmn
## toolcopilot -0.384
## toolgemini  -0.384  0.500
## toolchatgpt -0.384  0.500  0.500
```

```
summary(model10)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ tool + (1 | id)
##    Data: df10_long
##
## REML criterion at convergence: 199.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8219 -0.5079 -0.1636  0.5975  2.3730
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 105.11   10.252
##  Residual              26.27    5.125
## Number of obs: 32, groups:  id, 8
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 8.775e+01  4.052e+00  21.654
## toolcopilot 8.250e+00  2.563e+00   3.219
## toolgemini  3.077e-15  2.563e+00   0.000
## toolchatgpt 1.550e+01  2.563e+00   6.049
##
## Correlation of Fixed Effects:
##             (Intr) tlcplt tolgmn
## toolcopilot -0.316
## toolgemini  -0.316  0.500
## toolchatgpt -0.316  0.500  0.500
```

```
summary(model20)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ tool + (1 | id)
##    Data: df20_long
##
## REML criterion at convergence: 207.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0319 -0.5557 -0.1593  0.5919  2.1680
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 88.73    9.419
##  Residual             40.95    6.399
## Number of obs: 32, groups:  id, 8
```

```
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   96.625      4.026  24.000
## toolcopilot    2.875      3.200   0.899
## toolgemini    -8.875      3.200  -2.774
## toolchatgpt    7.125      3.200   2.227
##
## Correlation of Fixed Effects:
##             (Intr) tlcplt tolgmn
## toolcopilot -0.397
## toolgemini  -0.397  0.500
## toolchatgpt -0.397  0.500  0.500
```

```
summary(model30)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ tool + (1 | id)
##    Data: df30_long
##
## REML criterion at convergence: 207.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9131 -0.6117 -0.1066  0.7064  1.6965
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 39.37    6.274
##  Residual             50.66    7.118
## Number of obs: 32, groups:  id, 8
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   81.500      3.355  24.295
## toolcopilot   14.625      3.559   4.110
## toolgemini    -0.875      3.559  -0.246
## toolchatgpt   20.500      3.559   5.760
##
## Correlation of Fixed Effects:
##             (Intr) tlcplt tolgmn
## toolcopilot -0.530
## toolgemini  -0.530  0.500
## toolchatgpt -0.530  0.500  0.500
```

```
summary(model40)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ tool + (1 | id)
##    Data: df40_long
##
## REML criterion at convergence: 215.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.3451 -0.2603 -0.1130  0.5553  2.2106
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 132.68   11.518
##  Residual              52.87    7.271
## Number of obs: 32, groups:  id, 8
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   95.625      4.816  19.856
## toolcopilot   10.000      3.636   2.751
## toolgemini     4.125      3.636   1.135
## toolchatgpt   11.875      3.636   3.266
##
## Correlation of Fixed Effects:
##             (Intr) tlcplt tolgmn
## toolcopilot -0.377
## toolgemini  -0.377  0.500
## toolchatgpt -0.377  0.500  0.500
```

```
# Install and load lme4, if not already loaded
if (!require(emmeans)) {
  install.packages("emmeans")
  library(emmeans)
}
```

```
## Loading required package: emmeans
```

```
## Warning: package 'emmeans' was built under R version 4.2.3
```

```
# Post hoc test for tool comparisons
post_hocs1 <- emmeans(model1, pairwise ~ tool)
post_hocs1$contrasts
```

```
##  contrast          estimate   SE df t.ratio p.value
##  guide - copilot     -35.75 11.5 21  -3.100  0.0257
##  guide - gemini       -0.25 11.5 21  -0.022  1.0000
##  guide - chatgpt     -55.00 11.5 21  -4.770  0.0006
##  copilot - gemini     35.50 11.5 21   3.079  0.0270
##  copilot - chatgpt   -19.25 11.5 21  -1.669  0.3638
##  gemini - chatgpt    -54.75 11.5 21  -4.748  0.0006
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```
post_hocs10 <- emmeans(model10, pairwise ~ tool)
post_hocs10$contrasts
```

```
##  contrast          estimate   SE df t.ratio p.value
##  guide - copilot     -8.25 2.56 21  -3.219  0.0198
##  guide - gemini       0.00 2.56 21   0.000  1.0000
##  guide - chatgpt     -15.50 2.56 21  -6.049  <.0001
##  copilot - gemini     8.25 2.56 21   3.219  0.0198
##  copilot - chatgpt   -7.25 2.56 21  -2.829  0.0458
##  gemini - chatgpt    -15.50 2.56 21  -6.049  <.0001
##
```

```
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```
post_hocs20 <- emmeans(model20, pairwise ~ tool)
post_hocs20$contrasts
```

```
##  contrast          estimate  SE df t.ratio p.value
##  guide - copilot      -2.88 3.2 21  -0.899  0.8057
##  guide - gemini        8.88 3.2 21   2.774  0.0514
##  guide - chatgpt      -7.12 3.2 21  -2.227  0.1485
##  copilot - gemini     11.75 3.2 21   3.672  0.0072
##  copilot - chatgpt    -4.25 3.2 21  -1.328  0.5559
##  gemini - chatgpt    -16.00 3.2 21  -5.001  0.0003
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```
post_hocs30 <- emmeans(model30, pairwise ~ tool)
post_hocs30$contrasts
```

```
##  contrast          estimate   SE df t.ratio p.value
##  guide - copilot    -14.625 3.56 21  -4.110  0.0026
##  guide - gemini       0.875 3.56 21   0.246  0.9946
##  guide - chatgpt    -20.500 3.56 21  -5.760  0.0001
##  copilot - gemini    15.500 3.56 21   4.355  0.0015
##  copilot - chatgpt   -5.875 3.56 21  -1.651  0.3734
##  gemini - chatgpt   -21.375 3.56 21  -6.006  <.0001
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```
post_hocs40 <- emmeans(model40, pairwise ~ tool)
post_hocs40$contrasts
```

```
##  contrast          estimate   SE df t.ratio p.value
##  guide - copilot     -10.00 3.64 21  -2.751  0.0539
##  guide - gemini       -4.12 3.64 21  -1.135  0.6729
##  guide - chatgpt     -11.88 3.64 21  -3.266  0.0179
##  copilot - gemini      5.88 3.64 21   1.616  0.3915
##  copilot - chatgpt    -1.88 3.64 21  -0.516  0.9544
##  gemini - chatgpt     -7.75 3.64 21  -2.132  0.1757
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```