WCST analysis

Check data

The size of the raw data table is (number of moves x number of variables):

```
## [1] 1092   11
```

There should be 4 types of rules: c, n, s and spxxxx (x stands for A, B, C or D). If there is more than 4, calculations based on the rules will be wrong!

```
##   c   n   s  sp
## 210 181 179 522
```

There should be 24+1=25 types of source cards if we use unambiguous cards or 64-4+1=61 types if we use ambiguous cards. The number of source card types:

```
## [1] 100
```

All source cards should have a 4-digit code. The source card codes and frequencies were:

```
##     sy1     qy2     tb4     cb1     qg4     tr2     cr2     cr4     cy2
##      17      16      16      15      15      15      14      14      14
##     cy4     qb4     qg1     qg2     qg3     sb2     sb3     sb4     sr1
##      14      14      14      14      14      14      14      14      14
##     sr2     tg1     tr3     ty1     ty4     cb2     cg1     cg2     cg3
##      14      14      14      14      14      13      13      13      13
##     cr1     cr3     cy1     cy3     qb3     qr2     qr4     qy1     sb1
##      13      13      13      13      13      13      13      13      13
##     sg1     sy3     tb3     tg2     ty3     cg4     qb2     qr1     qy4
##      13      13      13      13      13      12      12      12      12
##     sg4     sr3     sr4     sy2     tg3     tg4     cb3     qb1     qr3
##      12      12      12      12      12      12      11      11      11
##     sg3     sy4     tb1     tb2     tr4     ty2     hm0     hm1    hm41
##      11      11      11      11      11      11       9       8       7
##    hm42    hm43    hm44    hm45    hm35    hm36    hm37    hm38    hm39
##       7       7       7       7       6       6       6       6       6
##    hm40    hm46    hm47    hm48    hm49    hm50    hm51    hm52    hm53
##       6       6       6       6       6       6       6       6       6
##    hm54    hm55    hm56    hm57    hm58     hm2    hm29     hm3    hm30
##       6       6       6       6       6       5       5       5       5
##    hm31    hm32    hm33    hm34     hm4     hm5     hm6    hm10    hm11
##       5       5       5       5       5       5       5       4       4
## (Other)
##      78
```

There should be 4 types of target cards, with the codes: tr1A, sg2B, qy3C, cb4D. Target card codes and frequencies were:

```
##  cb4 div1 div2 div3 div4  qy3  sg2  tr1
##  216    3    3    6    2  368  262  232
```

Variables

Additional variables per move

From the raw data, I calculated some additional variables, such as:

- Think time: total time between two drop action minus the time spent with drag-and-drop
- Current rule: this is the same as "category", but recoded with numbers, where s, c, n and sp have the codes: 1, 2, 3 and 4, respectively.
- Applied rule: the rule that the move conforms to. THIS WON'T BE CORRECT UNTIL THE SOURCE CARD CODE DOES NOT INCLUDE THE INDEX FOR THE SPATIAL RULE
- Number of applied rules: integer from 0 to 4 (actually 3, because we excluded the key cards from the deck)
- Random search: this is TRUE if none of the rules applies to the current rule (if the number of applied rules = 1)
- Perseveration error: after a "success", we check if the participant keeps using the same rule (if the applied rule - or one of the applied rules - is the same as the current rule in the successful step) for any number of moves

