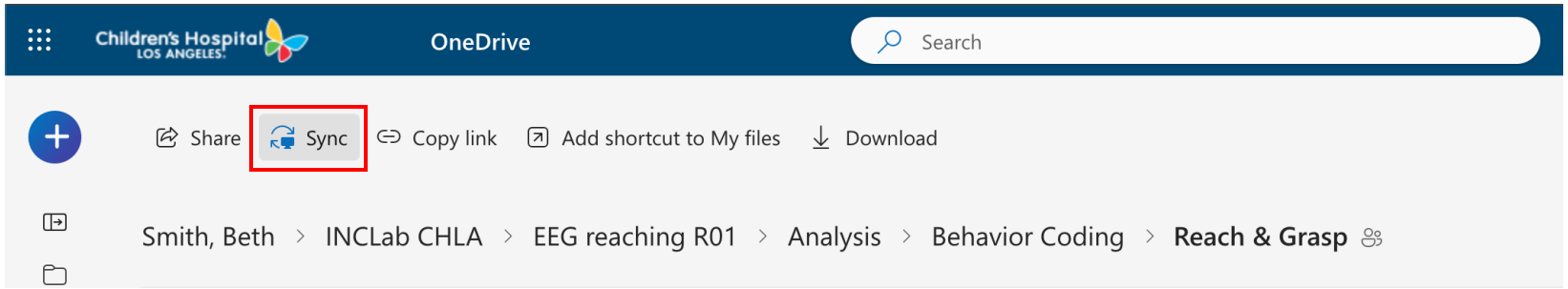
Data Quality Check

자동화를 해보아요

Synchronize the folder: Reach & Grasp

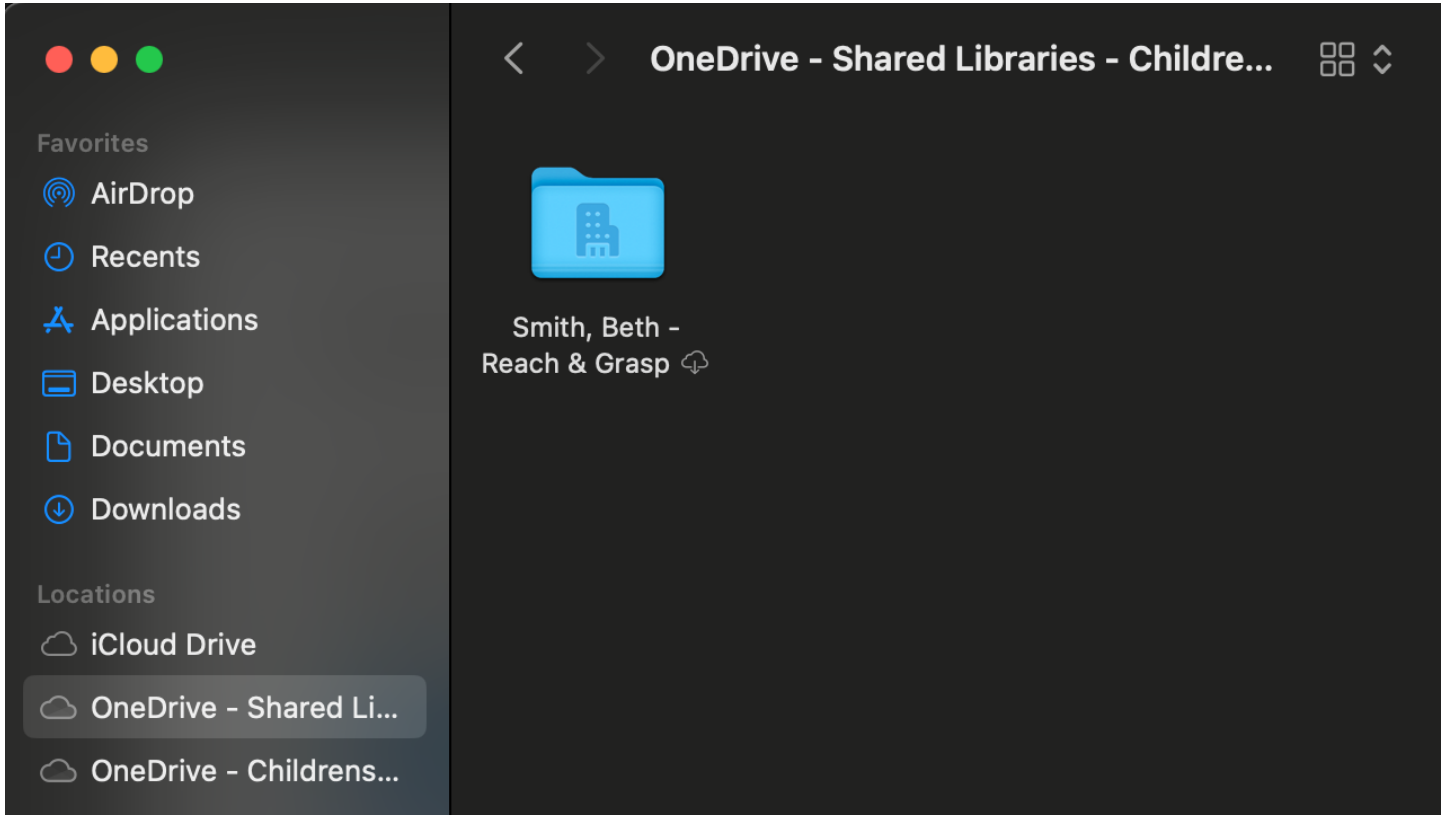
- Sync this folder: Smith, Beth/INCLab CHLA/EEG reaching R01/Analysis/Behavior Coding/**Reach & Grasp**



- Synchronization may take some time.

