preschool_data_wrangling

Organize data for MRS project

Import data from old tracking sheet and reduce to kids with MRS & relevant columns

```
import numpy as np
import pandas as pd

df = pd.read_csv("/Users/meaghan/OneDrive - University of Calgary/Preschool_data/preschool_mri_data_shee

mrs = df.loc[df["spec"] == '1', ['study_code','subj_id', 'spec_id', 'date_scan', 'hand', 'female', 'mri
```

Recode t1_quality from text to numeric (1=good, 2=medium, 3=bad) & convert date_scan to datetime format

```
mrs = mrs.replace({"good": 1, "medium": 2, "bad":3, "good ":1, "medium ":2, "bad ":3})

mrs['date_scan_dt'] = pd.to_datetime(mrs['date_scan'])
mrs['date_scan'] = mrs['date_scan_dt'].dt.date
```

Then import data from REDCap export (2019-present), rename variables to match mrs dataframe and reduce to kids with MRS & relevant columns & convert date format to datetime format

```
df2 = pd.read_csv("/Users/meaghan/OneDrive - University of Calgary/Preschool_data/redcap_report_mri_data

df2 = df2.rename(columns={"meta_subj_id": "subj_id", "mri_studycode": "study_code", "mri_date": "date_s

mrs2 = df2.loc[df2["spec"] == 1, ['study_code', 'subj_id', 'spec_id', 'date_scan', 'mri_age_y', 'spec',

mrs2['date_scan_dt'] = pd.to_datetime(mrs2['date_scan'], yearfirst=True)
mrs2['date_scan'] = mrs2['date_scan_dt'].dt.date
```

Merge mrs and mrs2 dataframes, clean up, and create new separate columns indicating yes/no for acquisition of ACG and LAG voxels, count total data sets per voxel and how many subjects have MRS data for each voxel

```
mrs_all = mrs.append(mrs2)

mrs_all = mrs_all.replace({"ACG": 1, "LAG": 2, "LAG, ACG": 3, "LAG and ACG": 3})

mrs_all['spec_location'].value_counts()
```

```
## 2.0    324
## 1.0    131
## 3.0      3
## Name: spec_location, dtype: int64
```

```
mrs_all.loc[(mrs_all['spec_location'] == 1) | (mrs_all['spec_location'] == 3), 'spec_acg'] = 1
mrs_all.loc[(mrs_all['spec_location'] == 2), 'spec_acg'] = 0

mrs_all.loc[(mrs_all['spec_location'] == 2) | (mrs_all['spec_location'] == 3), 'spec_lag'] = 1
mrs_all.loc[(mrs_all['spec_location'] == 1), 'spec_lag'] = 0

mrs_all['spec_acg'].value_counts()
```

```
## 0.0    324
## 1.0    134
## Name: spec_acg, dtype: int64
```

```
mrs_all['spec_lag'].value_counts()
```

```
## 1.0    327
## 0.0    131
## Name: spec_lag, dtype: int64
```

```
mrs_all.groupby('spec_acg')['subj_id'].nunique()
```

```
## spec_acg
## 0.0    111
## 1.0    130
## Name: subj_id, dtype: int64
```

```
mrs_all.groupby('spec_lag')['subj_id'].nunique()
```

```
## spec_lag
## 0.0    130
## 1.0    111
## Name: subj_id, dtype: int64
```

Save mrs_all dataframe to .csv

```
mrs_all.to_csv("/Users/meaghan/OneDrive - University of Calgary/Preschool_data/MRS/data/mrs_data_summary
```

Create separate data frames for ACG and LAG data and write to .csv

```
mrs_acg = mrs_all.query('spec_acg == 1')
mrs_acg.to_csv("/Users/meaghan/OneDrive - University of Calgary/Preschool_data/MRS/data/mrs_data_ACG_Oct

mrs_lag = mrs_all.query('spec_lag == 1')
mrs_lag.to_csv("/Users/meaghan/OneDrive - University of Calgary/Preschool_data/MRS/data/mrs_data_LAG_Oct
```

Age range per voxel location

```
mrs_acg['mri_age_y'].min()
```

```
## 2.3381
```

```
mrs_acg['mri_age_y'].max()
```

```
## 8.0246
```

```
mrs_lag['mri_age_y'].min()
```

```
## 2.4887
```

```
mrs_lag['mri_age_y'].max()
```

```
## 10.44
```

Create variable to track data collected after ISMRM abstract and check count

```
mrs_all.loc[(mrs_all['date_scan_dt'] > '2019-07-09'), 'after_jul092019'] = 1
mrs_all.loc[(mrs_all['date_scan_dt'] <= '2019-07-09'), 'after_jul092019'] = 0

mrs_all['after_jul092019'].value_counts()
```

```
## 0.0    403
## 1.0     56
## Name: after_jul092019, dtype: int64
```

```
mrs_all.groupby('after_jul092019')['subj_id'].nunique()
```

```
## after_jul092019
## 0.0    132
## 1.0     50
## Name: subj_id, dtype: int64
```

```
mrs_all.to_csv("/Users/meaghan/OneDrive - University of Calgary/Preschool_data/MRS/data/mrs_data_summary
```

Create list of subject folders for downloading pfiles first create a column of the folder name prefix and a separator make sure mrs_lag is pulled from code as above (reading in from .csv gets the date time format wrong for the string we need) concatenate columns to produce folder name where p file is located & write to a .txt file

```
mrs_lag['pfile_prefix'] = "SPECT-EXAM"
```

```
## <string>:1: SettingWithCopyWarning:
## A value is trying to be set on a copy of a slice from a DataFrame.
## Try using .loc[row_indexer,col_indexer] = value instead
##
## See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
```

```
mrs_lag['sep'] = "-"

mrs_lag['pfile_path'] = mrs_lag['pfile_prefix'].map(str) + mrs_lag['spec_id'].astype(str).str.split('.'

pfile_path_lag=mrs_lag['pfile_path']
pfile_path_lag=pfile_path_lag[~pfile_path_lag.str.contains("nan")]

pfile_path_lag.to_csv("/Users/meaghan/OneDrive - University of Calgary/Preschool_data/MRS/data/mrs_lag_
```