Identifying Risk Factors for Cognitive Decline using Statistical Learning Techniques and Functional Data Analysis

A thesis submitted to the
College of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Mathematics and Statistics
University of Saskatchewan
Saskatoon

By
Hao Hu

Permission to Use

Disclaimer

Abstract

Background: Numerous studies have shown that older adults' cognitive abilities are age-related and likely to decline at a certain age. Based on these indications, this work uses functional principal component analysis (FPCA) to explore changes in cognitive function with age. This study aims to describe the longitudinal cognitive function of elderly trajectory patterns between 65 and 80 years of baseline age using FPCA and identify risk factors for cognitive decline using machine learning algorithms.

Methods: We used FPCA to extract the overall pattern change and use elastic-net, decision tree, and random forest models to find risk factors. In a sample of elderly (n = 944) with 6608 measurements (7 waves) from the Survey of Health, Aging, and Retirement in Europe (SHARE), by using age at interview as time, longitudinal cognitive function trajectory patterns for the elderly were extracted using FPCA. Zou and Hastie (2005) proposed the elastic net regression method, which effectively implements feature selection by setting coefficients of non-significant variables to zero [56]. Random forest is a tree-based machine learning algorithm that harnesses the power of multiple decision trees to make decisions. Random forests combine the outputs of individual decision trees to generate the final result. [15]. We modelled the first two functional principal components (FPCs) with these machine learning algorithms and used the selected covariates in the baseline wave to identify risk factors.

Results and Conclusions: We have obtained four FPCs explained by 78.0, 14.2, 6.7 and 1.1 % of the variation respectively for the cognitive function. The mean function of FPCA shows that cognitive decline is generally divided into two stages (early decline and late decline). By analyzing and comparing a set of models at the national level, the cognitive function of each country is slightly different. Older people in Italy and Spain have significantly lower cognitive abilities. The Predictive R^2 of FPC1 and FPC2 is around 0.5 and 0.2 with covariate delayed recall score (DRS) and reduced to 0.4 and 0.1 without covariate DRS. From the individual point of view, many risk factors are modifiable and can be prevented in advance. Our results show that immediate recall score, education level, country, numeracy score, reading score, and household income are associated with cognitive patterns in the elderly.

Acknowledgements

First of all, I would like to express to my supervisor Dr. Li Xing and Longhai Li, that without their continuous guidance and encouragement, I could not complete my thesis. It is my great honour to work with them. I'm also grateful to Dr. Steven Rayan, the Graduate Chair in the Department of Mathematics & Statistics and Prof. Junxin Liu, my committee member, for academic support during my master's program. Besides, this thesis is supported by the University of Saskatchewan (USask) through a USask Internal Research Grant and the USask College of Arts and Science. This thesis was partially enabled by support provided by Compute Canada (www.computecanada.ca). The data used in this paper is from SHARE Waves 1 [6], 2 [7], 4 [8], 5 [9], 6 [10], 7 [11], and 8 [12].

Last, I am thankful to my parents, who have loved me and supported me all the time.

Contents

List of Tables

List of Figures

List of Abbreviations

SHARE	Survey of Health, Aging, and Retirement in Europe
PCA	Principal component analysis
FPCA	Functional principal component analysis
BMI	Body mass index
IQR	Interquartile Range=Upper Qualitle-Lower Qualitle
USask	University of Saskatchewan
AD	Alzheimer's disease
PD	Parkinson's Disease
FPCs	Functional principal components
LHC	Life History Calendar
DRS	Delayed recall score
IRS	Immediate recall Score
FCS	Conditional specification method
E	Expectation
Var	Variance
MLTC	median longitudinal trajectory curves
CP	Complexity Parameter
FDA	Functional Data Analysis
EFs	Orthonormal eigenfunctions
i.i.d.	Independent and identically distributed
%IncMSE	Mean Decrease Accuracy
RSS	Residual Sum of Square
PRESS	Predicted Residual Sum of Squares
EHCI	Euro Health Consumer Index
PRIS	Patient rights and information score
WTTS	Waiting times for treatment score
OS	Outcomes score
RRSS	Range and reach of services score
PS	Pharmaceuticals score
RMSE	Root Mean Square Error
MAE	Mean Absolute Error

1 Introduction

Cognitive functions refer to various mental abilities, which are more related to how people learn, think, remember, pay attention, make decisions and solve problems [31]. Cognitive functions have been distinguished by how they change throughout adulthood, namely cognitive mechanics and cognitive pragmatics. Cognitive mechanics are constructed by information processing rate, working memory and inhibition(ability to automatically inhibit goal-irrelevant information). The information processing rate is most sensitive to age differences. In this work, we focus on cognitive mechanics, which is typically decline with age and accelerate in old age. In contrast to the mechanics, the pragmatics of cognition are verbal and numerical abilities that rely on the accumulation of lifespan development and knowledge-based forms of intelligence, which are well maintained during adulthood and will not decline obviously in advanced age [2].

Cognitive abilities are critical to living independently as people age, such as whether a person can live independently and drive safely. Furthermore, cognition is essential for effective human communication, including processing information given by others and responding appropriately to others. Cognitive impairment and memory loss, which tend to decline with age [37], are characteristics of diseases such as dementia. Dementia is a clinical syndrome characterized by a loss of cognitive functioning (thinking, remembering, and reasoning) that interferes with a person's daily life and activities [25]. In 2015, it was estimated that nearly 47 million people worldwide were affected by dementia, and the number continues to rise, which is expected to reach 75 million by 2030 and 131 million by 2050 [51]. Alzheimer's disease (AD) is the most common form of dementia, accounting for around 50% -70% of cases. In other words, cognitive decline is the most common cause of AD in older adults. A group analyzed the prevalence and incidence of AD in European countries by using meta-analyses. They found the prevalence of AD in Europe was estimated at 5.05% and the incidence was 11.08 per 1000 persons per year [38]. For AD or other dementia, there is no cure or treatment that substantially relieves symptoms. AD can profoundly impact the lives of patients and their families over many years, but it doesn't stop there. AD can also have a significant economic impact on individuals and their families. According to statistics, the cumulative cost of an AD patient starting from primary care to specialist care in Sweden has exceeded €$5,000$ [51]. However, this cost is only a small fraction of an AD patient's lifetime medical and nursing costs. The total social cost of dementia in Europe in 2010 was estimated to be between US $238.6 billion and €105.6 billion [53]. And by 2050, the overall mean cost burden for AD and Parkinson's Disease(PD) is estimated to be €357 billion [34]. The global economic cost of dementia was estimated at more than US $600 billion in 2010 [50], in excess of US $1 trillion in 2020 and forecast to double by 2030 [55].

Declining cognitive functions has become one of the primary concerns for European countries. Explosive growth in care costs and associated societal burdens on AD, PD, and other dementia challenges European countries' health care and long-term care systems. Since the primary risk factor for cognitive decline is age, as life expectancy increases worldwide, the prevalence of the disease is increasing dramatically. However, Cognitive impairment has multifactorial etiology, and partial risk factors are modifiable [51]. In 2011, Barnes D E and Yaffe K identified the evidence for seven potentially modifiable risk factors associated with AD: diabetes, midlife hypertension, midlife obesity, smoking, depression, cognitive inactivity or low educational attainment, and physical inactivity. They also predicted that if all seven risk factors were reduced by 10 %-25 %, as many as 1.1 to 3 million cases of AD could be prevented globally, and $184,000$ to $492,000$ cases in the United States [3]. In 2014, [39] suggested that around one-third of AD cases might be attributed to potentially modifiable risk factors.

This thesis examines the association between age and cognitive decline among people, especially the elderly, living in different European countries. We model the cognitive function trajectory as a function of age to see the overall pattern and if there is a pattern change. In addition, we intend to identify all aspects of risk factors from the overall design and the pattern change. To this end, we used the data from the Survey of Health, Aging, and Retirement in Europe. SHARE is a research infrastructure providing internationally comparable longitudinal data on various health, economic, and social factors [13].

Longitudinal studies employ continuous or repeated methods for tracking the same individual to detect any changes that might occur over a long period, usually, years or decades [19]. Longitudinal data are collected naturally on any combination of exposures and outcomes, without any external influence [18]. The type of study is most commonly used in medicine, economics, medical sciences and epidemiology. It's useful for assessing risk factors, disease progression, and treatment outcomes over different lengths of time. Mixed-effects regression (MER) is one of the preferred methods for longitudinal data, which allows time-invariant(s.g., gender) and time-varying predictors (s.g., age) [36]. Furthermore, MER explicitly models individual changes over time and is flexible in repeated measures without requiring the same number of observations for each subject. Nevertheless, MER is not appropriate for our study, we instead propose an estimation procedure based on principal component bases and extend FPCA to the longitudinal setting. In MER models, we assume that our outcome variable is a linear function of age, and then there will only be a positive or negative linear relationship between them. We believe that age-based cognitive function trajectories should be more complex with a nonlinear quadratic functional format.

Functional data analysis (FDA) is a statistical field that studies models and analysis methods for data recorded over continuous time for each subject. Likewise, it can be described as the study of a sample of trajectories or time courses. In FDA, where we consider repeated observations over time courses and do not rely on any stationarity assumptions, FDA is particularly well suited for analyzing temporal dynamics and longitudinal data that are abundantly found in applications such as biomedicine. In general, any time-related data that is repeatedly observed across independent individuals or units can be analyzed using FDA's

methods. A basic paradigm of the FDA is that the observed data is regarded as an independent and identically distributed random sample from a stochastic process. However, it is also possible to incorporate dependencies between the realizations of stochastic processes. The underlying stochastic process that is assumed to lie in L_2 space is generally assumed to be smooth over a continuum, which is the target interest of FDA [23] [28]. Functional principle component analysis is one of the most popular multivariate analysis techniques for the extraction of information from the FDA.

FPCA is a powerful tool used to model longitudinal data observed at different time points. FPCA is based on the principle of component analysis(PCA); however, rather than using variables, the FPCA uses functions. It is a dimension reduction method to decompose the latent stochastic process into a linear combination of FPCs [54]. And it represents functional data in the most parsimonious way, which maximizes the variation in the randomly observed curves. The top few FPCs explain most of the variability in the underlying stochastic process [44]. FPCA can work with sparse data and does not require that the observations are taken at the same time points. In this regard, FPCA offers significant advantages for a better understanding of trends. FPCA has successfully been applied to real life, such as weather and climate prediction, fetal movement monitoring data [52], genetic growth [27] and child growth study [30].

In this thesis, because the number of repeated measurements for cognitive function per elderly is irregular and small, FPCA is performed by a nonparametric method to analyze and characterize cognitive trajectory data. Unlike classic FPCA, which requires many regularly spaced measurements to express the random curves per subject, this method works well with a small number of repeatable measurements per subject. We assume the repeatable measurements are randomly located with a random number of repetitions for each subject and are determined by an underlying smooth random trajectory plus measurement errors. Even if only one or few measurements are available for a subject, it can estimate individual smooth trajectories satisfactorily. This approach assumes the repeated measures are determined by an underlying smooth random (subject-specific) trajectory plus measurement errors [45] and principal component scores (FPCs) [54]. Simultaneously, like classic FPCA, it is suitable for extracting the pattern of the entire cognitive abilities as a function that will otherwise be lost with applying some traditional statistical techniques. By treating the whole curve as a single entity, there is no concern about correlations between repeated measurements. We assumed that an underlying functional relationship governs the data. In addition to the computational advantages, we are thus able to extract the main differences between subjects in their average cognitive function and how their cognitive function over time. We also detect the associations between one or more factors and longitudinal cognitive growth data.

The primary aim of this study is to characterize individual cognitive trajectories of the elderly using FPCA and identify the risk factors by using machine learning models (including elastic-net, decision tree and random forest). The results of FPCA show that cognitive function is an up-and-down process, divided into two stages (early and late decline), with an increase in cognitive function at about age 70, and then an accelerated decline after age 75. Based on the results of these machine learning models, we found many

preventable risk factors, such as alcohol drinking behaviour, physical activity, household income level, and education level.

The remainder of the thesis is organized as follows. Section 1 is the introduction. In Section 2, we present the data, including the source of data, prime outcome, covariates and study sample. The FPCA, Cross-Validation, Elastic-net, Decision tree, and Random Forest methods are described in Section 3. The result of FPCA and risk factors are presented in Section 4. Section 5 is the conclusion.

2 The data

2.1 Source of data

Our sample data is from SHARE. The baseline study of SHARE took place in 2004 and continued today. With the exception of the SHARE, Corona Survey collected via telephone in 2020, all other SHARE data collection consisted of face-to-face computer-assisted interviews. $140,000$ people aged 50 or older from various regions of Europe and Israel participated and conducted $530,000$ in-depth interviews. The baseline wave involved 12 countries(Austria, Belgium, Denmark, France, Germany, Greece, Israel, Italy, Netherlands, Spain, Sweden, and Switzerland). Switzerland and Belgium contribute the least and the most respondents at around 3.33% and 12.68% of the whole population, respectively. Approximately 8% to 10% of respondents are from other countries. In contrast, Spain and Italy have seriously ageing populations, with the average respondent reaching over 65 years old, while the average age of respondents in Greece is relatively low, around 62 years old. Respondents from other countries are about 63 to 64 years old. Overall, the male-to-female ratio in these participating countries is around 5 to 5. Several other countries have been added in the following waves. Until now, 28 European countries and Israel joined SHARE. The persons aged 50 years and over are the target population of SHARE. New participants Will be enrolled in each wave as refreshment samples to compensate for the dropout of participants. The collection method in wave 3 differs from other waves; it is called SHARELIFE. The questioning method in SHARELIFE is based on the Life History Calendar (LHC). Respondents' lives are graphically represented by a grid that automatically populates during the interview. The LHC helps respondents remember primarily by asking about life events that are likely to be reflected accurately. Usually includes the respondent's child's name, date of birth and partner's history. Since our outcome variable has not been provided in the third wave, the sample data include wave 1 (2011), wave2 (2011), wave4 (2013), wave5 (2015), wave6 (2017), wave7 (2019) and wave8 (2020). Wave 1 was selected as the baseline, and wave 8 was a new wave when collection took place in 2020.

2.2 Primary Outcome

Cognition function was assessed using the "ten words list learning" test in SHARE. The test is conducted with immediate recall and delayed recall. Immediate recall was the number of recalled words after the interviewer read a list of 10 words, so the score of immediate recall ranges from 0 to 10. At the end of the test, the interviewer will ask the participants to recall the words again from the list as the delayed recall

score(DRS). From Table 1, the mean of DRS decreases as the wave increases, from 3.44 in wave 1 to 2.54 in wave 8, showing a downward trend; as we mentioned in the introduction, cognitive function is closely related to age. We will use the normalized delayed recall score as our outcome variable and only keep the subjects without any missing value in DRS. According to our introduction to cognitive function in section 1, DRS belongs to cognitive mechanics, which is sensitive to age differences.

2.3 Covariates

SHARE is an extensive database that covers thousands of variables; we initially identified covariates from multi-aspect, including demographic information, household composition, social support, network, childhood conditions, health, behavioural risk, work, and money. Our descriptive statistics for the final covariates across waves are depicted in Tables 2.1 and 2.2. Note there are 944 people in each wave, no missing values in the wave1, but other waves might have it, and the number in the bracket is the percentage of an index for a variable (Table 2.1). The number in the bracket after the description is the index in the dataset. For example, in the original dataset, Male(1) means Male represent by index 1.

In descriptive Table 2.1, we found that the participants were more female (58%), and the elderly living alone were increasing, from 29% in wave 1 to 54% in wave 8. At the same time, the health of the elderly is declining, the number of hospitalisations is increasing, and some activities cannot be completed, which is in line with the laws of nature. It is worth noting that when self-rated writing, reading and orientation time skills have dropped significantly, the numeracy score of the elderly has not shown a significant drop; it remained unchanged. From the descriptive Table 2.2, we found that the cognitive function of the elderly decreased with age, which was consistent with expectations. In addition, BMI, maximum grip strength, and income levels all decreased over time.