Synchronize the folder: Reach & Grasp

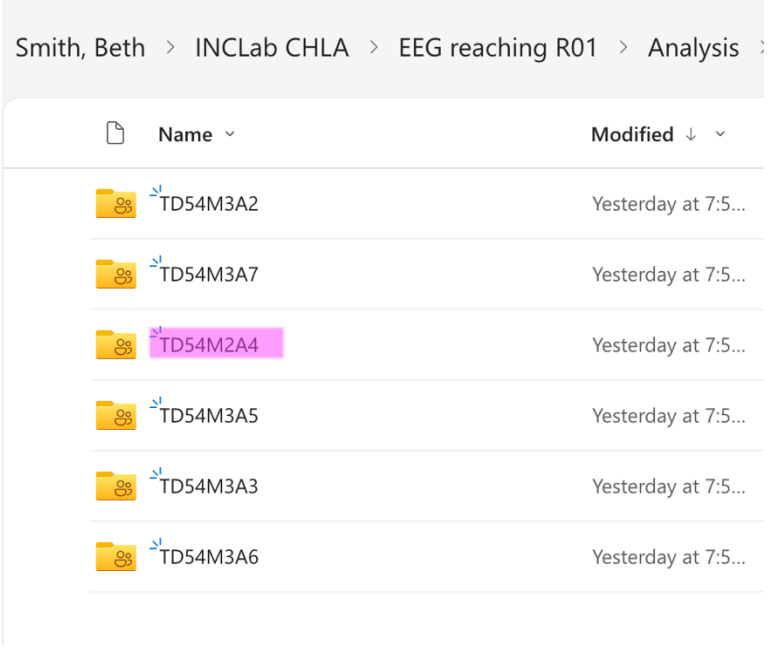
- You will see **Reach & Grasp** folder from finder.



Names corrected

- TD27_M3 -> TD27M3

- TD54M2A4 -> TD54M3A4



Question #1

Answer

#1: Kashish coded the TD27M3 files so it wasn't labeled CC.

- TD27M3 – there's no ***_CC.txt files**

Excel sheet says that there should be TD27M3 coding done by CC

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
			M1	Month 1 (M1)						M2	Month 2 (M2)						M3	Month 3 (M3)					
ID	**CODER**	**Priority**	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	A3	A4	A5	A6	A7
TD13	CC	high	6	X	X	X	X	X	X	6	X	X	x				6	X	X	X	X	X	X
TD17	CC	high	6							5							6	X	X	X	X	X	X
TD24	CC	high	6							5							4	X	X	X	X	--	--
TD26	CC									5							6						
TD27	CC	high	5							6							6	x	x	x	x	x	x
TD30	CC	high	3							6							6	x	x	x	x	x	x

This is the reality

Smith, Beth > INCLab CHLA > EEG reaching R01 > Analysis > Behavior Coding > Reach & Grasp > Data > TD27 > **TD27M3**

Name	Modified	Modified By	File size	Sharing	Activity
TD27M3A7R6_KC.txt	October 4, 2024	Chainani, Kashish	690 bytes	Shared	
TD27M3A6R5_KC.txt	October 4, 2 10/4/2024 2:43 AM	Kashish	749 bytes	Shared	
TD27M3A5R4_KC.eaf	October 4, 2024	Chainani, Kashish	12.2 KB	Shared	
TD27M3A5R4_KC.txt	October 4, 2024	Chainani, Kashish	513 bytes	Shared	

Question #1

- TD27M3 – ok, ***_KC.txt files** exist, but they are not in the correct format; ONLY FOUR COLUMNS

Incorrect output

TD27M3A2R1_KC.txt

LA	0	3550	N
LA	3550	5010	MO
LA	5010	7820	MDT
LA	7820	8900	MO
LA	8900	12340	N
LA	12340	13150	MDX
LA	13150	16030	N
LA	16030	16880	MO
LA	16880	24100	N
LA	24100	25410	MDX
LA	25410	26060	MO
LA	26060	26690	N
LA	26690	28040	MDX
LA	28040	32360	MO
LA	32360	42250	N

Correct output

TD27-M4_A3R2_CC.txt

RA	0	3600	3600	N
RA	3600	4210	610	MO
RA	4210	5020	810	MDX
RA	5020	5460	440	MO
RA	5460	15130	9670	N
RA	15130	15930	800	MO
RA	15930	20600	4670	MDT

Question #2

- What are the accepted labels? (ex. MDX, MO, N, MDG…)

Answer

- Moving (**M**):
  - Direction of toy (**D**)
    - Outcome (*if in direction of toy = D*)
      - Touch (**T**) – any part of the hand/fingers in contact with any part of the toy *without* fingers wrapped around some portion of the toy, ends when there is a clear space or separation between the hand/fingers and the toy.
      - Grasp (**G**) – 2+ fingers wrapped around some portion of the toy, with or without thumb.
        - Consider the reach outcome when coding: if reach results in grasp code as a grasp, *cannot* have (**T**) preceding.
        - Grasp ends when the hand is removed from the toy (i.e., clear separation of space between toy and hand which reverts to **M, O** or **N**).
      - No touch (**X**) – hand does not contact toy; ends when arm starts moving away from the toy (revert to **N** or **O**)
  - **NOT** moving in toy direction (**O**)
    - i.e. can be moving away from the toy.
- Not moving (**N**) – when watching video at regular speed, the arm is not moving.
- Research Assistant pause **(Z)** – on occasions where baby drops toy and is not in the vicinity of the baby, or RA is cleaning it then you use a Z code.
  - This would also hold true if at the beginning the toy is not in the proper position.
- **NOT** visible (**Q**)- hand/arms are being blocked from camera and no clear code can be determined.
  - **Use this code sparingly.

