

This is a bunch of random attempts to analyze the data. The biggest problem I ran into was dealing with the different names for the different services. I kept getting errors while attempting to upend each cohort. Im not sure if my analysis was right or not, but it appears those that did S14, did it in conjunction with other services. However, my analysis of those using both S10 and S14 appears to include many other id, thus showing that those using the Family Strengthening are also using the Mental Health Services Looking at the coreelation tables from lecture 5, there appears to be 0.042 correlation, which could be the case given how large our data set was.

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import dhs_util
from dhs_util import *

os.chdir('C:/Users/christian fink/OneDrive/Math485')
df = pd.read_csv('dhs_service_records_synthesized_final.csv')
df = dhs_preprocessing(df)
df, service_map = add_service_label(df)
df = add_age_bin(df)

recipient = get_recipient_attribute(df)

def get_service_attribute(df):
    service = df.groupby(['serv']).agg(
        num_recipient = ('id', 'count'),
        distinct_recipient = ('id', 'nunique'),
        num_month = ('month', 'count'),
        distinct_month = ('month', 'nunique')
    ).reset_index()
    return service
```

```
In [4]: s14_id = df.query("serv == 'S14'")['id'].unique()
one_serv_id = recipient.query("distinct_service == 1").id.to_list()
s14_only = list(set(s14_id) & set(one_serv_id))
```

```
In [5]: s10_id = df.query("serv == 'S10'")['id'].unique()
one_serv_id = recipient.query("distinct_service == 1").id.to_list()
s10_only = list(set(s10_id) & set(one_serv_id))
```