Table 2.1: Descriptive Statistics Across Waves (discrete variable)

Variable	**wave** 1	**wave** 2	**wave** 4	**wave** 5	**wave** 6	**wave** 7	**wave** 8
gender							
Male(1)	401(42)	401(42)	401(42)	401(42)	401(42)	401(42)	401(42)
Female(2)	543(58)	543(58)	543(58))	543(58)	543(58)	543(58)	543(58)
partnerinhh							
Living with spouse/partner(1)	670(71)	642(68)	581(62)	545(58)	506(54)	475(50)	432(46)
Living w.o spouse/partner(3)	274(29)	302(32)	363(38)	399(42)	438(46)	469(50)	512(54)
Marital status(mstat_m)							
married and living together with spouse(1)	645(68)	25(45)	16(27)	24(34)	24(37)	10(17)	7(11)
registered partnership(2)	10(1)	0(0)	0(0)	0(0)	0(0)	0(0)	0 (0)
married, living separated from spouse(3)	7(1)	0(0)	0(0)	0(0)	1(2)	1(2)	1(2)
never married(4)	38(4)	1(2)	1(2)	1(1)	2(3)	0(0)	1(2)
divorced(5)	54(6)	0(0)	3(5)	2(3)	1(2)	1(2)	2(3)
widowed(6)	190(20)	29(53)	40(67)	44(62)	37(57)	46(79)	50(82)
Current job situation(ep005)							
employed(1)	39(4)	18(2)	13(1)	7(1)	6(1)	8(1)	4(0)
unemployed,retired(0)	904(96)	918(98)	914(99)	919(99)	913(99)	902(99)	880(100)
Number of Child(child)							

0(1)	76(11)	79(11)	81(11)	87(12)	90(12)	88(12)	88(11)
1 to 3(2)	511(72)	519(72)	535(72)	530(72)	557(73)	540(73)	568(73)
over 4(3)	126(18)	120(17)	124(17)	117(16)	120(16)	113(15)	121(16)
Number of Received help(ghelp)							
0(0)	758(80)	789(84)	758(80)	741(78)	685(73)	584(65)	557(59))
1 to 3(1)	186(20)	155(16)	186(20)	203(22)	259(27)	318(35)	387(41)
Number of Given help(ghelp)							
0(1)	609(65)	645(68)	553(75)	558(76)	750(79)	735(81)	808(86)
1 to 3(2)	335(35)	299(32)	187(25)	179(24)	194(21)	167(19)	136(14)
Self-perceived health(health)							
Excellent(1)	112(12)	94(10)	66(7)	56(6)	51(5)	39(4)	34(4)
Good(2)	636(67)	567(60)	581(62)	547(58)	526(56)	505(53)	445(47)
Poor(3)	196(21)	283(30)	297(31)	341(36)	367(39)	400(42)	465(49)
Hospital stay(hospital_m)							
No(0)	849(90)	814(86)	791(84)	780(84)	775(82)	757(80)	724(77)
Yes(1)	95(10)	130(14)	151(16)	164(17)	169(18)	187(20)	220(23)
Drinking behavior(drink)							
not at all(1)	273(31)	249(29)	265(31)	281(33)	NaN	NaN	NaN
less than twice a month(2)	20(2)	72(8)	73(9)	78(9)	NaN	NaN	NaN
less than four days a week(3)	293(34)	246(29)	228(27)	225(27)	NaN	NaN	NaN
almost every day(4)	281(32)	286(34)	277(33)	262(31)	NaN	NaN	NaN
ever smoking(esmoked_m)							
No(0)	589(62)	592(63)	592(63)	593(63)	593(63)	567(63)	607(64)
Yes(1)	355(38)	351(37)	351(37)	351(37)	351(37)	335(37)	337(36)
N of chronic diseases(chronic)							
0(1)	198(21)	191(20)	159(17)	145(15)	134(14)	117(12)	100(11)
1(2)	316(33)	299(32)	289(31)	256(27)	235(25)	231(24)	226(24)
over 1(2)	430(46)	454(48)	496(53)	543(58)	575(61)	596(63)	618(65)
N of doctor visit in 12 month(doctorS)							
0 to 3(1)	430(46)	393(42)	343(36)	331(35)	328(35)	327(35)	290(31)
4 to 12(2)	417(44)	426(45)	474(50)	468(50)	497(53)	485(51)	492(52)
13 to 98(3)	97(10)	125(13)	127(13)	145(15)	119(13)	132(14)	162(17)
depression scale(depression)							
0 to 3(1)	691(76)	730(79)	671(74)	633(70)	644(71)	590(68)	586(64)
4 to 12(2)	220(24)	190(21)	237(26)	275(30)	265(29)	273(32)	330(36)
Mobility Index(mob)							
0(1)	473(50)	496(53)	420(45)	378(40)	375(40)	326(35)	266(28)
1 to 4(2)	471(50)	448(47)	522(55)	566(60)	569(60)	618(65)	678(72)
Eyesight reading(reading)							
poor(0)	180(19)	161(17)	174(18)	214(23)	182(19)	161(18)	208(22)
good(1)	764(81)	783(83)	770(82)	730(77)	762(81)	741(82)	736(78)
Self-rated writing skills(writing)							
poor(0)	169(18)	167(18)	167(18)	166(18)	166(18)	219(24)	222(24)
good(1)	775(82)	777(82)	777(82)	778(82)	778(82)	683(76)	722(76)
Self-rated reading skills(reading)							
poor(0)	143(15)	144(15)	144(15)	143(15)	143(15)	153(17)	155(16)
good(1)	801(85)	800(85)	800(85)	801(85)	801(85)	749(83)	789(84)
hearing							
poor (0)	171(18)	188(20)	215(23)	261(28)	260(28)	271(30)	306(32)
good(1)	773(82)	756(80)	729(77)	683(72)	684(72)	631(70)	638(68)
Large Muscle Index(muscle)							
0(1)	567(60)	597(63)	522(55)	487(52)	490(52)	432(46)	64(36)
1 to 2(2)	229(24)	203(22)	225(24)	245(26)	219(23)	246(26)	34(19)
3 to 4(3)	148(16)	144(15)	197(21)	212(22)	235(25)	266(28)	80(45)

Vigorous activities(vigorous)							
never(1)	348(37)	395(42)	463(49)	508(54)	529(56)	535(59)	124(70)
more than once(2)	596(63)	548(58)	480(51)	436(46)	415(44)	367(41)	54(30)
active of Daily Living Index(active)							
0(1)	899(95)	886(94)	857(91)	846(90)	820(87)	809(86)	714(76)
1 to 5(2)	45(5)	58(6)	85(9)	98(10)	124(13)	135(14)	230(24)
Instrumental Activities Index(active2)							
0(1)	844(89)	848(90)	794(84)	762(81)	700(74)	670(71)	536(57))
1 to 5(2)	100(11)	96(10)	148(16)	182(19)	244(26)	274(29)	408(43)
Fine Motor Skills Index(finemotor)							
1	891(94)	882(93)	852(90)	837(89)	829(88)	820(87)	135(76)
2	50(5)	55(6)	85(9)	96(10)	94(10)	102(11)	30(17)
3	3(0)	7(1)	7(1)	10(1)	18(2)	18(2)	11(6)
4	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
Gross Motor Skills Index(grossmotor)							
1	851(90)	843(89)	790(84)	766(81)	747(79)	674(71)	96(54)
2	74(8)	70(7)	107(11)	107(11)	100(11)	158(17)	39(22)
3	15(2)	23(2)	37(4)	52(6)	66(7)	70(7)	22(12)
4	2(0)	7(1)	8(1)	10(1)	22(2)	31(3)	13(7)
5	2(0)	1(0)	2(0)	9(1)	8(1)	10(1)	8(4)
Score of orientation in time test(orient)							
1	0(0)	0(0)	0(0)	0(0)	4(0)	9(1)	30(3)
2	0(0)	4(0)	5(0)	5(0)	6(1)	13(1)	19 (2)
3	16(2)	6(1)	8(0)	10(1)	15(2)	31(3)	46(5)
4	108(11)	98(10)	100(11)	102(11)	111(12)	125(14)	163(17)
5	820(87)	836(89)	831(88)	827(88)	808(86)	724(80)	686(73)
Numeracy Score(percentage)(numeracy_m)							
1	44(5)	45(5)	46(5)	46(5)	46(5)	43(5)	46(5)
2	154(16)	156(17)	156(17)	155(16)	155(16)	151(17)	155(16)
3	312(33)	274(29)	273(29)	273(29)	272(29)	259(29)	272(29)
4	290(31)	303(32)	302(32)	303(32)	304(32)	286(32)	304(32)
5	144(15)	166(18)	167(18)	167(18)	167(18)	163(18)	167(18)
mother alive							
Yes(1)	70(7)	45(23)	15(2)	15(2)	9(47)	4(0)	0(0)
No(5)	867(93)	153(77)	750(98)	929(98)	10(53)	927(100)	20(100)
father alive							
Yes(1)	9(1)	3(2)	1(0)	1(0)	1(12)	0(0)	0(0)
No(5)	930(99)	146(98)	760(100)	939(100)	7(88)	934(100)	14(100)
country							
Austria	38(4)	38(4)	38(4)	38(4)	38(4)	38(4)	38(4)
Belgium	137(15)	137(15)	137(15)	137(15)	137(15)	137(15)	137(15)
Denmark	88(9)	88(9)	88(9)	88(9)	88(9)	88(9)	88(9)
France	126(13)	126(13)	126(13)	126(13)	126(13)	126(13)	126(13)
Germany	95(10)	95(10)	95(10)	95(10)	95(10)	95(10)	95(10)
Italy	133(14)	133(14)	133(14)	133(14)	133(14)	133(14)	133(14)
Sweden	137(15)	137(15)	137(15)	137(15)	137(15)	137(15)	137(15)
Switzerland	75(8)	75 (8)	75(8)	75(8)	75(8)	75(8)	75(8)
Spain	115(12)	115(12)	115(12)	115(12)	115(12)	115(12)	115(12)

Table 2.2: Descriptive Statistics Across Waves (continues variable)

Variable	Summary	wave 1	wave 2	wave 4	wave 5	wave 6	wave 7	wave 8
Immediate recall Score	mean(sd)	4.85(1.68)	4.92(1.67)	4.93(1.58)	4.83(1.63)	4.74(1.65)	4.54(1.69)	4.03(1.77)
Age for time interview	mean(sd)	69.31(3.74)	71.54(3.78)	75.88(3.74)	77.83(3.76)	79.78(3.76)	81.86(3.75)	84.48(3.76)
BMI	mean(sd)	26.41(3.89)	26.50(3.99)	26.45(4.08)	26.60(4.15)	26.49(4.30)	26.23(4.29)	25.80(4.43)
Income(thousand)(thinc_m)	median	25.11	21.78	24.24	23.58	21.64	20.42	21.00
	IQR	26.67	22.95	25.46	22.74	20.04	19.45	18.87
Education year	mean(sd)	9.38(4.81)	10.38(4.90)	10.37(4.92)	10.38(4.89)	10.38(4.87)	10.37(4.88)	10.37(4.88)
Maximum of grip strength(maxgrip)	median	31	30	29	27	27	26	24
	IQR	16	16	15	15	14	14	13
Household size(hhsize)	median	2	2	2	2	2	2	2
	IQR	1	1	1	1	1	1	1
Number of siblings alive	median	1	1	1	1	1	1	2
	IQR	2	2	2	2	2	1	3

2.4 Study sample

Due to filters/routing or an abandoned interview, there are plenty of missing values in the main SHARE database. Besides, apart from the above reasons, SHARE uses missing codes to represent the missing values, such as -1, -2, and -99. We regard all these missing codes as missing values. Independently of the chosen imputation method. SHARE provide five multiple imputations [42] of the missing values on some variables by using the simple hot-deck method or jointly by the fully conditional specification method (FCS) [49] for users to account for additional variability caused by the imputation process when evaluating the accuracy of their estimators. Hot-deck imputations are carried out separately by country, while FCS imputations are carried out by country and sample type. In our study, we use some variables that have been imputed by SHARE to reduce the proportion of missing values. For continuous variables, we calculate the mean of the five multiple imputations. And for discrete variables, we use the most frequent value.

Researchers in different fields often wonder what percentage of missing data should be removed. Yet, there is generally no established cut-off value for an acceptable portion of missing data in a dataset. By referring to various literature, we find that Schafer asserts that a missing rate of 5% or less is irrelevant [43] and Bennett believes that when more than 10% of the data are missing, the statistical analysis is likely to be biased [5]. Considering our dataset, we think 10% is an appropriate threshold. We further filter out variables if there is more than 10% missing data [33].

We suspected an accelerated decline in cognitive function after the age of 65, So we kept only participants whose baseline age is from 65 to 80. Among these, we find the sample size for the age over 95 is too small, which causes the outliers in the FPCA model. Therefore we remove the age over 95 in wave 8. In addition, we filter out subjects with DRS less than zero and do not consider interviewees with missing values in DRS. Since many respondents did not participate in all wave interviews, we exclude those respondents who did not have all 7 measurements (all waves except wave 3). The FPCA method sample includes 944 individuals with 6608 measurements. After filtering out variables with missing data, the sample size reduces to 713.

3 Methods

3.1 Sparse Functional principal component analysis

In sparse FPCA, sparse functional data as noisy sampled points are assumed to be independent realizations of a smooth random function with the mean is $\mathrm{E}(x_i(t)) = \mu_t$ and $\mathrm{cov}(x_i(s), x_i(t)) = G(s,t) = \sum_k \lambda_k \phi_k(s)\phi_k(t)$, $t, s \in \boldsymbol{T}$, where λ_k are non-increasing eigenvalues. The t and s are index variables as time which belong to the closed time interval $\boldsymbol{T}$. In classical FPCA, $\xi_{ik} = \int (x_i(t) - \mu(t))\phi_k(t)\, dt$ and the ith random curve is expressed as $X_i(t) = \mu(t) + \sum_k \xi_{ik}\phi_k(t)$. $t \in \boldsymbol{T}$, where ξ_{ik} is the kth FPCs for i^{th} random curve with $\mathrm{E}(\xi_{ik}) = 0$, $\mathrm{Var}(\xi_{ik}) = \lambda_k$, $\sum_k \lambda_k < \infty$, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots$, and $\phi_k(t)$ is the kth eignfunction. X_i can be represented by the sequence of $\xi_{i1}, \xi_{i2}, \ldots \xi_{ik}$. For any k, the first k term is the best k-dimensional linear approximation for $X(t)$ in L_2 space, which explains the highest proportion of variance in data with a given number of components.

Now we consider adding uncorrelated measurement errors with mean zero and constant variance σ^2 to reflect additive measurement errors and the model will become [54]:

$$Y_{ij} = X_i(T_{ij}) + \epsilon = \mu(T_{ij}) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(T_{ij}) + \epsilon_{ij} \tag{3.1}$$

where $T_{ij} \in \boldsymbol{T}$, ϵ_{ij} is the i.i.d. measurement error and is also independent of ξ_{ik}, where $i = 1, \ldots, n$, $j = 1, \ldots, N_i$, $k = 1, 2, \ldots$. N_i is the number of measurements on the ith subject.

Note Y_{ij} is the jth observation of the random function $X_i(t)$, made at a random time T_{ij}. In our study, Y_{ij} is a jth cognitive data point for i^{th} person and T_{ij} is the age in the jth measurement for i^{th} person, where i and j are integers. $j \in \{1, 2, 3, 4, 5, 6, 7\}$ and $i \in \{1, 2, 3, 4, \ldots, 944\}$

In sparse FPCA, we use local linear smoothers for function and surface estimation. The local lines and planes are fitted by weighted least squares. Mean, covariance and eigenfunctions are all assumed to be smooth [22]. The mean function $\hat{\mu}$ is estimated based on the pooled data from all individuals. The local linear scatterplot smoother for $\hat{\mu}$ is minimizing

$$\sum_{i=1}^{n} \sum_{j=1}^{N_i} \boldsymbol{k_1}\left(\frac{T_{ij} - t}{h_\mu}\right)(Y_{ij} - \beta_0 - \beta_1(t - T_{ij}))^2 \tag{3.2}$$

with respect to β_0, β_1. The estimate of $\mu(t)$ is $\hat{\mu}(t) = \hat{\beta}_0(t)$, $\boldsymbol{k_1}$ is a kernel function of order $(\boldsymbol{v}, l)$ and is compactly supported, $\|\boldsymbol{k_1}\|^2 = \int \boldsymbol{k_1^2}(u)\, du < \infty$, where $\boldsymbol{v}$ is a multi-index $\boldsymbol{v} = (\boldsymbol{v_1}, \boldsymbol{v_2})$. $\boldsymbol{v_1}.\boldsymbol{v_2}.\boldsymbol{l}$ are the given integers, with $0 \leq \boldsymbol{v_1} + \boldsymbol{v_2} \leq l$

The local linear surface smoother for $G(s,t)$ is defined by minimizing

$$\sum_{i=1}^{n} \sum_{1 \le j \ne l \le N_i} \boldsymbol{k_2}(\frac{T_{ij}-s}{h_G}, \frac{T_{il}-t}{h_G})(G_i(T_{ij},T_{il}) - f(\beta,(s,t),(T_{ij},T_{il})))^2 \tag{3.3}$$

where $f(\beta,(s,t),(T_{ij},T_{il})) = \beta_0 + \beta_{11}(s-T_{ij}) + \beta_{12}(t-T_{il})$. Minimization is with regard to $\beta = (\beta_0, \beta_{11}, \beta_{12})$. $\boldsymbol{k_2}$ is a kernel function of order $(\boldsymbol{v}, l)$ and is compactly supported, $\|\boldsymbol{k_2}\|^2 = \int\int \boldsymbol{k_2^2}(u,\boldsymbol{v})\, du d\boldsymbol{v} < \infty$. Usually, the smoothing parameter for this surface smoothing step is chosen by one-curve-leave-out cross-validation.

The covariance function between Y_{ij} and Y_{il} is given by

$$\text{cov}(Y_{ij}, Y_{il}|T_{ij}, T_{il}) = \text{cov}(X_i(T_{ij}), X_i(T_{il})) + \sigma^2 \delta_{jl} \tag{3.4}$$

where $\delta_{jl} = 1$ if $j = l$ and 0 otherwise.

Now let $G_i(T_{ij}, T_{il}) = (Y_{ij} - \hat{\mu}(T_{ij}))(Y_{il} - \hat{\mu}(T_{il}))$ be the raw covariances, where $\hat{\mu}(t)$ is the estimated mean function. Then the $\text{E}\left[G_i(T_{ij}, T_{il})\right]$ becomes

$$\text{E}\left[G_i(T_{ij}, T_{il})\right] \approx \text{cov}(X_i(T_{ij}), X_i(T_{il})) + \sigma^2 \delta_{jl} \tag{3.5}$$

By estimating the diagonal covariance function $V(t) = G(t,t)$ with the local linear smoother on the diagonal raw covariances $G_i(T_{ij}, T_{ij})$ (obtained from equation (3.2) by using $\{G_i(T_{ij}, T_{ij})\}$ as input) [29] [21] [24], if $\sigma^2 > 0$, the σ^2 of the measurement errors in Equation (3.1) is estimated by

$$\hat{\sigma}^2 = \frac{2}{|\boldsymbol{T}|} \int_{\boldsymbol{T}_1} \left[\hat{V}(t) - \hat{G}(t,t)\right] dt \tag{3.6}$$

where $|\boldsymbol{T}|$ denote the length of $\boldsymbol{T}$ and $\boldsymbol{T}_1$ is the interval $\boldsymbol{T}_1 = [inf\{x : x \in \boldsymbol{T}\} + |\boldsymbol{T}|/4, sup\{x : x \in \boldsymbol{T}\} - |\boldsymbol{T}|/4]$

The eigenfunctions and eigenvalues correspond to the $\hat{\phi}_k$ and $\hat{\lambda}_k$ can be estimated by the following equation with the constraints $\left\|\hat{\phi}_k(t)\right\| = 1$ and $\langle \hat{\phi}_k(t)\ ,\ \hat{\phi}_m(t)\rangle = 0$, for $m < k$. $\hat{G}(s,t)$ denote the smooth surface estimate of $G(s,t) = \text{cov}(X(s), X(t))$ (see Equation (3.3))

$$\int_{\boldsymbol{T}} \hat{G}(s,t)\hat{\phi}_k(s)\, ds = \hat{\lambda}_k \hat{\phi}_k(t) \tag{3.7}$$

For the classical FPCA, FPC scores are estimated by $\xi_{ik} = \int (x_i(t) - \mu(t))\phi_k(t)\, dt$, which works well for the sufficient density of the grid of measurements for each subject. However, for sparse FPCA, since the Y_{ij} are only available at discrete random times T_{ij}, this numerical integration can not provide a reasonable approximation. The FPCs for sparse FPCA should be estimated under the assumption that ξ_{ik} and ϵ_{ij} are jointly Gaussian, and the best prediction is given by

$$\tilde{\xi}_{ik} = \text{E}(\xi_{ik}|Y_i) = \lambda_k \phi_{ik} \textstyle\sum_{Y_i}^{-1} (Y_i - \mu_i) \tag{3.8}$$

where $Y_i = (Y_{i1}, \ldots, Y_{iN_i})$, $\sum_{Y_i} = \text{cov}(Y_i, Y_i)$, $\phi_{ik} = (\phi_k(T_{i1}), \ldots, \phi_k(T_{iN_i}))$ and $\mu_i = (\mu(T_{i1}), \ldots, \mu(T_{iN_i}))$, Note the true values of λ_k, ϕ_{ik}, $\sum_{Y_i}$ and μ_i are unknown. $\tilde{\xi_{ik}}$ is the best linear prediction of ξ_{ik}, given the information from the ith subject, no matter whether the Gaussian assumption holds or not.