TD27-M4_A2R1_CC.txt

1	RA	0	26220	26220	N
2	RA	26220	27850	1630	MDX
3	RA	27850	28900	1050	MO
4	RA	28900	29390	490	MDT
5	RA	29390	32100	2710	MO
6	RA	32100	43110	11010	N
7	RA	43110	46870	3760	MO
8	RA	46870	47260	390	N
9	RA	47260	53490	6230	MDG
10	RA	53490	56010	2520	N
11	RA	56010	59340	3330	MO
12	RA	59340	61169	1829	N
13	LA	0	53030	53030	N
14	LA	53030	54510	1480	MO
15	LA	54510	57660	3150	N
16	LA	57660	59070	1410	MO
17	LA	59070	61169	2099	N

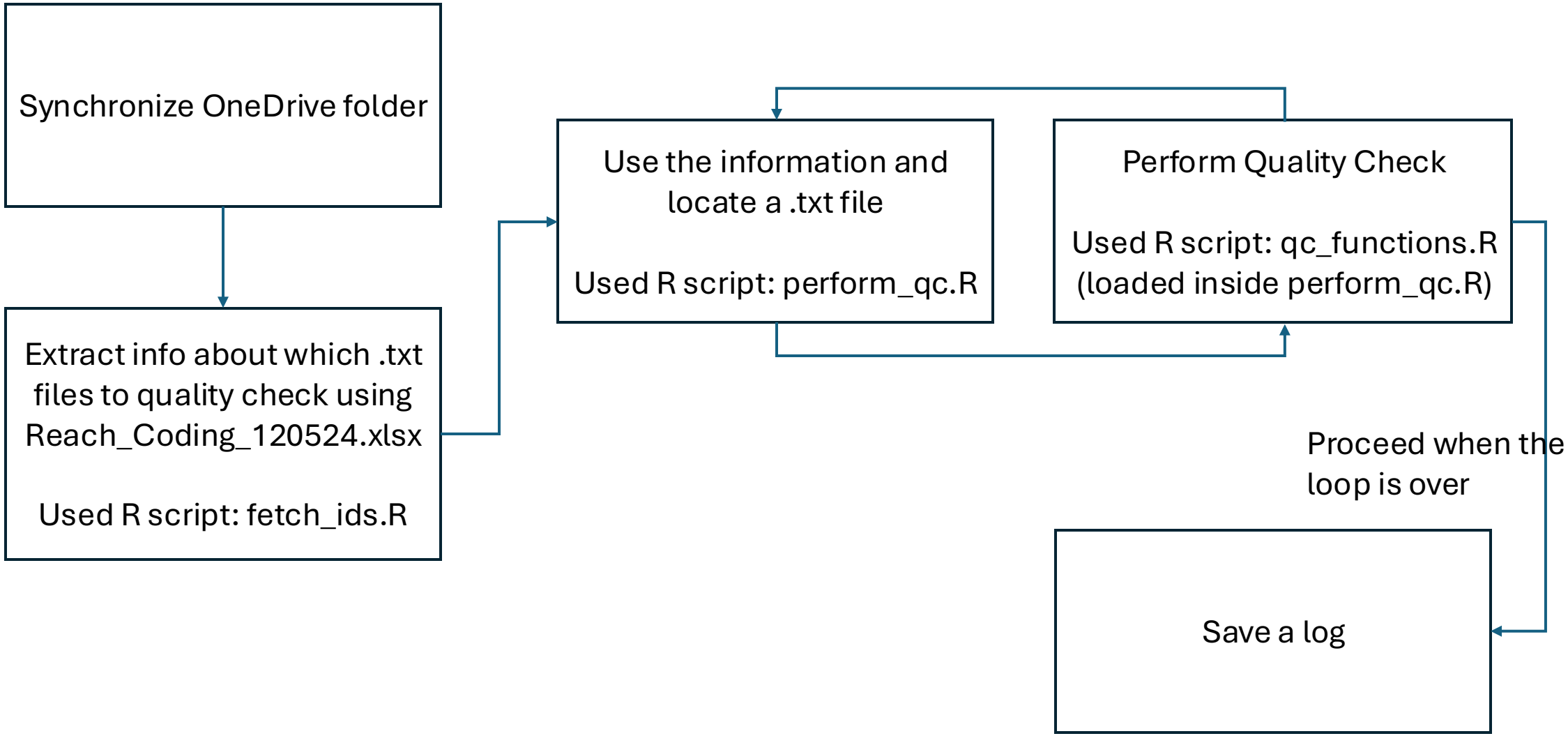
Question #2

- Accepted labels are:
  - N
  - Z
  - Q
  - MO
  - MDT
  - MDG
  - MDX

- No more, right?

- Moving (**M**):
  - Direction of toy (**D**)
    - Outcome (*if in direction of toy = D*)
      - <u>Touch</u> (**T**) – any part of the hand/fingers in contact with any part of the toy *without* fingers wrapped around some portion of the toy, ends when there is a clear space or separation between the hand/fingers and the toy.
      - <u>Grasp</u> (**G**) – 2+ fingers wrapped around some portion of the toy, with or without thumb.
        - Consider the reach outcome when coding: if reach results in grasp code as a grasp, *cannot* have (**T**) preceding.
        - Grasp ends when the hand is removed from the toy (i.e., clear separation of space between toy and hand which reverts to **M, O** or **N**).
      - <u>No touch</u> (**X**) – hand does not contact toy; ends when arm starts moving away from the toy (revert to **N** or **O**)
  - **NOT** moving in toy direction (**O**)
    - i.e. can be moving away from the toy.
- Not moving (**N**) – when watching video at regular speed, the arm is not moving.
- Research Assistant pause **(Z)** – on occasions where baby drops toy and is not in the vicinity of the baby, or RA is cleaning it then you use a Z code.
  - This would also hold true if at the beginning the toy is not in the proper position.
- **NOT** visible (**Q**)- hand/arms are being blocked from camera and no clear code can be determined.
  - **Use this code sparingly.

