Descriptives

Contents

1 Metrics

Labeling factors and exploring scales.

1.1 Data preliminaries

Initial point of departure - the databox of the selected variables, described in the Methods chapter.

VARIABLE TITLE	Units	Codename															
CV_SAMPLE_TYPE	1/0	*sample*	1997														
PUBID, YOUTH CASE IDENTIFICATION CODE	integers	*id*	1997														
KEY!**SEX**, RS GENDER	m/f	*sex*	1997														
KEY!**RACE**_ETHNICITY, COMBINED RACE AND ETHNICITY	b/h/m/o	*race*	1997														
KEY!BDATE, RS BIRTHDATE **MONTH**/YEAR	01-12	*bmonth*	1997														
KEY!BDATE, RS BIRTHDATE MONTH/**YEAR**	years	*byear*	1997														
HOW OFTEN PR **ATTEND** CHURCH IN LAST YEAR?	1-8	*attendPR*	1997														
WHAT IS **PR**S CURRENT **RELI**GIOUS **PREF**ERENCE?	1-8	*relprefPR*	1997														
WHAT **RELI**GION WAS **PR RAISED** IN?	1-8	*relraisedPR*	1997														
RS **AGE** IN **MON**THS AS OF INTERVIEW DATE	months	*agemon*	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
RS **AGE** AT INTERVIEW DATE	years	*ageyear*	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
# DAYS PER WEEK TYPICALLY **FAM**ILY DOES SOMETHING **REL**IGIOUS	# days	*famrel*	1997	1998	1999	2000											
HOW OFTEN R **ATTEND**ED WORSHIP SERVICE IN PAST 12 MONTHS	1-8	*attend*				2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
R DOES NOT NEED RELIGION FOR GOOD **VALUES**	y/n	*values*						2002			2005			2008			2011
GOD NOTHING **TO DO** HAPPENS TO R	y/n	*todo*						2002			2005			2008			2011
R BELIEVES RELIGIOUS TEACHINGS ARE TO BE **OBEYED** EXACTLY AS WRITTEN	y/n	*obeyed*						2002			2005			2008			2011
R **PRAY**S MORE THAN ONCE A DAY	y/n	*pray*						2002			2005			2008			2011
R ASKS GOD HELP MAKE **DECISIONS**	y/n	*decisions*						2002			2005			2008			2011
WHAT IS R'S CURRENT **RELIGIOUS** PREFERENCE?	cats:35	*relpref*									2005			2008			2011
R A **BORN-AGAIN** EVANGELICAL CHRISTIAN?	y/n	*bornagain*												2008			2011
IMPORTANCE OF RELIGIOUS **FAITH** IN DAILY LIFE	1-5	*faith*												2008			2011
HOW OFTEN R FELT CALM AND **PEACEFUL** IN PAST MONTH	1-4	*calm*				2000		2002		2004		2006		2008		2010	
HOW OFTEN R FELT DOWN OR **BLUE** IN PAST MONTH	1-4	*blue*				2000		2002		2004		2006		2008		2010	
HOW OFTEN R HAS BEEN A **HAPPY** PERSON IN PAST MONTH	1-4	*happy*				2000		2002		2004		2006		2008		2010	
HOW OFTEN R **DEPRESSED** IN LAST MONTH	1-4	*depressed*				2000		2002		2004		2006		2008		2010	
HOW OFTEN R HAS BEEN A **NERVOUS** PERSON IN PAST MONTH	1-4	*nervous*				2000		2002		2004		2006		2008		2010	
HOW MANY HOURS PER WEEK DOES R WATCH **TELEVISION**	cats:6	*tv*						2002					2007	2008	2009	2010	2011
HOW MANY HOURS PER WEEK DOES R USE A **COMPUTER**	cats:6	*computer*						2002					2007	2008	2009	2010	2011
CURRENTLY HAVE ACCESS TO **INTERNET**?	y/n	*internet*						2002	2003	2004	2005	2006	2007	2008	2009	2010	2011

This databox corresponds to the dataset **dsL** produced by Derive_dsL_from_Extract report, given in the Appendix.

```
dsL<-readRDS("./Data/Derived/dsL.rds")
```

Time Invariant | Time Variant

sample	id	sex	race	bmonth	byear	attendPR	relprefPR	relraisedPR	year	agemon	ageyear	famrel	attend	values	todo	obeyed	pray	decisions	relpref	bornagain	faith	calm	blue	happy	depressed	nervous	tv	computer	internet
1	1	2	4	9	1981	7	21	21	1997	190	15	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	1	2	4	9	1981	7	21	21	1998	206	17	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	1	2	4	9	1981	7	21	21	1999	219	18	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	1	2	4	9	1981	7	21	21	2000	231	19	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	3	3	3	3	3	NA	NA	NA
1	1	2	4	9	1981	7	21	21	2001	243	20	NA	6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	1	2	4	9	1981	7	21	21	2002	256	21	NA	2	1	1	1	0	1	NA	NA	NA	4	2	3	2	1	2	5	NA
1	1	2	4	9	1981	7	21	21	2003	266	22	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1
1	1	2	4	9	1981	7	21	21	2004	279	23	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	4	1	4	1	1	NA	NA	0
1	1	2	4	9	1981	7	21	21	2005	290	24	NA	1	0	1	0	0	1	21	NA	NA	NA	NA	NA	NA	NA	NA	NA	1
1	1	2	4	9	1981	7	21	21	2006	302	25	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	4	1	4	1	1	NA	NA	1
1	1	2	4	9	1981	7	21	21	2007	313	26	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2	6	1
1	1	2	4	9	1981	7	21	21	2008	325	27	NA	1	0	1	0	0	1	21	NA	3	3	3	3	3	3	NA	NA	1
1	1	2	4	9	1981	7	21	21	2009	337	28	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2	6	1
1	1	2	4	9	1981	7	21	21	2010	350	29	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	3	3	3	3	3	1	6	1
1	1	2	4	9	1981	7	21	21	2011	360	29	NA	1	0	1	0	0	1	21	NA	1	NA	NA	NA	NA	NA	NA	6	1
1	2	1	2	7	1982	NA	NA	NA	1997	178	14	3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	2	1	2	7	1982	NA	NA	NA	1998	196	16	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Figure 3.3 Generic dataset used in the current study, view for one respondent

Note that the variable **year** serves as a natural devided between time invariant (TIvars) and time variant (TVvars) variables. All modeling operations beging with subsetting this dataset. For the grammer rules of operations with relevant data see Data Manipulation Guide.

1.2 Labeling Factor Levels

Review of the item reference cards shows that initially, all items were recorded on some discrete scale, either counting occasions or assigning an intiger to a category of response. However, data were saved as numerical values or intigers

```r
ds<- dsL %>%
  dplyr::select(
        sample, id, sex, race, bmonth,byear, attendPR, relprefPR,relraisedPR,
    year,
        agemon, ageyear, famrel, attend,
        values, todo, obeyed, pray, decisions,
        relpref, bornagain, faith,
        calm, blue, happy, depressed, nervous,
        tv, computer, internet)
str(ds)
```

```
'data.frame':   134745 obs. of  30 variables:
 $ sample     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ id         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ sex        : int  2 2 2 2 2 2 2 2 2 2 ...
 $ race       : int  4 4 4 4 4 4 4 4 4 4 ...
 $ bmonth     : int  9 9 9 9 9 9 9 9 9 9 ...
 $ byear      : int  1981 1981 1981 1981 1981 1981 1981 1981 1981 1981 ...
 $ attendPR   : int  7 7 7 7 7 7 7 7 7 7 ...
 $ relprefPR  : int  21 21 21 21 21 21 21 21 21 21 ...
 $ relraisedPR: int  21 21 21 21 21 21 21 21 21 21 ...
 $ year       : int  1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 ...
 $ agemon     : num  190 206 219 231 243 256 266 279 290 302 ...
 $ ageyear    : num  15 17 18 19 20 21 22 23 24 25 ...
 $ famrel     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ attend     : num  NA NA NA 1 6 2 1 1 1 1 ...
 $ values     : num  NA NA NA NA NA 1 NA NA 0 NA ...
 $ todo       : num  NA NA NA NA NA 1 NA NA 1 NA ...
 $ obeyed     : num  NA NA NA NA NA 1 NA NA 0 NA ...
 $ pray       : num  NA NA NA NA NA 0 NA NA 0 NA ...
 $ decisions  : num  NA NA NA NA NA 1 NA NA 1 NA ...
 $ relpref    : num  NA NA NA NA NA NA NA NA 21 NA ...
 $ bornagain  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ faith      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ calm       : num  NA NA NA 3 NA 4 NA 4 NA 4 ...
 $ blue       : num  NA NA NA 3 NA 2 NA 1 NA 1 ...
 $ happy      : num  NA NA NA 3 NA 3 NA 4 NA 4 ...
 $ depressed  : num  NA NA NA 3 NA 2 NA 1 NA 1 ...
 $ nervous    : num  NA NA NA 3 NA 1 NA 1 NA 1 ...
 $ tv         : num  NA NA NA NA NA 2 NA NA NA NA ...
 $ computer   : num  NA NA NA NA NA 5 NA NA NA NA ...
 $ internet   : num  NA NA NA NA NA NA 1 0 1 1 ...
```