By substituting estimates of μ_i, λ_k, ϕ_{ik} and $\sum_{Y_i}$ from the entire data ensemble, Yao, Muller and Wang (2005) proposed to use the conditional expectation to estimate scores

$$\hat{\xi}_{ik} = \hat{\lambda}_k \hat{\phi}_{ik} \hat{\sum}_{Y_i}^{-1} (Y_i - \hat{\mu}_i) \tag{3.9}$$

The prediction of trajectory $X_i(t)$ is given by

$$\hat{X}_i^K(t) = \hat{\mu}(t) + \sum_{k=1}^{K} \hat{\xi}_{ik} \hat{\phi}_k(t) \tag{3.10}$$

The K is chosen by cross-validation based on the one-cure-leave-out prediction error [54]. This conditioning method provides the best predictors under Gaussian assumptions and works in the presence of both measurement errors and sparsity.

All statistical analyses were performed using R studio version 4.0.4 and FPCA was performed using the *fdapace* package [21] [54] [1] [20]. The first step toward using the FPCA function in the fdapace package is to restructure the data so that each respondent's time and DRS data are stored as lists in separate columns, where each row contains all of the data for a single id. The work-flow for sparse FPCA is as follows:

- Calculate the smoothed mean $\hat{\mu}$ (local linear smoothing) by aggregating all available cognitive curves together (see Equation (3.2)).
- Calculate each cognitive curve's own raw covariance separately and aggregate all these raw covariances to form the sample raw covariance (see Equation (3.4) and (3.5)).
- Use the off-diagonal raw covariances to estimate the smooth covariance (see Equation (3.3)).
- Obtain the eigenfunctions $\hat{\phi}$ and eigenvalues $\hat{\lambda}$ by doing eigenanalysis on the smooth covariance and then project the smooth covariance on a positive semi-definite surface (see Equation (3.7)) [26].
- Use conditional expectation to estimate the $\hat{\xi}$ (see Equation (3.8) and (3.9)).

In functional data analysis, FPCA plays a vital role. The top FPCs explain major sources of total variation among the whole dataset. We use FPCA to catch major variance, avoid overfitting, and smooth the fitted curves [44]. The FPCA method can accurately capture the time fluctuation characteristics of cognitive function, especially the changing direction and form in time, which provides a scientific basis for modelling the cognitive function of the elderly. In the next section, we introduce some machine learning models and methods, namely cross-validation, Lasso, ridge, elastic network model, decision tree and random forest model.

3.2 Statistical learning

3.2.1 Shrinkage Models

After applying the FPCA method for dimension reduction, we use shrinkage models to identify the risk factors. In the machine learning field, it is called feature selection. Our dependent variable is FPC1 or

FPC2 (y_i) which we obtain from the FPCA technique and independent variables (x_{ij}) are the covariates in Table 1 and 2. Note that since after removing all missing values there are 713 individuals and 39 covariates remaining, so $i \in \{1, 2, 3, \ldots, 713\}$ and $j \in \{1, 2, 3, \ldots, 39\}$. After fitting the shrinkage model with significant coefficients(β_j), the remaining covariates are the risk factors we want to recognise.

Ridge and Lasso Regression

The ridge regression and Lasso regression are the two best-known techniques for the shrinkage of the regression coefficients. By adding the L_1 or L_2 penalty into the loss function, both methods could shrink the regression coefficients toward zero. These two methods are very similar to least squares ($RSS = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$), except that the coefficients are estimated by minimizing a slightly different quantity. The ridge regression provides the minimum lambda to minimize the

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{3.11}$$

where λ is a tuning parameter that greater or equal to zero. The shrinkage penalty $\lambda\sum_{j=1}^{p}\beta_j^2$ has the effect of shrinking the estimates of β_j towards zero, but it never shrinks the β_j exactly to zero. The parameter λ is used to control these two terms' relative influence on the regression coefficients' estimation. That is, when λ is zero, the penalty term has no effect, and when λ approaches infinite, the penalty becomes more impactive, ridge regression coefficient tends to be zero. Ridge regression has the obvious disadvantage that the final ridge regression model will contain all predictors. However, the Lasso regression can overcome this disadvantage.

The Lasso regression provides the minimum lambda to minimize the

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{3.12}$$

L_1 penalty equal to $\lambda\sum_{j=1}^{p}|\beta_j|$ can estimate all the coefficients exactly to zero when the tuning parameter λ is sufficiently large. Therefore, the Lasso performs variable selection, and the Lasso model is much easier to interpret than the ridge. The λ usually is selected by cross-validation methods. By choosing a grid of λ values, we select the λ for which the cross-validation error is the smallest. Lasso also has obvious shortcomings. When there is a multicollinearity problem, Lasso randomly selects a multicollinear variable, so Lasso cannot overcome the multicollinearity problem. The shrinkage regressions (e.g. ridge, Lasso regression) usually significantly reduce the prediction variance and achieve the purpose of coefficient shrinkage and variable selection, but they all have limitations.

Elastic Net

The elastic net is a hybrid between the Lasso and ridge model for improving the shortcomings of these two techniques; it uses the penalties as a combination of a L_1 norm penalty and a L_2 norm penalty. Ridge

with L_2 norm $= \sum_{j=1}^{p} \beta_j^2$ includes all the predictors in the final model and will not perform feature selection. Lasso with L_1 norm $= \sum_{j=1}^{p} |\beta_j|$ only takes a few samples for high-dimensional data. Like ridge and Lasso, we attempt to minimize the residual sum of squares with different penalty terms:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \left[(1-\alpha)(\sum_{j=1}^{p} \beta_j^2)/2 + \alpha \sum_{j=1}^{p} |\beta_j| \right] \tag{3.13}$$

note that dividing the ridge penalty by 2 is convenient for optimization.

The new penalty is $\lambda \left[(1-\alpha)(\sum_{j=1}^{p} \beta_j^2)/2 + \alpha \sum_{j=1}^{p} |\beta_j| \right]$, which balance the two penalties of Lasso and ridge and can result in a better performance on some problems. Notice that α and λ are always positive numbers. It is more useful to think of α as controlling the mixing between the two penalties and λ controlling the amount of penalization. α takes values between 0 and 1. When $\alpha = 1$, it gives Lasso and when $\alpha = 0$, it gives ridge. We use R Package *caret* to fit the elastic-net model [32]. In that package, we set up a 10 fold cross-validation strategy and use $train()$ with $method =$ "$glmnet''$ to fit the elastic net. We allow *caret* automatically choose the best tuning parameters α and λ based on the minimizing cross-validation error.

If a group of variables is highly correlated, Lasso tends to select only one in this group and ignore the rest. In elastic-net, strongly correlated predictors tend to appear together in or removed from the model [56]. Hence, elastic-net performs feature selection and regularization simultaneously. In our case, some selected variables may be highly correlated. With the grouping effect of elastic-net regularization, those correlated variables will be selected together, and non-zero coefficients will become more interpretable.

3.2.2 Decision tree

The decision tree method is a predictive algorithm that can perform both classification and regression based on multiple covariates. Common usages of decision tree models include feature selection, assessing the relative importance of variables, handling of missing values, prediction, and data manipulation. Besides, decision trees are fundamental components of random forests. The main components of a decision tree model are nodes and branches. Each node represents a "test", and each branch represents the result of the test. The node in the tree model shows the predicted value, the number of data points reaching this node, and the population percentage in this node.

The Figure named Decision Tree shows the prediction of the FPC1 from the categorical variables. This decision tree starts with a single node (base. country = Austria, Denmark, France, Germany, Sweden, Switzerland), which branches into possible outcomes. Each outcome leads to additional nodes, which branch off into other possibilities. 0.032 is a predicted value for the first FPCs, 713 is the number of data points reaching this node, and 100% is the percentage of the population in this node. For example, if you choose "Yes" in the first node, it comes to the node base.numeracy = 4 or 5 (with probability 58%) and if you choose "No", it comes to the node base.writing = good (with probability 42%).

All statistical analyses about decision tree are using R Package *rpart* [48] and all visualization plots are

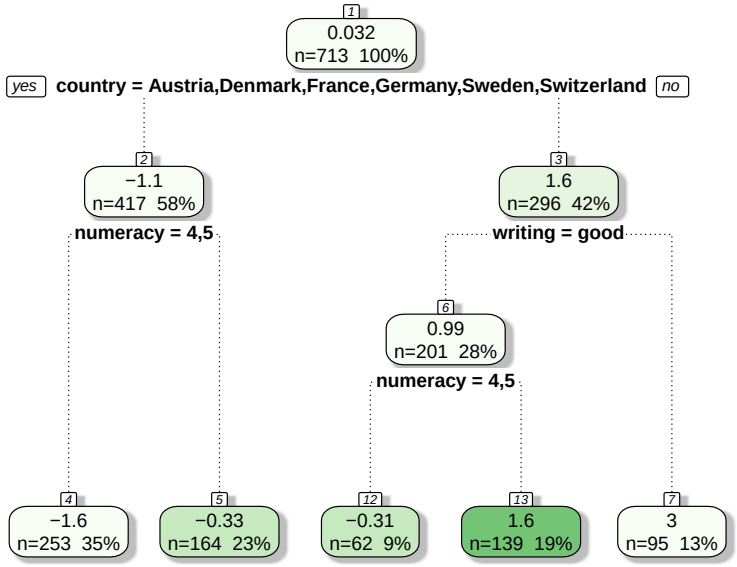

Figure 3.1: Decision Tree

using R Package *rpart.plot* [35]. Since the outcome variables, FPC1 or FPC2 are continuous. We need to build the regression tree rather than the classification tree. The process of building a regression tree is roughly two steps. First, we construct distinct and non-overlapping nodes and put each observation onto these nodes. Second, we make the same prediction for every observation that falls into the node by simply calculating the mean response values for the training observations. For example, if there are node 1 and node 2 obtained in the first step. The response mean of the training observations in the first node is 5 and in the second node is 10. Then if given observations $a_1 \in$ node 1 and $a_2 \in$ node 2, we will predict a_1 as 5 and a_2 as 10. We construct the node by minimizing the residual sum of square (RSS), given by

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{3.14}$$

where R_j represents j^{th} node, $y_i - \hat{y}_{R_j}$ is the mean response for the training observations within the jth node. The decision tree first considers all predictors and all possible values of cutpoint for each predictor. Then it chooses the predictor and cutpoint such that the resulting tree has the lowest RSS.

The resulting decision tree might be too complex, which is likely to overfit the data, leading to poor test set performance. Therefore, we will prune a tree to obtain a subtree. Rather than considering every possible subtree, cost complexity (CP) determines the best prune way to prune the tree. We consider a sequence of

trees indexed by a nonnegative tuning parameter α. For each value of α, there corresponds a subtree $T \in T_0$ such that

$$\sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \tag{3.15}$$

is as small as possible. Here T_0 is the tree before pruning, and $|T|$ is the number of terminal nodes of the tree T. R_m is the subset of predictor space (rectangle) corresponding to the mth terminal node. $\hat{y}_{R_m}$ is the predicted response associated with R_m. The parameter α controls the trade-off between the complexity of the subtree and the fit to the training data. When $\alpha = 0$, $T = T_0$. Usually, we use cross-validation to select the value of α and then return it to the dataset for obtaining the subtree corresponding to α.

The final predicted value for the decision tree is FPC1 or FPC2, which obtain from the FPCA technique, and the nodes are picked from the covariates in Table 1 and 2. Our objective is to find the risk factor, which is the name of each node in the decision tree. The step for the decision tree:

- Pick the variables based on the lowest RSS and grow a large tree on the training data.
- Snipping off the least important splits based on the CP (prune the tree).
- Use cross-validation (Repeat step 1 and step 2 on all but the kth fold of the training data) to choose the α that minimizes the mean squared prediction error.
- Obtain the subtree in step 2 with the α in step 3.

The decision tree model is straightforward to understand and interpret, and it can handle heavily skewed without doing transformation, and missing data without imputation [47]. However, decision trees are not stable; a tiny change in data may lead to a significant difference in the structure of the optimal decision tree. Decision trees also tend to have an over-fitting problem, which random forests can handle, but they are not as easily explained as decision trees.

3.2.3 Random forest

The Random Forest algorithm proposed by L. Breiman in 2001 is a powerful machine learning classifier that performs classification by constructing numerous decision trees by bootstrapping and aggregating their results to create the final model (called bagging). Bootstrapping is resampling the observed dataset (and of equal size to the observed dataset), each of which is obtained by random sampling with replacement from the original dataset [14] [46]. Figure 3.2 depicts how Bootstrapping works. The right part is the original dataset, and the left details are the training datasets bootstrapping. It displays a small sample size $n = 3$. Each bootstrap dataset contains 3 observations, and it chooses samples with replacements from the original dataset. $\hat{\alpha}^{\star i}$ are estimated by each bootstrap dataset.

By generating B different bootstrapped training data sets, we train our method on the bth bootstrapped training data set to get $\hat{f}^{\star b}(x)$. After averaging all the predictions, we obtain the following equation called

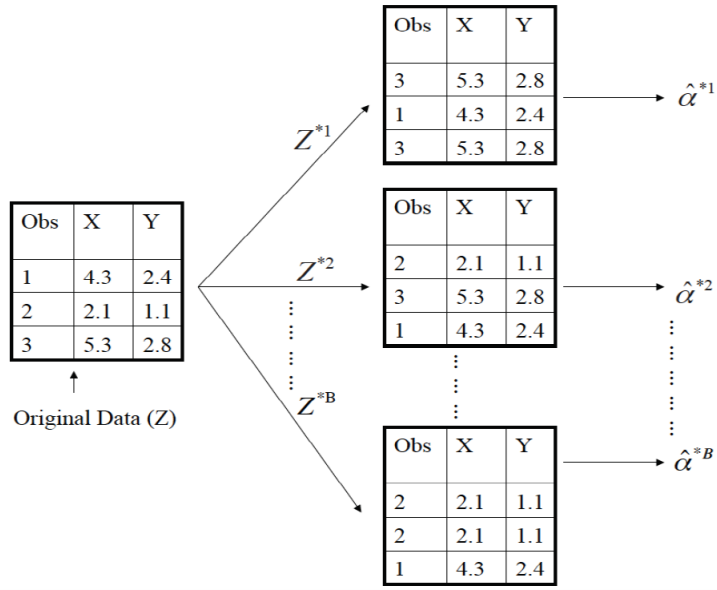

Figure 3.2: Bootstrapping

bagging. With bigger values of B, the variance of the decision trees will decrease while the computational time will grow substantially. An optimal number of trees B can be found using cross-validation to balance precision and computational cost. However, bagging contains a large number of decision trees, it is no longer possible to exhibit a process like a single decision tree, and it is no longer clear which variables are critical to the overall process. Thus, the improved predictive power of bagging comes at the expense of interpretability.

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{\star b}(x) \tag{3.16}$$

We build several decision trees on bootstrapped training samples in the random forest model. When building these decision trees, a split in a tree is considered at each time, and a random sample with m predictors is chosen from the full set of p predictors. Splitting allows only one of the m predictors to be used. The new sample of m predictors is taken for each split. Usually, we choose $m \approx \sqrt{p}$. Note that m is the number of predictors considered at each split, and p is the total number of predictors. The difference between bagging and the random forest is the predictor subset size m. If a random forest is built using the m equal to p, this amounts simply to bagging.

Because of the law of large numbers, random forest rectifies the overfitting problem in decision trees for the training set. And the right kind of randomness makes them accurate classifiers and regressors. The output of the random forests for classification is the class chosen by the majority of the trees, and the regression task returns the mean prediction of the individual trees [41]. Excellent performance is shown by random forest when the number of variables is much larger than the number of observations; it allows hundreds of input variables without variable deletion. Furthermore, random forests have extremely high classification rates,

and they can make feature selection by determining the importance of variables [15]. All statistical analyses about random forest are using R Package *randomForest* [40]. The sample size of each bootstrap replicate is not the same as the original data, which is around 480 each time. There are 500 number of trees grown and 12 number of predictors sampled for splitting at each node.

Steps involved in random forest algorithm:

- Create a Bootstrapped Data Set.
- Individual decision trees are constructed for each sample.
- Each decision tree will generate an output.
- Go back to Step 1 and Repeat
- Bootstrapped the data and used the aggregate from all the trees to make a decision. This process is known as Bagging (Majority Voting or Averaging for Classification and regression, respectively).

Usually, there are two measures of variable importance for random forests: Mean Decrease Gini and Mean Decrease Accuracy. In this thesis, we use Mean Decrease Accuracy(%IncMSE), the most robust and informative measure obtained from RSS (for bagging regression trees). For calculating the %IncMSE for a given predictor, we record the total amount of RSS decrease due to splits over this given predictor and average overall bagging trees. The higher value of Mean Decrease Accuracy, the higher importance of the variable in the random forest model.