An example: Data for participant 242040

```
##    tar_card src_card match current_rule applied_rule randsearch persev
## 1       sg2      qg4 false            1            2      FALSE  FALSE
## 2       qy3      qr4  true            1            1      FALSE  FALSE
## 3       cb4      cy3  true            1            1      FALSE  FALSE
## 4       sg2      sb2  true            1         1, 3      FALSE  FALSE
## 5       cb4      cr2  true            1            1      FALSE  FALSE
## 6       qy3      qr2  true            1            1      FALSE  FALSE
## 7       sg2      sy1  true            1            1      FALSE  FALSE
## 8       tr1      ty2  true            1            1      FALSE  FALSE
## 9       sg2      sg3  true            1         1, 2      FALSE  FALSE
## 10      tr1      ty4  true            1            1      FALSE  FALSE
## 11      tr1      tr2  true            2         1, 2      FALSE   TRUE
## 12      tr1      tr3  true            2         1, 2      FALSE   TRUE
## 13      tr1      tr4  true            2         1, 2      FALSE   TRUE
## 14      sg2      sy2 false            2         1, 3      FALSE   TRUE
## 15      tr1      tb2 false            2            1      FALSE   TRUE
## 16      sg2      qg3  true            2            2      FALSE  FALSE
## 17      cb4      cb3  true            2         1, 2      FALSE  FALSE
## 18      qy3      sy3  true            2         2, 3      FALSE  FALSE
## 19      sg2      sg1  true            2         1, 2      FALSE  FALSE
## 20      sg2      tg3  true            2            2      FALSE  FALSE
## 21      cb4      qb1  true            2            2      FALSE  FALSE
## 22      sg2      cg3  true            2            2      FALSE  FALSE
## 23      sg2      tg2  true            3         2, 3      FALSE   TRUE
## 24      qy3      ty1 false            3            2      FALSE   TRUE
## 25      cb4      tb3 false            3            2      FALSE   TRUE
## 26      cb4      tg4  true            3            3      FALSE  FALSE
## 27      cb4      sb4  true            3         2, 3      FALSE  FALSE
## 28      cb4      sg4  true            3            3      FALSE  FALSE
## 29      tr1      tb1  true            3         1, 3      FALSE  FALSE
## 30      qy3      qr3  true            3         1, 3      FALSE  FALSE
## 31      tr1      tg1  true            3         1, 3      FALSE  FALSE
```

```
## 32      sg2      qg2  true           3         2, 3      FALSE  FALSE
## 33      cb4      cy4  true           3         1, 3      FALSE  FALSE
## 34      cb4      qy4 false           4            3      FALSE   TRUE
## 35      tr1      cy1 false           4            3      FALSE   TRUE
## 36      cb4      cr3 false           4            1      FALSE  FALSE
## 37      qy3      qy2  true           4         1, 2      FALSE  FALSE
## 38      tr1      sr2  true           4            2      FALSE  FALSE
## 39      qy3      qy1 false           4         1, 2      FALSE  FALSE
## 40      tr1      cr1 false           4         2, 3      FALSE  FALSE
## 41      sg2      sb1 false           4            1      FALSE  FALSE
## 42      sg2      cy2 false           4            3      FALSE  FALSE
## 43      qy3      sr1  true           4            0       TRUE  FALSE
## 44      sg2      qb3 false           4            0       TRUE  FALSE
## 45      sg2      ty3 false           4            0       TRUE  FALSE
## 46      tr1      qb4 false           4            0       TRUE  FALSE
## 47      cb4      tb4 false           4         2, 3      FALSE  FALSE
## 48      cb4      sy4 false           4            3      FALSE  FALSE
## 49      qy3      qr1  true           4            1      FALSE  FALSE
## 50      qy3      cr4 false           4            0       TRUE  FALSE
## 51      sg2      sb3 false           4            1      FALSE  FALSE
## 52      tr1      cg1 false           4            3      FALSE  FALSE
## 53      sg2      cg2 false           4         2, 3      FALSE  FALSE
## 54      sg2      sr3 false           4            1      FALSE  FALSE
## 55      cb4      cg4 false           4         1, 3      FALSE  FALSE
## 56      cb4      cb1 false           4         1, 2      FALSE  FALSE
## 57      qy3      qg1  true           4            1      FALSE  FALSE
## 58      qy3      qb2 false           4            1      FALSE  FALSE
## 59      tr1      sr4 false           4            2      FALSE  FALSE
## 60      tr1      tr2 false           4         1, 2      FALSE  FALSE
## 61      tr1      sb4 false           4            0       TRUE  FALSE
## 62      tr1      qb3 false           4            0       TRUE  FALSE
## 63      tr1      tb4 false           4            1      FALSE  FALSE
## 64      tr1      tg1 false           4         1, 3      FALSE  FALSE
```

