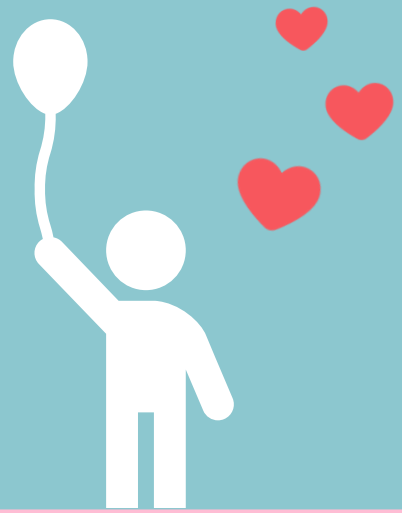
행복지수 데이터로 서울 시민의 현재 행복도 예측하기

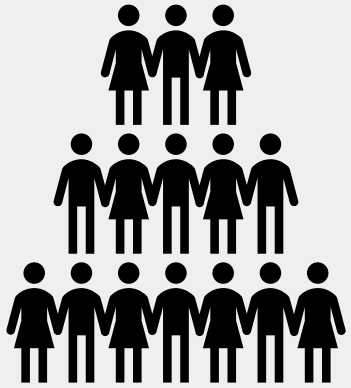


곽지석 김채성 김채현 선대운 신유진 이재환

CONTENTS

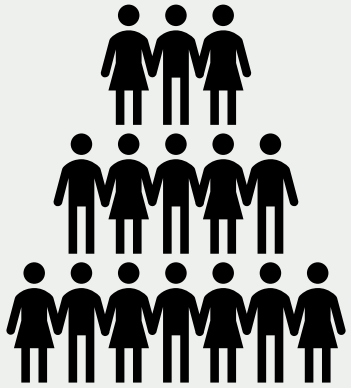
Intro

서울형 행복지수 관련 시민의식 조사



998명 대상

서울형 행복지수 관련 시민의식 조사



998명 대상

얼마나 행복하신가요? **[개인별 주관적 행복도]**

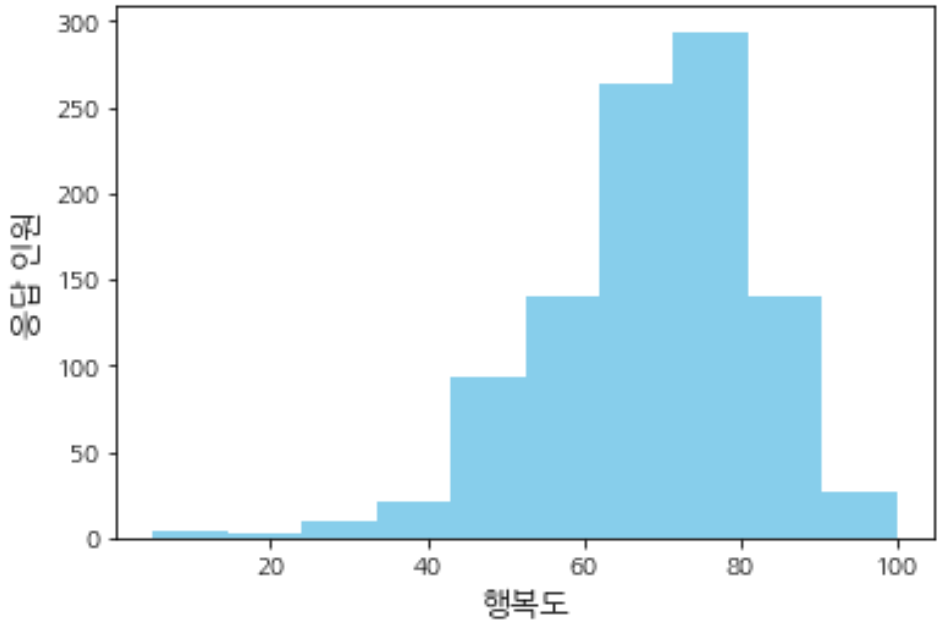
이러이러한 부분에 대해 얼마나 만족하시나요?
[사회경제적 요인에 대한 만족도]

이건 행복에 얼마나 영향을 미칠까요?
[사회경제적 요인의 중요성]