Workflow


Synchronize OneDrive folder
Extract info about which .txt files to quality check using Reach_Coding_120524.xlsx
Used R script: fetch_ids.R
Use the information and locate a .txt file
Used R script: perform_qc.R
Perform Quality Check
Used R script: qc_functions.R (loaded inside perform_qc.R)
Proceed when the loop is over
Save a log

Workflow: step 2a

- **Read** *Reach_Coding_120524.xlsx*
  - Sheet name: *CC*
  - Columns to attend: *M3* & *M4*

			M1	Month 1 (M1)						M2		M3	Month 3 (M3)						M4	Month 4 (M4)					
ID	**CODER**	**Priority**	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	A3	A4	A5	A6	A7
TD13	CC	high	6	X	X	X	X	X	X	6	X	6	X	X	X	X	X	X	5	x	x	x	x	x	--
TD17	CC	high	6							5		6	X	X	X	X	X	X	6	x	x	x	x	x	x
TD24	CC	high	6							5		4	X	X	X	X	--	--	5	x	x	x	x	x	--
TD26	CC									5		6							4						
TD27	CC	high	5							6		6	x	x	x	x	x	x	4	x	x	x		x	
TD30	CC	high	3							6		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD31	CC	high	3							4		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD53	CC	high	6							6		7													
TD54	CC	high	6							6		6	x	x	x	x	x	x							
TD55	CC	high	6							6		4	x	x	x	x	--	--							
TD16	CC	low	5							No Data		No Data							6						
TD20	CC	low	6							Withdrawn		Withdrawn							Withdrawn						
TD25	CC	low	5							Withdrawn		Withdrawn							Withdrawn						
TD03	CC	medium	6							6		6							6						
TD11	CC	medium	6							No Data		4							6						
TD26	CC	high	5							5		6							4						

Trial Count | NR | CC | SB | KC | EGM | Template | Notes | Training | Reliabilities | +

Workflow: step 2a

You **can't** read the table as intended

1. These are merged in Excel –> a programming language (ex. R) does not recognize merged cells.

			M1	Month 1 (M1)						M2		M3	Month 3 (M3)						M4	Month 4 (M4)					
ID	**CODER**	**Priority**	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	A3	A4	A5	A6	A7
TD13	CC	high	6	X	X	X	X	X	X	6	X	6	X	X	X	X	X	X	5	x	x	x	x	x	--
TD17	CC	high	6							5		6	X	X	X	X	X	X	6	x	x	x	x	x	x
TD24	CC	high	6							5		4	X	X	X	X	--	--	5	x	x	x	x	x	--
TD26	CC									5		6							4						
TD27	CC	high	5							6		6	x	x	x	x	x	x	4	x	x	x		x	
TD30	CC	high	3							6		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD31	CC	high	3							4		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD53	CC	high	6							6		7													
TD54	CC	high	6							6		6	x	x	x	x	x	x							
TD55	CC	high	6							6		4	x	x	x	x	--	--							
TD16	CC	low	5							No Data		No Data							6						
TD20	CC	low	6							Withdrawn		Withdrawn							Withdrawn						
TD25	CC	low	5							Withdrawn		Withdrawn							Withdrawn						
TD03	CC	medium	6							6		6							6						
TD11	CC	medium	6							No Data		4							6						
TD26	CC	high	5							5		6							4						

Trial Count | NR | CC | SB | KC | EGM | Template | Notes | Training | Reliabilities | +

2. You can *skip* the first row when reading this sheet, **knowing** that the first A2~A7 is of M1, the next A2~A7 is of M2, and so on.

You can also make it more explicit so that each set of columns A2~A7 can indicate which Month it is associated with (ex. A2 -> M1A2)

Workflow: step 2a

You **can't** read the table as intended

New column names will be:
'ID', 'CODER', 'PRIORITY', 'M1#_of_trials', 'M1A2', 'M1A3',...

			M1	Month 1 (M1)						M2		M3	Month 3 (M3)						M4	Month 4 (M4)					
ID	CODER	Priority	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	A3	A4	A5	A6	A7
TD13	CC	high	6	X	X	X	X	X	X	6	X	6	X	X	X	X	X	X							

fetch_ids.R

```
# Column names of the excel sheet
colnames = c('ID', 'CODER', 'Priority',
             paste0('M1', c('#_of_trials', paste0('A', 2:7))),
             paste0('M2', c('#_of_trials', paste0('A', 2:7))),
             paste0('M3', c('#_of_trials', paste0('A', 2:7))),
             paste0('M4', c('#_of_trials', paste0('A', 2:7))),
             paste0('M5', c('#_of_trials', paste0('A', 2:7))))
```

paste0(...) is a function concatenating vectors after converting to character.

paste0('A', 2:7) in R will concatenate 'A' with each element of the series (2, 3, 4, 5, 6, 7). Thus the command will return a character vector: ('A2', 'A3', 'A4', 'A5', 'A6', 'A7')

c(...) is a function which combines its arguments.

c('#_of_trials', paste0('A', 2:7)) returns a character vector: ('#_of_trials', 'A2', ..., 'A7')

paste0('M3', c('#_of_trials', paste0('A', 2:7')))
>>> ('M3#_of_trials', 'M3A2', 'M3A3', ..., 'M3A7')

Workflow: step 2a

This is how you read the spreadsheet

fetch_ids.R

```r
# sheet='CC'
record = read_excel(onedrive_path,
                    sheet='CC', skip=1)
colnames(record) = colnames
```

Path to this spreadsheet

Read from the second row

Renaming columns to be more specific about months

Attend to this sheet: CC

fetch_ids.R

```r
# Column names of the excel sheet
colnames = c('ID', 'CODER', 'Priority',
             paste0('M1', c('#_of_trials', paste0('A', 2:7))),
             paste0('M2', c('#_of_trials', paste0('A', 2:7))),
             paste0('M3', c('#_of_trials', paste0('A', 2:7))),
             paste0('M4', c('#_of_trials', paste0('A', 2:7))),
             paste0('M5', c('#_of_trials', paste0('A', 2:7))))
```