Variables per participant

From the per-move variables, I calculated averaged and summed values for each participant, separately for each rule:

- success
- failure
- number of moves
- number of correct moves
- number of incorrect moves
- average time between two drop actions
- average move time
- average think time
- time spent with the task (sec)

```
## Source: local data frame [48 x 6]
## Groups: condition, part [12]
##
##    condition   part category success_bin failure_bin moves_nb
```

```
##       (fctr)  (int)   (fctr)       (int)       (int)    (dbl)
## 1       wcst 242040        c           1           0       12
## 2       wcst 242040        n           1           0       11
## 3       wcst 242040        s           1           0       10
## 4       wcst 242040       sp           0           1       31
## 5       wcst 329422        c           1           0       11
## 6       wcst 329422        n           1           0        7
## 7       wcst 329422        s           1           0       10
## 8       wcst 329422       sp           0           1       31
## 9       wcst 757143        c           1           0       28
## 10      wcst 757143        n           1           0       10
## ..       ...    ...      ...         ...         ...      ...
```

```
## Source: local data frame [48 x 6]
##
##    correctmoves_nb incorrectmoves_nb totaltime_avg movetime_avg
##              (int)             (int)         (dbl)        (dbl)
## 1               10                 2      2540.750     923.3333
## 2                9                 2      2324.455     866.4545
## 3                9                 1      2554.300     769.9000
## 4                5                26      3609.935    1062.7419
## 5                9                 2      3217.455    1143.3636
## 6                7                 0      3132.857    1128.7143
## 7                8                 2      2977.000    1097.1000
## 8                4                27      3167.323    1057.5806
## 9               13                15      4169.107     772.3214
## 10               7                 3      2629.300     751.5000
## ..             ...               ...           ...          ...
## Variables not shown: thinktime_avg (dbl), time_spent (dbl)
```

- *Would perseveration be interesting? Do we need it?*
- *Distribution of rules used?*

Variables per condition

I coded the following variables, separately for each rule:

- Number of participants
- Number of solvers
- Number of nonsolvers
- Percentage of solvers
- Average number of moves
- Average time spent with the task

Other ideas?

```
## Source: local data frame [16 x 6]
## Groups: condition [4]
##
##    condition category partic_nb solvers_nb nonsolvers_nb solvers_perc
##       (fctr)   (fctr)     (int)      (int)         (int)        (dbl)
## 1       wcst        c         3          3             0          100
```

```
## 2       wcst        n         3         3             0         100
## 3       wcst        s         3         3             0         100
## 4       wcst       sp         3         0             3           0
## 5      wcstc        c         4         3             1          75
## 6      wcstc        n         4         3             1          75
## 7      wcstc        s         4         4             0         100
## 8      wcstc       sp         4         0             4           0
## 9      wcstf        c         2         0             2           0
## 10     wcstf        n         2         2             0         100
## 11     wcstf        s         2         1             1          50
## 12     wcstf       sp         2         0             2           0
## 13    wcstfc        c         2         2             0         100
## 14    wcstfc        n         3         3             0         100
## 15    wcstfc        s         3         3             0         100
## 16    wcstfc       sp         3         0             3           0
```

```
## Source: local data frame [16 x 2]
##
##    nbofmoves_avg timespsent_avg
##            (dbl)          (dbl)
## 1      17.000000       60.87200
## 2       9.333333       24.59733
## 3      12.666667       39.26733
## 4      31.000000      109.60800
## 5      16.750000       49.53475
## 6      17.750000       62.58600
## 7      10.000000       38.44975
## 8      31.000000       95.50650
## 9      31.000000      113.39450
## 10     18.500000       56.42500
## 11     26.000000       83.37750
## 12     61.000000      212.45800
## 13     10.333333       19.73167
## 14     15.000000       28.49167
## 15     16.333333       33.23867
## 16     61.000000      117.22267
```