```
In [6]: s14_id_and_s10_id = list(set(s14_id) & set(s10_id))
```

```
In [7]: print("s10_only IDs:", s10_only)
print("s14_only IDs:", s14_only)
print("s14 + s10", s14_id_and_s10_id)
```

s10_only IDs: [243008, 384384, 153698, 243140, 471941, 450311, 56296, 321008, 192415, 30997, 338102, 24728, 526425, 129118, 297119]
s14_only IDs: []
s14 + s10 [233473, 163842, 262149, 186374, 20492, 389132, 366604, 284688, 520213, 452630, 460824, 354329, 319516, 471075, 501798, 143399, 528423, 432174, 471087, 217137, 215091, 528435, 411701, 20534, 45111, 208952, 229433, 518201, 258110, 73799, 53251, 55369, 520273, 342098, 469076, 317524, 98392, 262235, 446557, 47198, 100451, 69731, 47203, 258150, 323687, 16488, 469102, 426097, 118904, 168057, 368765, 366722, 51334, 336007, 178312, 413840, 98449, 270482, 204948, 139415, 524441, 313499, 315547, 428189, 430235, 465055, 376992, 102562, 356515, 82088, 331950, 82109, 133309, 430276, 221385, 489674, 514250, 155853, 319697, 86228, 299226, 385243, 430304, 110821, 366821, 20711, 172265, 239856, 49397, 256246, 248056, 194813, 188670, 246018, 182531, 45318, 334091, 151823, 121108, 190741, 442646, 205079, 375067, 8476, 450850, 67875, 436515, 176420, 366882, 96550, 272682, 522538, 203051, 512302, 129326, 24882, 98610, 76082, 483637, 174386, 375096, 207171, 475460, 434504, 459083, 108878, 168272, 391504, 147794, 319826, 518482, 440661, 373080, 219480, 131419, 377180, 229726, 340321, 262501, 487784, 334188, 323950, 282992, 334194, 108926, 217470, 207235, 401799, 18823, 287113, 162188, 500116, 110997, 371094, 315798, 27034, 426395, 154011, 41374, 104868, 287142, 106919, 76198, 235946, 197035, 111020, 285098, 358833, 496050, 106931, 59830, 84407, 149945, 184762, 219577, 350658, 63939, 465347, 442821, 145869, 195024, 313809, 256467, 395731, 479701, 455134, 68064, 96736, 244194, 367074, 102884, 149992, 518633, 322024, 467434, 57837, 516592, 51698, 145906, 96755, 328183, 78327, 68094, 363011, 440837, 408070, 205327, 217625, 236061, 471583, 498209, 281124, 453157, 148007, 451115, 209455, 234031, 518705, 113199, 506417, 117302, 533048, 162363, 412221, 72256, 240193, 485953, 449089, 506433, 397893, 436810, 522826, 385615, 23120, 299604, 287319, 143959, 29272, 51802, 25179, 180827, 354908, 119388, 203356, 436832, 166498, 10853, 45672, 535145, 170601, 518764, 174701, 29292, 289391, 248435, 35446, 207478, 31350, 60027, 371324, 240258, 332421, 60039, 217735, 475786, 313998, 311952, 236178, 369299, 35478, 420504, 74393, 125596, 357020, 129699, 191141, 332454, 422567, 17066, 381613, 35506, 156339, 363186, 37554, 242364, 393917, 117440, 80581, 531150, 72401, 215764, 465620, 68310, 371415, 86742, 316120, 170715, 305885, 33508, 324328, 135913, 336618, 84715, 455407, 391920, 324338, 197376, 420609, 13057, 398082, 64264, 189200, 535317, 144149, 312087, 348949, 486169, 381718, 277276, 375580, 121630, 162591, 4897, 514850, 183076, 484135, 475945, 373546, 209707, 303921, 305973, 336694, 19254, 92983, 310069, 248638, 406334, 166720, 535363, 473925, 465736, 109385, 426827, 228171, 246609, 258900, 191317, 289621, 144213, 62304, 152420, 332646, 371559, 439147, 220012, 414573, 195437, 279407, 334702, 101234, 4984, 355203, 95111, 228233, 324493, 480143, 308114, 88982, 459672, 21404, 332701, 179101, 388001, 299939, 89002, 535472, 156598, 164791, 15288, 318392, 392121, 62402, 439234, 447428, 252869, 531396, 121799, 424904, 515015, 498638, 232399, 201683, 506840, 281562, 500699, 226271, 52197, 187365, 236520, 140266, 101354, 254956, 158701, 254958, 275439, 230384, 381937, 236529, 533491, 433137, 246782, 439297, 203777, 451587, 140292, 48133, 461831, 85000, 433159, 132114, 347155, 521234, 523299, 91172, 209963, 443442, 259125, 519230, 377919, 50240, 35906, 58436, 310340, 359493, 392261, 529489, 142424, 236634, 224347, 72796, 89181, 121948, 521314, 425058, 44132, 447593, 459888, 54384, 351347, 296052, 377980, 224380, 31870, 420991, 410754, 285826, 476292, 492676, 175237, 191623, 482448, 146581, 255128, 404634, 181413, 285865, 189612, 474288, 521398, 394423, 52408, 119994, 318653, 62654, 357573, 42184, 367816, 498889, 257230, 21714, 396499, 122069, 466134, 390365, 40157, 148707, 208100, 38115, 130277, 267497, 173291, 421100, 226539, 171245, 247033, 255228, 322813, 494847, 312583, 234759, 353545, 386318, 349454, 34065, 202002, 318737, 455956, 54549, 255254, 175382, 265496, 146708, 449813, 27931, 378143, 36133, 23846, 335143, 351529, 54571, 372017, 507189, 384311, 259394, 243012, 357703, 265544, 91466, 478543, 154961, 66901, 220502, 355672, 273757, 355677, 152928, 284001, 124258, 507235, 443748, 3429, 136549, 177509, 193896, 480617, 290154, 298346, 267628, 50541, 163178, 171375, 226675, 497014, 513403, 318845, 464259, 505220, 157062, 3464, 66955, 251276, 347538, 150934, 99735, 355738, 157084, 161184, 357796, 16528

9, 435628, 322988, 13740, 71088, 142769, 484787, 243710, 62905, 116154, 497086, 523710, 144832, 58817, 234944, 492997, 91590, 382407, 171461, 75213, 146896, 347601, 28119, 282072, 34266, 28123, 323036, 212444, 288222, 232929, 476642, 368100, 275941, 499174, 91628, 21997, 435694, 345584, 503282, 507902, 69111, 157177, 97788, 126461, 415229, 263680, 146953, 488972, 523791, 110097, 208401, 388625, 128534, 310807, 333336, 388635, 26144, 271908, 398884, 108070, 501286, 388647, 1577, 42537, 511530, 362028, 253479, 130604, 423474, 87607, 413242, 290363, 196154, 155200, 282185, 230987, 247371, 364107, 388684, 355917, 204363, 245328, 138837, 101976, 81496, 511580, 274017, 509540, 208488, 134763, 458349, 140910, 427631, 429678, 358005, 376440, 253566, 13955, 398984, 241292, 155278, 243343, 186000, 120466, 419475, 173716, 296602, 290460, 396956, 487070, 396959, 196270, 200367, 181936, 351918, 181939, 382643, 73395, 366265, 407237, 108231, 89800, 155337, 317132, 151245, 399056, 218834, 36563, 296665, 335579, 155357, 274142, 188128, 34531, 194276, 401124, 440040, 220905, 247532, 202476, 116462, 380655, 194286, 100083, 433908, 497397, 524021, 370424, 206589, 345853, 270077, 169732, 433926, 300815, 370448, 274193, 423696, 10004, 141079, 263959, 532247, 474904, 427803, 345884, 329509, 87845, 155429, 132905, 257833, 499500, 85805, 157487, 466738, 472891, 421692, 276284, 247616, 44865, 169792, 411460, 507717, 358222, 22351, 71503, 194384, 313171, 282456, 477017, 384857, 98139, 5982, 198495, 159589, 466790, 454505, 42860, 137070, 524153, 155517, 204673, 126853, 233354, 79754, 259978, 335756, 155533, 391055, 458638, 401300, 212886, 384921, 173981, 116638, 487326, 380831, 425885, 110498, 528291, 516004, 419751, 40878, 470959, 182195, 100279, 522171, 395196, 18365, 489408, 110531, 178116, 288712, 468937, 249807, 430032, 178128, 149458, 391129, 448474, 42973, 432096, 526306, 196578, 233444, 176101, 262117, 376805, 438250, 430061, 69617, 387060, 534516, 511991, 239609, 79869, 397310]