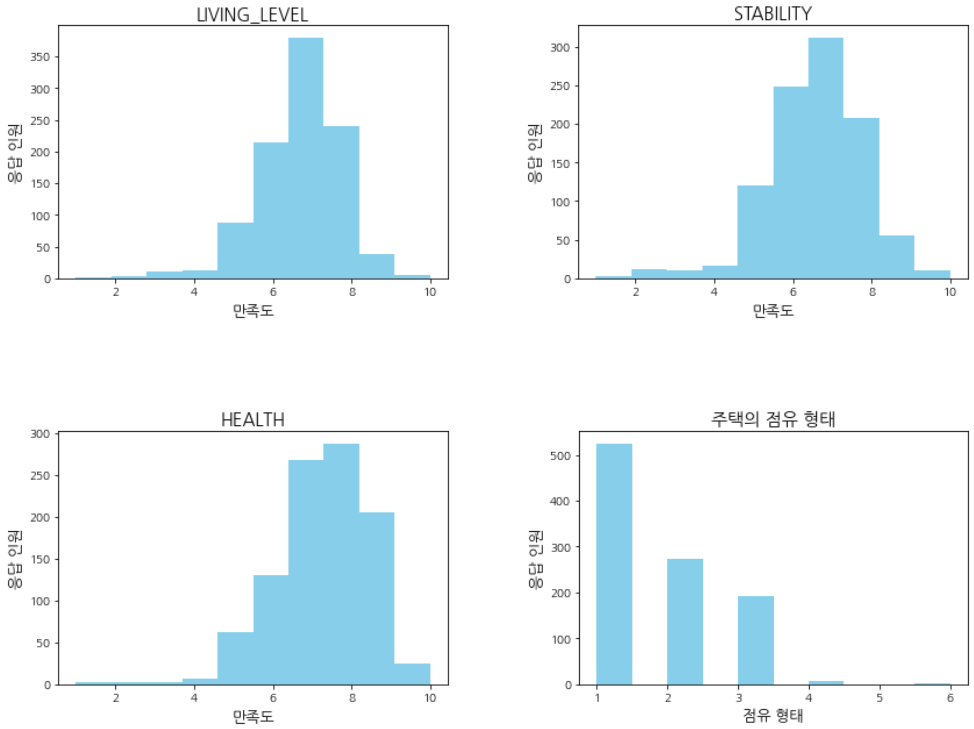
당신은 어떤 사람인가요? **[인적 사항]**

EDA

현재 행복도 분포

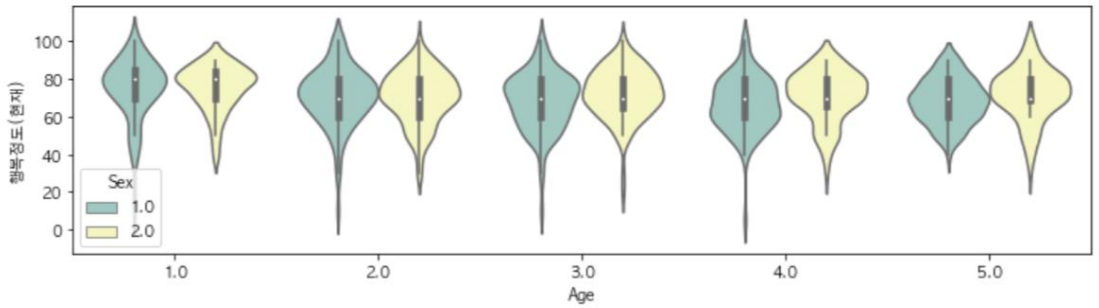


변수별 응답 분포



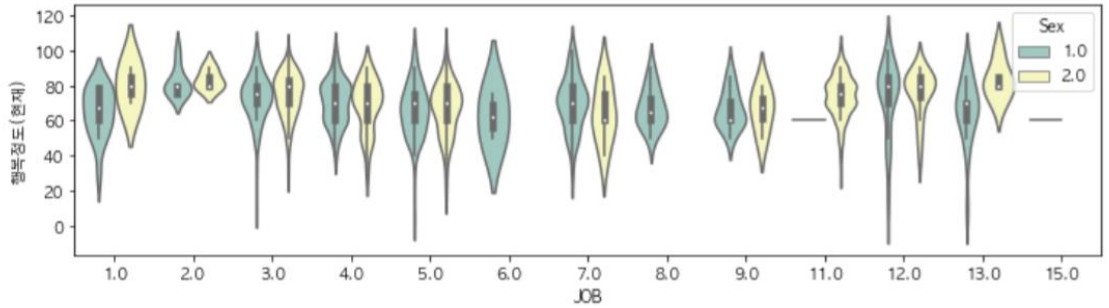
EDA

연령과 성별에 따른 행복도의 분포



남자의 경우 고연령층으로 갈수록 행복도의 중위값이 낮아지는 경향이 존재하나 여자의 경우는 연령에 대해 U자형 분포가 나타나는 것으로 확인

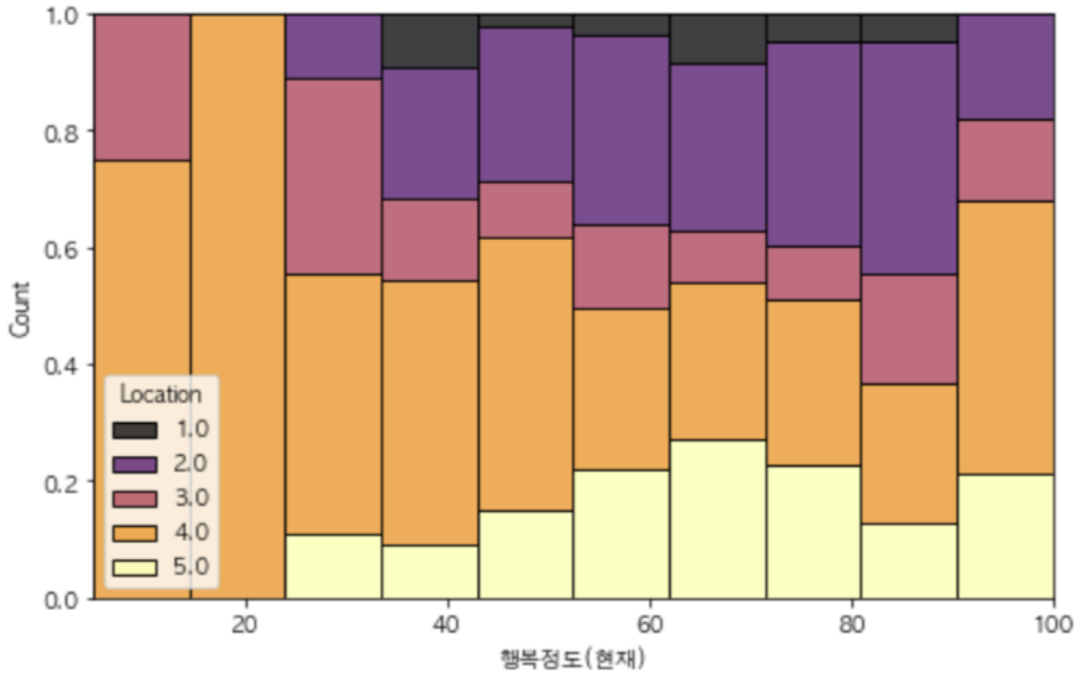
직업에 따른 행복도의 분포



전문직에서 평균적으로 가장 높은 행복도를 확인할 수 있었고 학생 및 무직/연금생활자 그룹(13번)도 높은 행복도를 보였다.

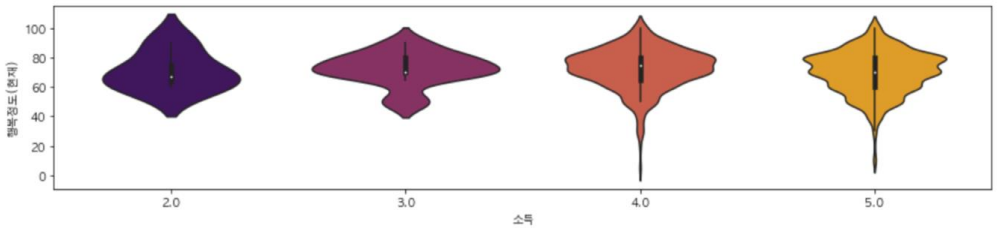
EDA

지역에 따른 행복도의 분포 차이



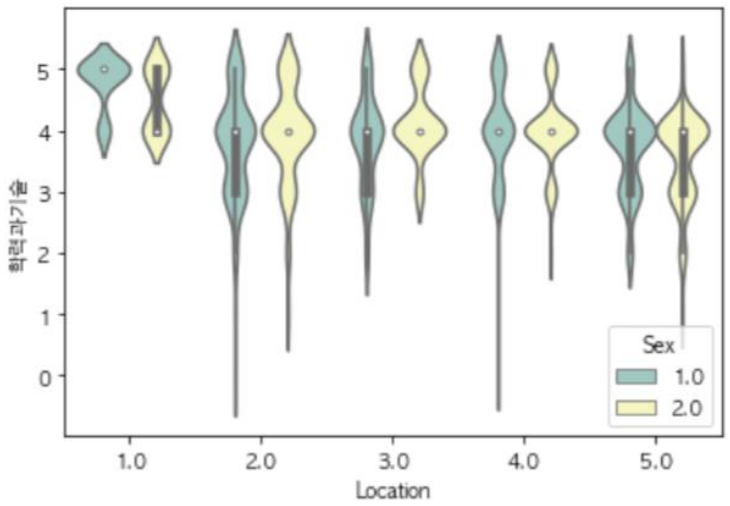
동북권(2번) 및 동남권(5번)의 경우 대체로 행복한 쪽에 치우쳐져 있음, 4번 서남권은 그 반대.

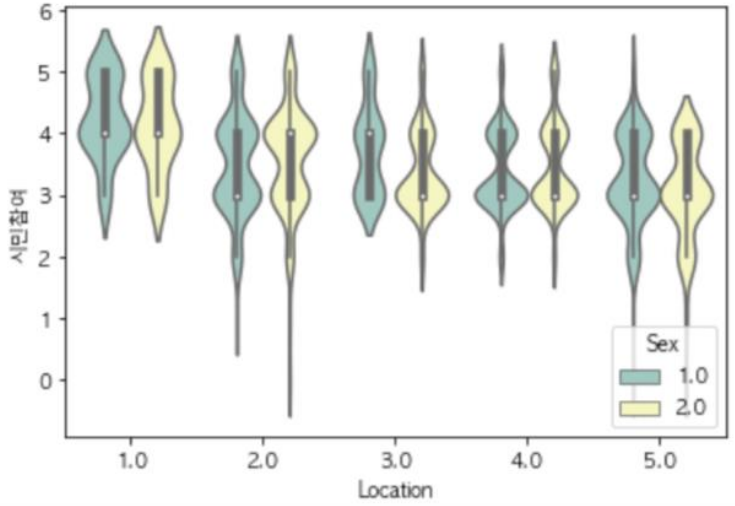
개인별 소득의 중요도에 따른 행복정도의 차이

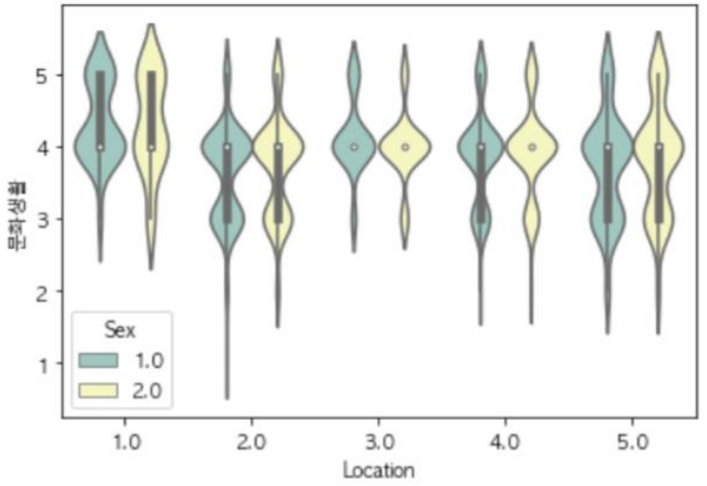


그룹별 중위값을 놓고 보면, 소득에 대한 높은 중요도를 가진 그룹에서 행복도가 크게 나타남. (이러한 경향은 모든 지역, 성별에서 똑같이 나타남)

EDA







학력과 기술/ 시민참여 / 문화생활의 중요도는 도심권 지역에서 특히 높게 나타남.

전처리

**
일반적 사항

A
삶의 질
인식 평가

B
행복의 항목별
중요도 평가

C
행복의 항목별
세부 지표
중요도 평가

D
항목별
만족도 평가

전처리

**

일반적 사항

- 성별
- 연령
- 거주지역
- 학교 교육
- 혼인상태
- 주택의 점유형태
- 주택유형
- 빚 유무

A

삶의 질
인식 평가

현재 행복한 정도

B

행복의 항목별
중요도 평가

C

행복의 항목별
세부 지표
중요도 평가

D

항목별
만족도 평가

- 생활수준
- 건강상태
- 삶의 성취도
- 대인관계

… 총 10개

전처리

처리 필요

HAPPY	생활수준	건강상태	삶의 성취도	•••	연령대	거주지역	학교 교육	•••
82	7	9	7		1	1	5	
90	8	7	6		3	1	3	
75	7	7	8		3	1	4	
⋮				⋮				⋮
70	7	8	7		4	1	4	
70	6	7	6		5	1	3	
65	6	7	5		5	1	3	

전처리

카테고리 단순화

연령, 거주지역, 학교교육, 혼인상태, 주택, 빚 유무

Ex) 학교 교육

① 초등학교 졸업 (무학, 중퇴포함)
② 중학교 졸업 (중퇴포함)
③ 고등학교 졸업 (중퇴포함)

⌄⌄

고졸 이하

④ 전문대 졸업 (재학, 중퇴포함)
⑤ 대학교 졸업 (재학, 중퇴포함)
⑥ 대학원 이상 (재학, 중퇴포함)

⌄⌄

대졸 이상

One-Hot Encoding

연령, 거주지역, 주택유형

Ex) 주택 유형 : 아파트 / 연립주택 / 단독주택

WHAT_HOUSE_연립	WHAT_HOUSE_단독
0	1
0	1
1	0
0	0

* drop_first = True

Linear Regression

Linear Regression

From **Frequentist Perspective**

실제 데이터의 현재 행복도 분포를 바탕으로
glm을 통해 구한 분포를 **ols와 비교**

Bayesian 회귀분석을 통해 현재 행복도에 영향을
미치는 **변수의 특징 및 관계 파악**

Frequentist (OLS)