			M1	Month 1 (M1)						M2		M3	Month 3 (M3)						M4	Month 4 (M4)					
ID	CODER	Priority	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	A3	A4	A5	A6	A7
TD13	CC	high	6	X	X	X	X	X	X	6	X	6	X	X	X	X	X	X	5	x	x	x	x	x	--
TD17	CC	high	6							5		6	X	X	X	X	X	X	6	x	x	x	x	x	x
TD24	CC	high	6							5		4	X	X	X	X	--	--	5	x	x	x	x	x	--
TD26	CC									5		6							4						
TD27	CC	high	5							6		6	x	x	x	x	x	x	4	x	x	x		x	
TD30	CC	high	3							6		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD31	CC	high	3							4		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD53	CC	high	6							6		7													
TD54	CC	high	6							6		6	x	x	x	x	x	x							
TD55	CC	high	6							6		4	x	x	x	x	--	--							
TD16	CC	low	5							No Data		No Data							6						
TD20	CC	low	6							Withdrawn		Withdrawn							Withdrawn						
TD25	CC	low	5							Withdrawn		Withdrawn							Withdrawn						
TD03	CC	medium	6							6		6							6						
TD11	CC	medium	6							No Data		4							6						
TD26	CC	high	5							5		6							4						

Trial Count | NR | CC | SB | KC | EGM | Template | Notes | Training | Reliabilities | +

Workflow: step 2b

- **Identify** which combinations of ID and month-specific *A* are completed

			M1	Month 1 (M1)						M2		M3	Month 3 (M3)						M4	Month 4 (M4)					
ID	**CODER**	**Priority**	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	# of Trials	A2	A3	A4	A5	A6	A7	# of Trials	A2	A3	A4	A5	A6	A7
TD13	CC	high	6	X	X	X	X	X	X	6	X	6	X	X	X	X	X	X	5	x	x	x	x	x	--
TD17	CC	high	6							5		6	X	X	X	X	X	X	6	x	x	x	x	x	x
TD24	CC	high	6							5		4	X	X	X	X	--	--	5	x	x	x	x	x	--
TD26	CC									5		6							4						
TD27	CC	high	5							6		6	x	x	x	x	x	x	4	x	x	x		x	
TD30	CC	high	3							6		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD31	CC	high	3							4		6	x	x	x	x	x	x	6	x	x	x	x	x	x
TD53	CC	high	6							6		7													
TD54	CC	high	6							6		6	x	x	x	x	x	x							
TD55	CC	high	6							6		4	x	x	x	x	--	--							
TD16	CC	low	5							No Data		No Data							6						
TD20	CC	low	6							Withdrawn		Withdrawn							Withdrawn						
TD25	CC	low	5							Withdrawn		Withdrawn							Withdrawn						
TD03	CC	medium	6							6		6							6						
TD11	CC	medium	6							No Data		4							6						
TD26	CC	high	5							5		6							4						

Trial Count | NR | CC | SB | KC | EGM | Template | Notes | Training | Reliabilities | +

Workflow: step 2b

This is what you read & renamed (columns are hidden for visual representation)

m34 = record[, c(1, 19:24, 25:31)]

A	S	T	U	V	W	X	Z	AA	AB			
ID	M3A2	M3A3	M3A4	M3A5	M3A6	M3A7	M4A2	M4A3	M4A4	M4A5	M4A6	M4A7
TD13	X	X	X	X	X	X	x	x	x	x	x	--
TD17	X	X	X	X	X	X	x	x	x	x	x	x
TD24	X	X	X	X	--	--	x	x	x	x	x	--

In R, this is header, not a data row

Start from the first data row and iterate. In each row, mark the columns whose values are **X** or **x**.
You can then combine ID and the corresponding column names and save them.

fetch_ids.R

```
# M3, M4 entries - S:X, Z:AE, or 19:24, 25:31
m34 = record[,c(1, 19:24, 25:31)]
m34colnames = colnames(m34)

subj = vector()      # ex. TD17
months = vector()    # ex. M3
acts = vector()      # ex. A4
prefixes = vector() # ex. TD17-M3_A4
                     # full filename ex: TD17-M3_A4R3_CC.txt
paths = vector()     # ex. TD17/TD17M3/TD17M3A4
```

Workflow: step 2b

Outer loop direction (**i** changes from 1 to the number of rows

	A	S	T	U	V
	ID	M3A2	M3A3	M3A4	M3A5
m34[1,]	TD13	X	X	X	X
	TD17	X	X	X	X
m34[3,]	TD24	X	X	X	X
	row[1]	row[2]	row[3]	row[4]	

Inner loop direction (**j** changes from 2 to the number of columns)

i = 3, j = 2

Value is **X** or **x**: **TRUE**

subjstr = 'TD24'
temp = ('M', '3', 'A', '2')
monstr = 'M3'
astr = 'A2'
txtstr = 'TD24-M3_A2'
pathstr = 'Data/TD24/TD24M3/TD24M3A2'

subj = ('TD13', 'TD13', ..., 'TD24')
months = ('M3', 'M3', ..., 'M3')
acts = ('A2', 'A3', 'A4', ..., 'A2')
prefixes = ('TD13-M3_A2', 'TD13-M3_A3', ..., 'TD24-M3_A2')
paths = ('Data/TD13/TD13M3/TD13M3A2', ..., 'Data/TD24/TD24M3/TD24M3A2')

fetch_ids.R

```r
for (i in 1:dim(m34)[1]){
    row = m34[i,]        # `row` is a tibble
    for (j in 2:length(row)){
        if (row[j] %in% c('X', 'x')){
            subjstr = row[1]$ID
            # `stringr::str_split()` splits a string into pieces
            temp = str_split(m34colnames[j], "")[[1]]
            # `stringr::str_c()` joins multiple strings into one
            monstr = str_c(temp[1], temp[2])
            astr = str_c(temp[3], temp[4])
            txtstr = str_c(subjstr, '-',
                           monstr, '_',
                           astr)
            # `stringr::str_c()` is very similar to `paste0()`
            # so you can join strings in the following way.
            pathstr = file.path('Data', subjstr,
                                paste0(subjstr, monstr),
                                paste0(subjstr, monstr, astr))
            # add items to vectors
            subj = c(subj, subjstr)
            months = c(months, monstr)
            acts = c(acts, astr)
            prefixes = c(prefixes, txtstr)
            paths = c(paths, pathstr)
        }
    }
}
```