Hypotheses

In our plans we had the following testable hypotheses:

1. Going through the standard rules first makes the task of finding the index rule harder
    a) Condition 1 (WCST) will be more difficult than condition 2 (WCSTF)
    b) Condition 3 (WCSTC) will be more difficult than condition 4 (WCSTFC)
    c) Condition 1 (WCST) will be more difficult than condition 4 (WCSTFC)
2. Using the standard cards for the index rule makes the task of finding the index rule harder
    a) Condition 1 (WCST) will be more difficult than condition 3 (WCSTC)
    b) Condition 2 (WCSTF) will be more difficult than condition 4 (WCSTFC)
    c) Condition 1 (WCST) will be more difficult than condition 4 (WCSTFC)
3. Some participants will report having an Heureka! moment, so this task can be categorized as an insight task

4. Is impasse associated with inactivity in this task? Is the average move time longer for the spatial task than for the other tasks?

Would you like to add something else? Can we justify Hypothesis 1/c and 2/c?

Analyses

Exclusion of participants

What exclusion criteria do we want to use?

- Exclude participants who did not finish the experiment?
- Exclude participants who did not find out the standard rules?
- Exclude participants who stay inactive for more than a certain amount of time?
- Exclude participants who respond too quickly?

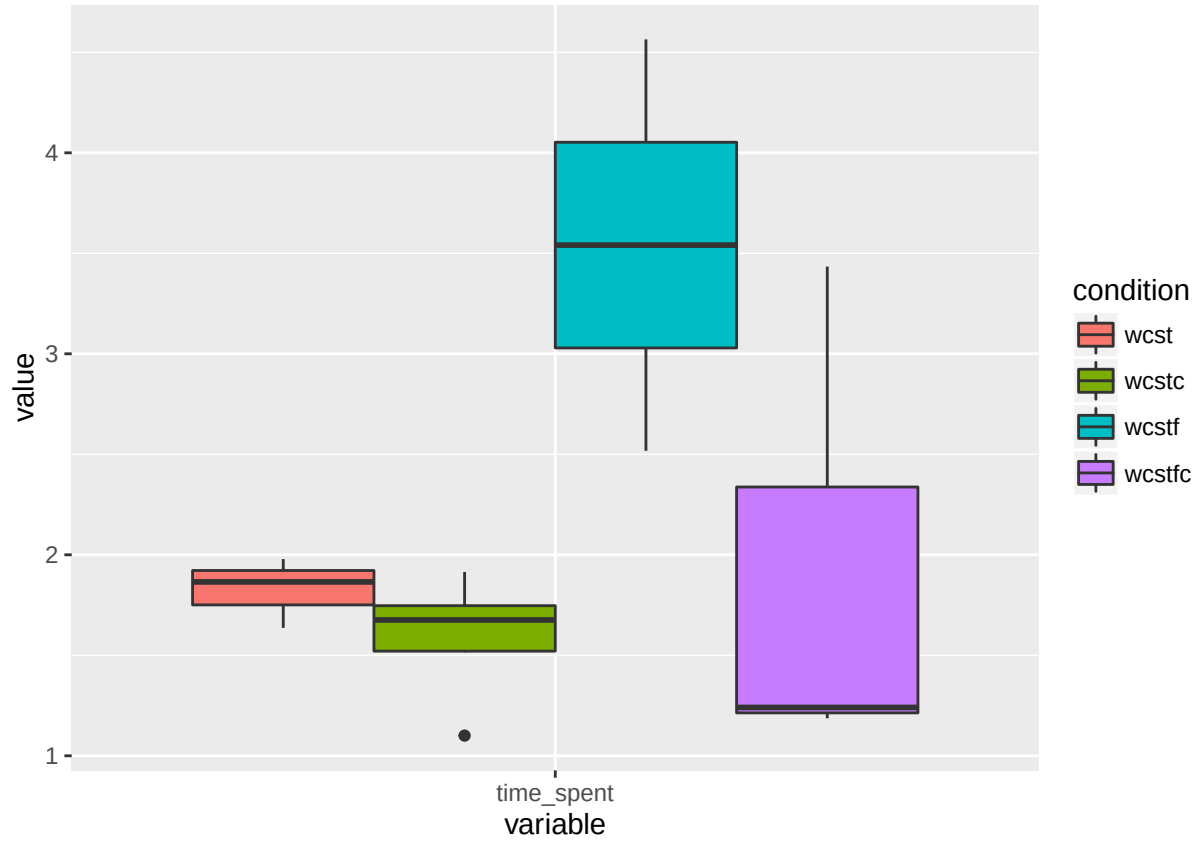
Statistical tests

For deciding whether solving the task in one condition is more difficult than solving it in another, we can use two variables: solution rates and solution times. As for the solution rates, we can construct a four-by-two contingency table (number of solvers and non-solvers in the four conditions). Analyzing a larger than two-by-two contingency table is possible with a Chi-square test, but it is only useful if the categories defining the rows are arranged in a natural order with equal spacing between rows. This is not true in our case, so we will have to analyze two-by-two contingency tables instead, separately for Hypothesis 1 and Hypothesis 2. We chose Fisher's exact test, which is supposed to be more exact than the Chi-square test, especially if the cells in the contingency table contain small values.

Solution rates in the four conditions:

```
## Source: local data frame [4 x 3]
## Groups: condition [4]
##
##   condition solvers_nb nonsolvers_nb
##      (fctr)      (int)         (int)
## 1      wcst          0             3
## 2     wcstc          0             4
## 3     wcstf          0             2
## 4    wcstfc          0             3
```