LabelingFactorLevels.R sourced at the end of Derive_dsL_from_Extract matches numeric values with response labels from the questionnaire and adds to **dsL** copies of the variables, saved as labeled factors. For estimations routines such as lme4 or graphing functions such as ggplot2, the data type (string,numeric, factor) is a meaningful input, so a quick access to both formats frequently proves useful. It is convenient to think that **dsL** contains only

```r
ncol(dsL)/2
```

```
[1] 30
```

variables, but each of them has a double, a labeled factor.

```r
str(dsL)
```

```
'data.frame':	134745 obs. of  60 variables:
 $ sample      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ id          : int  1 1 1 1 1 1 1 1 1 1 ...
 $ sex         : int  2 2 2 2 2 2 2 2 2 2 ...
 $ race        : int  4 4 4 4 4 4 4 4 4 4 ...
 $ bmonth      : int  9 9 9 9 9 9 9 9 9 9 ...
 $ byear       : int  1981 1981 1981 1981 1981 1981 1981 1981 1981 1981 ...
 $ attendPR    : int  7 7 7 7 7 7 7 7 7 7 ...
 $ relprefPR   : int  21 21 21 21 21 21 21 21 21 21 ...
 $ relraisedPR : int  21 21 21 21 21 21 21 21 21 21 ...
 $ year        : int  1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 ...
 $ agemon      : num  190 206 219 231 243 256 266 279 290 302 ...
 $ ageyear     : num  15 17 18 19 20 21 22 23 24 25 ...
 $ famrel      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ attend      : num  NA NA NA 1 6 2 1 1 1 1 ...
 $ values      : num  NA NA NA NA NA 1 NA NA 0 NA ...
 $ todo        : num  NA NA NA NA NA 1 NA NA 1 NA ...
 $ obeyed      : num  NA NA NA NA NA 1 NA NA 0 NA ...
 $ pray        : num  NA NA NA NA NA 0 NA NA 0 NA ...
 $ decisions   : num  NA NA NA NA NA 1 NA NA 1 NA ...
 $ relpref     : num  NA NA NA NA NA NA NA NA 21 NA ...
 $ bornagain   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ faith       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ calm        : num  NA NA NA 3 NA 4 NA 4 NA 4 ...
 $ blue        : num  NA NA NA 3 NA 2 NA 1 NA 1 ...
 $ happy       : num  NA NA NA 3 NA 3 NA 4 NA 4 ...
 $ depressed   : num  NA NA NA 3 NA 2 NA 1 NA 1 ...
 $ nervous     : num  NA NA NA 3 NA 1 NA 1 NA 1 ...
 $ tv          : num  NA NA NA NA NA 2 NA NA NA NA ...
 $ computer    : num  NA NA NA NA NA 5 NA NA NA NA ...
 $ internet    : num  NA NA NA NA NA NA 1 0 1 1 ...
 $ sampleF     : Ord.factor w/ 2 levels "Cross-Sectional"<..: 1 1 1 1 1 1 1 1 1 1 ...
 $ idF         : Factor w/ 8983 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ sexF        : Ord.factor w/ 3 levels "Male"<"Female"<..: 2 2 2 2 2 2 2 2 2 2 ...
 $ raceF       : Ord.factor w/ 4 levels "Black"<"Hispanic"<..: 4 4 4 4 4 4 4 4 4 4 ...
 $ bmonthF     : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 9 9 9 9 9 9 9 9 9 9 ...
 $ byearF      : Factor w/ 5 levels "1980","1981",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ attendPRF   : Ord.factor w/ 8 levels "Never"<"Once or Twice"<..: 7 7 7 7 7 7 7 7 7 7 ...
 $ relprefPRF  : Ord.factor w/ 33 levels "Catholic"<"Baptist"<..: 21 21 21 21 21 21 21 21 21 21 ...
 $ relraisedPRF: Ord.factor w/ 33 levels "Catholic"<"Baptist"<..: 21 21 21 21 21 21 21 21 21 21 ...
 $ yearF       : Factor w/ 15 levels "1997","1998",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ agemonF     : Factor w/ 244 levels "146","147","148",..: 45 61 74 86 98 111 121 134 145 157 ...
 $ ageyearF    : Factor w/ 21 levels "12","13","14",..: 4 6 7 8 9 10 11 12 13 14 ...
 $ famrelF     : Factor w/ 8 levels "0","1","2","3",..: NA NA NA NA NA NA NA NA NA NA ...
 $ attendF     : Ord.factor w/ 8 levels "Never"<"Once or Twice"<..: NA NA NA 1 6 2 1 1 1 1 ...
 $ valuesF     : Ord.factor w/ 2 levels "FALSE/less Religious"<..: NA NA NA NA NA 2 NA NA 1 NA ...
 $ todoF       : Ord.factor w/ 2 levels "FALSE/less Religious"<..: NA NA NA NA NA 2 NA NA 2 NA ...
 $ obeyedF     : Ord.factor w/ 2 levels "FALSE/less Religious"<..: NA NA NA NA NA 2 NA NA 1 NA ...
 $ prayF       : Ord.factor w/ 2 levels "FALSE/less Religious"<..: NA NA NA NA NA 1 NA NA 1 NA ...
 $ decisionsF  : Ord.factor w/ 2 levels "FALSE/less Religious"<..: NA NA NA NA NA 2 NA NA 2 NA ...
 $ relprefF    : Ord.factor w/ 33 levels "Catholic"<"Baptist"<..: NA NA NA NA NA NA NA NA 21 NA ...
 $ bornagainF  : Ord.factor w/ 2 levels "NO"<"YES": NA NA NA NA NA NA NA NA NA NA ...
```

```
 $ faithF     : Ord.factor w/ 5 levels "Exrtemely"<"Very"<..: NA NA NA NA NA NA NA NA NA NA ...
 $ calmF      : Ord.factor w/ 4 levels "All of the time"<..: NA NA NA NA NA NA NA NA NA NA ...
 $ blueF      : Ord.factor w/ 4 levels "All of the time"<..: NA NA NA NA NA NA NA NA NA NA ...
 $ happyF     : Ord.factor w/ 4 levels "All of the time"<..: NA NA NA NA NA NA NA NA NA NA ...
 $ depressedF : Ord.factor w/ 4 levels "All of the time"<..: NA NA NA NA NA NA NA NA NA NA ...
 $ nervousF   : Ord.factor w/ 4 levels "All of the time"<..: NA NA NA NA NA NA NA NA NA NA ...
 $ tvF        : Ord.factor w/ 6 levels "less than 2"<..: NA NA NA NA NA 2 NA NA NA NA ...
 $ computerF  : Ord.factor w/ 6 levels "None"<"less than 1"<..: NA NA NA NA NA 5 NA NA NA NA ...
 $ internetF  : Ord.factor w/ 2 levels "No"<"Yes": NA NA NA NA NA NA 2 1 2 2 ...
```