3.2.4 Predictive R^2 with CV

For some regularized methods (e.g. Lasso), we typically have numerous variables, and a model with all the variables will always have the largest R^2. As such, we are using R^2 is not recommended. With a continuous dependent variable, we first divide the whole dataset into 10 folds and use one of the folds as a testing set and the others as a training set. Each fold will become a test set, making a total of ten predictions. Then we would use the Predicted Residual Sum of Squares (PRESS) statistic, which is given by

$$\text{PRESS} = \sum_{k=1}^{K} \sum_{i \in \text{fold}_k} (y_i - \hat{y}_{i,-k})^2 \tag{3.17}$$

where $\hat{y}_{i,-k}$ is the ith predictive value for y_i without using test cases in fold_k. $k \in \{1, 2, 3, \ldots, 10\}$ $i \in \text{fold}_k$.

The predictive R^2 is denoted by

$$R^2 = 1 - \frac{\text{PRESS}}{\text{SS}} \tag{3.18}$$

where $\text{SS} = \sum_i (y_i - \bar{y})^2$, $\bar{y}$ is the mean of y_i in the whole dataset.

3.2.5 Cross Validation

Cross-validation is a statistical method used to evaluate the performance of a machine learning model. It is often used in applied machine learning to compare and select models for a given predictive modelling problem, as it helps us choose the model that performs best on unseen data, evaluate the quality of the model and avoid overfitting and underfitting. In this method, we split the data into train and test sets so that the sample between train and test set does not overlap[17]. As there is never enough data to train a model, removing a part of it for validation poses a problem of under-fitting. We risk losing essential patterns by reducing training data and increasing bias-induced error. K-Fold and Leave One Out Cross-Validation are popular and easy ways to understand. They generally result in a less biased model compared to other methods. Because it ensures that every observation from the original dataset has the chance of appearing in the training and test set, this is one of the best approaches if we have limited input data[16].

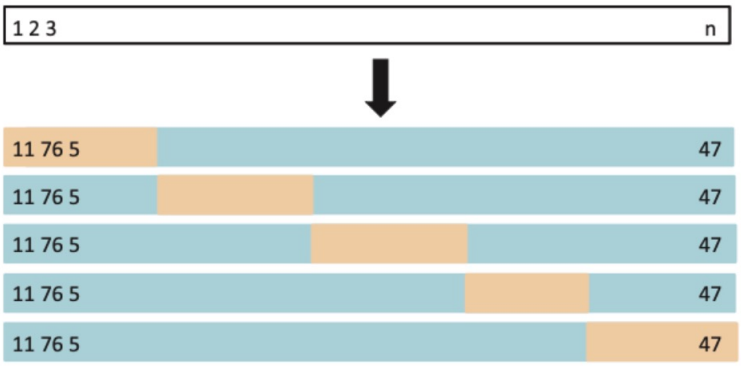

Figure 3.3: 5-fold Cross-Validation

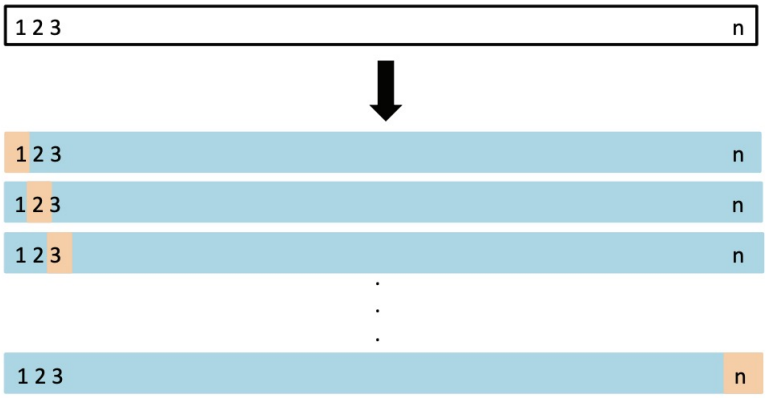

Figure 3.4: Leave One Out Cross-Validation

A set of n observations randomly divided the dataset into two segments. One segment is for training the model, and remains for testing the model. K-fold cross-validation means randomly partitioning all samples

into k equal-sized and non-overlapping parts. One of the k parts is regarded as the validation data for testing the model, and the remaining $k-1$ parts are used as the training set. The k-fold cross-validation process repeats k times, with each k subset being used once as the validation set. Figure 3.1 is a schematic display of a 5 Fold cross-validation.

In LOOCV, we divide the data set into two parts. In one part, we have a single observation, which is our test data, and in the other part, we have all the other observations from the dataset forming our training data. If we have a data set with n observations, then training data contains $n-1$ observation, and test data contains 1 observation. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth. This process is iterated for each data point as shown in Figure 3.2[16].

The test error is then estimated by averaging the n resulting MSEs. It is described as follows:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.19}$$

where y_i is the observation and $\hat{y}_i$ is the predictive value of this observation.

4 Result

4.1 FPCA

Elderly cognitive data contains one time-varying trait, the DRS. We obtained patterns of variation in the cognitive outcomes by using FPCA. Figure 4.1 shows the mean function, scree plot and the first three eigenfunctions. The y-axis in the mean function plot denotes the normalized DRS. The computed mean function for the data shows a dip in DRS near the beginning, then a clear upward trend at age 70, followed by an abrupt decline at age 75. In the scree plot, the y-axis represents FPCs, and the x-axis represents the fraction of variance explained. It can be seen that FPC1 explained a large fraction of the variation (78.0%). The second, third, and fourth FPCs explained 14.2, 6.7 and 1.1% of the variation, respectively. The first two FPC explained 92.2% of the variability, and 99.9% of the variability was described by the first 4 FPCs. The remaining are less important except for the first two FPCs. Eigenfunctions(EFs) are also known as weight functions in Chapter 3. Interpreting these EFs can be quite hard, as there might not be an obvious counterpart in the data. EFs of DRS seem to explain that EF1 is the reduction of the DRS from the mean function over age, EF2 is the reduction of DRS before and after the age of 80.

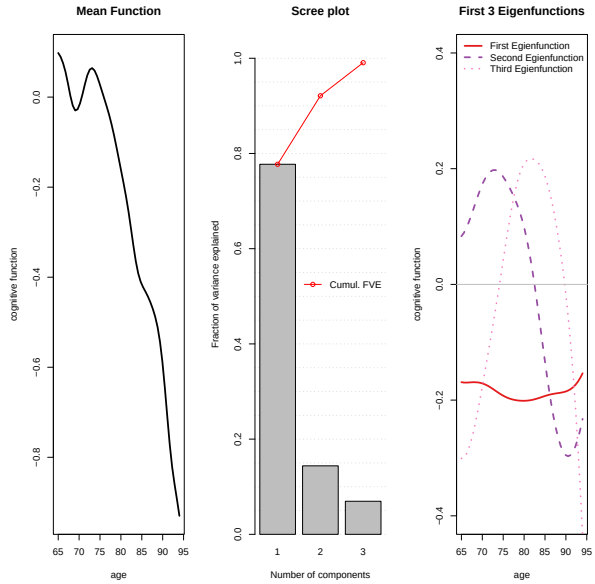

Figure 4.1: Summary Of FPCA Result

4.2 Comparison of FPC scores by countries

The sample data after filtering involved 9 European countries (Austria, Belgium, Denmark, France, Germany, Italy, Spain, Sweden, and Switzerland). We compare the first two FPCs and the median longitudinal trajectory for these nine countries in Figures $4.2 - 4.5$. The curves in Figures 4.2 are the mean function plus the median FPC1 of each country. Since the EF1 is a decline in the cognitive mean function, We can regard these curves as the mean cognitive curve after the first dip. After adding the median FPC2 of each country, the curves shown in Figure 4.3 show another descent of the curves 4.2. It can be seen from both Figures that the median longitudinal trajectory curves(MLTC) of Italy and Spain are at the bottom. However, from Figure 4.4, we find that Italy and Spain have higher FPC1 than other counties, and Sweden and Switzerland have relatively high FPC1. This is because EF1 is negative. The first FPC for each country has a negative correlation with the actual DRS, which can be verified in Figures 4.2 and 4.3. We guess it is because these two countries' economies are not great compared to other European countries, resulting in an imperfect health care system and a lack of medical conditions.

Table 4.1 shows the Euro Health Consumer Index(EHCI) in 2015 (from the Euro health consumer index 2015 report), which was a comparison of European health care systems based on waiting times, results, and generosity and included 37 European countries. Based on patient rights and information score (PRIS), pharmaceuticals score and overall score ranking in the table, we find that Spain and Italy are in eighth and ninth positions for these indexes, respectively. In addition, these two countries do not have high scores on waiting times for treatment (scores) (WTTS), outcomes (scores) (OS), range and reach of services (scores) (RRSS) and prevention scores, which provide evidence for our results that the MLTC of these two countries is at the bottom among these nine European countries. Figure 4.1 shows that 80 is a turning point, and FPC2 will become negative after about 80 years old. According to the boxplot in Figure 4.5, only the medians of Spain and Italy are below zero. This result confirms the summary result mentioned in section 2.1, the severe problem of population ageing in Spain and Italy.

The top 5 MLTC are from Sweden, Switzerland, Denmark, Germany, and France. The comparative figure of the FPC2 (Figure 4.5) shows that the overall country difference in pattern change is not as substantial as the difference in the first FPC scores.

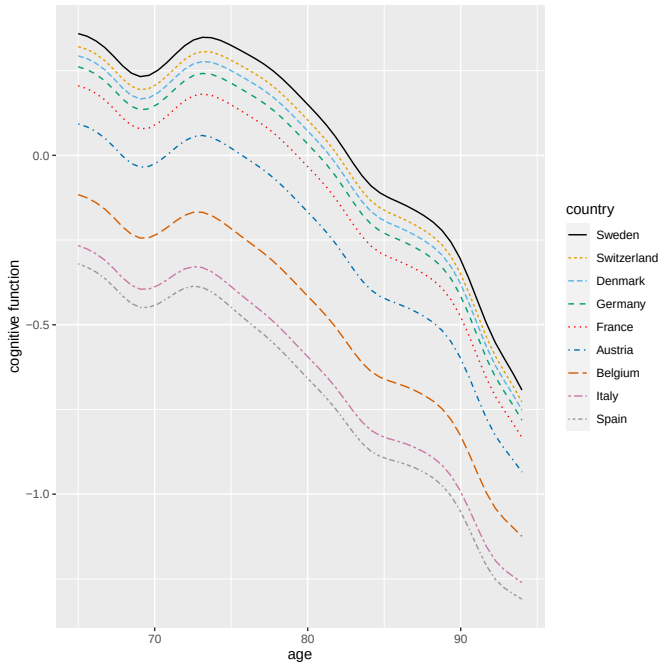

Figure 4.2: Overall Cognitive Function Trajectory Across Different European Countries Based On FPC1. (Mean function + Median FPC1 of each country)

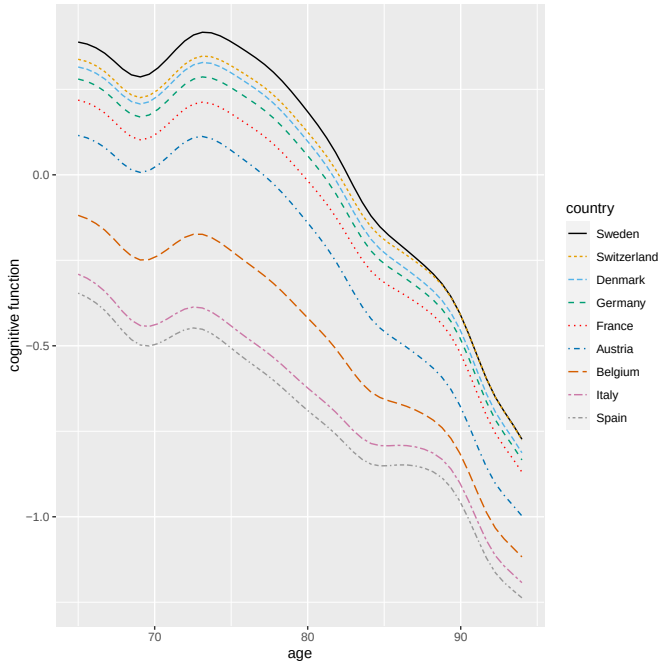

Figure 4.3: Overall Cognitive Function Trajectory Across Different European Countries Based On FPC1 and FPC2. (Mean function + Median FPC1 + Median FPC2 of each country)

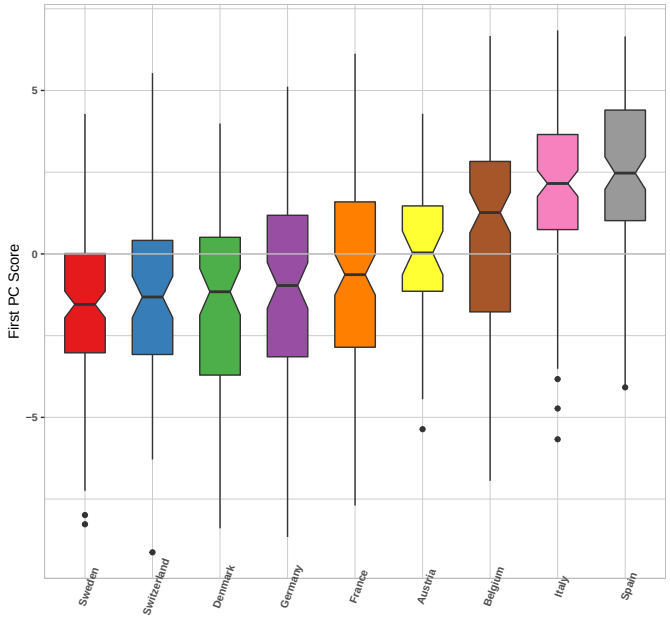

Figure 4.4: Bloxplot Of FPC1 Across Different European Countries

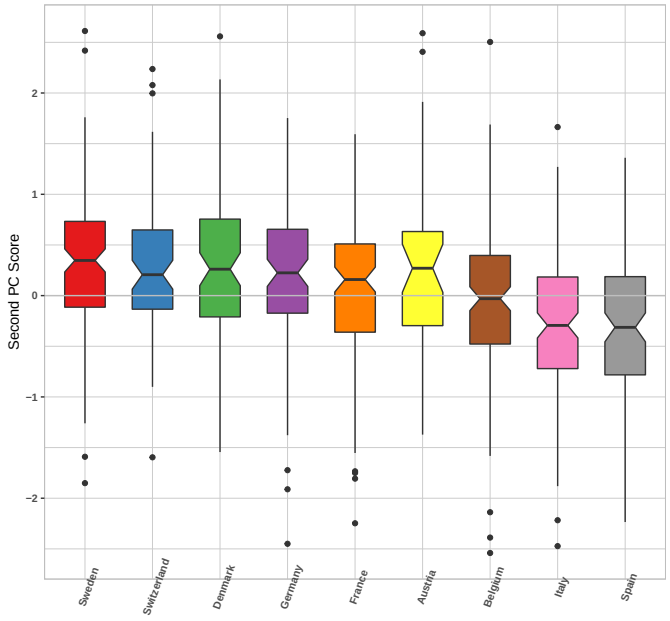

Figure 4.5: Bloxplot Of FPC2 Across Different European Countries

Table 4.1: Euro Health Consumer Index 2015

Country	Overall ranking	Total socre	PRIS	WTTS	OS	RRSS	Prevention score	PS
Switzerland	2	894	133	225	240	119	101	76
Belgium	5	836	117	225	198	131	89	76
Germany	7	828	125	188	229	94	107	86
Denmark	9	793	133	138	219	138	89	76
Sweden	10	786	125	100	229	144	107	81
France	11	775	113	188	208	106	89	71
Austria	12	774	121	188	188	119	83	76
Spain	19	695	104	113	198	113	101	67
Italy	22	667	96	138	188	88	101	57

4.3 Predictive R^2 for models

Table 4.2 shows the predictive R^2 of FPC1 and FPC2 for different models. We calculate predictive R^2 four times for each model (two with and two without covariate DRS). From the table, we conclude that the predictive R^2 for FPC1 with covariate DRS is around 0.4 to 0.5. If we remove the covariate DRS, the predictive R^2 drops from 0.4 to 0.2. Nevertheless, the Predictive R^2 of FPC2 for all models is very low, around 0.1, regardless of whether the covariate DRS is contained. The best result of predictive R^2 for predicting FPC1 is to use the Elastic-net model, which is 0.550 with covariate DRS and 0.450 without covariate DRS. Again, the best result of predictive R^2 for predicting FPC2 is to use the elastic-net model (0.132 with covariate DRS and 0.115 without covariate DRS). Since the predictive R^2 for FPC2 is low, we should not over-interpret the coefficients as we presented in the previous report. Many factors for explaining the reduction of cognitive functions after 80 years old haven't been included in this study, and this will be an exciting topic for future research.

Table 4.2: Predictive R^2 for machine learning models

Methods	For FPC1 with DRS	For FPC1 without DRS	For FPC2 with DRS	For FPC2 without DRS
Elastic-net	0.550	0.450	0.132	0.115
Lasso	0.540	0.400	0.130	0.080
Random Forest	0.479	0.370	0.122	0.100
Decision Tree	0.436	0.260	0.094	0.074

4.4 Indentification of Risk factors for cognitive decline

We set up a 10 fold cross-validation strategy for choosing the parameter α and λ in the elastic net model. And we use the argument "tuneLength" in R that tests different combinations of values for α and λ. There are ten α values from 0.10 to 1.00, and each tried ten times to select the optimal model. The smallest value

of the root mean square error (RMSE) is used to select the optimal model.

With FPC1 as dependent variable, the final final tuning parameter used for the model are $\alpha = 0.2$ and $\lambda = 0.28$. Generally, a model with all the variables will have a larger R^2. Hence for a regularized model, we prefer to use RMSE, mean absolute error(MAE) and predictive R^2 for goodness-of-fit measures. The $RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$ for the model include all covariates in Table 2.1 and 2.2 is 2.28 and $MAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n}$ is 1.87. If we include the covariate DRS the $RMSE$ and MAE decrease to 2.11 and 1.72, respectively. The predictive R^2, including all covariates, is 0.42 and reduced to 0.55 if covariate DRS is added. With FPC2 as dependent variable, the final tuning parameter used for the model are $\alpha = 0.2$ and $\lambda = 0.28$. However, the predictive R^2 for FPC2 is low (0.132 with DRS and 0.115 without DRS). The details of the predictive R^2 are in Table 4.2.