X = Preprocessed Variables

Y = Current Happiness

statsmodels.formula.api 사용

OLS Regression Results

Dep. Variable:	HAPPY	R-squared:	0.426
Model:	OLS	Adj. R-squared:	0.410
Method:	Least Squares	F-statistic:	27.58
Date:	Wed, 23 Nov 2022	Prob (F-statistic):	4.59e-98
Time:	06:55:36	Log-Likelihood:	-3776.0
No. Observations:	995	AIC:	7606.
Df Residuals:	968	BIC:	7738.
Df Model:	26		
Covariance Type:	nonrobust		

R-squared : 0.426

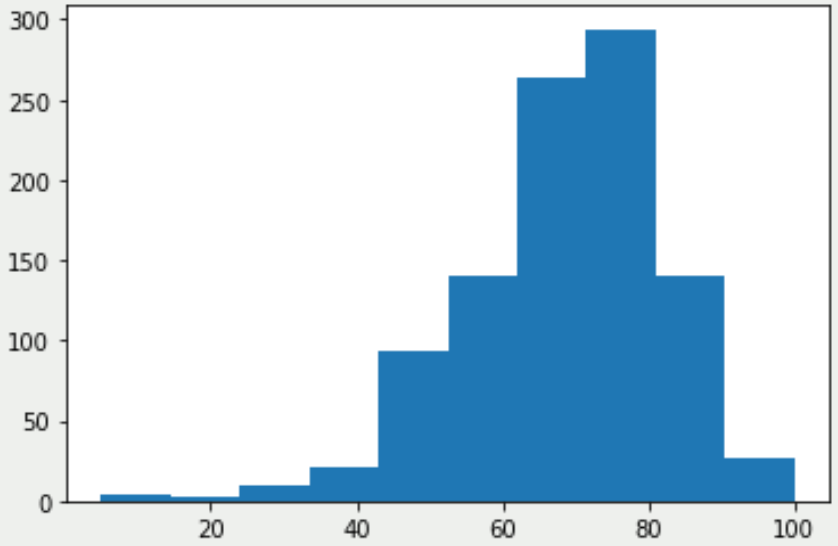
AIC : 7606

회귀계수

```
LIVING_LEVEL : 1.828025746555342
HEALTH : 0.1621363202402447
ACHIEVEMENT : 2.339089890702217
RELATIONSHIP : 0.7826752295318109
SAFETY : -0.0329391997592653
BELONGING : -0.12510036678588782
STABILITY : 1.7573332101859256
LEISURE : 1.4540200420375216
LOCAL : 0.6947588002194324
JOB_SATISFACTION_무응답 : 0.5661591074267058
JOB_SATISFACTION_불만족 : -1.8746808984366865
SEX : -0.5030431107026084
AGE_30대 : -2.9299404993750584
AGE_40대 : -2.5630777479487694
AGE_50대 : -4.672430752331455
AGE_60대 : -3.2672753623601953
LOC_동북 : 2.3235306784175105
LOC_서북 : 1.6774986306379367
LOC_서남 : 1.1045285575147403
LOC_동남 : 3.1151954773048285
EDU : -1.2879876544335285
MARRY : 0.5409163768193823
MY_HOUSE : 2.0195630674481926
WHAT_HOUSE_연립 : -1.0659579959934822
WHAT_HOUSE_단독 : -1.6219197103899683
DEBT : -1.2854824608544477
```

Bayesian I Modeling

현재 행복도



Normal + Left Skewed

Family

Gamma?

Gaussian?

Bayesian I Distribution

Gamma

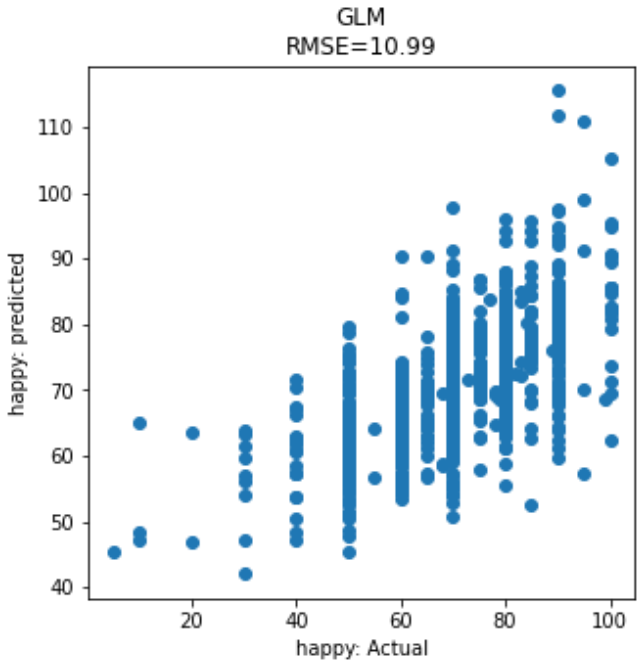
- Link function : inverse_power
- 선택 이유
  1. 왜도 형태 감안
  2. Gamma: Positive
- **AIC** : 7990

Gaussian

- Link function: identity function
- 선택 이유: 데이터 자체의 숫자가 낮은 건 거의 없고, 만약 점수가 낮게 나올 상황이라 음수로 예측해도 그건 행복하지 않은 거라고 판단하고 OO이랑 동일하게 본다고 생각함.
- **AIC** : 7605