Workflow: step 2b

Organize and save in a table format

fetch_ids.R

```
tab = data.frame(subj=subj,
                 month=months,
                 act=acts,
                 prefix=prefixes,
                 path=paths)

# save the reference to a tab separated file
write.table(tab, file='reference.tsv', sep= "\t",
            row.names=F, col.names=T,
            quote=F)
```

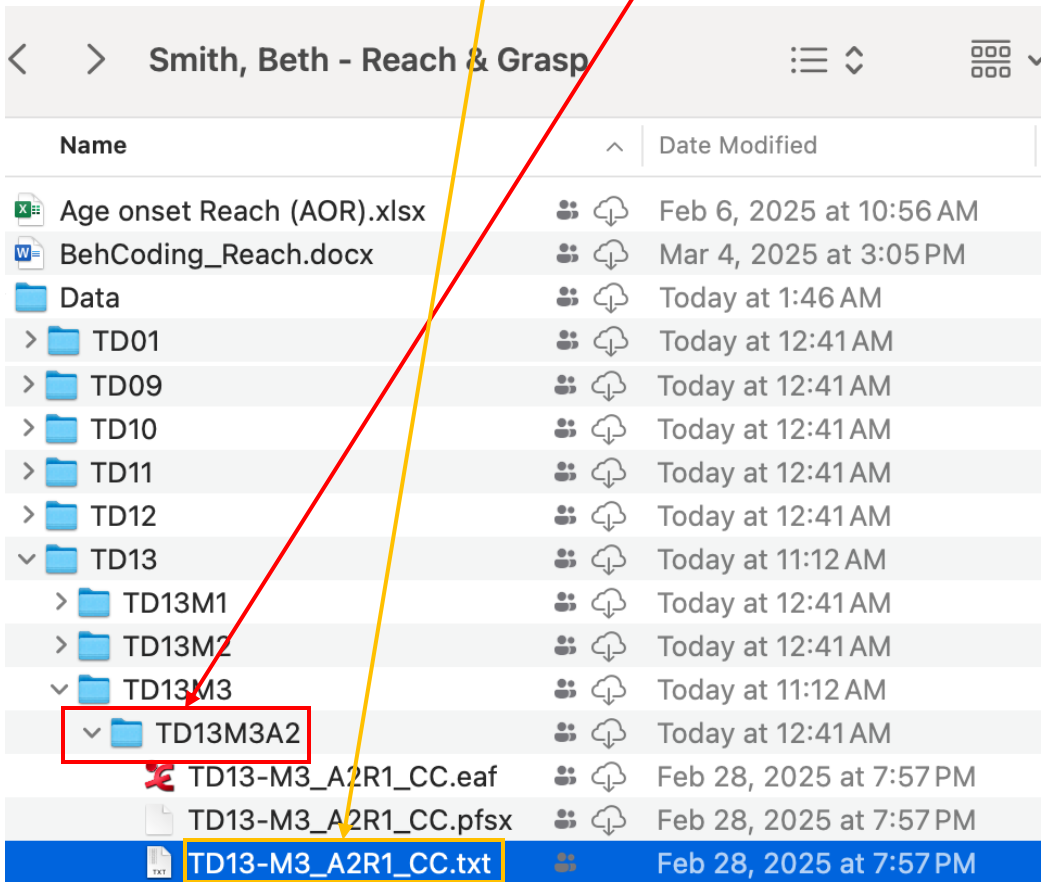
reference.tsv (open with Excel)

A	B	C	D	E
subj	month	act	prefix	path
TD13	M3	A2	TD13-M3_A2	Data/TD13/TD13M3/TD13M3A2
TD13	M3	A3	TD13-M3_A3	Data/TD13/TD13M3/TD13M3A3
TD13	M3	A4	TD13-M3_A4	Data/TD13/TD13M3/TD13M3A4
TD13	M3	A5	TD13-M3_A5	Data/TD13/TD13M3/TD13M3A5
TD13	M3	A6	TD13-M3_A6	Data/TD13/TD13M3/TD13M3A6
TD13	M3	A7	TD13-M3_A7	Data/TD13/TD13M3/TD13M3A7
TD13	M4	A2	TD13-M4_A2	Data/TD13/TD13M4/TD13M4A2
TD13	M4	A3	TD13-M4_A3	Data/TD13/TD13M4/TD13M4A3
TD13	M4	A4	TD13-M4_A4	Data/TD13/TD13M4/TD13M4A4

Workflow: step 3a

- **Locate** .txt files iteratively using the *path* column of *reference.tsv*.
  - There CAN be more than one .txt file in the path. Use *prefix* to match the pattern.

A	B	C	D	E
subj	month	act	prefix	path
TD13	M3	A2	TD13-M3_A2	Data/TD13/TD13M3/TD13M3A2
TD13	M3	A3	TD13-M3_A3	Data/TD13/TD13M3/TD13M3A3
TD13	M3	A4	TD13-M3_A4	Data/TD13/TD13M3/TD13M3A4
TD13	M3	A5	TD13-M3_A5	Data/TD13/TD13M3/TD13M3A5
TD13	M3	A6	TD13-M3_A6	Data/TD13/TD13M3/TD13M3A6
TD13	M3	A7	TD13-M3_A7	Data/TD13/TD13M3/TD13M3A7
TD13	M4	A2	TD13-M4_A2	Data/TD13/TD13M4/TD13M4A2
TD13	M4	A3	TD13-M4_A3	Data/TD13/TD13M4/TD13M4A3
TD13	M4	A4	TD13-M4_A4	Data/TD13/TD13M4/TD13M4A4