This give a certain flexibity in assembling needed dataset quickly and have access to factor labels. One can alternate between the raw metric and labeled factor by adding “F” suffix to the end of the variable name:

```
ds<- dsL %>%
  dplyr::filter(id==25) %>%
  dplyr::select(id,byear,year, attend,attendF)
ds
```

```
   id byear year attend            attendF
1  25  1983 1997     NA               <NA>
2  25  1983 1998     NA               <NA>
3  25  1983 1999     NA               <NA>
4  25  1983 2000      5  About twice/month
5  25  1983 2001      7 Several times/week
6  25  1983 2002      7 Several times/week
7  25  1983 2003      2      Once or Twice
8  25  1983 2004      7 Several times/week
9  25  1983 2005      5  About twice/month
10 25  1983 2006      7 Several times/week
11 25  1983 2007      5  About twice/month
12 25  1983 2008      7 Several times/week
13 25  1983 2009      7 Several times/week
14 25  1983 2010      7 Several times/week
15 25  1983 2011      7 Several times/week
```

Having quick access to factor labels will be especially useful during graph production.

1.3 Time metrics : Age, Period, Cohort

NLSY97 sample includes individuals from five cohorts, born between 1980 and 1984.The following graphics shows how birth cohort, age of respondents, and round of observation are related in NSLY97.

Wide age

	Age in years 13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Born in 1980					1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
1981				1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	
1982			1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011		
1983		1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011			
1984	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011				
															Wave				

Wide wave

	Waves of measurement 1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Born in 1980	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1981	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1982	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1983	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1984	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
															Age

Long wave

	Born in 1980	1981	1982	1983	1984	
Wave: 1997	17	16	15	14	13	
1998	18	17	16	15	14	
1999	19	18	17	16	15	
2000	20	19	18	17	16	
2001	21	20	19	18	17	
2002	22	21	20	19	18	
2003	23	22	21	20	19	
2004	24	23	22	21	20	
2005	25	24	23	22	21	
2006	26	25	24	23	22	
2007	27	26	25	24	23	
2008	28	27	26	25	24	
2009	29	28	27	26	25	
2010	30	29	28	27	26	
2011	31	30	29	28	27	Age

Long age

	Born in 1980	1981	1982	1983	1984	
Age years 13					1997	
14				1997	1998	
15			1997	1998	1999	
16		1997	1998	1999	2000	
17	1997	1998	1999	2000	2001	
18	1998	1999	2000	2001	2002	
19	1999	2000	2001	2002	2003	
20	2000	2001	2002	2003	2004	
21	2001	2002	2003	2004	2005	
22	2002	2003	2004	2005	2006	
23	2003	2004	2005	2006	2007	
24	2004	2005	2006	2007	2008	
25	2005	2006	2007	2008	2009	
26	2006	2007	2008	2009	2010	
27	2007	2008	2009	2010	2011	Wave
28	2008	2009	2010	2011		
29	2009	2010	2011			
30	2010	2011				
31	2011					

NSLY97 contains static (**bmonth**, **byear**) and dynamic (**agemon**, **ageyear**) indicators of age :

```
ds<- dsL %>%
  dplyr::filter(id==25, year %in% c(1997:2011)) %>%
  dplyr::select(id,byear,bmonthF,year,agemon,ageyear)
print(ds)
```

```
   id byear bmonthF year agemon ageyear
```

```
1  25  1983     Mar 1997    167      13
2  25  1983     Mar 1998    188      15
3  25  1983     Mar 1999    201      16
4  25  1983     Mar 2000    214      17
5  25  1983     Mar 2001    226      18
6  25  1983     Mar 2002    236      19
7  25  1983     Mar 2003    254      21
8  25  1983     Mar 2004    261      21
9  25  1983     Mar 2005    272      22
10 25  1983     Mar 2006    284      23
11 25  1983     Mar 2007    295      24
12 25  1983     Mar 2008    307      25
13 25  1983     Mar 2009    319      26
14 25  1983     Mar 2010    332      27
15 25  1983     Mar 2011    342      28
```

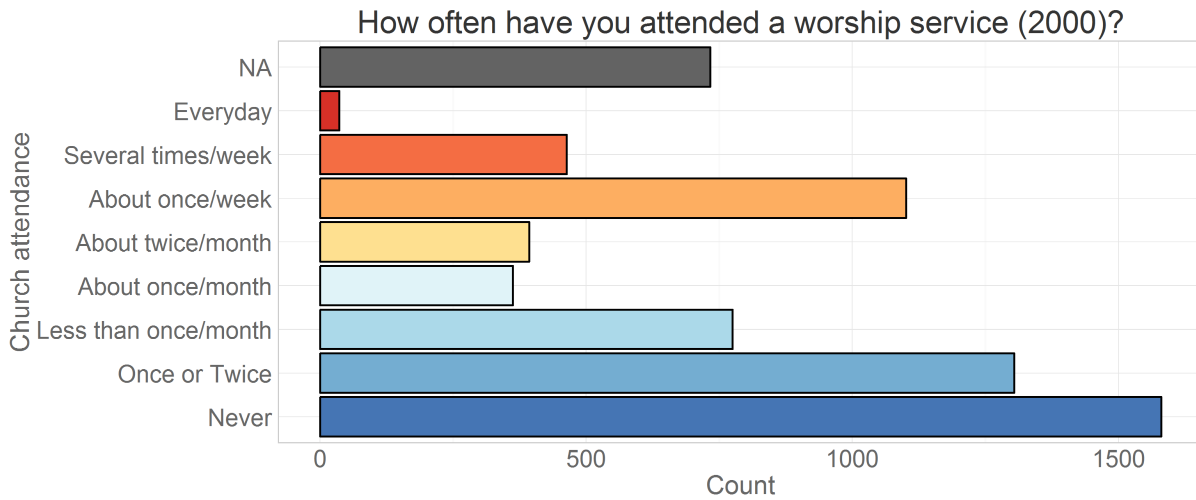
When transforming the metric of time, and using biological age instead of year of measurement as the temporal dimension, the value of age at the time of the interview will be computed as **age** = **agemon**/12

```
ds<- dsL %>%
  dplyr::filter(id==25, year %in% c(1997:2011)) %>%
  dplyr::select(id,bmonthF,byear,year, agemon,ageyear) %>%
  dplyr::mutate (age =  agemon/12)
print(ds)
```

```
   id bmonthF byear year agemon ageyear   age
1  25     Mar  1983 1997    167      13 13.92
2  25     Mar  1983 1998    188      15 15.67
3  25     Mar  1983 1999    201      16 16.75
4  25     Mar  1983 2000    214      17 17.83
5  25     Mar  1983 2001    226      18 18.83
6  25     Mar  1983 2002    236      19 19.67
7  25     Mar  1983 2003    254      21 21.17
8  25     Mar  1983 2004    261      21 21.75
9  25     Mar  1983 2005    272      22 22.67
10 25     Mar  1983 2006    284      23 23.67
11 25     Mar  1983 2007    295      24 24.58
12 25     Mar  1983 2008    307      25 25.58
13 25     Mar  1983 2009    319      26 26.58
14 25     Mar  1983 2010    332      27 27.67
15 25     Mar  1983 2011    342      28 28.50
```

1.4 Attendance

NLSY97 asked to report church attendance (**attend**)for the past 12 months preceding the interview date. The response offered a choice of 7 categories ordered by magnitude.