```
In [16]: s10_and_s14_only = list(set(s10_only) & set(s14_only))
s10_only_df = pd.DataFrame({'id': s10_only, 'in_s10_only': True})
s14_only_df = pd.DataFrame({'id': s14_only, 'in_s14_only': True})
s14_s10_df = pd.DataFrame({'id': s14_id_and_s10_id, 'in_s10_s14' : True})
s10_and_s14_only_df = pd.DataFrame({'id': s10_and_s14_only, 'in_s10_and_s14_only': False})
combined_df = pd.merge(s10_only_df, s14_only_df, on='id', how='outer')
combined_df = pd.merge(combined_df, s10_and_s14_only_df, on='id', how='outer')
combined_df = pd.merge(combined_df, s14_s10_df, on= 'id', how= 'outer')
combined_df['in_s10_only'] = combined_df['in_s10_only'].fillna(False)
combined_df['in_s14_only'] = combined_df['in_s14_only'].fillna(False)
combined_df['in_s10_and_s14_only'] = combined_df['in_s10_and_s14_only'].fillna(False)
combined_df['in_s10_s14'] = combined_df['in_s10_s14'].fillna(False)
print(combined_df)
#How many are in s10 only?
#How many are in s14 only?
#How many are in both s10 and s14 (maybe others)?
#How many are in only borth s10 and s14?
```

	<code>id</code>	<code>in_s10_only</code>	<code>in_s14_only</code>	<code>in_s10_and_s14_only</code>	<code>in_s10_s14</code>
0	1577	False	False	False	True
1	3429	False	False	False	True
2	3464	False	False	False	True
3	4897	False	False	False	True
4	4984	False	False	False	True
..
833	534516	False	False	False	True
834	535145	False	False	False	True
835	535317	False	False	False	True
836	535363	False	False	False	True
837	535472	False	False	False	True

[838 rows x 5 columns]

```
C:\Users\christian fink\AppData\Local\Temp\ipykernel_11508\2975886726.py:9: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will change in a future version. Call result.infer_objects(copy=False) instead. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
    combined_df['in_s10_only'] = combined_df['in_s10_only'].fillna(False)
C:\Users\christian fink\AppData\Local\Temp\ipykernel_11508\2975886726.py:10: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will change in a future version. Call result.infer_objects(copy=False) instead. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
    combined_df['in_s14_only'] = combined_df['in_s14_only'].fillna(False)
C:\Users\christian fink\AppData\Local\Temp\ipykernel_11508\2975886726.py:11: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will change in a future version. Call result.infer_objects(copy=False) instead. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
    combined_df['in_s10_and_s14_only'] = combined_df['in_s10_and_s14_only'].fillna(False)
C:\Users\christian fink\AppData\Local\Temp\ipykernel_11508\2975886726.py:12: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will change in a future version. Call result.infer_objects(copy=False) instead. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
    combined_df['in_s10_s14'] = combined_df['in_s10_s14'].fillna(False)
```

`s14_only_df=df[df['id'].isin(s14_only)] non_s14_df=df[~df['id'].isin(s14_only)]` 'date', 'service', 'age', 'gender', 'race', 'ethnicity', 'marital', 'education', 'month', 'serv', 'age_bin', 'cohort'

```
In [20]: print(df.columns)
recipient.groupby("distinct_month").id.nunique()
```

```
Index(['id', 'date', 'service', 'age', 'gender', 'race', 'ethnicity',
       'marital', 'education', 'month', 'serv', 'age_bin', 'cohort'],
      dtype='object')
```

```
Out[20]: distinct_month  
1      12774  
2      10685  
3      9970  
4      13509  
5      9051  
6      9428  
7      8556  
8      8531  
9      9205  
10     10096  
11     10094  
12    421900  
Name: id, dtype: int64
```