Workflow: step 3a

Load reference.tsv

A	B	C	D	E
subj	month	act	prefix	path
TD13	M3	A2	TD13-M3_A2	Data/TD13/TD13M3/TD13M3A2
TD13	M3	A3	TD13-M3_A3	Data/TD13/TD13M3/TD13M3A3
TD13	M3	A4	TD13-M3_A4	Data/TD13/TD13M3/TD13M3A4
TD13	M3	A5	TD13-M3_A5	Data/TD13/TD13M3/TD13M3A5
TD13	M3	A6	TD13-M3_A6	Data/TD13/TD13M3/TD13M3A6
TD13	M3	A7	TD13-M3_A7	Data/TD13/TD13M3/TD13M3A7
TD13	M4	A2	TD13-M4_A2	Data/TD13/TD13M4/TD13M4A2
TD13	M4	A3	TD13-M4_A3	Data/TD13/TD13M4/TD13M4A3
TD13	M4	A4	TD13-M4_A4	Data/TD13/TD13M4/TD13M4A4

perform_qc.R

```
# This will return error if you did not complete 2) above.
references = read.csv('reference.tsv', sep='\t')
# Some problematic folders rejected for now (March 5, 2025)
subdirs_temp = references$path
idx_spare = !grepl("Data/TD27/TD27M3", subdirs_temp)
subdirs = subdirs_temp[idx_spare]
prefixes_temp = references$prefix
prefixes = prefixes_temp[idx_spare]
```

Quick fix related to Question 1

Workflow: step 3a

perform_qc.R

```r
# You also need to load this R script to use functions I wrote.
source('qc_functions.R')

# save the current working directory in case you need to revisit
your_wkdir <- getwd()

###############
# PATH details #
###############

# User's HOME directory (ex. /Users/joh)
HOME = path_home()

# OneDrive specific
Mac_OneDrive_PATH = 'Library/CloudStorage/OneDrive-SharedLibraries-ChildrensHospitalLosAngeles/Smith, Beth - Reach & Grasp'

# combine the two
user_path = paste0(HOME, '/', Mac_OneDrive_PATH)

# paths to .txt files
# ex) /Users/joh/Library/ ... /TD17/TD17M3/TD17M3A2
txtpaths = file.path(user_path, subdirs)
```

Workflow: step 3a

perform_qc.R

```
# `txt_files` will store full file paths of the
# target .txt files
# `dir_ls()` is a function of `fs` package.
# It returns a named character vector.
# using `unname()` is not critical.
txt_files = vector()
for (i in 1:length(txtpaths)){
    txt_files = c(txt_files,
                  unname(dir_ls(txtpaths[i],
                                regex=paste0(prefixes[i],
                                             "R[0-9]_CC\\.txt$"))))
}
```

"[0-9]" means a single digit.
"$" means the end of the string.
"\\" makes sure that the pattern we want has ".txt" at the end.
You can read more about the regular expression in R: jfjelstul.github.io/regular-expressions-tutorial/

Workflow: step 3b

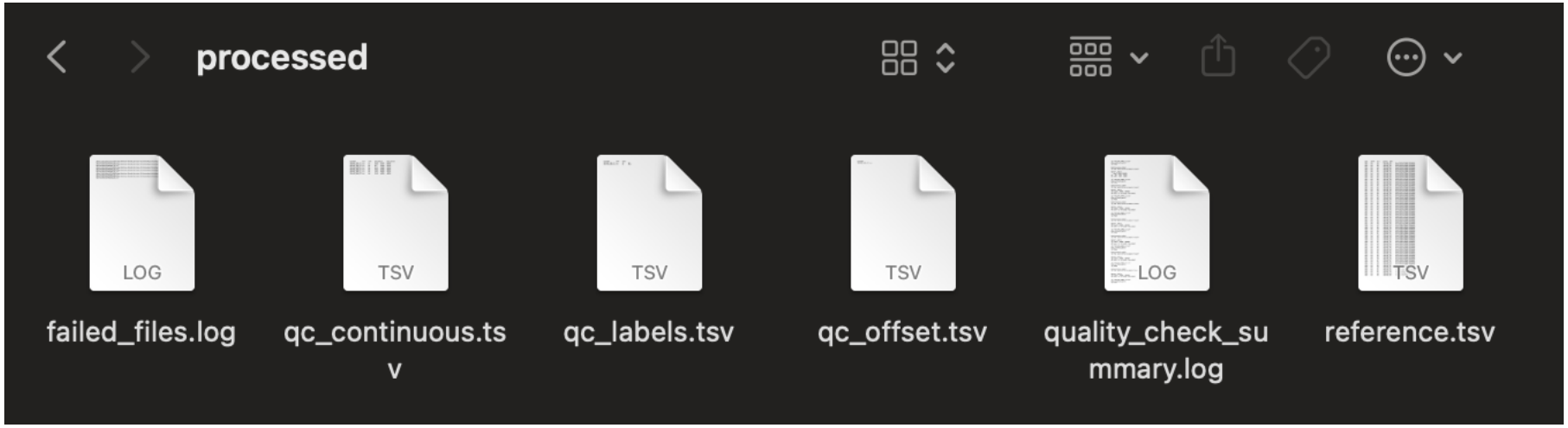
- **Perform quality checks** on the .txt files iteratively

```
# There can be different ways to report the quality check output.
# 1. You can create a long .log file.
#    Use `sink()` to log everything to a log file.
sink('../processed/quality_check_summary.log', append=TRUE, split=FALSE)
for (txt in txt_files){
    print(tail(str_split(txt, '/')[[1]], 1))
    # Logging improved - ChatGPT recommendation
    # Continue Processing even if one file fails
    result = tryCatch({
        qc.all(txt)
    }, error = function(e) {
        paste("Error processing:", txt, ";", conditionMessage(e))
    })
    print(result)
}
sink()
```