In Figures 4.6 and 4.7, we report the risk factors of baseline covariates by using elastic-net. The positive, negative, and no effects with an order of importance are represented by red, blue, and grey, respectively. The top five positive effects in baseline wave for FPC1 are Gross Motor Skills level 2 out of 4, Country Spain, Country Italy, Fine Motor Skills level 2 out of 4 and Instrumental Activities of Daily level 1 out of 2. And top five negative effects for FPC1 are Immediate recall score, country Sweden, numeracy score of 5 out of 5, Country France and gender Female. Notice since EF1 is negative. All effects are opposite for the original DRS. Namely, A negative effect on FPC1 is a positive effect on DRS.

In general, the elderly living in countries with excellent health care systems, being in good health, having a long education experience, being accompanied by family members, exercising regularly, being in a good mood without depression, having an average cognitive ability higher than the average of all people. More specifically, We find that the Gross Motor Skills Index level 2 out of 4 is the factor that decreases the most of the mean cognitive function (Fig. 4.6). This Index describes the sum of walking 100 meters, walking across a room, climbing one flight of stairs and bathing or showering. The higher the index, the better the cognitive ability of the participant. Similarly, Fine Motor Skills Index is the sum of picking up a small coin, eating/cutting up food and dressing. Instrumental Activities are the sum of telephone calls, taking medications, and managing money. Fine Motor Skills Index level 2 and Instrumental Activities level 1 are relatively low activity indexes. These results illustrate the importance of activity for cognitive decline. In addition, as we mentioned in section 4.2, the country is a significant factor affecting cognitive function, and the same conclusion can be drawn from Figure 4.6 as in section 4.2. The elderly from Spain, Italy, and Belgium showed slower mean cognitive function decline, while the elderly from Sweden, France and Denmark showed a faster reduction. Furthermore, we found that the mean cognitive function decreased more in the elderly hospitalized and with a high depression index. Cognitive decline can be mitigated by long years of education, good reading, writing skills, and eyesight. Immediate recall score is similar to delay recall score (DRS), so it greatly inhibits mean cognitive function decline as we expected. Compared with the top effects in Figure 4.7, we find the significant effects in both figures are very similar. Some covariates are continually significant. (e.g., country, Gross Motor Skill, immediate recall score)

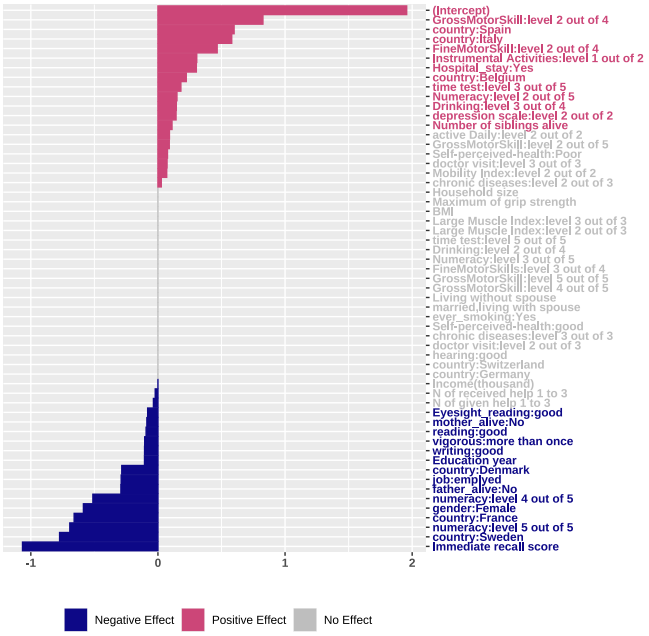

Figure 4.6: Covariate Effects on the First FPC Scores by Elastic-Net, Note all the covariates are in the baseline wave 1.

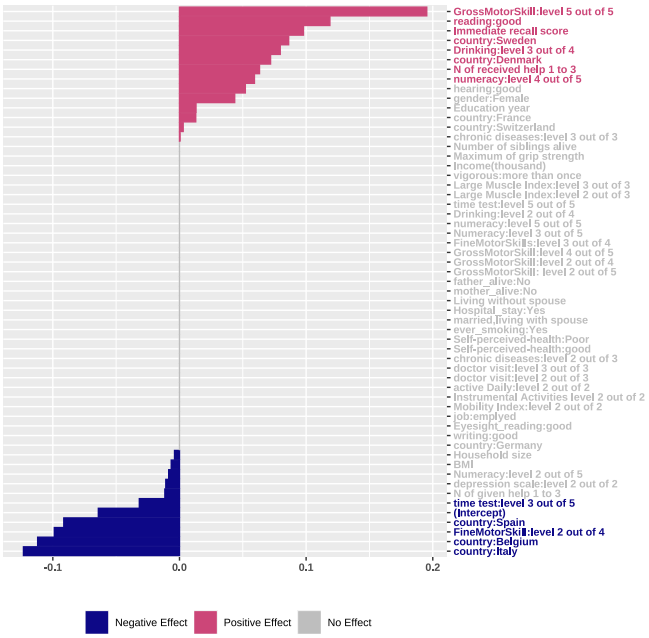

Figure 4.7: Covariate Effects on the Second FPC Scores by Elastic-Net

To validate and compare risk factors estimated by elastic-net shown in Figure 4.6-4.7, we report four graphs (see Fig. 4.8-4.9 in the main text and Figures in Appendix A) where we offer the tree-based method for predicting FPC1 and FPC2. We employ the tree's CP value with the smallest cross-validation error. The decision tree algorithm can naturally select which features are most crucial [47]. Besides, we prune parts of the tree that do not provide the power to classify instances to reduce the probability of overfitting problems. The features that impact our outcome variable DRS demonstrated in these four figures are comparable to those in the elastic-net model. (e.g., immediate recall score, country, numeracy score, reading)

Nevertheless, decision trees still suffer from overfitting the training data. Thus, we also use a random forest model to identify significant variables (risk factors). As mentioned above, a random forest model is a way of producing multiple decision trees with different parts of the same training set, which is insensitive to the overfitting of an individual decision tree [4]. Mean Decrease Accuracy ranks the random forest algorithm's essential features shown in Figure 4.13. Immediate recall score, country, numeracy score, reading and gender are still the top important variables.

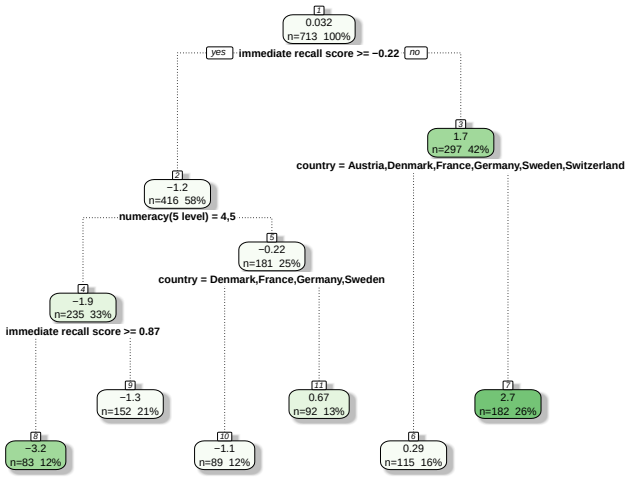

Figure 4.8: Pruned Tree based method for predicting FPC1 from all the variables

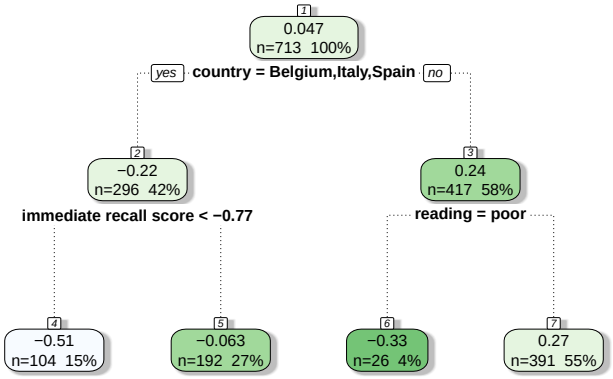

Figure 4.9: Pruned Tree based method for predicting FPC2 from all the variables

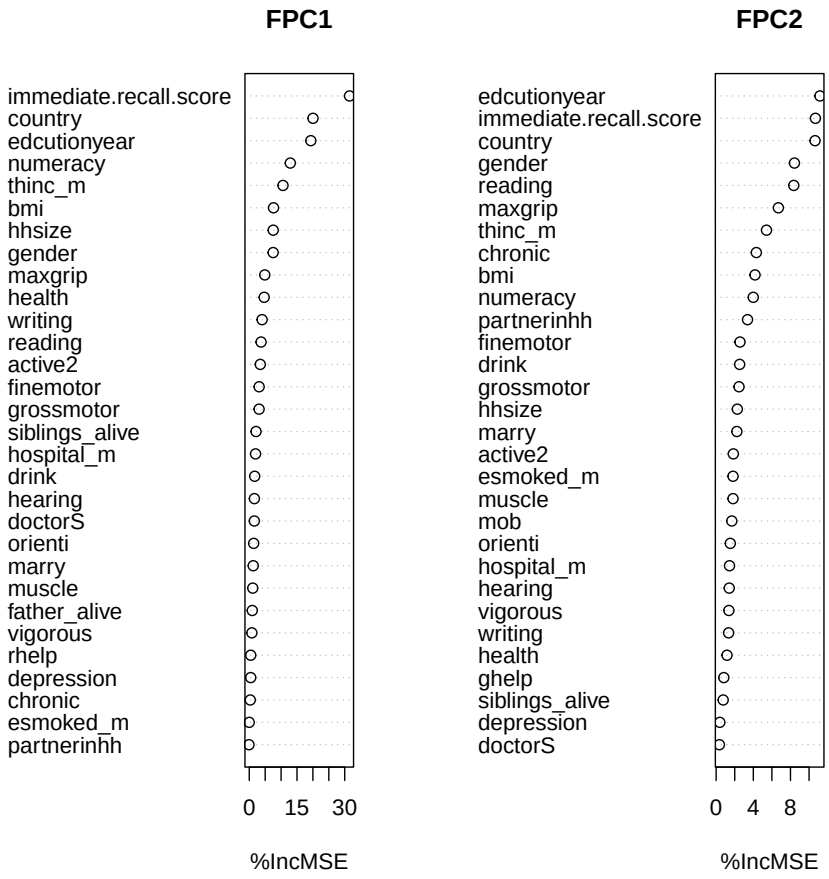

Figure 4.10: Top 30 ranked variables identified by Random Forest based on the Two FPC Scores

5 Conclusion

In this thesis, we have shown how the FPCA is helpful in the study of cognitive decline among age in the elderly, which can catch the major variance of the longitudinal cognitive measurements. Applying these functional techniques to European individuals aged 65 to 80 confirmed many previous conjectures and revealed many interesting findings. Our results emphasise the importance of education, Income, activities, drinking, and chronic diseases for cognitive decline. While some factors are unavoidable, others may not, such as lowering BMI through exercise or reducing alcohol consumption. Therefore, to prevent or delay cognitive decline in the elderly, interventions to prevent it may be an effective strategy.

Furthermore, according to our analysis, other European countries have similar cognitive abilities to people over 65, except for Italy and Spain. Nevertheless, they all go through two stages (early and late) of cognitive decline, and the decline will accelerate in the second stage (around 75). While the differences between Italy, Spain and other Europe countries are significant, further research is needed to discover the reason we find the link between country and cognitive decline. We believe this kind of analysis could be of considerable interest to cognitive patients since it allows them to develop a plan in accordance with early prevention. Based on the predictive R^2 we obtained, we believe that many other factors affect cognitive function in older adults, especially factors for explaining the reduction of cognitive functions after 80 years old, which is an exciting and worthwhile topic to explore in future research.

References

[1] Liu B and Müller H G. Estimating derivatives for samples of sparsely observed functions, with application to online auction dynamics. *Journal of the American Statistical Association*, 2009.

[2] Paul B Baltes, Ursula M Staudinger, and Ulman Lindenberger. Lifespan psychology: Theory and application to intellectual functioning. *Annual review of psychology*, 50:471–507, 1999.

[3] Deborah E Barnes and Kristine Yaffe. The projected effect of risk factor reduction on alzheimer's disease prevalence. *The Lancet Neurology*, 10(9):819–828, 2011.

[4] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114:24–31, 2016.

[5] Bennett and Derrick A. How can i deal with missing data in my study? *Australian and New Zealand journal of public health*, 25(5):464–469, 2001.

[6] A. Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 1. release version: 8.0.0. *SHARE-ERIC. Data set.*, 2022.

[7] A. Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 2. release version: 8.0.0. *SHARE-ERIC. Data set.*, 2022.

[8] A. Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 4. release version: 8.0.0. *SHARE-ERIC. Data set.*, 2022.

[9] A. Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 5. release version: 8.0.0. *SHARE-ERIC. Data set.*, 2022.

[10] A. Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 6. release version: 8.0.0. *SHARE-ERIC. Data set.*, 2022.

[11] A. Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 7. release version: 8.0.0. *SHARE-ERIC. Data set.*, 2022.

[12] A. Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 8. release version: 8.0.0. *SHARE-ERIC. Data set.*, 2022.

[13] Axel Börsch-Supan, Martina Brandt, Christian Hunkler, Thorsten Kneip, Julie Korbmacher, Frederic Malter, Barbara Schaan, Stephanie Stuck, and Sabrina Zuber. Data resource profile: the survey of health, ageing and retirement in europe (share). *International journal of epidemiology*, 42(4):992–1001, 2013.

[14] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[15] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[16] Michael W Browne. Cross-validation methods. *Journal of mathematical psychology*, 44(1):108–132, 2000.

[17] Jason Brownlee. A gentle introduction to k-fold cross-validation. *Machine learning mastery*, 2019, 2018.

[18] Edward Joseph Caruana, Marius Roman, Jules Hernández-Sánchez, and Piergiorgio Solli. Longitudinal studies. *Journal of thoracic disease*, 7(11):E537, 2015.

[19] Kehui Chen, Xiaoke Zhang, Alexander Petersen, and Hans-Georg Müller. Quantifying infinite-dimensional data: Functional data analysis in action. *Statistics in Biosciences*, 9(2):582–604, 2017.

[20] Castro P E, Lawton W H, and Sylvestre E A. Principal modes of variation for processes with continuous sample curves. *Technometrics*, 1986.

[21] Yao F, Müller H G, and et al. Clifford A J. Shrinkage estimation for functional principal component scores with application to the population kinetics of plasma folate. *Biometrics*, 2003.

[22] Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications*. Routledge, 2018.

[23] Frédéric Ferraty and Philippe Vieu. *Nonparametric functional data analysis: theory and practice*, volume 76. Springer, 2006.

[24] Staniswalis J G and Lee J J. Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 1998.

[25] David S Geldmacher and Peter J Whitehouse. Evaluation of dementia. *New England Journal of Medicine*, 335(5):330–336, 1996.

[26] Peter Hall, Hans-Georg Müller, and Fang Yao. Modelling sparse generalized longitudinal observations with latent gaussian processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):703–723, 2008.

[27] Kyunghee Han, Pantelis Z Hadjipantelis, Jane-Ling Wang, Michael S Kramer, Seungmi Yang, Richard M Martin, and Hans-Georg Müller. Functional principal component analysis for identifying multivariate patterns and archetypes of growth, and their association with long-term cognitive development. *PloS one*, 13(11):e0207073, 2018.

[28] Lajos Horváth and Piotr Kokoszka. *Inference for functional data with applications*, volume 200. Springer Science & Business Media, 2012.

[29] Fan J and Gijbels I. Local polynomial modelling and its applications. *Routledge*, 2018.

[30] Reka Karuppusami, Belavendra Antonisamy, and Prasanna S Premkumar. Functional principal component analysis for identifying the child growth pattern using longitudinal birth cohort data. *BMC Medical Research Methodology*, 22(1):1–10, 2022.

[31] Kim Kiely. Cognitive function. *Encyclopedia of Quality of Life and Well-Being Research. Springer*, 2014.

[32] Max Kuhn, Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, R Core Team, et al. Package 'caret'. *The R Journal*, 223:7, 2020.

[33] Paul Madley-Dowd, Rachael Hughes, Kate Tilling, and Jon Heron. The proportion of missing data should not be used to guide decisions on multiple imputation. *Journal of clinical epidemiology*, 110:63–73, 2019.

[34] Petra Maresova, Blanka Klimova, Michal Novotny, and Kamil Kuca. Alzheimer's and parkinson's diseases: Expected economic impact on europe—a call for a uniform european strategy. *Journal of Alzheimer's Disease*, 54(3):1123–1133, 2016.

[35] Stephen Milborrow and Maintainer Stephen Milborrow. Package 'rpart. plot'. *Plot'rpart'Models: An Enhanced Version of'plot. rpart*, 2019.

[36] Hans-Georg Müller. Peter hall, functional data analysis and random objects. *The Annals of Statistics*, 44(5):1867–1887, 2016.

[37] DL Murman. Theimpactofageoncognition. In *Seminars in Hearing*, volume 36, pages 111–121, 2015.

[38] HI Niu, F Álvarez-Álvarez, and I Guillén-Grima. Aguinaga-ontoso. prevalence and incidence of alzheimer's disease in europe: A meta-analysis. *Neurología*, 32(8):523–532, 2017.

[39] Sam Norton, Fiona E Matthews, Deborah E Barnes, Kristine Yaffe, and Carol Brayne. Potential for primary prevention of alzheimer's disease: an analysis of population-based data. *The Lancet Neurology*, 13(8):788–794, 2014.

[40] Suggests RColorBrewer and Maintainer Andy Liaw. Package 'randomforest'. *University of California, Berkeley: Berkeley, CA, USA*, 2018.