2 Descriptives

Basic descriptives reports on selected NLSY97 items

2.1 Basic demographics

A clean dataset **dsL** contains data on

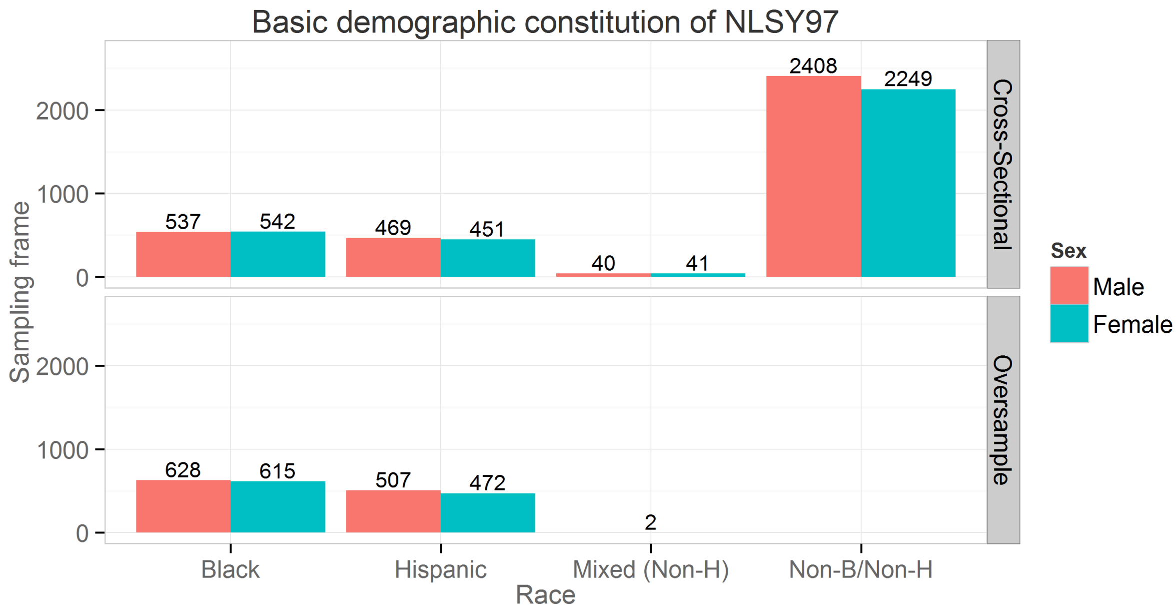
```
dplyr::summarize(dsL,N=n_distinct(id))
```

```
     N
1 8983
```

respondents. Of them one (id = 467) was removed from the dataset due to abberant age score that seemed as a coding mistake. NLSY97 contains representative household sample and the oversample of racial minorities.

```
ds<- dsL %>%
  dplyr::group_by(sampleF) %>%
  dplyr::summarize (count=n_distinct(id))
ds
```

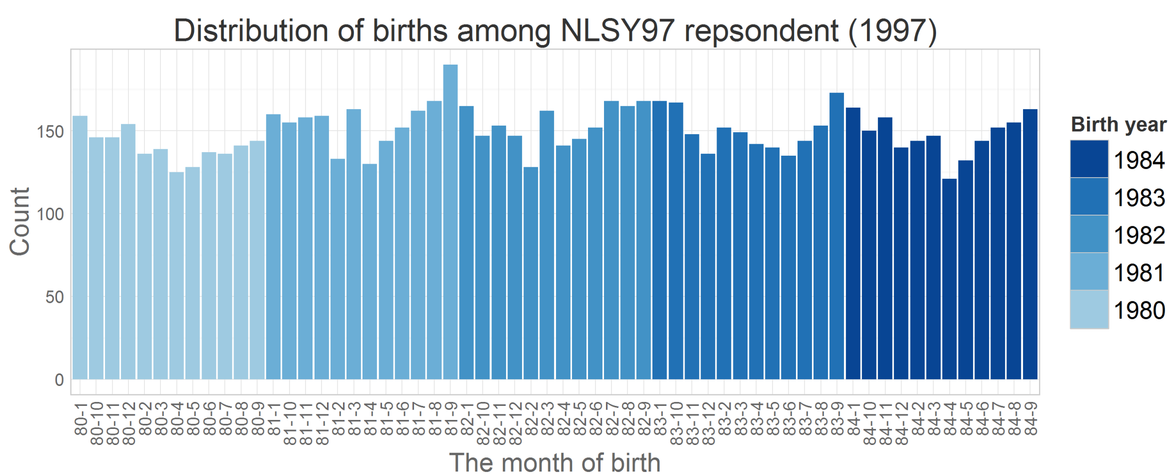
```
Source: local data frame [2 x 2]

          sampleF count
1 Cross-Sectional  6747
2      Oversample  2236
```



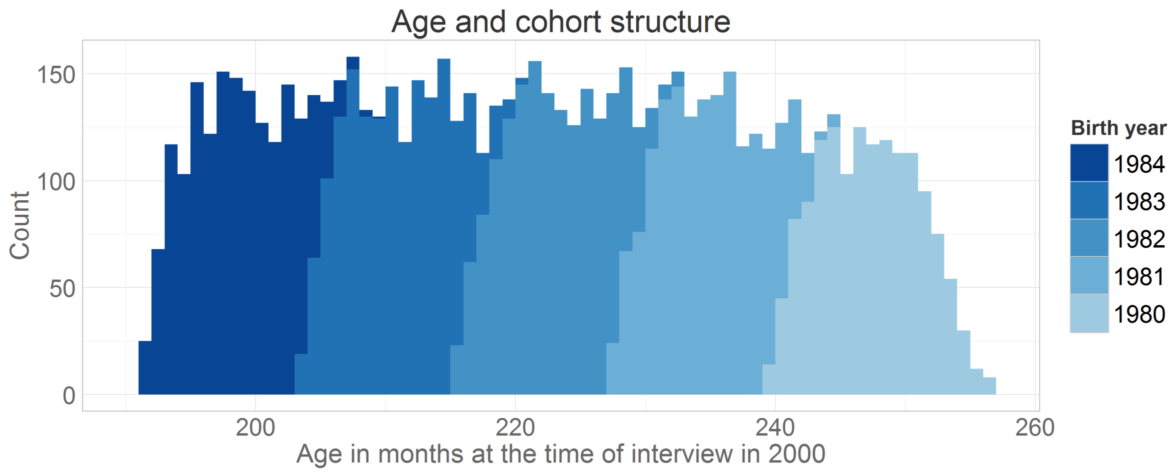
2.2 Distribution of age variables

The age of respondents was of particular interest and was entered as a predictor of the model outcome. NSLY97 contains static and dynamic indicators of age age. Variables byear and bmonth were recorded once in 1997 (static) and contain respondents birth year and birth month respectively. Two age variables were recorded continuously at each interview (dynamic): age at the time of the interview in months agemon and in years ageyear. Next graph shows how births in the NLSY97 sample (static age) was distributed over calendric months from 1980 to 1984:

2.2.1 Months of births



2.2.2 Age and cohort structure

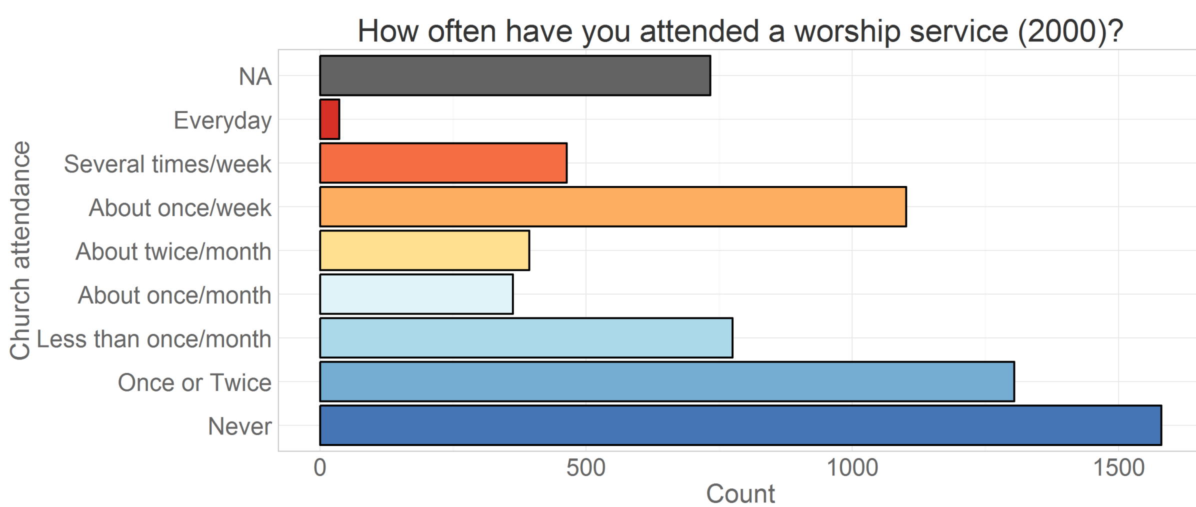


3 Attendance

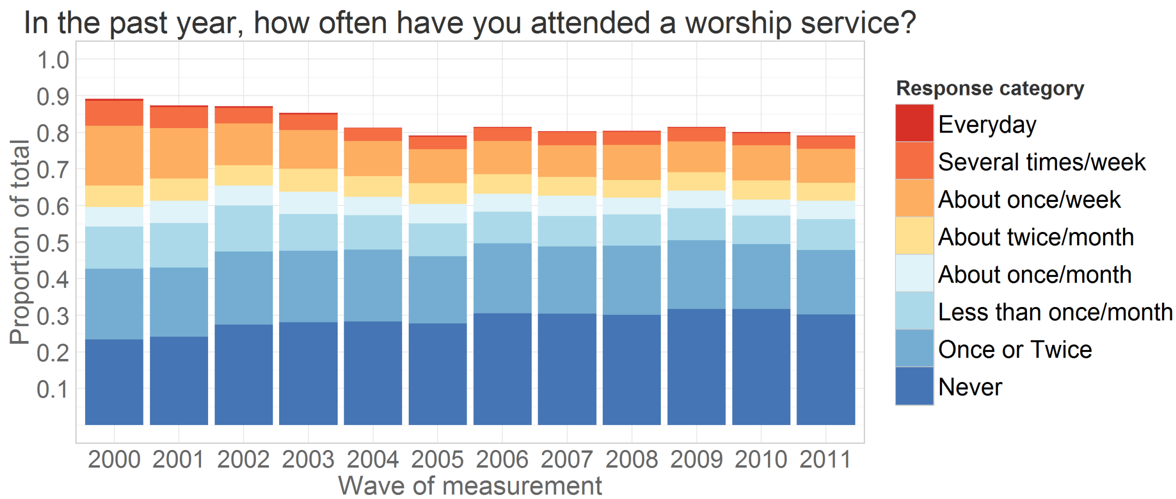
Mapping church attendance in time

3.1 Cross-Sectional View

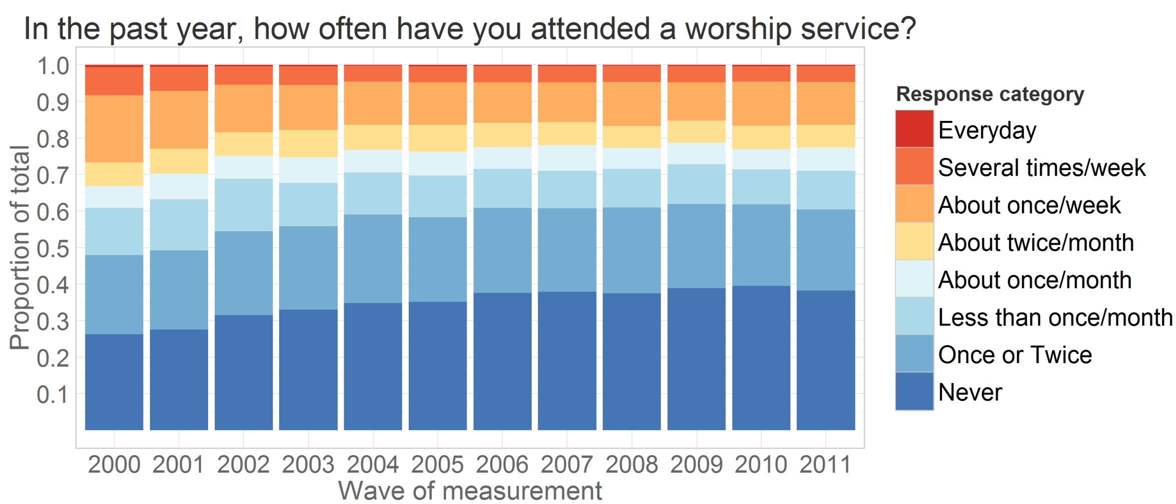
The focal variable of interest is **attend**, the item measuring church attendance for the year that preceded the interview date. The questionnaire recorded the responses on the ordinal scale.



Creating frequency distributions for each of the measurement wave we have:

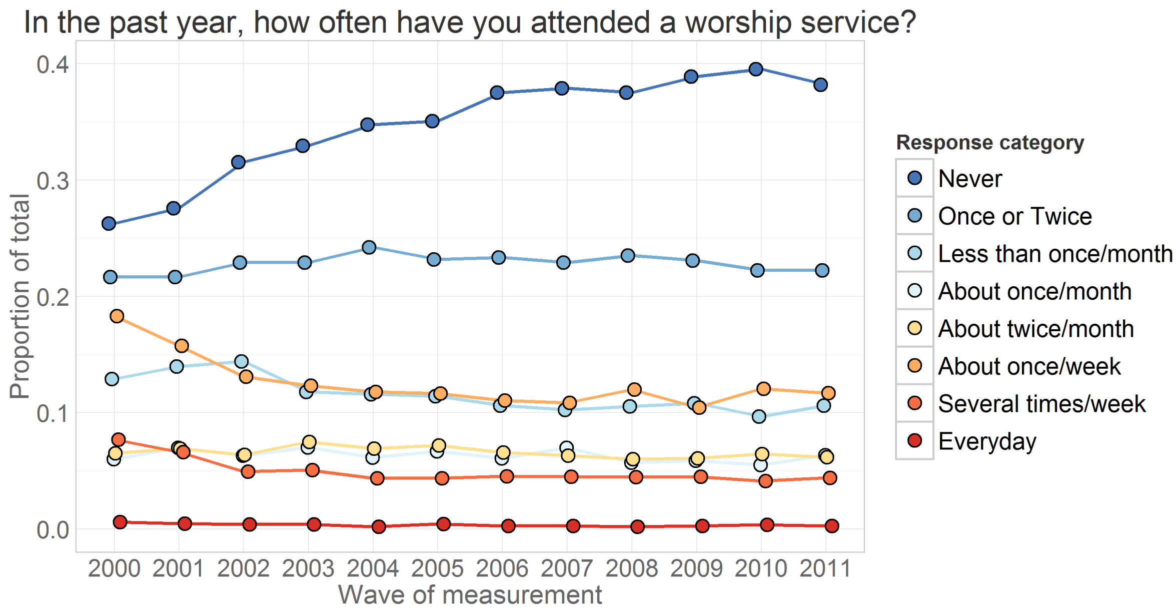


Here, missing values are used in the calculation of total responses to show the natural attrition in the study. Assuming that attrition is not significantly associated with the outcome measure, we can remove missing values from the calculation of the total and look at prevalence of response endorsements over time.