Bayesian I Distribution

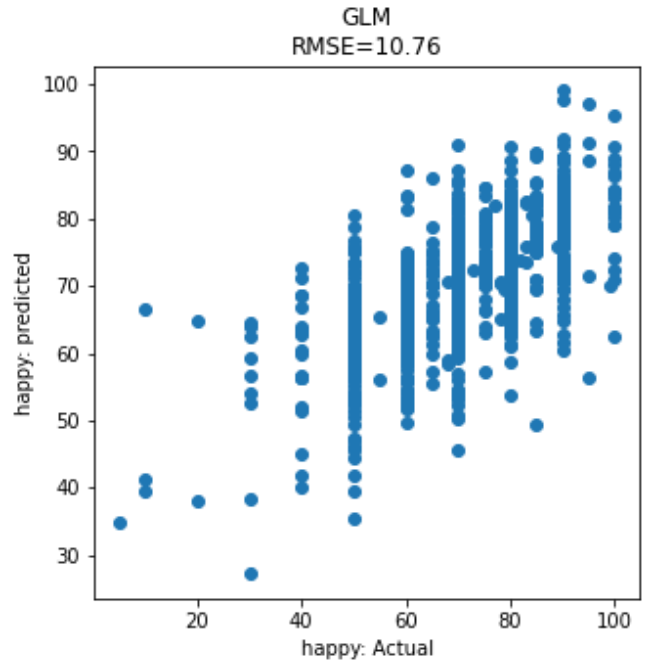
Gamma



RMSE = 10.99

실제 행복도
VS
예상 행복도

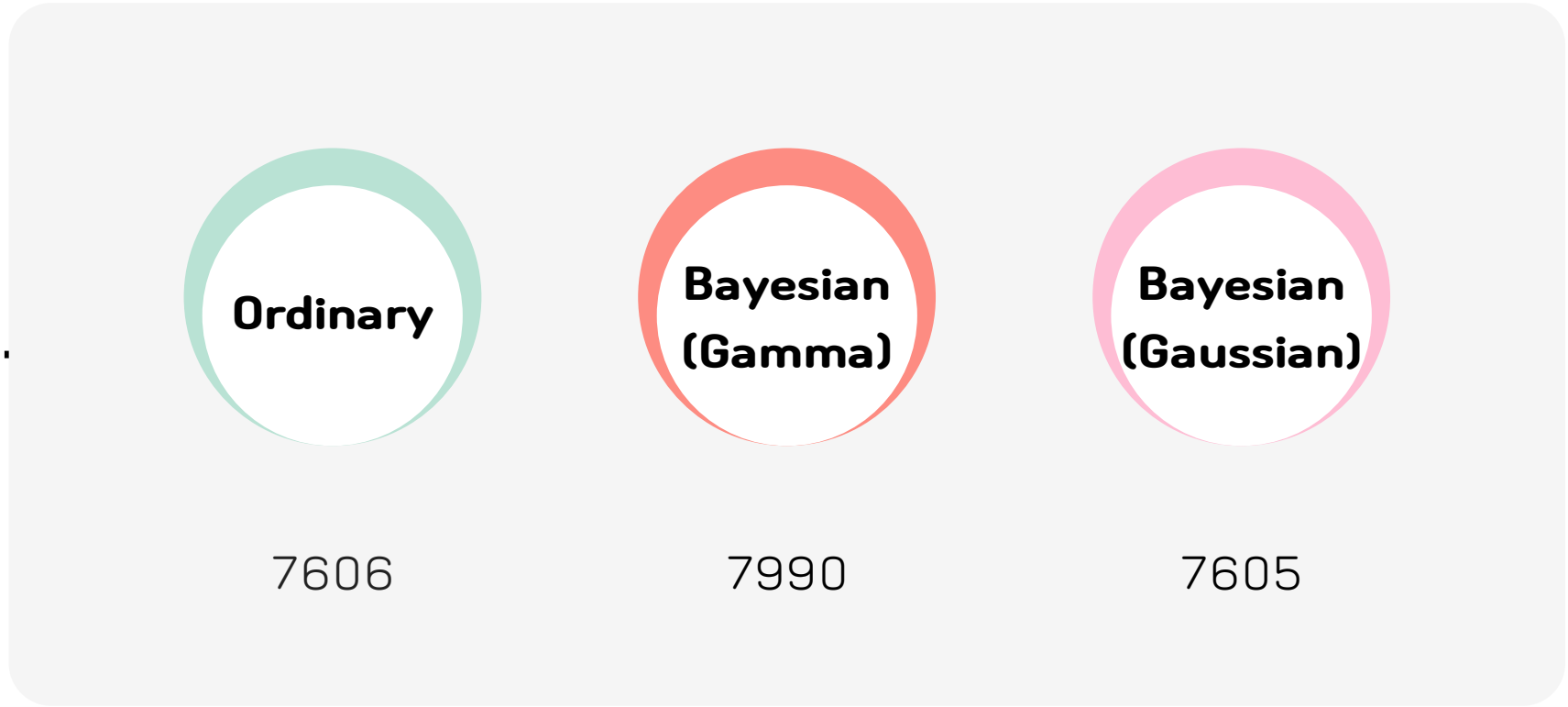
Gaussian



RMSE = 10.76

Bayesian I Distribution

AIC



Bayesian I MCMC

Use Pymc3 Package

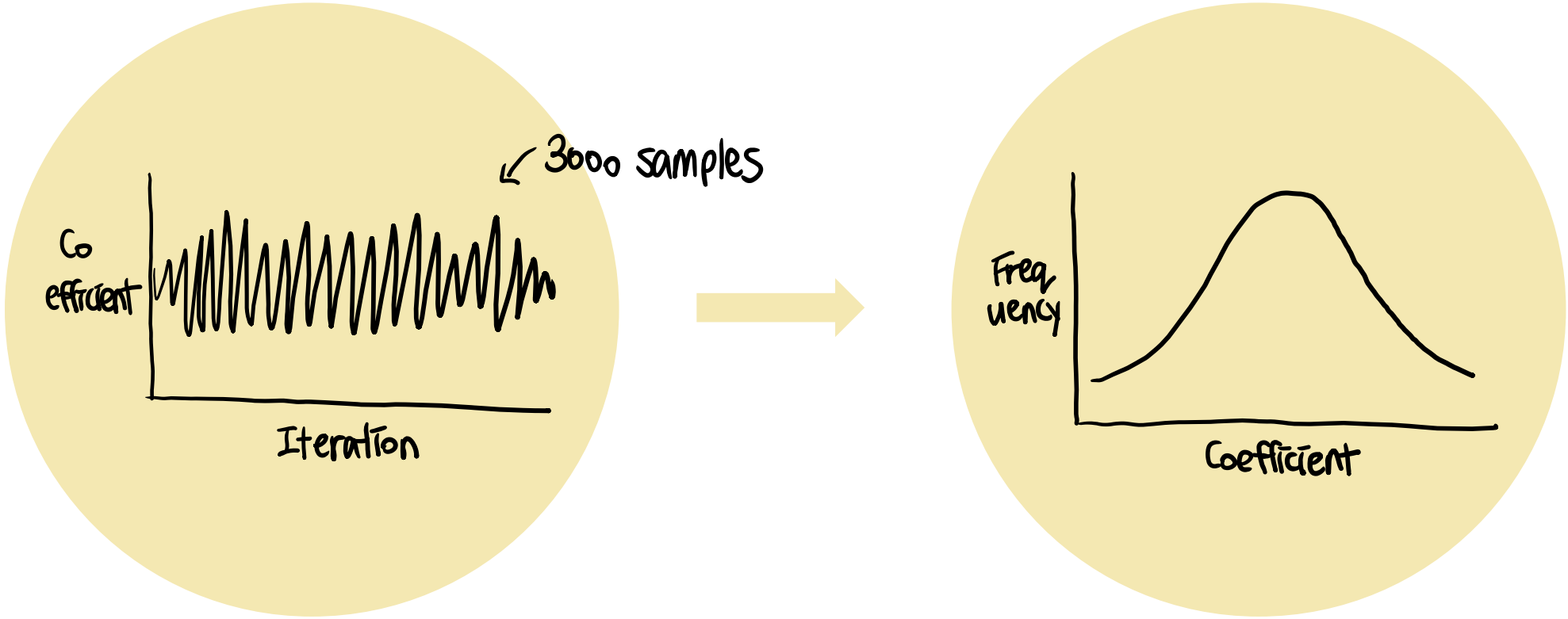
pm.NUTS를 통해 MCMC 알고리즘을 구현하고,
샘플링을 통해 사후 3000개의 샘플 생성

```
bglm = pm.Model()
with bglm:
    family = pm.glm.families.Normal()

    pm.GLM.from_formula(formula.data=happy, family=family)

    start = pm.find_MAP()

    step = pm.NUTS(scaling=start)

    trace = pm.sampling.sample(3000, step=step, start=start,
progressbar=False, return_inferencedata=True)
```

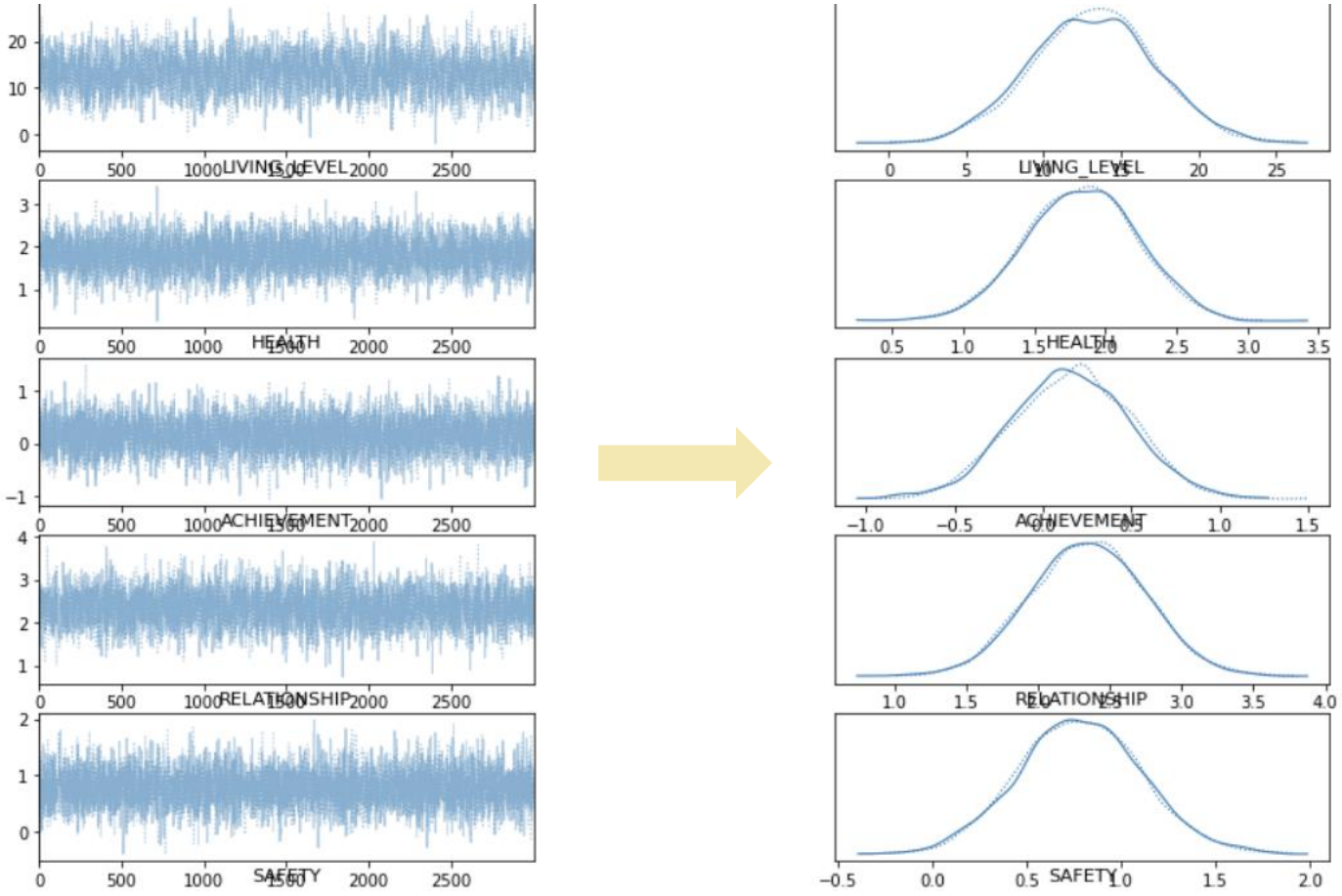
AIC가 더 낮은 Gaussian 사용

- **Normally** Distributed Priors
- 모델 생성
- 최적화를 사용하여 초기값 추정
- NUTS MCMC 샘플링 알고리즘 인스턴스 생성
- 샘플링을 사용하여 3000개의 사후 샘플 생성

Bayesian | MCMC



Bayesian I MCMC



Frequentist

하나의 특정 Point Estimate로 회귀계수가 결정됨

Bayesian

특정 Point가 아닌, **분포의 형태**로 회귀계수들이 추정됨

Bayesian I MCMC

Posterior Mean

LIVING_LEVEL	1.828	AGE_50대	-4.682
HEALTH	0.158	AGE_60대	-3.309
ACHIEVEMENT	2.346	LOC_동북	2.315
RELATIONSHIP	0.780	LOC_서북	1.652
SAFETY	-0.033	LOC_서남	1.086
BELONGING	-0.125	LOC_동남	3.110
STABILITY	1.755	EDU	-1.306
LEISURE	1.451	MARRY	0.530
LOCAL	0.692	MY_HOUSE	2.032
JOB_SATISFACTION_무응답	0.553	WHAT_HOUSE_연립	-1.072
JOB_SATISFACTION_불만족	-1.880	WHAT_HOUSE_단독	-1.613
SEX	-0.496	DEBT	-1.280
AGE_30대	-2.924		
AGE_40대	-2.573		