[41] Victor Francisco Rodriguez-Galiano, Bardan Ghimire, John Rogan, Mario Chica-Olmo, and Juan Pedro Rigol-Sanchez. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS journal of photogrammetry and remote sensing*, 67:93–104, 2012.

[42] Donald B Rubin. Multiple imputation for nonresponse in surveys. hoboken, 1987.

[43] Joseph L Schafer. Multiple imputation: a primer. *Statistical methods in medical research*, 8(1):3–15, 1999.

[44] Haolun Shi, Jianghu Dong, Liangliang Wang, and Jiguo Cao. Functional principal component analysis for longitudinal data with informative dropout. *Statistics in Medicine*, 40(3):712–724, 2021.

[45] Joan G Staniswalis and J Jack Lee. Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 93(444):1403–1418, 1998.

[46] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1):1–21, 2007.

[47] Steinbach M TAN PN and Vipin Kumar. Introduction to data mining, 2014.

[48] Terry Therneau, Beth Atkinson, Brian Ripley, and Maintainer Brian Ripley. Package 'rpart'. *Available online: cran. ma. ic. ac. uk/web/packages/rpart/rpart. pdf (accessed on 20 April 2016)*, 2015.

[49] Stef Van Buuren, Jaap PL Brand, Catharina GM Groothuis-Oudshoorn, and Donald B Rubin. Fully conditional specification in multivariate imputation. *Journal of statistical computation and simulation*, 76(12):1049–1064, 2006.

[50] Anders Wimo, Linus Jönsson, John Bond, Martin Prince, Bengt Winblad, and Alzheimer Disease International. The worldwide economic impact of dementia 2010. *Alzheimer's & dementia*, 9(1):1–11, 2013.

[51] Bengt Winblad, Philippe Amouyel, Sandrine Andrieu, Clive Ballard, Carol Brayne, Henry Brodaty, Angel Cedazo-Minguez, Bruno Dubois, David Edvardsson, Howard Feldman, et al. Defeating alzheimer's disease and other dementias: a priority for european science and society. *The Lancet Neurology*, 15(5):455–532, 2016.

[52] Brita Askeland Winje, Jo Røislien, Eli Saastad, Jorid Eide, Christopher Finne Riley, Babill Stray-Pedersen, and J Frederik Frøen. Wavelet principal component analysis of fetal movement counting data preceding hospital examinations due to decreased fetal movement: a prospective cohort study. *BMC pregnancy and childbirth*, 13(1):1–11, 2013.

[53] Hans-Ulrich Wittchen, Frank Jacobi, Jürgen Rehm, Anders Gustavsson, Mikael Svensson, Bengt Jönsson, Jes Olesen, Christer Allgulander, Jordi Alonso, Carlo Faravelli, et al. The size and burden of mental disorders and other disorders of the brain in europe 2010. *European neuropsychopharmacology*, 21(9):655–679, 2011.

[54] Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American statistical association*, 100(470):577–590, 2005.

[55] John Zeisel, Kirsty Bennett, and Richard Fleming. World alzheimer report 2020: Design, dignity, dementia: Dementia-related design and the built environment. 2020.

[56] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

Appendix A

Plots for predicting the first two FPC scores from the categorical variables

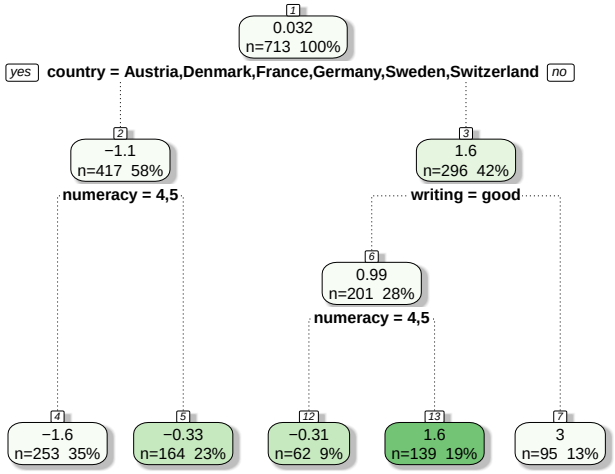

Figure A.1: Pruned Tree based method for predicting FPC1 from the categorical variables

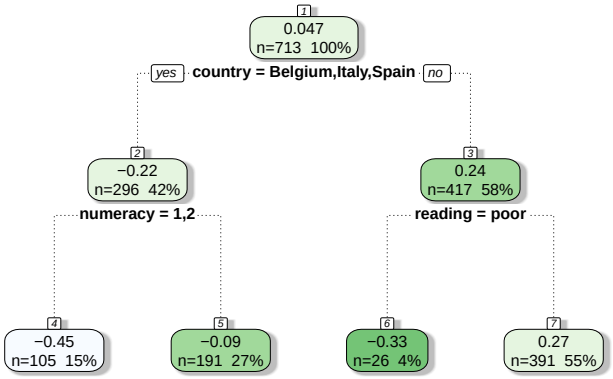

Figure A.2: Pruned Tree based method for predicting FPC2 from the categorical variables

Appendix B

R Code

B.1 Data Management for Wave8

B.2 Load data and data clearing

```
#Note this is only the data management for wave 8, the data management for wave1-7 data is similar to this procedure.

#clean function
cleanup <- function(myvar = Alcohol$ALQ110,  mymin = 1,  mymax = 2, exclude = c(999)){
  ID1 <- (myvar < mymin)        # ids with less than min values
  ID2 <- (myvar > mymax)        # ids with greater than max values
  if (length(exclude) > 0)      # ids with strange values
  { ID3 <- myvar%in%exclude }

  # set all the above three values to be NA.
  RmID <- ((ID1 + ID2 + ID3) > 0)
  myvar[RmID] = NA
  return(myvar)
}

data=read.csv("wave1_7.csv")
library(haven)
cr <- read_dta('sharew8_rel8-0-0_cv_r.dta')
cf <- read_dta('sharew8_rel8-0-0_cf.dta')
gv_health<-read_dta('sharew8_rel8-0-0_gv_health.dta')
names(cr)[names(cr)=="hhid8"]="hhid"
names(cr)[names(cr)=="mergeidp8"]="mergeidp"
names(cr)[names(cr)=="coupleid8"]="coupleid"
names(gv_health)[names(gv_health)=="hhid8"]="hhid"
names(gv_health)[names(gv_health)=="mergeidp8"]="mergeidp"
names(gv_health)[names(gv_health)=="coupleid8"]="coupleid"
names(gv_health)[names(gv_health)=="cf008tot"]="recall_1"
names(gv_health)[names(gv_health)=="cf016tot"]="recall_2"
cr$wave <- 8
gv_health$wave <- 8
inter<-c(intersect(colnames(cr),  colnames(data)))
inter2<-c(intersect(colnames(gv_health),  colnames(data)))

wave8cr<-cr[,c(inter)]
wave8gv_health<-gv_health[,c(inter2)]
wave81 <- merge(wave8cr,wave8gv_health,by = intersect(names(wave8cr), names(wave8gv_health)),all = TRUE)

ac=read_dta('sharew8_rel8-0-0_ac.dta')
names(ac)[names(ac)=="hhid8"]="hhid"
```

```
names(ac)[names(ac)=="mergeidp8"]="mergeidp"
names(ac)[names(ac)=="coupleid8"]="coupleid"
ac$wave=8
ac$ac002d1=NA
ac$ac002d2=NA
ac$ac002d3=NA
ac$ac002d4=NA
ac$ac002d5=NA
ac$ac002d6=NA
ac$ac002d7=NA
ac$ac002dno=NA
selectac=c("mergeid","hhid","coupleid","wave","ac002d1","ac002d2","
    ac002d3","ac002d4","ac002d5",
            "ac002d6","ac002d7","ac002dno")
wave8ac<-ac[,selectac]
wave81new <- merge(wave81,wave8ac,by = intersect(names(wave81), names(
    wave8ac)),all = TRUE)
#write.csv(wave81,file = "Desktop/2022\ share/wave81.csv",row.names = F)
##############
###############part 2
dn<-read_dta('sharew8_rel8-0-0_dn.dta')
names(dn)[names(dn)=="hhid8"]="hhid"
names(dn)[names(dn)=="mergeidp8"]="mergeidp"
names(dn)[names(dn)=="coupleid8"]="coupleid"
names(dn)[names(dn)=="dn014_"]="mar_stat"
names(dn)[names(dn)=="dn026_1"]="mother_alive"
names(dn)[names(dn)=="dn026_2"]="father_alive"

dn <- dn %>% rowwise() %>%
  mutate(siblings_alive = sum(dn036_,dn037_))
dn$wave<-8
selectdn=c("mergeid","hhid","coupleid","dn002_","dn003_","dn004_",
           "dn007_","mar_stat","mother_alive","father_alive","dn037_","
               siblings_alive","wave")
wave8dn<-dn[,selectdn]

ch<-read_dta('sharew8_rel8-0-0_ch.dta')
names(ch)[names(ch)=="hhid8"]="hhid"
names(ch)[names(ch)=="mergeidp8"]="mergeidp"
names(ch)[names(ch)=="coupleid8"]="coupleid"

ch$ch007_1[is.na(ch$ch007_1)]=0
ch$ch007_2[is.na(ch$ch007_2)]=0
ch$ch007_3[is.na(ch$ch007_3)]=0
ch$ch007_4[is.na(ch$ch007_4)]=0
ch$ch007_5[is.na(ch$ch007_5)]=0
ch$ch007_6[is.na(ch$ch007_6)]=0
ch$ch007_7[is.na(ch$ch007_7)]=0
ch$ch007_8[is.na(ch$ch007_8)]=0
ch$ch007_9[is.na(ch$ch007_9)]=0
ch$ch007_10[is.na(ch$ch007_10)]=0
ch$ch007_11[is.na(ch$ch007_11)]=0
ch$ch007_12[is.na(ch$ch007_12)]=0
ch$ch007_13[is.na(ch$ch007_13)]=0
```

```
ch$ch007_14[is.na(ch$ch007_14)]=0
ch$ch007_15[is.na(ch$ch007_15)]=0
ch$ch007_16[is.na(ch$ch007_16)]=0

ch$ch007_hh <- ifelse(ch$ch007_1 ==1| ch$ch007_2==1
                              | ch$ch007_3==1| ch$ch007_4==1
                              | ch$ch007_5==1| ch$ch007_6==1
                              | ch$ch007_7==1| ch$ch007_8==1
                              | ch$ch007_9==1| ch$ch007_10==1
                              |ch$ch007_11==1|ch$ch007_12==1
                              |ch$ch007_13==1|ch$ch007_14==1
                              |ch$ch007_15==1|ch$ch007_16==1|
                                ch$ch007_1 ==2| ch$ch007_2==2
                              | ch$ch007_3==2| ch$ch007_4==2
                              | ch$ch007_5==2| ch$ch007_6==2
                              | ch$ch007_7==2| ch$ch007_8==2
                              | ch$ch007_9==2| ch$ch007_10==2
                              |ch$ch007_11==2|ch$ch007_12==2
                              |ch$ch007_13==2|ch$ch007_14==2
                              |ch$ch007_15==2|ch$ch007_16==2,1,
                              ifelse(ch$ch007_1==0&ch$ch007_2==0
                                     &ch$ch007_3==0&ch$ch007_4==0
                                     &ch$ch007_5==0&ch$ch007_6==0
                                     &ch$ch007_7==0&ch$ch007_8==0&
                                        ch$ch007_9==0&ch$ch007_10==0&
                                        ch$ch007_11==0&ch$ch007_12==0&
                                        ch$ch007_13==0&ch$ch007_14==0&
                                        ch$ch007_15==0&ch$ch007_16==0,NA,5))
table(ch$ch007_hh,useNA = "always")

ch$ch007_km <- ifelse(ch$ch007_1 ==3| ch$ch007_2==3
                              | ch$ch007_3==3| ch$ch007_4==3
                              | ch$ch007_5==3| ch$ch007_6==3
                              | ch$ch007_7==3| ch$ch007_8==3
                              | ch$ch007_9==3| ch$ch007_10==3
                              |ch$ch007_11==3|ch$ch007_12==3
                              |ch$ch007_13==3|ch$ch007_14==3
                              |ch$ch007_15==3|ch$ch007_16==3,1,
                              ifelse(ch$ch007_1==0&ch$ch007_2==0
                                     &ch$ch007_3==0&ch$ch007_4==0
                                     &ch$ch007_5==0&ch$ch007_6==0
                                     &ch$ch007_7==0&ch$ch007_8==0&
                                        ch$ch007_9==0&ch$ch007_10==0&
                                        ch$ch007_11==0&ch$ch007_12==0&
                                        ch$ch007_13==0&ch$ch007_14==0&
                                        ch$ch007_15==0&ch$ch007_16==0,NA,5))
table(ch$ch007_km)

inter<-c(intersect(colnames(dn), colnames(data)))
inter
inter2<-c(intersect(colnames(ch), colnames(data)))
inter2

wave8dn<-dn[,c(inter)]
```

```
wave8ch<-ch[,c(inter2)]
wave82 <- merge(wave8dn,wave8ch,by = intersect(names(wave8dn), names(wave8ch)),all = TRUE)
#write.csv(wave82,file = "Desktop/2022\ share/wave82.csv",row.names = F)

###########
##########part 3
sp<-read_dta('sharew8_rel8-0-0_sp.dta')
names(sp)[names(sp)=="hhid8"]="hhid"
names(sp)[names(sp)=="mergeidp8"]="mergeidp"
names(sp)[names(sp)=="coupleid8"]="coupleid"
sp$wave<-8
selectsp<-c("mergeid","hhid","coupleid","wave","sp002_","sp003_1","sp003_2","sp003_3","sp008_",
            "sp009_1","sp009_2","sp009_3")
wave8sp<-sp[,selectsp]


ph <-read_dta('sharew8_rel8-0-0_ph.dta')
ph$wave=8
names(ph)[names(ph)=="hhid8"]="hhid"
names(ph)[names(ph)=="mergeidp8"]="mergeidp"
names(ph)[names(ph)=="coupleid8"]="coupleid"
selectph=c("mergeid","hhid","coupleid","wave",'ph003_',paste0("ph006d",1:6),paste0("ph006d",10:14))
wave8ph<-ph[,selectph]


mh <-read_dta('sharew8_rel8-0-0_mh.dta')
mh$wave=8
names(mh)[names(mh)=="hhid8"]="hhid"
names(mh)[names(mh)=="mergeidp8"]="mergeidp"
names(mh)[names(mh)=="coupleid8"]="coupleid"
selectmh=c("mergeid","hhid","coupleid","wave","mh002_",
           "mh003_","mh004_","mh005_","mh007_","mh008_",'mh010_',"mh011_",
           "mh013_","mh014_","mh015_","mh016_","mh017_")

wave8mh<-mh[,selectmh]
wave8mh$mh002_<-ifelse(wave8mh$mh002_==1,1,ifelse(wave8mh$mh002_==5,0,NA))
wave8mh$mh003_<-ifelse(wave8mh$mh003_==1,0,ifelse(wave8mh$mh003_==2,1,NA))
wave8mh$mh004_<-ifelse(wave8mh$mh004_==1,1,ifelse(wave8mh$mh004_==2,0,NA))
wave8mh$mh005_<-ifelse(wave8mh$mh005_==2,0,ifelse(wave8mh$mh005_==1,1,ifelse(wave8mh$mh005_==3,1,NA)))

wave8mh$mh007_<-ifelse(wave8mh$mh007_==1,1,ifelse(wave8mh$mh007_==2,0,NA))
wave8mh$mh008_<-ifelse(wave8mh$mh008_==2,0,ifelse(wave8mh$mh008_==1,1,ifelse(wave8mh$mh008_==3,1,NA)))
wave8mh$mh010_<-ifelse(wave8mh$mh010_==1,1,ifelse(wave8mh$mh010_==2,0,NA))
```

```
wave8mh$mh011_<-ifelse(wave8mh$mh011_==1,1,ifelse(wave8mh$mh011_==2,0,
    ifelse(wave8mh$mh011_==3,0,NA)))
wave8mh$mh013_<-ifelse(wave8mh$mh013_==1,1,ifelse(wave8mh$mh013_==5,0,NA
    ))
wave8mh$mh014_<-ifelse(wave8mh$mh014_==1,1,ifelse(wave8mh$mh014_==2,0,NA
    ))
wave8mh$mh015_<-ifelse(wave8mh$mh015_==1,1,ifelse(wave8mh$mh015_==2,0,NA
    ))
wave8mh$euro10=ifelse(wave8mh$mh014_==1|wave8mh$mh015_==1,1,0)
wave8mh$mh016_=ifelse(wave8mh$mh016_==1,1,ifelse(wave8mh$mh016_==2,0,NA)
    )
wave8mh$mh017_=ifelse(wave8mh$mh017_==1,1,ifelse(wave8mh$mh017_==5,0,NA)
    )

names(wave8mh)[names(wave8mh)=="mh002_"]="euro1"
names(wave8mh)[names(wave8mh)=="mh003_"]="euro2"
names(wave8mh)[names(wave8mh)=="mh004_"]="euro3"
names(wave8mh)[names(wave8mh)=="mh005_"]="euro4"
names(wave8mh)[names(wave8mh)=="mh007_"]="euro5"
names(wave8mh)[names(wave8mh)=="mh008_"]="euro6"
names(wave8mh)[names(wave8mh)=="mh010_"]="euro7"
names(wave8mh)[names(wave8mh)=="mh011_"]="euro8"
names(wave8mh)[names(wave8mh)=="mh013_"]="euro9"
names(wave8mh)[names(wave8mh)=="mh016_"]="euro11"
names(wave8mh)[names(wave8mh)=="mh017_"]="euro12"

wave8mh=subset(wave8mh,select = -c(mh014_,mh015_))
library(dplyr)
wave8mh <- wave8mh %>% rowwise() %>%
  mutate(eurod = sum(c_across(euro1:euro10)))

hc<-read_dta('sharew8_rel8-0-0_hc.dta')
hc$wave=8
names(hc)[names(hc)=="hhid8"]="hhid"
names(hc)[names(hc)=="mergeidp8"]="mergeidp"
names(hc)[names(hc)=="coupleid8"]="coupleid"
names(hc)[names(hc)=="hc602_"]="hc002_"
selecthc<-c("mergeid","hhid","coupleid","wave","hc002_","hc012_","hc029_
    ")
wave8hc<-hc[,selecthc]

ph<-read_dta('sharew8_rel8-0-0_ph.dta')
ph$wave=8
names(ph)[names(ph)=="hhid8"]="hhid"
names(ph)[names(ph)=="mergeidp8"]="mergeidp"
names(ph)[names(ph)=="coupleid8"]="coupleid"
selectph=c("mergeid","hhid","coupleid","wave","ph049d1","ph049d2","
    ph049d3","ph049d4","ph049d5",
           "ph049d8","ph049d9','ph049d10","ph049d11","ph049d13","ph048d1
               ","ph048d2","ph048d3","ph048d4","ph048d5",
           "ph048d6","ph048d8","ph048d10")
wave8ph<-ph[,selectph]
wave8ph$ph049d1=cleanup(wave8ph$ph049d1,0,1)
wave8ph$ph049d3=cleanup(wave8ph$ph049d3,0,1)
```

```
wave8ph$ph049d4=cleanup(wave8ph$ph049d4,0,1)
wave8ph$ph049d2=cleanup(wave8ph$ph049d2,0,1)
wave8ph$ph049d5=cleanup(wave8ph$ph049d5,0,1)
wave8ph$ph049d10=cleanup(wave8ph$ph049d10,0,1)
wave8ph$ph049d11=cleanup(wave8ph$ph049d11,0,1)
wave8ph$ph049d13=cleanup(wave8ph$ph049d13,0,1)
wave8ph$ph049d9=cleanup(wave8ph$ph049d9,0,1)
wave8ph$ph049d8=cleanup(wave8ph$ph049d8,0,1)
wave8ph$ph048d1=cleanup(wave8ph$ph048d1,0,1)
wave8ph$ph048d4=cleanup(wave8ph$ph048d4,0,1)
wave8ph$ph048d5=cleanup(wave8ph$ph048d5,0,1)
wave8ph$ph048d2=cleanup(wave8ph$ph048d2,0,1)
wave8ph$ph048d3=cleanup(wave8ph$ph048d3,0,1)
wave8ph$ph048d6=cleanup(wave8ph$ph048d6,0,1)
wave8ph$ph048d8=cleanup(wave8ph$ph048d8,0,1)
wave8ph$ph048d10=cleanup(wave8ph$ph048d10,0,1)

library(dplyr)
wave8ph <- wave8ph %>% rowwise() %>%
  mutate(adlwa = sum(c_across(ph049d1:ph049d4)))
table(wave8ph$adlwa,useNA = "always")

wave8ph <- wave8ph %>% rowwise() %>%
  mutate(adla = sum(c_across(ph049d1:ph049d5)))


wave8ph <- wave8ph %>% rowwise() %>%
  mutate(iadla = sum(c_across(ph049d10:ph049d13)))


wave8ph <- wave8ph %>% rowwise() %>%
  mutate(iadlza = sum(c_across(ph049d8:ph049d10)))

wave8ph <- wave8ph %>% rowwise() %>%
  mutate(mobilityind = sum(ph048d1,ph049d2,ph048d4,ph048d5))
table(wave8ph$mobilityind,useNA = "always")


wave8ph <- wave8ph %>% rowwise() %>%
  mutate(lgmuscle = sum(ph048d2,ph048d3,ph048d6,ph048d8))


wave8ph <- wave8ph %>% rowwise() %>%
  mutate(grossmotor = sum(ph048d1,ph049d2,ph048d5,ph049d3))


wave8ph <- wave8ph %>% rowwise() %>%
  mutate(finemotor = sum(ph048d10,ph049d4,ph049d1))

select=c("mergeid","hhid","coupleid","wave","adlwa","adla","iadla",
         "iadlza","mobilityind","lgmuscle","grossmotor",'finemotor')

wave8ph=wave8ph[,select]