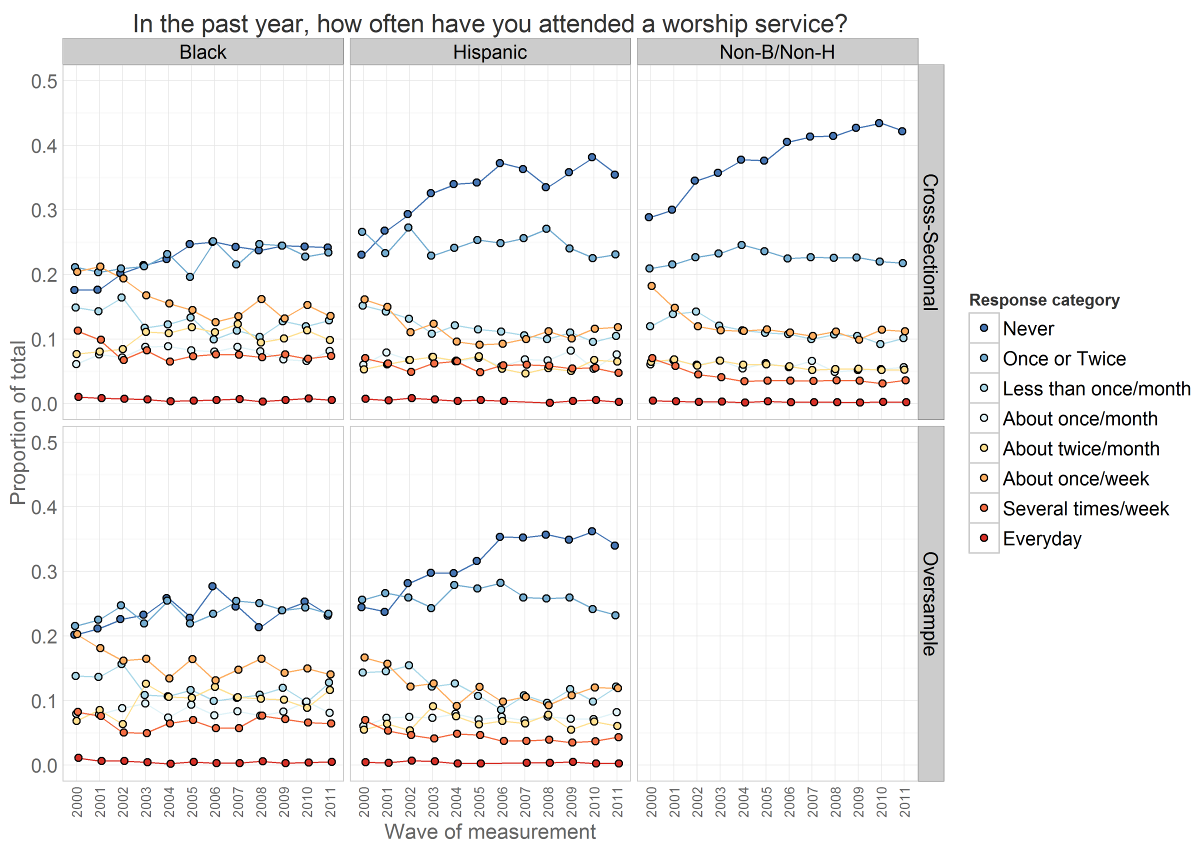
3.1.1 Change in prevalences

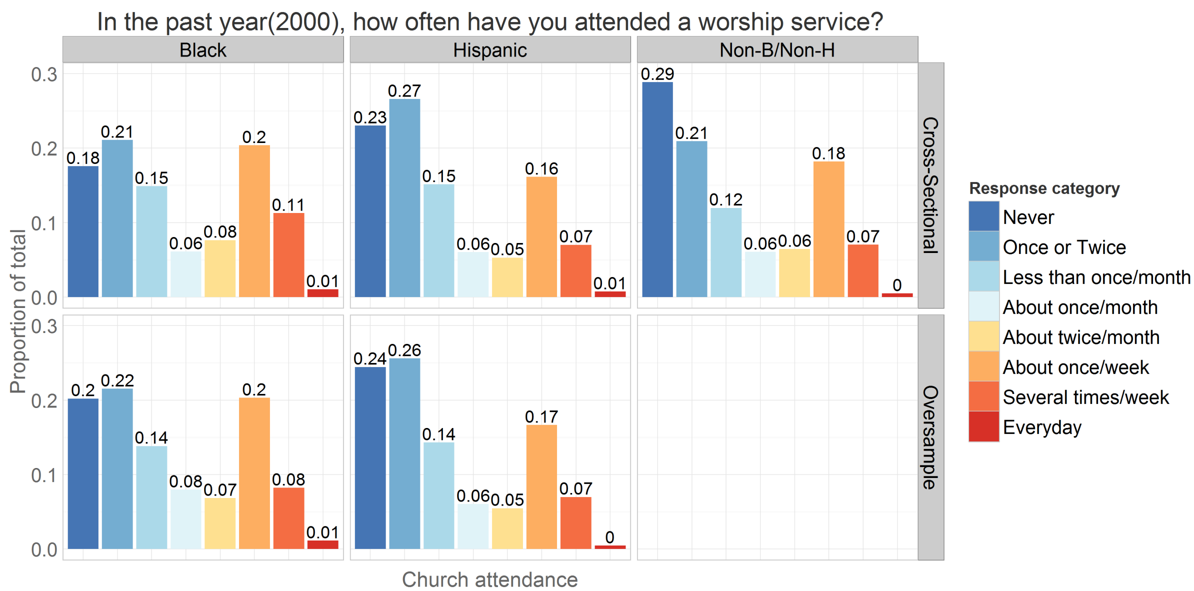
Tracing the rate of change of prevalence in a line graph, we see more clearly which categores increase over time (e.g. “Never”), which decline (e.g. “”About once/week), and which stay relatively stable (e.g. “About twice/month”).



3.1.2 Prevalence change and race

Inspecting the prevalence trajectories across races.





3.2 Longitudinal View

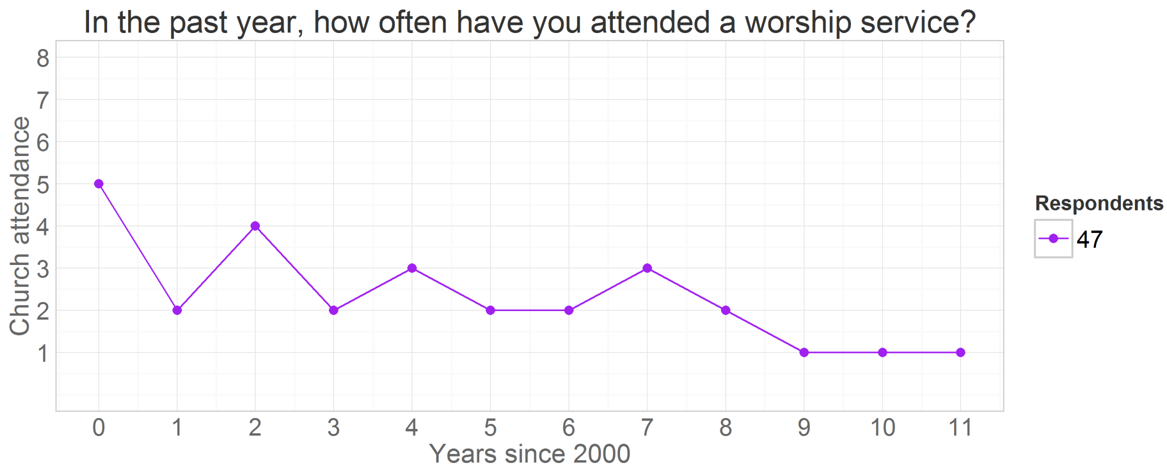
Graphs above shows change in the cross-sectional distribution of responses over the years. Modeling the change in these response frequencies is handled well by Markov models. LCM, however, works with longitudinal data, modeling the trajectory of each individual and treating attendance as a continuous outcome.