Solution time is calculated for each participant as the time spent with trying to solve the sequence rule (in minutes). We analyzed the solution time with a two-way ANOVA. Median solution times for the four conditions are shown by the box plots in the following figure:



Number of solvers

Does the order of rules influence the difficulty of finding the sequence rule?

We analyzed the contingency table containing the number of solvers and non-solvers in condition WCST and WCSTF and then separately, for condition WCSTC and WCSTFC. A $p<0.05$ means that the row/column association is statistically significant.

```
##
##  Fisher's Exact Test for Count Data
##
## data:  SR1234[c(1, 3), c(2, 3)]
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0

##
##  Fisher's Exact Test for Count Data
##
## data:  SR1234[c(2, 4), c(2, 3)]
## p-value = 1
```

```
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0
```

Does the order of rules influence the difficulty of finding the sequence rule?

We analyzed the contingency table containing the number of solvers and non-solvers in condition WCST and WCSTC and then separately, for condition WCSTF and WCSTFC. A $p<0.05$ means that the row/column association is statistically significant.

```
##
##  Fisher's Exact Test for Count Data
##
## data:  SR1234[c(1, 2), c(2, 3)]
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0

##
##  Fisher's Exact Test for Count Data
##
## data:  SR1234[c(3, 4), c(2, 3)]
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0
```

Solution time

We checked whether the data was normally distributed with Kolmogorov-Smirnoff test:

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  ST1
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  ST2
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  ST3
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  ST4
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

If the data is normally distributed, we use ANOVA, if it is not, we use Wilcoxon.

```
## Warning in wilcox.test.default(ST1, ST2, alternative = "two.sided", paired
## = FALSE, : Requested conf.level not achievable
```

```
##
##  Wilcoxon rank sum test
##
## data:  ST1 and ST2
## W = 0, p-value = 0.2
## alternative hypothesis: true location shift is not equal to 0
## 80 percent confidence interval:
##  -716421  -18957
## sample estimates:
## difference in location
##                -490369
```