```

```
M1<-merge(wave8hc,wave8sp,by = intersect(names(wave8hc), names(wave8sp)),all = TRUE)
M2=merge(M1,wave8ph,by = intersect(names(M1), names(wave8ph)),all = TRUE)
wave83=merge(M2,wave8mh,by = intersect(names(M2), names(wave8mh)),all = TRUE)

wave83$hc002_=cleanup(wave83$hc002_,0,365)
wave83$hc012_=cleanup(wave83$hc012_,0,5)
wave83$hc029_=cleanup(wave83$hc029_,1,5)
wave83$sp002_=cleanup(wave83$sp002_,1,5)
wave83$sp003_1=cleanup(wave83$sp003_1,1,96)
wave83$sp008_=cleanup(wave83$sp008_,1,5)
wave83$sp009_1=cleanup(wave83$sp009_1,1,96)
wave83$sp009_2=cleanup(wave83$sp009_2,1,96)

inter<-c(intersect(colnames(wave83), colnames(data)))
inter
#write.csv(wave83,file = "wave83.csv",row.names = F)

##########
#########Part 4
br<-read_dta('sharew8_rel8-0-0_br.dta')
names(br)[names(br)=="hhid8"]="hhid"
names(br)[names(br)=="mergeidp8"]="mergeidp"
names(br)[names(br)=="coupleid8"]="coupleid"
br$wave <- 8
br$br010_mod<-NA
names(br)[names(br)=="br002_"]="smoking"
names(br)[names(br)=="br001_"]="ever_smoked"
selectbr<-c("mergeid","hhid","coupleid","wave",
            "smoking","ever_smoked","br010_mod","br015_")
wave8br<-br[,selectbr]
wave83new=merge(wave83,wave8br,by = intersect(names(wave83), names(wave8br)),all = TRUE)
#write.csv(wave83new,file = "wave83.csv",row.names = F)

##########
#########Part 5
ep<-read_dta('sharew8_rel8-0-0_ep.dta')
names(ep)[names(ep)=="hhid8"]="hhid"
names(ep)[names(ep)=="mergeidp8"]="mergeidp"
names(ep)[names(ep)=="coupleid8"]="coupleid"
ep$wave <- 8
ep$ep011_=NA
selectep<-c("mergeid","hhid","coupleid",
            "wave","ep005_","ep009_","ep011_","ep013_","ep026_","ep036_")
wave8ep<-ep[,selectep]
wave83new2=merge(wave83new,wave8ep,by = intersect(names(wave83new), names(wave8ep)),all = TRUE)
#library(sjlabelled)
#a=as.list(get_label(ep))
#write.csv(wave83new2,file = "wave83.csv",row.names = F)
```

```
ph<-read_dta('sharew8_rel8-0-0_ph.dta')
ph$wave=8
names(ph)[names(ph)=="hhid8"]="hhid"
names(ph)[names(ph)=="mergeidp8"]="mergeidp"
names(ph)[names(ph)=="coupleid8"]="coupleid"
inter=c(intersect(colnames(ph),  colnames(data)))
inter
wave8phnew=ph[,inter]
wave8phnew <- wave8phnew %>% rowwise() %>%
  mutate(chronic_mod = sum(c_across(ph006d1:ph006d14)))

wave83new3=merge(wave83new2,wave8phnew,by = intersect(names(wave83new2),
    names(wave8phnew)),all = TRUE)
#write.csv(wave83new3,file = "wave83.csv",row.names = F)


##########
#########Part 6

co<-read_dta('sharew8_rel8-0-0_co.dta')
co$wave=8
names(co)[names(co)=="hhid8"]="hhid"
names(co)[names(co)=="mergeidp8"]="mergeidp"
names(co)[names(co)=="coupleid8"]="coupleid"
selectco<-c("mergeid","hhid","coupleid","wave","co007_")
wave8co<-co[,selectco]

gv_isced<-read_dta('sharew8_rel8-0-0_gv_isced.dta')
names(gv_isced)[names(gv_isced)=="hhid8"]="hhid"
names(gv_isced)[names(gv_isced)=="mergeidp8"]="mergeidp"
names(gv_isced)[names(gv_isced)=="coupleid8"]="coupleid"
gv_isced$wave<-8
selectr<-c("mergeid","hhid","coupleid","wave","isced1997_r")
wave8gv_isced<-gv_isced[,selectr]

iv<-read_dta('sharew8_rel8-0-0_iv.dta')
names(iv)[names(iv)=="hhid8"]="hhid"
names(iv)[names(iv)=="mergeidp8"]="mergeidp"
names(iv)[names(iv)=="coupleid8"]="coupleid"
iv$wave<-8
selectiv<-c("mergeid","hhid","coupleid","wave","iv009_")
wave8iv<-iv[,selectiv]


im <- read_dta("sharew8_rel8-0-0_gv_imputations.dta")
im$wave=8
names(im)[names(im)=="hhid8"]="hhid"
names(im)[names(im)=="mergeidp8"]="mergeidp"
names(im)[names(im)=="coupleid8"]="coupleid"
colSums(is.na(im))
wave8im<-im[,c("mergeid","hhid","coupleid","wave","implicat","thinc","
    sphus","mstat","nchild","chronic",
```

```
                          "esmoked","eurod","doctor","rhfo","ghto","fdistress"
                              ,"yedu")]
wave8im$fdistress[wave8im$fdistress==-99]=NA
wave8im$esmoked[wave8im$esmoked==-99]=NA
wave8im$ghto[wave8im$ghto==-99]=NA
wave8im=wave8im %>% group_by(mergeid) %>%
  dplyr::summarise(thinc_m=mean(thinc),
                        nchild_m=as.integer(mean(nchild)),chronic_m=as.
                            integer(mean(chronic)))
                        doctor_m=as.integer(mean(doctor)),rhfo_m=as.integer(
                            mean(rhfo)),ghto_m=as.integer(mean(ghto)),
                        fdistress_m=as.integer(mean(fdistress)),yedu_m=mean(
                            yedu))
wave8im <- wave8im %>%
  mutate(
    sphus_m = as.factor(sphus_m),
    mstat_m = as.factor(mstat_m),
    esmoked_m = as.factor(esmoked_m),
    rhfo_m = as.factor(rhfo_m),
    ghto_m = as.factor(ghto_m),
    fdistress_m=as.factor(fdistress_m)
  )
wave8im$wave=8

m1=merge(wave8co,wave8gv_isced,by = intersect(names(wave8co), names(
    wave8gv_isced)),all = TRUE)
m2=merge(m1,wave8iv,by = intersect(names(m1), names(wave8iv)),all = TRUE
    )
wave84=merge(m2,wave8im,by = intersect(names(m2), names(wave8im)),all =
    TRUE)

#write.csv(wave84,file = "wave84.csv",row.names = F)


#######
########Merge
MM1<-merge(wave81new,wave82,by = intersect(names(wave81new), names(
    wave82)),all = TRUE)
MM2<-merge(MM1,wave83new3,by = intersect(names(MM1), names(wave83new3)),
    all = TRUE)
wave8<-merge(MM2,wave84,by = intersect(names(MM2), names(wave84)),all =
    TRUE)

#write.csv(wave8,file = "wave8.csv",row.names = F)
wave1_8=merge(data,wave8,by = intersect(names(data), names(wave8)),all =
    TRUE)
```

B.3 Load data and data clearing

```
library("RColorBrewer")
library("rattle")
library(fdapace)
```

```r
library("ggplot2")
library(caret)
library(tidyr)
library(glmnet)
library(viridis)
library(rpart)
library(rpart.plot)
library(randomForest)
library(cvTools)

#read data
data=read.csv("wave1_8_5.24.2022.csv")

#set the color
mycolor<-brewer.pal(9, "Set1")
mycolor.alpha <- scales::alpha(mycolor, 80/100)
mygray <- scales::alpha("gray", 40/100)
myeffectcolor<-viridis_pal(option = "C")(3)[c(1, 3, 2)]
myeffectcolor[2] <- "gray"
mycolor.alpha <- scales::alpha(mycolor, 80/100)


# set all the above three values to be NA.
  RmID <- ((ID1 + ID2 + ID3) > 0)
  myvar[RmID] = NA
  return(myvar)


#Only keep subjects without missing cogintive functions
subdata <- subset(data, (recall_1>=0)&(recall_2>=0))

#Only keep the people age 50 and older
baseline <- subset(subdata, wave == 1)
ID1 <- baseline$mergeid[which(baseline$age >=50)]
subdata <- subset(subdata, mergeid%in%ID1)

#Scale my cognitive functions
subdata[, c("recall_1.scale", "recall_2.scale")] <- scale(subdata[, c("
    recall_1", "recall_2")])

# Only keep those with 7 measurements
myt <- table(subdata$mergeid)
mytt <- table(myt)
length(myt)
print(mytt/length(myt)*100)

myID <- names(myt)[myt >= 7]
subdata1 <- subset(subdata, mergeid%in%myID)
dim(subdata1)
#22876     54

#only keep the age bw. 60 and 85 in wave1
ID <- baseline$mergeid[which((baseline$age_int <= 80)&(baseline$age_int
    >= 65))]
```

```r

#exclude the age over 95 in the last wave(the sample size is too small, outlier)
wave8 <- subset(subdata, wave == 8)
ID2 <- wave8$mergeid[which(wave8$age_int < 95)]

subdata2 <- subset(subdata1, mergeid%in%ID)
subdata2 <- subset(subdata2, mergeid%in%ID2)
dim(subdata2)
#6608    55

# baseline covariates
base02=subset(subdata2,select = -c(recall_1,recall_2))
base2 <- subset(base02, wave == 1)

colnames(base2)[2:53] <- paste("base.", colnames(base2)[2:53], sep = "")
}
```

B.4 FPCA

```r
#FPCA
BFPCA2 <- MakeFPCAInputs(IDs = subdata2$mergeid, subdata2$age_int, subdata2$recall_2.scale)
FPCA2 <- FPCA(BFPCA2$Ly, BFPCA2$Lt)

plot(FPCA2)


#Figure 4.1 code
par(mfrow = c(1, 3))
#time            #mean
plot(FPCA2$workGrid, FPCA2$mu, type = "l", xlab = "age",
     ylab = "cognitive function", lwd = 2, col = mycolor[1], main = "Mean Function")
CreateScreePlot(FPCA2)
plot(FPCA2$workGrid, FPCA2$phi[,1], type = "l", col = mycolor[1],
     ylim = c(-0.4, 0.4), lwd = 2, xlab = "age", ylab = "cognitive function", lty = 1,
     main = "First 3 Eigenfunctions")
lines(FPCA2$workGrid, FPCA2$phi[,2], col = mycolor[4], lwd = 2, lty = 2)
lines(FPCA2$workGrid, FPCA2$phi[,3], col = mycolor[8], lwd = 2, lty = 3)
abline(h = 0, col = "gray")
legend("top", paste(c("First", "Second", "Third"), "Egienfunction"),
       lwd=2, lty = 1:3, col = mycolor[c(1, 4, 8)], bty = "n")



#Figure effect of the First FPC
par(mfrow=c(1,2))
plot(FPCA2$workGrid, FPCA2$mu, type = "l", xlab = "age", ylab = "cognitive function",
     main = "Effect of First Eigenfunction", lwd = 1.5)
```

```r
lines(FPCA2$workGrid, FPCA2$mu - 0.3*FPCA2$phi[,1], col = myeffectcolor[1], lwd = 1.5)
lines(FPCA2$workGrid, FPCA2$mu + 0.3*FPCA2$phi[,1], col = myeffectcolor[3], lwd = 1.5)

plot(FPCA2$workGrid, FPCA2$mu, type = "l", xlab = "age", ylab = "cognitive function",
     main = "Effect of Second Eigenfunction", lwd = 1.5)
lines(FPCA2$workGrid, FPCA2$mu - 0.3*FPCA2$phi[,2], col = myeffectcolor[1], lwd = 1.5)
lines(FPCA2$workGrid, FPCA2$mu + 0.3*FPCA2$phi[,2], col = myeffectcolor[3], lwd = 1.5)


#merge FPC1 and FPC2 with baseline covariates dataset
First2PC2 <- data.frame(cbind(names(BFPCA2$Ly), FPCA2$xiEst[, 1:2]))
colnames(First2PC2) <- c("mergeid", "PC1", "PC2")
First2PC2$PC1 <- as.numeric(as.character(First2PC2$PC1))
First2PC2$PC2 <- as.numeric(as.character(First2PC2$PC2))
PCs2 <- merge(First2PC2, base2, by = "mergeid")
```

B.5 PC score V.s. nine countries

```r
###PC1 for nine countries (Figure 4.4)
ggplot(PCs2, aes(x=country, y=PC1)) +
  geom_boxplot(fill = mycolor, width = 0.6, notch = T)+
  geom_hline(yintercept = 0 , col = "gray") +
  theme(axis.text.x = element_text(face = "bold", angle = 70, size = 8),
        axis.ticks.x = element_blank(),
        axis.text.y = element_text(face = "bold", size = 8),
        panel.background = element_rect(fill = "white", colour = NA),
        panel.border = element_rect(fill = "NA", color = "gray"),
        panel.grid.major = element_line(colour = "gray80", size = 0.25),
        panel.grid.minor = element_line(colour = "gray80", size = 0.25)
  )+
  labs(x="",y="First PC Score")

###PC2 for nine countries (Figure 4.5)
ggplot(PCs2, aes(x=country, y=PC2)) +
  geom_boxplot(fill = mycolor, width = 0.6, notch = T)+
  geom_hline(yintercept = 0 , color = "gray") +
  theme(axis.text.x = element_text(face = "bold", angle = 70, size = 8),
        axis.ticks.x = element_blank(),
        axis.text.y = element_text(face = "bold", size = 8),
        panel.background = element_rect(fill = "white", colour = NA),
        panel.border = element_rect(fill = "NA", color = "gray"),
        panel.grid.major = element_line(colour = "gray80", size = 0.25),
        panel.grid.minor = element_line(colour = "gray80", size = 0.25)
  )+labs(x="",y="Second PC Score")
Median11 <- with(PCs2, tapply(PC1, country, median))##median of PC1 for each country
```

```r
Median22 <- with(PCs2, tapply(PC2, country, median))##median of PC2 for each country


mymedcurves22 <- data.frame(cbind(FPCA2$workGrid,FPCA2$mu+
                                  outer(FPCA2$phi[,1], Median11))) # eigenfunction one times median11


data_long22 <- gather(mymedcurves22, country, value, Austria:Switzerland, factor_key=TRUE)
### Figure 4.2 code
ggplot(data_long22, aes(x=V1, y=value, group = country, color = country))+
  geom_line()+
  scale_color_manual(values=mycolor)  +
  labs(x = "age", y = "cognitive function")

mymedcurves33 <- data.frame(cbind(FPCA2$workGrid,
                                  FPCA2$mu+
                                  # outer(FPCAsparse$phi[,1], Median1)+
                                  outer(FPCA2$phi[,2], Median22)))
data_long33 <- gather(mymedcurves33, country, value, Austria:Switzerland, factor_key=TRUE)


ggplot(data_long33, aes(x=V1, y=value, group = country, color = country))+
  geom_line()+
  scale_color_manual(values=mycolor) +
  labs(x = "age", y = "cognitive function")

 ### Figure 4.3 code
data_long44 <- gather(mymedcurves44, country, value, Austria:Switzerland, factor_key=TRUE)

data_long44$country <- factor(data_long44$country, levels = c("Sweden", "Switzerland", "Denmark",
                                                              "Germany", "France","Austria","Belgium",
                                                              "Italy","Spain"))
pdf("median3.pdf")
ggplot(data_long44, aes(x=V1, y=value, group = country))+
  geom_line(aes(linetype=country, color = country))+
  scale_color_manual(values=c("#000000", "#E69F00", "#56B4E9",
                              "#009E73", "red", "#0072B2", "#D55E00",
```

```
                                        "#CC79A7","#999999"))+
  labs(x = "age", y = "cognitive function")
dev.off()
```