To demonstrate mapping of individual trajectories to time, let's select a dataset that would include personal identifyer (**id**), cohort indicator (**byear**), wave of measurement (**year**) and the focal variable of interest - worship attendance (**attend**).

```
ds<- dsL %>%  dplyr::filter(year %in% c(2000:2011), id==47) %>%
              dplyr:: select(id, byear, year, attend, attendF)
print(ds)
```

```
   id byear year attend              attendF
1  47  1982 2000      5    About twice/month
2  47  1982 2001      2        Once or Twice
3  47  1982 2002      4     About once/month
4  47  1982 2003      2        Once or Twice
5  47  1982 2004      3 Less than once/month
6  47  1982 2005      2        Once or Twice
7  47  1982 2006      2        Once or Twice
8  47  1982 2007      3 Less than once/month
9  47  1982 2008      2        Once or Twice
10 47  1982 2009      1                Never
11 47  1982 2010      1                Never
12 47  1982 2011      1                Never
```

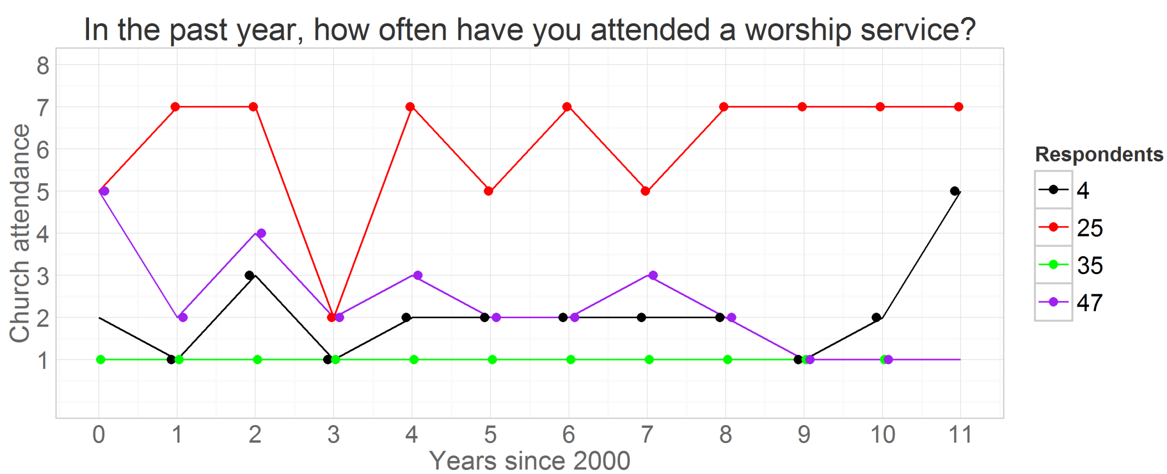
The view above lists attendance data for subjust with id = 47. Mapping his attendance to time we have .



where vertical dimension maps the outcome value and the horizontal maps the time. There will be a trajecory for each of the respondents. Each of such trajectories imply a story, a life scenario. Why one person grows in his religious involvement, while other declines, or never develops an interest in the first place? To demostrate how interpretations of trajectories can vary among individuals consider the following example.

3.2.1 Attendance over waves

Attendance trajectories of subjects with **id**s 4, 25, 35, and 47 are plotted in the next graph



The respondent **id** = 35 reported attending no worship services in any of the years, while respodent **id** = 25 seemed to frequent it, indicating weekly attendance in 8 out of the 12 years. Individual **id** = 47 started as a fairly regular attendee of religious services in 2000 (5 = "about twice a month"), then gradually declined his involvement to nill in 2009 and on. Respondent **id** = 4, on the other hand started off with a rather passive involvement, reporting attended church only "Once or twice" in 2000, maintained a low level of participation throughout the years, only to surge his attendance in 2011. Latent curve models will describe intraindividual trajectories of change, while summarizinig the interindividual similarities and trends.

3.2.2 Changing the metric of time

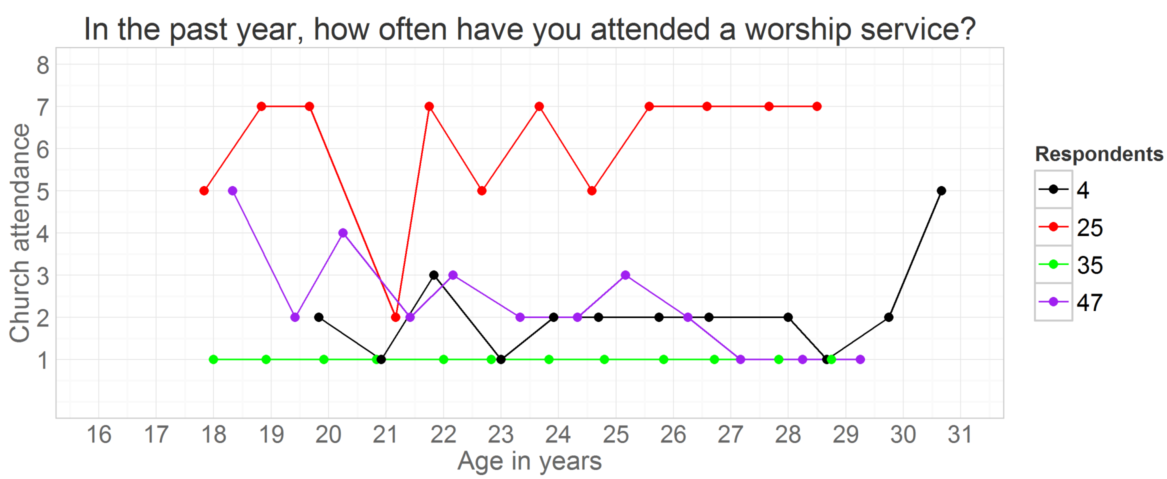
Previous research in religiousity indicated that age might be one of the primary factors explaining interindividual differences in church attendance. To examine the role of age, we change the metric of time from waves of measurement, as in the previous graph, to biological age. Consult Metrics report for details on measurement of age.

```
ds<- dsL %>% dplyr::filter(id %in% c(4,25,35,47),year %in% c(2000:2011)) %>%
  dplyr::select(idF,byear,bmonth,year,ageyear,agemon) %.%
  dplyr::mutate(time=year-2000, age=agemon/12)
print(ds[ds$idF==25,])
```

```
   idF byear bmonth year ageyear agemon time   age
13  25  1983      3 2000      17    214    0 17.83
14  25  1983      3 2001      18    226    1 18.83
15  25  1983      3 2002      19    236    2 19.67
16  25  1983      3 2003      21    254    3 21.17
17  25  1983      3 2004      21    261    4 21.75
18  25  1983      3 2005      22    272    5 22.67
19  25  1983      3 2006      23    284    6 23.67
20  25  1983      3 2007      24    295    7 24.58
21  25  1983      3 2008      25    307    8 25.58
22  25  1983      3 2009      26    319    9 26.58
23  25  1983      3 2010      27    332   10 27.67
24  25  1983      3 2011      28    342   11 28.50
```

3.3 Attendance over ages

Plotting individual trajectories, with age as the metric of time.



4 Data Manipulation Guide

Demonstrating the language of data manipulation in dplyr packages using **dsL** as an example

4.1 Five basic functions in data handling

For a more detailed discussion of basic verbs and operations consult the [R-Studio guide][1] or internal [vignette][2]

```
vignette("introduction",package="dplyr")
```

The following is a brief demonstration of dplyr syntax using **dsL** dataset as an example. I attach prefix dplyr:: to avoid possible conflicts with plyr package on which ggplot2 package relies. I recommend such practice in all dplyr expressions in sharable publications.

One of the innovations in dplyr is the ability to chain phrases in the data manipulationsentence. The operator %>% (or %.%), accomplishes this, turning x %>% f(y) into f(x, y) .