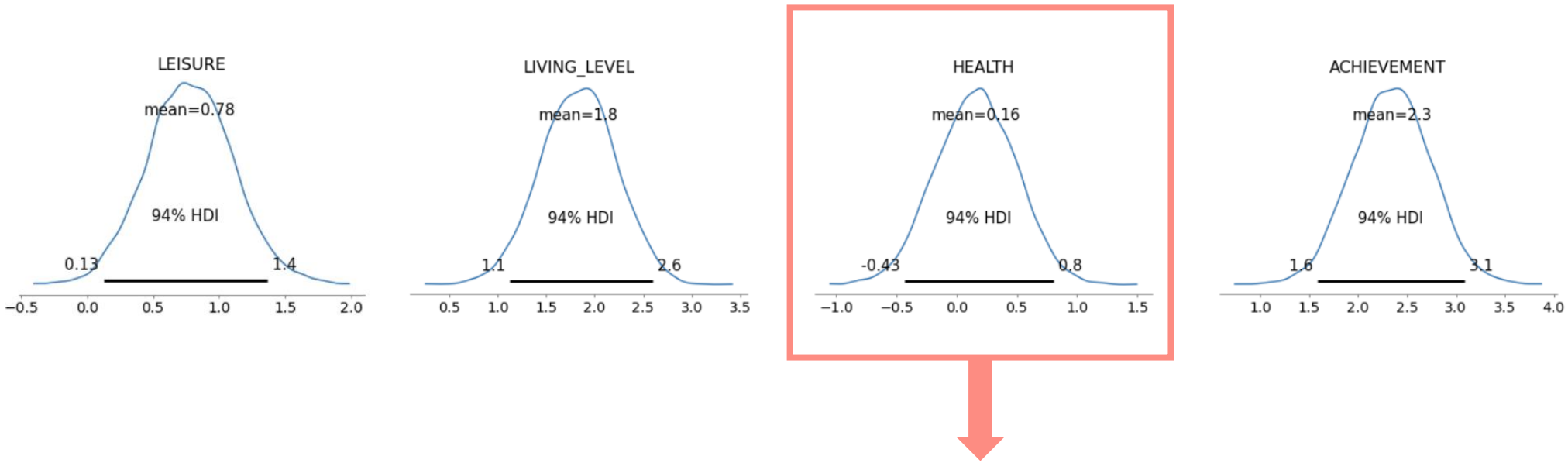
+ 대인관계, 삶의 성취도, 생활수준, 삶의 안정성, 문화생활

− 직업 만족도 (불만족)

× 건강상태, 성별, 소속감, 지역사회, 학교 교육, 결혼 유무, 안전

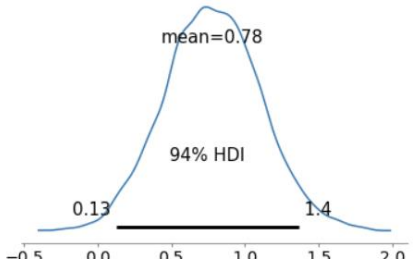
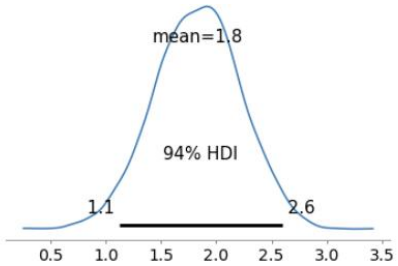
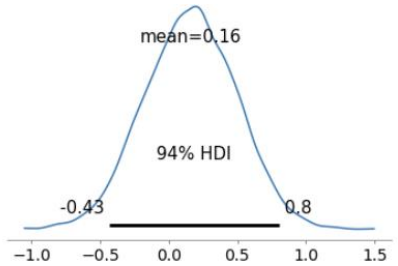
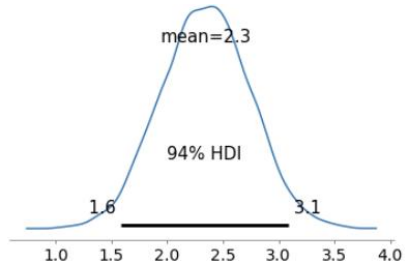
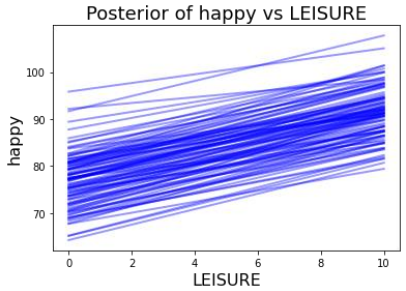
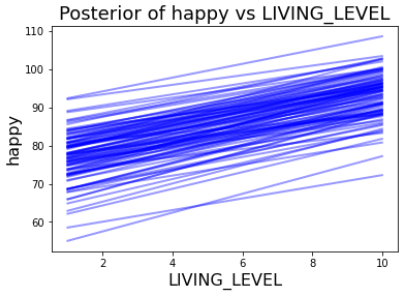
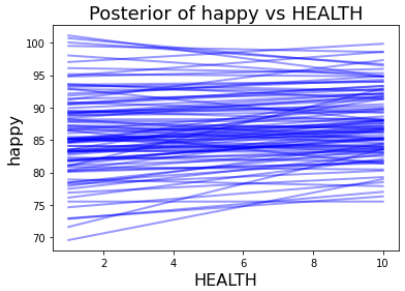
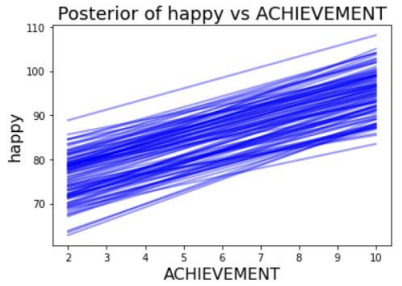
✓ Bayesian과 Frequentist 방법에 있어 큰 차이가 없음.

Bayesian I Interpretation



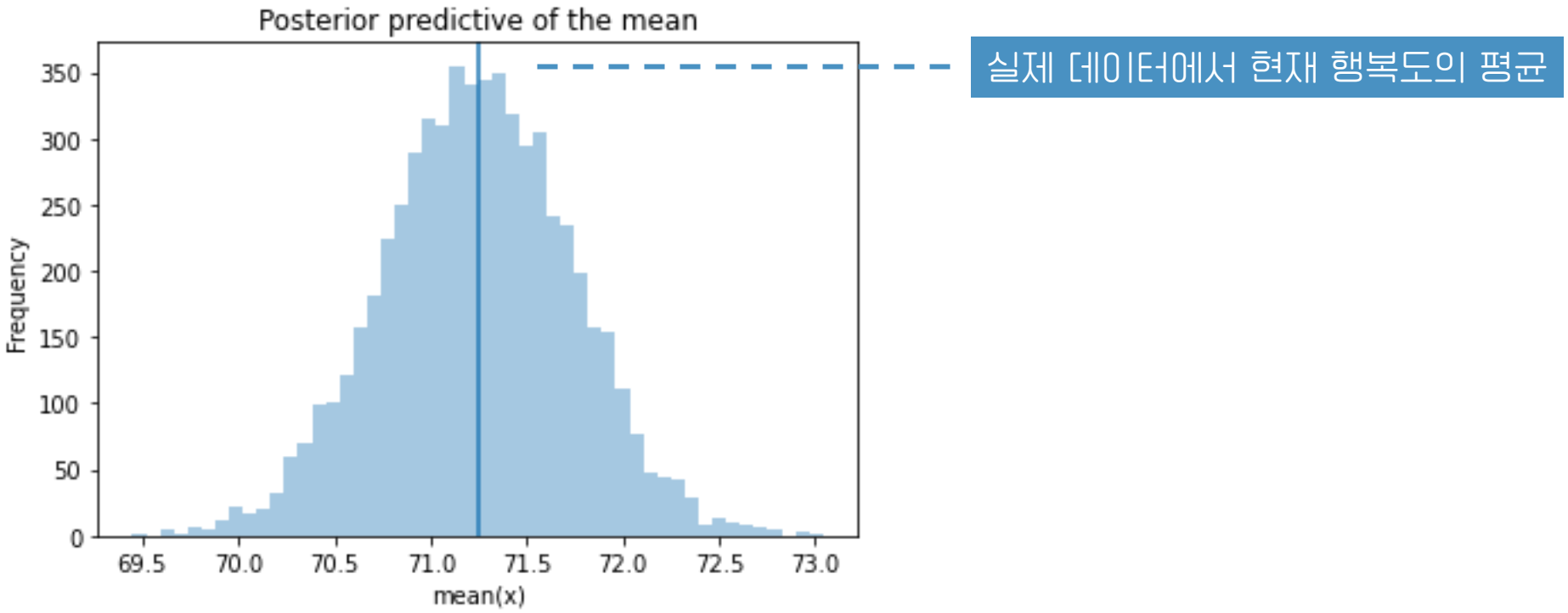
0 포함 → 유의한 변수가 아니라고 판단 가능

Bayesian I Interpretation

	여가생활	생활수준	건강	삶의 성취도
Distribution (Sampled by MCMC)				
Predictor Variable - Response				