B.6 data management before model fitting

```
#select covariates
select.co<-c("base.bmi", "base.thinc_m", "base.yedu_m","base.maxgrip","base.hhsize","base.recall_1.scale","base.recall_2.scale','base.siblings_alive')
select.dis <- c("base.country", "base.gender", "base.hearing", "base.reading",
                "base.writing", "base.eyesight", "base.ep005", "base.mob",
"base.active2", "base.active", "base.doctorS", "base.depression",
"base.chronic", "base.health", "base.ghelp", "base.rhelp",
"base.esmoked_m", "base.marry", "base.hospital_m",
"base.partnerinhh","base.mother_alive","base.father_alive",
"base.hc029_","base.grossmotor","base.finemotor","base.orienti",
'base.numeracy','base.drink","base.muscle","base.vigorous")

mynewPCs0 <- PCs2
mynewPCs0[select.dis] <- lapply(mynewPCs0[select.dis],factor)

#new dataset with the select coulmn
mynewPCs3=cbind(mynewPCs0[1:3],mynewPCs0[,c(select.dis,select.co)])
mynewPCs3=mynewPCs3[complete.cases(mynewPCs3), ]
```

B.7 Model fitting

B.7.1 Elastic Net

```
# Fitted Model based on minimum Mean Squared Error
set.seed(1000)
myfit3 <- train(
  PC1 ~.-mergeid-PC2, data = mynewPCs3, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneLength = 30
)

get_best_result(myfit3)
mycoef13 <- coef(myfit3$finalModel, myfit3$bestTune$lambda)

mycoef3 <- data.frame(rep(0, length(mycoef13@Dimnames[[1]])), mycoef13@Dimnames[[1]])
mycoef3[mycoef13@i + 1, 1] <- mycoef13@x
colnames(mycoef3) <- c("V1", "V2")

newcoef3 <- mycoef3[order(mycoef3$V1), ]
```

```r
newcoef3$V2 <- factor(newcoef3$V2, level = unique(as.character(newcoef3$V2)) )
newcoef3$color <- myeffectcolor[sign(newcoef3$V1) + 2]

## Figure 4.6 code
ggplot(newcoef3, aes(x=V2, y=V1, fill=color, color = color)) +
  geom_bar(stat="identity")+
  coord_flip()+
  scale_x_discrete(position = "top")+
  scale_fill_manual(name = "",
                    labels = c("Negative Effect", "Positive Effect", "No Effect"),
                    values=myeffectcolor[c(1,3,2)])+
  scale_color_manual(name = "",
                     labels = c("Negative Effect", "Positive Effect", "No Effect"),
                     values=myeffectcolor[c(1,3,2)])+
  theme(#axis.text.x = element_blank(),
    axis.text.x = element_text(face = "bold"),
    axis.text.y = element_text(face = "bold", color = newcoef3$color),
    legend.position="bottom")+
  labs(x = "", y = "", title = "Covariate Effects on the First FPC Scores")
#####################
#######################
set.seed(100000)
myfit23 <- train(
  PC2 ~.-mergeid-PC1, data = mynewPCs3, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneLength = 10
)
mycoef23 <- coef(myfit23$finalModel, myfit23$bestTune$lambda)
mycoef2.23 <- data.frame(rep(0, length(mycoef23@Dimnames[[1]])), mycoef23@Dimnames[[1]])
mycoef2.23[mycoef23@i + 1, 1] <- mycoef23@x
colnames(mycoef2.23) <- c("V1", "V2")

newcoef23 <- mycoef2.23[order(mycoef2.23$V1), ]
newcoef23$V2 <- factor(newcoef23$V2, level =  unique(as.character(newcoef23$V2)))
newcoef23$color <- myeffectcolor[sign(newcoef23$V1) + 2]
#newcoef2[which(sign(newcoef2$V1)==-1),]
#newcoef2[which(sign(newcoef2$V1)==1),]

## Figure 4.7 code
ggplot(newcoef23, aes(x=V2, y=V1, fill=color, color = color)) +
  geom_bar(stat="identity")+
  coord_flip()+
  scale_x_discrete(position = "top")+
  scale_fill_manual(name = "",
                    labels = c("Negative Effect", "Positive Effect", "No Effect"),
                    values=myeffectcolor[c(1,3,2)])+
  scale_color_manual(name = "",
```

```
                        labels = c("Negative Effect", "Positive Effect", "No Effect"),
                        values=myeffectcolor[c(1,3,2)])+
  theme(text = element_text(size=10),
    axis.text.x = element_text(face = "bold"),
    axis.text.y = element_text(face = "bold", color = newcoef23$color),
    legend.position="bottom")+
  labs(x = "", y = "", title = "Covariate Effects on the Second FPC Scores")
```

B.7.2 Dicision Tree model with discrete variables

```
# fit a big tree
dat02 <- subset(mynewPCs3, select = c("PC1", select.dis))
dat02[2:31] <- lapply(dat02[2:31], factor)

fit02 <- rpart(PC1~., data=dat02)
model02 <- prune(fit02, cp=fit02$cptable[which.min(fit02$cptable[,"xerror"]),"CP"])
# This function returns the optimal cp value associated with the minimum error.
rpart.plot(model02, type=0,cex=1.2)

#Figure A.1
fancyRpartPlot(model02, uniform=TRUE,sub="")

dat12 <- subset(mynewPCs3, select = c("PC2", select.dis))
dat12[2:31] <- lapply(dat12[2:31], factor)

fit12 <- rpart(PC2~., data=dat12)
model12 <- prune(fit12, cp=fit12$cptable[which.min(fit12$cptable[,"xerror"]),"CP"])

rpart.plot(model12, type=0,cex=1.2)

#Figure A.2
fancyRpartPlot(model12, uniform=TRUE,sub="")
```

B.7.3 Dicision Tree model with whole selected variables

```
dat002 <- subset(mynewPCs3, select = c("PC1", select.dis, select.co))
fit02 <- rpart(PC1~., data=dat002)
model002 <- prune(fit02, cp=fit02$cptable[which.min(fit02$cptable[,"xerror"]),"CP"])
rpart.plot(model002, type=0,cex=1.2)

#Figure 4.8
fancyRpartPlot(model002, uniform=TRUE,sub="")

dat112 <- subset(mynewPCs3, select = c("PC2", select.dis, select.co))
fit112 <- rpart(PC2~., data=dat112)
```

```
model112 <- prune(fit112, cp=fit112$cptable[which.min(fit112$cptable[,"xerror"]),"CP"])
rpart.plot(model112, type=0,cex=1.2)

#Figure 4.9
fancyRpartPlot(model112, uniform=TRUE,sub="")
```

B.7.4 Random Forest with whole selected variables

```
fit2 <- randomForest(PC1 ~ ., importance = TRUE, data=dat002)
print(fit2)
fit12 <- randomForest(PC2 ~ ., importance = TRUE,  data=dat112)
print(fit12)
#Figure 4.10
par(mfrow = c(1,2))
varImpPlot(fit2, main = "For the First FPC Score" ,type=1)
varImpPlot(fit12, main = "For the Second FPC Score",type=1 )
```

B.8 PredictiveR^2 for different models

B.8.1 Elastic-net

```
#For FPC1
submynewPCs3=subset(mynewPCs3,select = -c(mergeid,PC2))
k <- 10 #the number of folds
folds <- cvFolds(NROW(submynewPCs3), K=k)
submynewPCs3$holdoutpred <- rep(0,nrow(submynewPCs3))

for(i in 1:k){
  train <- submynewPCs3[folds$subsets[folds$which != i], ] #Set the training set
  validation <- submynewPCs3[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(1000000)
  newlm <- train(
    PC1 ~.-holdoutpred, data = train, method = "glmnet", #change to -holdoutpred-base.recall_2.scale
    tuneGrid =expand.grid(alpha=0.1,lambda = 0.190567)) #the best tuning parameter
  newpred <- predict(newlm,newdata=validation)
  submynewPCs3[folds$subsets[folds$which == i], ]$holdoutpred <- newpred
}

submynewPCs3$holdoutpred #do whatever you want with these predictions
ppress=sum((submynewPCs3$PC2 - submynewPCs3$holdoutpred)^2)
ss=sum((submynewPCs3$PC2-mean(submynewPCs3$PC2))^2)
rsquare=1-ppress/ss
rsquare
#0.45 elastic net predictive R^2 for PC2 without drs
#0.55 elastic net predictive R^2 for PC2 with drs
##############################################
```

```r
#For FPC2
submynewPCs3=subset(mynewPCs3,select = -c(mergeid,PC1))
k <- 10 #the number of folds
folds <- cvFolds(NROW(submynewPCs3), K=k)
submynewPCs3$holdoutpred <- rep(0,nrow(submynewPCs3))

for(i in 1:k){
  train <- submynewPCs3[folds$subsets[folds$which != i], ] #Set the training set
  validation <- submynewPCs3[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(1000000)
  newlm <- train(
    PC2 ~.-holdoutpred, data = train, method = "glmnet", #change to -holdoutpred-base.recall_2.scale
    tuneGrid =expand.grid(alpha=0.1,lambda = 0.190567)) #the best tuning parameter
  newpred <- predict(newlm,newdata=validation)
  submynewPCs3[folds$subsets[folds$which == i], ]$holdoutpred <- newpred
}

submynewPCs3$holdoutpred #do whatever you want with these predictions
ppress=sum((submynewPCs3$PC2 - submynewPCs3$holdoutpred)^2)
ss=sum((submynewPCs3$PC2-mean(submynewPCs3$PC2))^2)
rsquare=1-ppress/ss
rsquare
#0.115 elastic net predictive R^2 for PC2 without drs
#0.132 elastic net predictive R^2 for PC2 with drs
```

B.8.2 Lasso

```r
##select best tuning parameter lambda
set.seed(10000)
lasso1 <- train(
  PC1 ~.-mergeid-PC2-base.recall_2.scale, data = mynewPCs3, method = "glmnet",
  trControl = trainControl(method = "repeatedcv",
                           number = 10,
                           repeats = 5,
                           verboseIter = TRUE),
  tuneGrid = expand.grid(alpha = 1, lambda=seq(0, 0.5, by = 0.1)),
  tuneLength = 10)
get_best_result(lasso1)

datlas <- subset(mynewPCs3, select = c("PC1", select.dis, select.co))

#For PFC1
datlas <- subset(mynewPCs3, select = c("PC1", select.dis, select.co))

k <- 10 #the number of folds
folds <- cvFolds(NROW(datlas), K=k)
datlas$holdoutpred <- rep(0,nrow(datlas))

```

```
for(i in 1:k){
  train <- datlas[folds$subsets[folds$which != i], ] #Set the training set
  validation <- datlas[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(10000)
  newlm <- train(PC1 ~.-holdoutpred, data = train, method = "glmnet",
  tuneGrid =expand.grid(alpha=1,lambda = 0.1))   #change to -holdoutpred-base.recall_2.scale
  newpred <- predict(newlm,newdata=validation)
  datlas[folds$subsets[folds$which == i], ]$holdoutpred <- newpred
}

datlas$holdoutpred #do whatever you want with these predictions
ppress=sum((datlas$PC1 - datlas$holdoutpred)^2)
ss=sum((datlas$PC1-mean(datlas$PC1))^2)
rsquare=1-ppress/ss
rsquare
##0.54 lasso predictive R^2 for PC1 with drs
##0.4 lasso predictive R^2 for PC1 without drs
#########################################
datlas2 <- subset(mynewPCs3, select = c("PC2", select.dis, select.co))

k <- 10 #the number of folds
folds <- cvFolds(NROW(datlas2), K=k)
datlas2$holdoutpred <- rep(0,nrow(datlas2))

for(i in 1:k){
  train <- datlas2[folds$subsets[folds$which != i], ] #Set the training set
  validation <- datlas2[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(10000)
  newlm <- train(PC2 ~.-holdoutpred, data = train, method = "glmnet",
                 tuneGrid =expand.grid(alpha=1,lambda = 0.1)) #change to -holdoutpred-base.recall_2.scale
  newpred <- predict(newlm,newdata=validation)
  datlas2[folds$subsets[folds$which == i], ]$holdoutpred <- newpred t
}

datlas2$holdoutpred #do whatever you want with these predictions
ppress=sum((datlas2$PC2 - datlas2$holdoutpred)^2)
ss=sum((datlas2$PC2-mean(datlas2$PC2))^2)
rsquare=1-ppress/ss
rsquare
##0.13 lasso predictive R^2 for PC2 with drs
##0.08 lasso predictive R^2 for PC2 without drs
```

B.8.3 Random Forest

```
datc00 <- subset(mynewPCs3, select = c("PC1", select.dis, select.co))

k <- 10 #the number of folds
```

```
folds <- cvFolds(NROW(datc00), K=k)
datc00$holdoutpred <- rep(0,nrow(datc00))

for(i in 1:k){
  train <- datc00[folds$subsets[folds$which != i], ] #Set the training set
  validation <- datc00[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(10000)
  newlm <- randomForest(PC1 ~ .-holdoutpred,data=train)  #change to -holdoutpred-base.recall_2.scale
  newpred <- predict(newlm,newdata=validation)
  datc00[folds$subsets[folds$which == i], ]$holdoutpred <- newpred
}
datc00$holdoutpred #do whatever you want with these predictions
ppress=sum((datc00$PC1 - datc00$holdoutpred)^2)
ss=sum((datc00$PC1-mean(datc00$PC1))^2)
rsquare=1-ppress/ss
rsquare
#0.479 random forest predictive R^2 for PC1 with drs
#0.37 random forest predictive R^2 for PC1 without drs
##################################################
datc112 <- subset(mynewPCs3, select = c("PC2", select.dis, select.co))
folds <- cvFolds(NROW(datc112), K=k)
datc112$holdoutpred <- rep(0,nrow(datc112))

for(i in 1:k){
  train <- datc112[folds$subsets[folds$which != i], ] #Set the training set
  validation <- datc112[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(1000000)
  newlm <- randomForest(PC2 ~ .--holdoutpred,data=train) #change to -holdoutpred-base.recall_2.scale
  newpred <- predict(newlm,newdata=validation)
  datc112[folds$subsets[folds$which == i], ]$holdoutpred <- newpred }

datc112$holdoutpred #do whatever you want with these predictions
ppress=sum((datc112$PC2 - datc112$holdoutpred)^2)
ss=sum((datc112$PC2-mean(datc112$PC2))^2)
rsquare=1-ppress/ss
rsquare
#0.122 random forest predictive R^2 for PC2 with drs
#0.1 random forest predictive R^2 for PC2 without drs
```

B.8.4 Decision Tree

```
k <- 10 #the number of folds
folds <- cvFolds(NROW(datc00), K=k)
datc00$holdoutpred <- rep(0,nrow(datc00))

for(i in 1:k){
```

```
  train <- datc00[folds$subsets[folds$which != i], ] #Set the training set
  validation <- datc00[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(10000)
  newlm <- prune(rpart(PC1~.-holdoutpred, data=train), cp=rpart(PC1~.-holdoutpred, data=train
  $cptable[which.min(rpart(PC1~.-holdoutpred, data=train)$cptable[,"xerror"]),"CP"])
  #change to -holdoutpred-base.recall_2.scale
  newpred <- predict(newlm,newdata=validation)
  datc00[folds$subsets[folds$which == i], ]$holdoutpred <- newpred
  }

datc00$holdoutpred #do whatever you want with these predictions
ppress=sum((datc00$PC1 - datc00$holdoutpred)^2)
ss=sum((datc00$PC1-mean(datc00$PC1))^2)
rsquare=1-ppress/ss
rsquare
#0.436 dicision tree predictive R^2 for PC1 with drs
#0.26 dicision predictive R^2 for PC1 without drs
#################################################
for(i in 1:k){
  train <- datc112[folds$subsets[folds$which != i], ] #Set the training set
  validation <- datc112[folds$subsets[folds$which == i], ] #Set the validation set
  set.seed(1000000)
  newlm <- prune(rpart(PC2~.-holdoutpred, data=train), cp=rpart(PC2~.-holdoutpred, data=train)$cptable[which.min(rpart(PC2~.-holdoutpred, data=train)$cptable[,"xerror"]),"CP"])
    #change to -holdoutpred-base.recall_2.scale
  newpred <- predict(newlm,newdata=validation)
  datc112[folds$subsets[folds$which == i], ]$holdoutpred <- newpred
  }

datc112$holdoutpred #do whatever you want with these predictions
ppress=sum((datc112$PC2 - datc112$holdoutpred)^2)
ss=sum((datc112$PC2-mean(datc112$PC2))^2)
rsquare=1-ppress/ss
rsquare
#0.094 decision tree predictive R^2 for PC2 with drs
#0.074 decision tree predictive R^2 for PC2 without drs
```