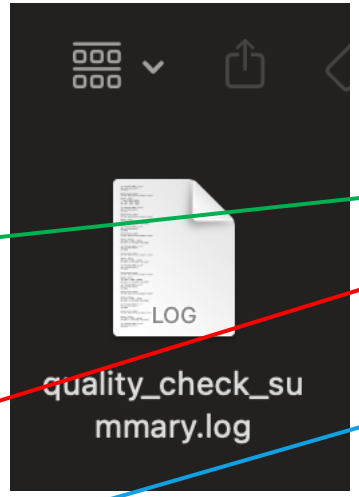
A log created after the looping is finished

quality_check_summary.log

Output


processed
LOG
failed_files.log
TSV
qc_continuous.tsv
TSV
qc_labels.tsv
TSV
qc_offset.tsv
LOG
quality_check_summary.log
TSV
reference.tsv

Summary log file



Under a .txt filename
(ex. [1] “TD13-M3_A3R2_CC.txt”)

three items are listed:

$last_offsets_match

if FALSE, last offsets differ among tiers

$continuously_coded

if no mismatch between adjacent rows,
“No onset-offset mismatch found”
Otherwise, list where the rows mismatch

```
$continuously_coded
[1] "Tier: LA; rows: 19-20; values differ: 59700 vs. 60640"
[2] "Tier: LA; rows: 20-21; values differ: 61377 vs. 59700"
```

$proper_labels

3 column table printed
- label: label a coder put
- UPPER: if FALSE, label is not in uppercase
- PROPER: if FALSE, label is not correct

quality_check_summary.log

```
[1] "TD13-M3CA3R2_CC.txt"
$last_offsets_match
[1] TRUE

$continuously_coded
[1] "No onset-offset mismatch found"

$proper_labels
   label UPPER PROPER
19    N?  TRUE  FALSE
34   MDX?  TRUE  FALSE

[1] "TD13-M3_A4R3_CC.txt"
$last_offsets_match
[1] TRUE

$continuously_coded
[1] "No onset-offset mismatch found"

$proper_labels
[1] label  UPPER  PROPER
<0 rows> (or 0-length row.names)
```

Summary in separate categories

Single column: filename
If a filename is listed, this file has non-unique last offsets

```
filename
TD30-M3_A2R1_CC.txt
```

Three columns: filename, row#, label

filename	row#	label
TD13-M3_A3R2_CC.txt	19	N?
TD13-M3_A3R2_CC.txt	34	MDX?

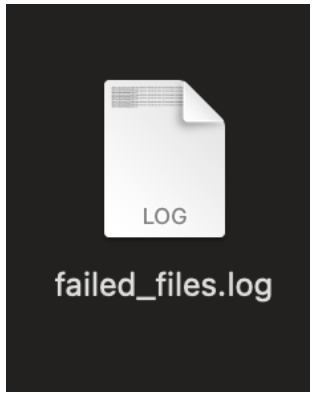
Only problematic files are listed in the files

Five columns: filename, tier, rows, prev_value, next_value

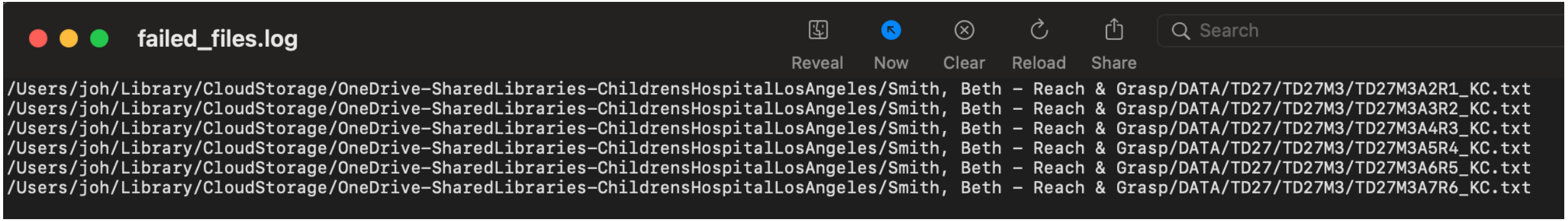
qc_continuous.tsv

filename	tier	rows	prev_value	next_value
TD30-M3_A2R1_CC.txt	LA	19-20	59700	60640
TD30-M3_A2R1_CC.txt	LA	20-21	61377	59700
TD31-M3_A7R6_CC.txt	RA	8-9	24130	25120

Failed_files log



Show file paths where files were not processed – currently all TD27M3 files




```
/Users/joh/Library/CloudStorage/OneDrive-SharedLibraries-ChildrensHospitalLosAngeles/Smith, Beth - Reach & Grasp/DATA/TD27/TD27M3/TD27M3A2R1_KC.txt
/Users/joh/Library/CloudStorage/OneDrive-SharedLibraries-ChildrensHospitalLosAngeles/Smith, Beth - Reach & Grasp/DATA/TD27/TD27M3/TD27M3A3R2_KC.txt
/Users/joh/Library/CloudStorage/OneDrive-SharedLibraries-ChildrensHospitalLosAngeles/Smith, Beth - Reach & Grasp/DATA/TD27/TD27M3/TD27M3A4R3_KC.txt
/Users/joh/Library/CloudStorage/OneDrive-SharedLibraries-ChildrensHospitalLosAngeles/Smith, Beth - Reach & Grasp/DATA/TD27/TD27M3/TD27M3A5R4_KC.txt
/Users/joh/Library/CloudStorage/OneDrive-SharedLibraries-ChildrensHospitalLosAngeles/Smith, Beth - Reach & Grasp/DATA/TD27/TD27M3/TD27M3A6R5_KC.txt
/Users/joh/Library/CloudStorage/OneDrive-SharedLibraries-ChildrensHospitalLosAngeles/Smith, Beth - Reach & Grasp/DATA/TD27/TD27M3/TD27M3A7R6_KC.txt
```