Bayesian I Diagnosis

Posterior Predictive Check 실제 데이터와 베이지안 모델의 예측값 비교



Lasso

About LASSO

기본 회귀모형

$$y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I_n),$$

OLS

$$\underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

LASSO

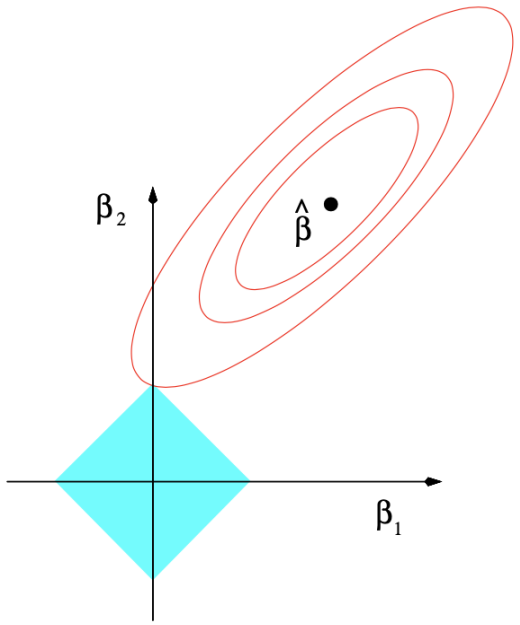
$$\underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

'각 계수의 절댓값의 합'을 수식에 포함

(penalty…)

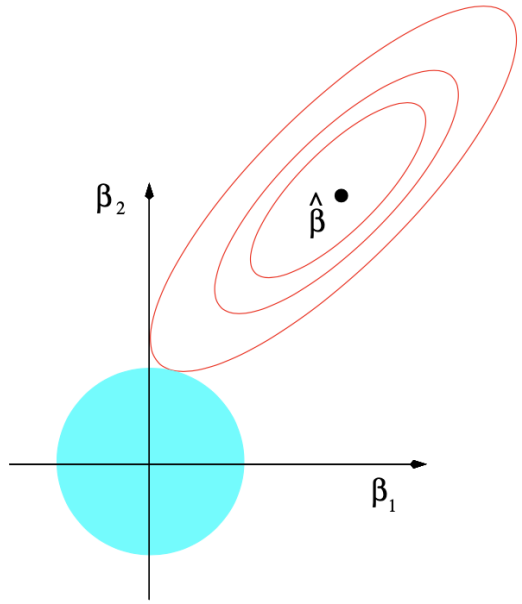
About LASSO

LASSO



$$\text{subject to} \sum_{j=1}^{p} |\beta_j| \leq t.$$

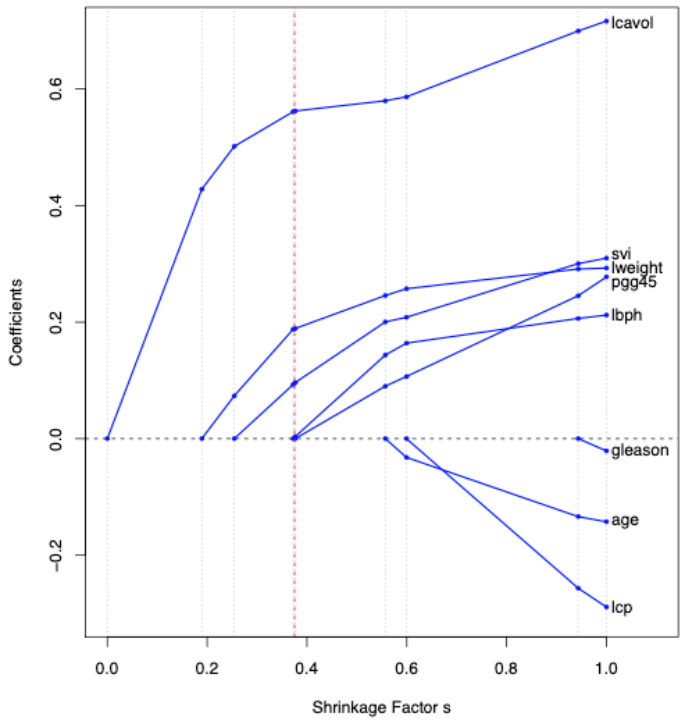
RIDGE



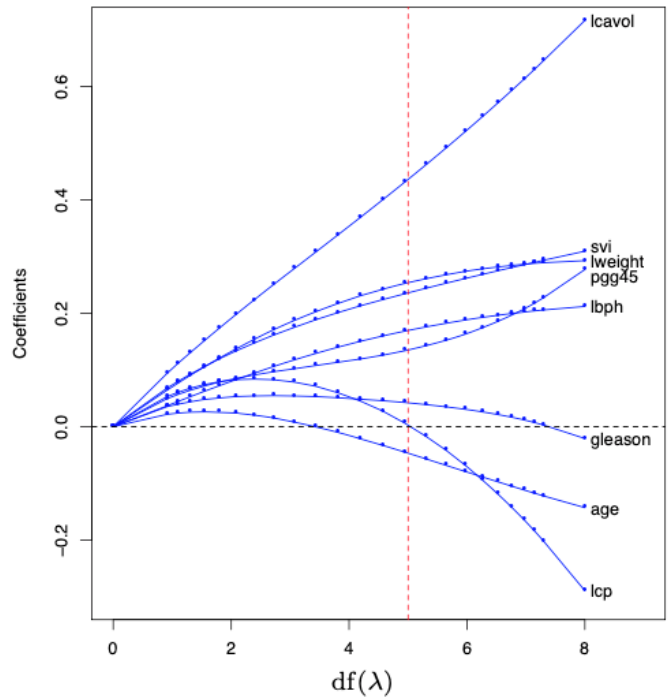
$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 \leq t.$$

About LASSO

LASSO



RIDGE



Implementation

Best penalty (lambda) 값 선택
(sklearn.LassoCV, alpha_)

≫

Best penalty 값을 대입하여
Lasso Regression 실행

≫

회귀계수 비교
(어떤 요인이 가장 영향이 큰가?)

Ordinary / Bayesian 비교

Implementation

1. Best Penalty 계산 | Penalty function의 규제에 대한 최적값 도출

$$\hat{\beta}^{lasso} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Lambda!

```
lasso_cv = LassoCV().fit(x_train, y_train)
```

```
lasso_cv.alpha_
```

```
0.27712168496119505
```

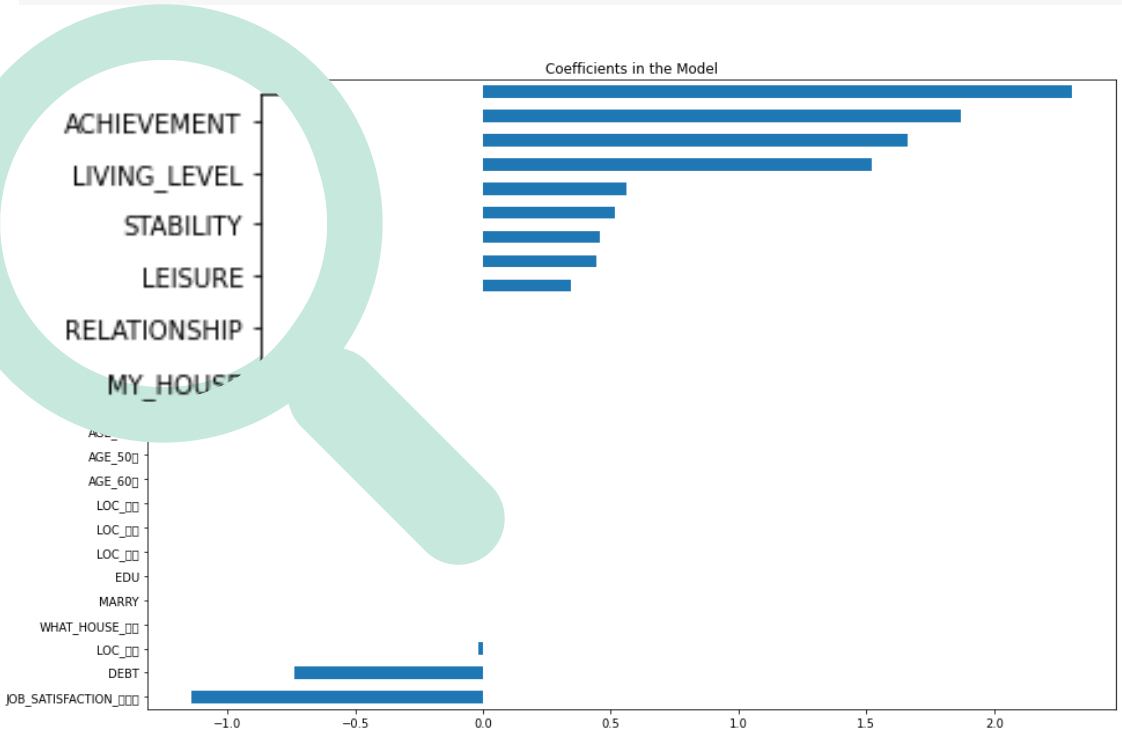
Best lambda : 0.27712168

Implementation

2. Lasso Regression 실행

First, Ordinary!