4.1.1 select()

selects variables into a smaller data set

```
ds<-dsL
dim(ds)
```

```
[1] 134745     60
```

```
ds<- dsL %>%
  dplyr::select(id,year, byear, attend, attendF)
head(ds,13)
```

```
   id year byear attend          attendF
1   1 1997  1981     NA             <NA>
2   1 1998  1981     NA             <NA>
3   1 1999  1981     NA             <NA>
4   1 2000  1981      1            Never
5   1 2001  1981      6 About once/week
6   1 2002  1981      2   Once or Twice
7   1 2003  1981      1            Never
8   1 2004  1981      1            Never
9   1 2005  1981      1            Never
10  1 2006  1981      1            Never
11  1 2007  1981      1            Never
12  1 2008  1981      1            Never
13  1 2009  1981      1            Never
```

```
dim(ds)
```

```
[1] 134745      5
```

4.1.2 filter()

Removes observations that do not meet criteria. The following code selects observation based on the type of sample

```
  sample         sampleF
1      1 Cross-Sectional
2      0      Oversample
```

and only between years 2000 and 2011, as only during those years the outcome of interest attend was recorded.

```
ds<- dsL %>%
  dplyr::filter(sample==1, year %in% c(2000:2011))%>%
  dplyr::select(id, year, attend, attendF)
head(ds,13)
```

```
   id year attend          attendF
1   1 2000      1            Never
2   1 2001      6 About once/week
3   1 2002      2    Once or Twice
4   1 2003      1            Never
5   1 2004      1            Never
6   1 2005      1            Never
7   1 2006      1            Never
8   1 2007      1            Never
9   1 2008      1            Never
10  1 2009      1            Never
11  1 2010      1            Never
12  1 2011      1            Never
13  2 2000      2    Once or Twice
```

4.1.3 arrange()

Sorts observations

```
ds<- dsL %>%
  dplyr::filter(sample==1, year %in% c(2000:2011)) %>%
  dplyr::select(id, year, attend) %>%
  dplyr::arrange(year, desc(id))
head(ds,13)
```

```
     id year attend
1  9022 2000      1
2  9021 2000      2
3  9020 2000      2
4  9018 2000      4
5  9017 2000      6
6  9012 2000      5
7  9011 2000      6
8  9010 2000      1
9  9009 2000      2
10 9008 2000      6
11 8992 2000     NA
12 8991 2000      3
13 8987 2000      6
```

```
ds<- dplyr::arrange(ds, id, year)
head(ds, 13)
```

```
   id year attend
1   1 2000      1
2   1 2001      6
3   1 2002      2
4   1 2003      1
5   1 2004      1
6   1 2005      1
7   1 2006      1
8   1 2007      1
```

```
9    1  2008       1
10   1  2009       1
11   1  2010       1
12   1  2011       1
13   2  2000       2
```

4.1.4 mutate()

Creates additional variables from the values of existing.

```
ds<- dsL %>%
  dplyr::filter(sample==1, year %in% c(2000:2011)) %>%
  dplyr::select(id, byear, year, attend) %>%
  dplyr::mutate(age = year-byear,
                timec = year-2000,
                linear= timec,
                quadratic= linear^2,
                cubic= linear^3)
head(ds,13)
```

```
   id byear year attend age timec linear quadratic cubic
1   1  1981 2000      1  19     0      0         0     0
2   1  1981 2001      6  20     1      1         1     1
3   1  1981 2002      2  21     2      2         4     8
4   1  1981 2003      1  22     3      3         9    27
5   1  1981 2004      1  23     4      4        16    64
6   1  1981 2005      1  24     5      5        25   125
7   1  1981 2006      1  25     6      6        36   216
8   1  1981 2007      1  26     7      7        49   343
9   1  1981 2008      1  27     8      8        64   512
10  1  1981 2009      1  28     9      9        81   729
11  1  1981 2010      1  29    10     10       100  1000
12  1  1981 2011      1  30    11     11       121  1331
13  2  1982 2000      2  18     0      0         0     0
```

4.1.5 summarize()

collapses data into a single value computed according to the aggregate functions.

```
require(dplyr)
ds<- dsL %>%
  dplyr::filter(sample==1) %>%
  dplyr::summarize(N= n_distinct(id))
ds
```

```
     N
1 6747
```

Other functions one could use with summarize() include:

From base

- min()
- max()
- mean()
- sum()
- sd()
- median()
- IQR()

Native to dplyr

- n() - number of observations in the current group
- n_distinct(x) - count the number of unique values in x.
- first(x) - similar to x[1] + control over NA
- last(x) - similar to x[length(x)] + control over NA
- nth(x, n) - similar to x[n] + control over NA

4.2 Grouping and Combining

The function group_by() is used to identify groups in split-apply-combine (SAC) procedure: it splits the initial data into smaller datasets (according to all possible interactions between the levels of supplied variables). It is these smaller datasets that summarize() will individually collapse into a single computed value according to its formula.

```
ds<- dsL %>%
  dplyr::filter(sample==1, year %in% c(2000:2011)) %>%
  dplyr::select(id, year, attendF) %>%
  dplyr::group_by(year,attendF) %>%
  dplyr::summarise(count = n()) %>%
  dplyr::mutate(total = sum(count),
              percent= count/total)
head(ds,10)
```

```
Source: local data frame [10 x 5]
Groups: year

   year              attendF count total  percent
1  2000                Never  1580  6747 0.234178
2  2000        Once or Twice  1304  6747 0.193271
3  2000 Less than once/month   775  6747 0.114866
4  2000      About once/month   362  6747 0.053653
5  2000     About twice/month   393  6747 0.058248
6  2000       About once/week  1101  6747 0.163184
```

```
7  2000   Several times/week   463   6747 0.068623
8  2000             Everyday    36   6747 0.005336
9  2000                   NA   733   6747 0.108641
10 2001                Never  1626   6747 0.240996
```

To verify :

```
dplyr::summarize( filter(ds, year==2000), should.be.one=sum(percent))
```

```
Source: local data frame [1 x 2]

  year should.be.one
1 2000             1
```

4.3 Base subsetting

Generally, we can compose any desired dataset by using matrix calls. The general formula is of the form: **ds**[*rowCond* , *colCond*], where **ds** is a dataframe, and *rowCond* and *colCond* are conditions for including rows and columns of the new dataset, respectively. One can also call a variable by attaching $ followed variable name to the name of the dataset: **ds***$variableName*.

```
ds<-dsL[dsL$year %in% c(2000:2011),c('id',"byear","year","agemon","attendF","ageyearF")]
print(ds[ds$id==1,])
```

```
   id byear year agemon          attendF ageyearF
4   1  1981 2000    231            Never       19
5   1  1981 2001    243 About once/week       20
6   1  1981 2002    256   Once or Twice       21
7   1  1981 2003    266            Never       22
8   1  1981 2004    279            Never       23
9   1  1981 2005    290            Never       24
10  1  1981 2006    302            Never       25
11  1  1981 2007    313            Never       26
12  1  1981 2008    325            Never       27
13  1  1981 2009    337            Never       28
14  1  1981 2010    350            Never       29
15  1  1981 2011    360            Never       29
```

The following is a list of operators that can be used in these calls.

basic math operators: +, -, *, /, %%, ^

math functions: abs, acos, acosh, asin, asinh, atan, atan2, atanh, ceiling, cos, cosh, cot, coth, exp, floor, log, log10, round, sign, sin, sinh, sqrt, tan, tanh

logical comparisons: <, <=, !=, >=, >, ==, %in%

boolean operations: &, &&, |, ||, !, xor

basic aggregations: mean, sum, min, max, sd, var

dplyr can translate all of these into SQL. For more of on dplyr and SQL compatibility consult another built-in [vignette][3]

```
vignette("database",package="dplyr")
```

4.4 Base Reference

The following unary and binary operators are defined for base. They are listed in precedence groups, from highest to lowest.

- :: ::: - access variables in a namespace
- $ @ - component / slot extraction
- [[[- indexing
- ˆ - exponentiation (right to left)
- - + - unary minus and plus
- : - sequence operator
- %any% - special operators (including %% and %/%)
- * / - multiply, divide
- + - - (binary) add, subtract
- < > <= >= == != - ordering and comparison
- ! - negation
- & && - and
- | || - or
- ~ - as in formulae
- -> ->> - rightwards assignment
- <- <<- - assignment (right to left)
- = - assignment (right to left)
- ? - help (unary and binary)