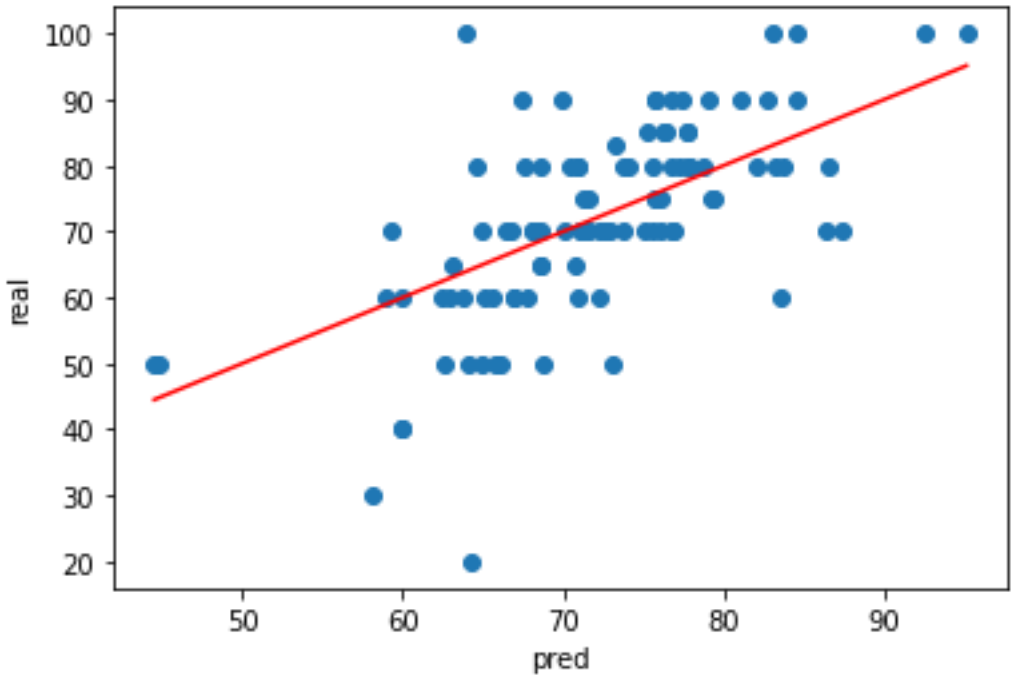
```
model_lasso = Lasso(alpha=0.3391800797246021, max_iter=50000, fit_intercept=False).fit(x_train, y_train)
```



- 삶의 성취도, 생활수준, 미래안정성, 여가/취미/오락
- 직업 불만족, 빚 유무
- 삶의 성취도, 생활수준, 미래안정성의 영향이 가장 큼
- 나이, 거주지역, 교육수준, 혼인 상태는 행복도에 영향을 미치지 않음.

Implementation

2. Lasso Regression 실행 First, Ordinary!



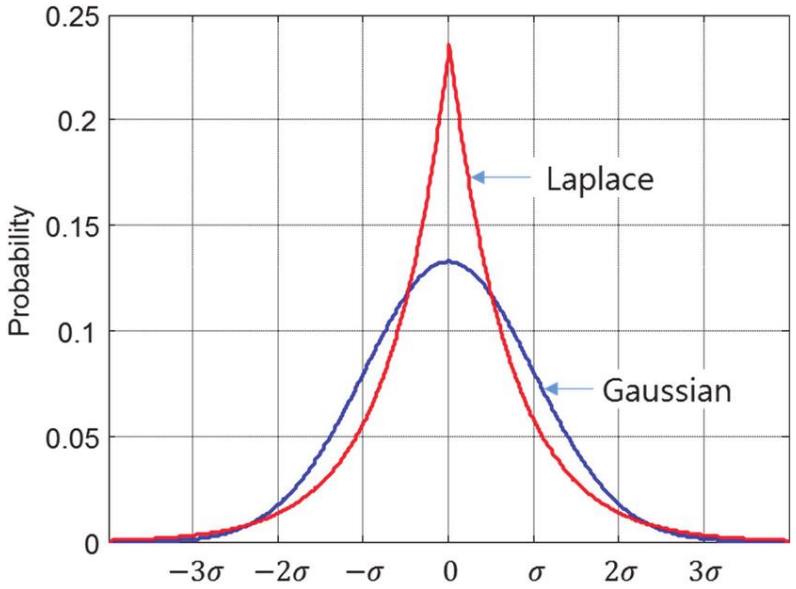
MSE : 129.97873345127318

R2 score: 0.4189559299102529

Implementation

2. Lasso Regression 실행 Second, Bayesian!

➡ 문제 해결을 위해 **Laplace Distribution** (Double Exponential Distribution) 도입



$$\text{Laplace}(\mu, b)$$

- pdf: $f(x \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$

- μ: location parameter, b: scale parameter

Implementation

Prior and Likelihood Selection

Jeffrey's Prior (Summer WEEK3)

$$\beta|\sigma^2 \sim Laplace(0, b) \qquad \sigma^2 \sim 1/\sigma^2$$

Joint Prior $p(\beta, \sigma^2) = p(\beta|\sigma^2)p(\sigma^2)$

Likelihood $y|\beta, \sigma^2 \sim N(X\beta, \sigma^2 I)$

하이퍼파라미터까지 추정할 수 있는 더 복잡한 모델도 있지만, 구현을 위하여 단순한 모델을 사용하였고
하이퍼파라미터는 **ordinary LASSO에서 구한 값과 동일하게 사용**

(Park and Casella, 2008)

Why Laplace?

(Summer WEEK3)

$$\begin{aligned}(\text{posterior}) &\propto (\text{likelihood}) \times (\text{prior}) \\ &\propto \exp\Big(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2\Big)\exp\Big(-\frac{1}{b}\sum_{j=1}^{p}|\beta_j - 0|\Big) \\ &= \exp\Big(-\frac{1}{2\sigma^2}\Big(\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \frac{2\sigma^2}{b}\sum_{j=1}^{p}|\beta_j|\Big)\Big)\end{aligned}$$

Likelihood $y|\beta,\sigma^2 \sim N(X\beta,\sigma^2 I)$

$2\sigma^2/b = \lambda$ 를 만족하도록 lambda를 선택하면, $\Big(\sum(y_i - x_i^T\beta)^2 + \frac{2\sigma^2}{b}\sum|\beta_j|\Big)$

가 Lasso 손실이 된다.

(Park and Casella, 2008)

Implementation

Model Defining with RStan

추정하고자 하는 파라미터와
그 파라미터를 변환하여 얻는 파라미터 정의

초기값 설정 | 입력 데이터 타입 지정 | 파라미터 정의 | Prior, Likelihood 정의

```
> n = nrow(df)
> p = ncol(df) - 1
> X = as.matrix(df[, -(p+1)])
> y = df$HAPPY
> lambda = 0.27712168496
119505
```

```
> model = '
+   data {
+     int n;
+     int p;
+     matrix[n,p] X;
+     real y[n];
+     real lambda;
+   }
```

```
+   parameters {
+     vector[p] beta;
+     real<lower=0> sigma2;
+   }
+
+   transformed parameters {
+     real<lower=0> sigma;
+     vector[n] mu;
+     sigma = sqrt(sigma2);
+     mu = X * beta;
+   }
```

```
+   model {
+     target += 1/sigma2;
+     target += double_
        exponential_lpdf(beta
        | 0, 2*sigma2/lambda);
+     target += normal_lpdf
      (y | mu, sigma);
+   }
+ '
```

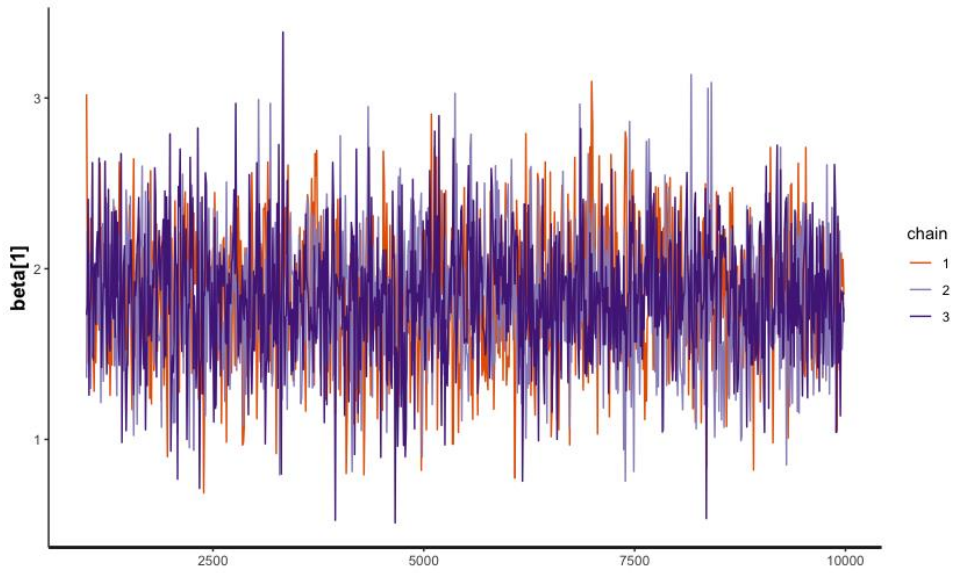
Implementation

Sampling Use NUTS Sampler for implementation

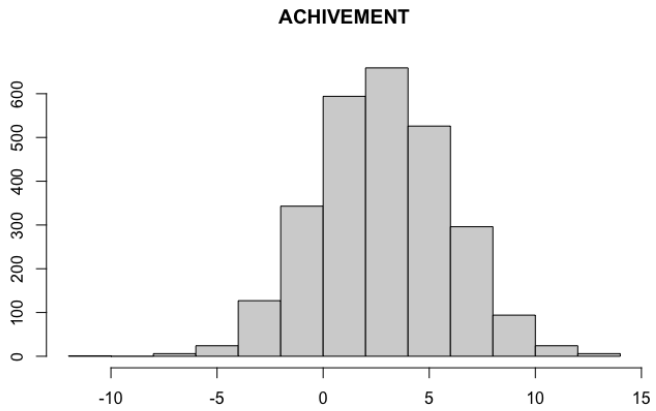
3개의 Markov Chain
→ Convergence 확인이 용이

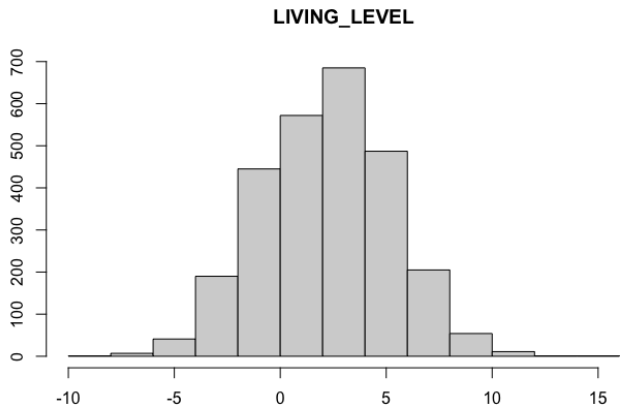
```
data = list(X=X, y=y, n=n, p=p, lambda=lambda)
m = stan_model(model_code = model)
fit = stan(model_code = model, data=data, iter=10000, warmup=1000, thin=10, chains=3)
```

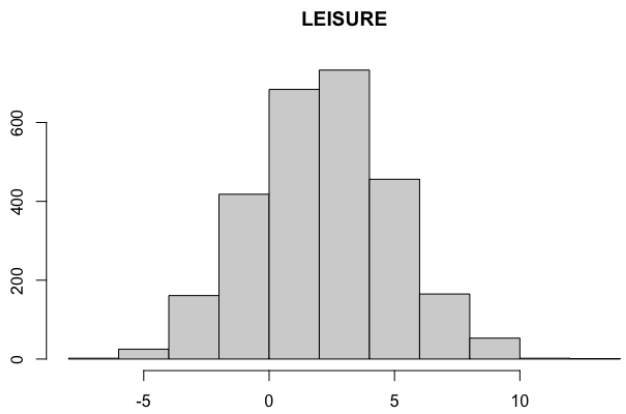
```
SAMPLING FOR MODEL 'f473bd209c20cc3332f41a6db701f5e0' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000256 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.56 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
Chain 1: Iteration: 1001 / 10000 [ 10%]  (Sampling)
Chain 1: Iteration: 2000 / 10000 [ 20%]  (Sampling)
Chain 1: Iteration: 3000 / 10000 [ 30%]  (Sampling)
Chain 1: Iteration: 4000 / 10000 [ 40%]  (Sampling)
Chain 1: Iteration: 5000 / 10000 [ 50%]  (Sampling)
Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 17.378 seconds (Warm-up)
Chain 1:                134.398 seconds (Sampling)
Chain 1:                151.776 seconds (Total)
Chain 1:
```

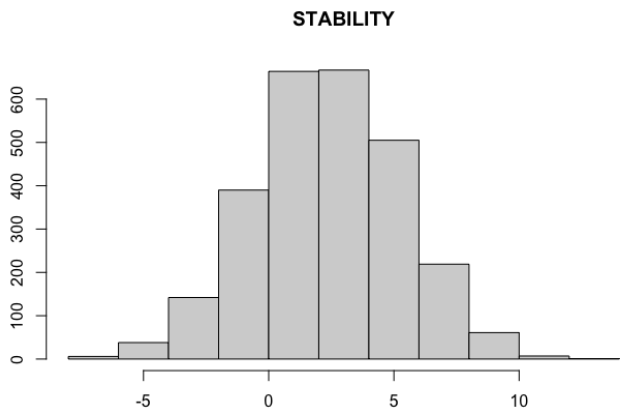


Implementation









Implementation

Optimization

```
f = optimizing(m, data=data)
opt.beta = f$par
```

주어진 모델 내에서 자동으로
파라미터 최적화

	optimal beta
LIVING_LEVEL	2.15945404
HEALTH	0.20586289
ACHIEVEMENT	2.84549596
RELATIONSHIP	1.04795425
SAFETY	-0.04356259
BELONGING	-0.15027966
STABILITY	2.39427748
LEISURE	2.14958673
LOCAL	0.79635843
JOB_SATISFACTION_무응답	0.26086052
JOB_SATISFACTION_불만족	-0.83079949
SEX	-0.26031633
AGE_30대	-1.21883155
AGE_40대	-1.07774805
AGE_50대	-1.88056196
AGE_60대	-1.39119626
LOC_동북	0.99296808
LOC_서북	0.47964465
LOC_서남	0.42998674
LOC_동남	1.19451052
EDU	-0.65156764
MARRY	0.26764076
MY_HOUSE	1.01138049
WHAT_HOUSE_연립	-0.51297827
WHAT_HOUSE_단독	-0.55860159
DEBT	-0.64162077

Implementation

3. 회귀계수 비교 | Ordinary vs Bayesian

MSE

Ordinary	Bayesian
129.9778	115.8231

Sparse?

Sparse?

The Bayesian Lasso estimates appear to be a compromise between the Lasso and ridge regression estimates; the paths are smooth, like ridge regression, but are more similar in shape to the Lasso paths, particularly when the L1 norm is relatively small.

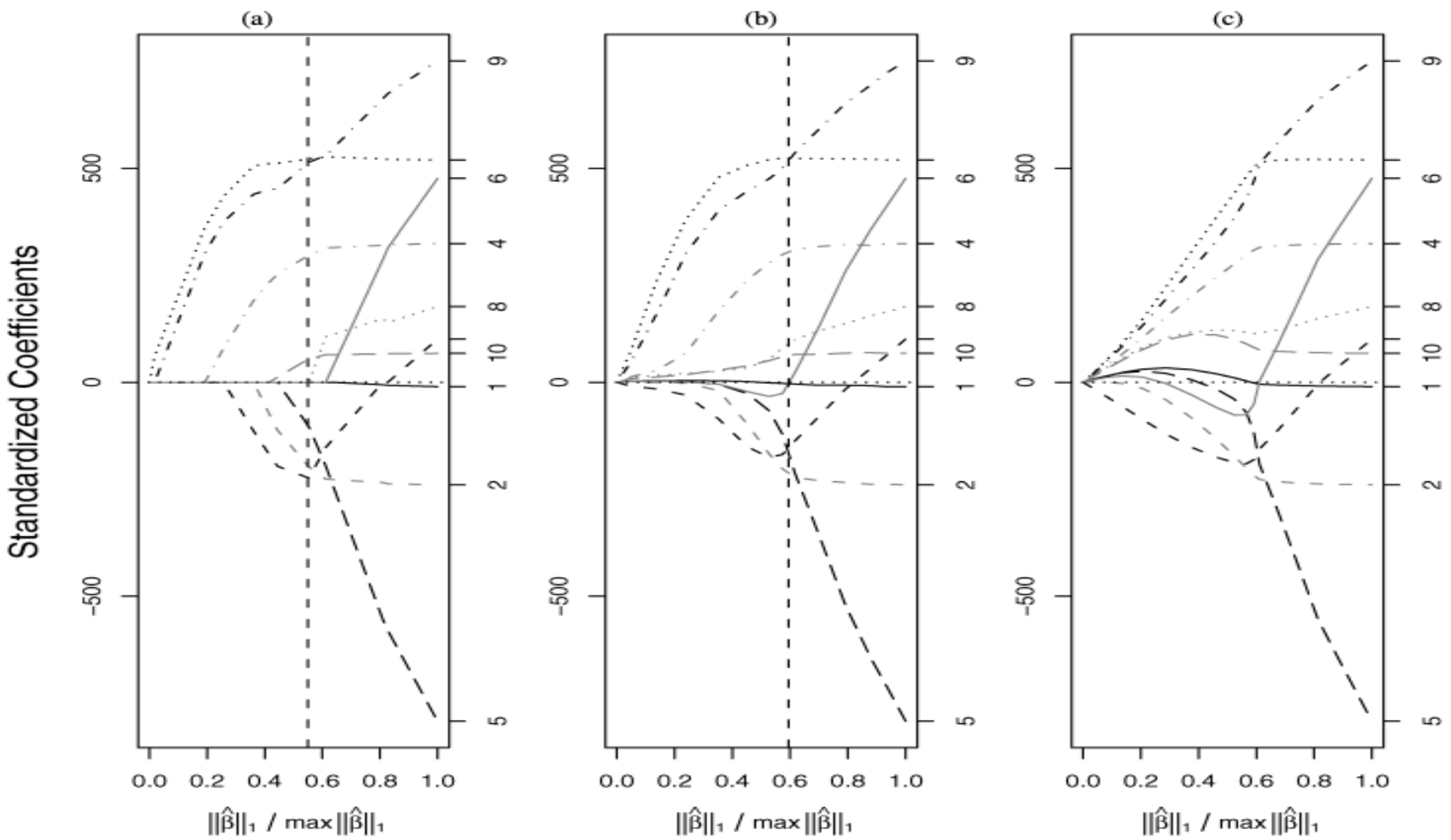
(Park and Casella, 2008)

LASSO처럼 완전히 0으로 만들어지지 않지만(smooth), **하이퍼 파라미터의 변화에 따라 LASSO와 비슷한 모양으로 회귀 계수 값이 변화한다.**

Ordinary LASSO

Bayesian LASSO

Ridge



Conclusion

결론

변수	특징	Why?
생활수준, 빚, 자가	회귀계수 ↑	경제적 요건이 행복한 삶에 중요하다
성취감, 여가, 직업 만족도	회귀계수 ↑, 직업 무응답자의 회귀계수가 작음	직업이 큰 영향을 미친다
교육	회귀계수 -	현대인의 비교 심리
연령	50대의 회귀계수 - ↓	노후 및 은퇴 준비, 심리적 압박감
거주지역	동남권 회귀계수 + ↑	강남...!

감사합니다!