

# Predicting Total Wealth: A Predictive Analysis Using the 1991 SIPP Data

Xueshan (Kevin) Peng

## Introduction

### Loading and Inspecting the Data

Let's take a look at the first 6 rows of the data.

```
data <- read.table('data_tr.txt', head = T)[-1]
head(data)
```

```
##      tw  ira e401  nifa  inc hmort  hval hequity educ male twoearn nohs hs
## 1  53550    0    0   100 28146 60150  69000   8850   12    0      0    0    1
## 2 124635    0    0 61010 32634 20000  78000  58000   16    0      0    0    0
## 3 192949 1800    0   7549 52206 15900 200000 184100   11    1      1    1    0
## 4   -513    0    0   2487 45252    0      0      0   15    0      1    0    0
## 5 212087    0    0 10625 33126 90000 300000 210000   12    0      0    0    1
## 6  24400    0    0   9000 76860 99600 120000  20400   15    0      1    0    0
##  smcol col age fsize marr
## 1     0  0  31     5     1
## 2     0  1  52     5     0
## 3     0  0  50     3     1
## 4     1  0  28     4     1
## 5     0  0  42     3     0
## 6     1  0  49     6     1
```

The variables in this dataset is defined as follows:

- tw: Total wealth (in US \$), which is defined as “net financial assets, including Individual Retirement Account (IRA) and 401(k) assets, plus housing equity plus the value of business, property, and motor vehicles.”
- ira: individual retirement account (IRA) balance (in US \$).
- e401: Binary variable, where 1 indicates eligibility for a 401(k)-retirement plan, and 0 indicates otherwise.
- nifa: Non-401k financial assets (in US \$).
- inc: Income (in US \$).
- hmort: Home mortgage (in US \$).
- hval: Home value (in US \$).
- hequity: Home value minus home mortgage.
- educ: Education (in years).
- male: Binary variable, where 1 indicates male and 0 indicates otherwise.
- twoearn: Binary variable, where 1 indicates two earners in the household, and 0 indicates otherwise.

- nohs, hs, smcol, col: Dummy variables for education levels - no high school, high school, some college, college.
- age: Age.
- fsize: Family size.
- marr: Binary variable, where 1 indicates married and 0 indicates otherwise.

```
colSums(is.na(data))
```

```
##      tw      ira    e401     nifa      inc    hmort      hval hequity      educ      male
##      0       0       0       0       0       0       0       0       0       0
## twoearn    nohs      hs    smcol      col      age    fsize      marr
##      0       0       0       0       0       0       0       0
```

```
any(duplicated(data))
```

```
## [1] FALSE
```

We can see that the data is in good shape, where categorical variables are already transformed into dummy variables. We can also see that there exists multi-collinearity in education levels (**nohs**, **hs**, **smcol**, **col**) and home-ownership-related variables (**hmort**, **hval**, and **hequity**).

```
summary(data)
```

```
##      tw      ira      e401      nifa
## Min.   : -502302  Min.   : 0  Min.   : 0.0000  Min.   : 0
## 1st Qu.:  3246    1st Qu.: 0  1st Qu.: 0.0000  1st Qu.: 200
## Median : 25225    Median : 0  Median : 0.0000  Median : 1687
## Mean   : 63629    Mean   : 3471  Mean   : 0.3714  Mean   : 13611
## 3rd Qu.: 82173    3rd Qu.: 0  3rd Qu.: 1.0000  3rd Qu.: 8875
## Max.   :1887115    Max.   :100000  Max.   :1.0000  Max.   :1425115
##      inc      hmort      hval      hequity
## Min.   : -9      Min.   : 0  Min.   : 0  Min.   : -40000
## 1st Qu.: 19413    1st Qu.: 0  1st Qu.: 0  1st Qu.: 0
## Median : 31575    Median : 8000  Median : 50000  Median : 10000
## Mean   : 37177    Mean   : 30207  Mean   : 63965  Mean   : 33757
## 3rd Qu.: 48615    3rd Qu.: 52000  3rd Qu.: 95000  3rd Qu.: 48000
## Max.   :242124    Max.   :150000  Max.   :300000  Max.   :300000
##      educ      male      twoearn      nohs
## Min.   : 1.0      Min.   : 0.0000  Min.   : 0.0000  Min.   : 0.0000
## 1st Qu.:12.0      1st Qu.: 0.0000  1st Qu.: 0.0000  1st Qu.: 0.0000
## Median :12.0      Median : 0.0000  Median : 0.0000  Median : 0.0000
## Mean   :13.2      Mean   : 0.2018  Mean   : 0.3808  Mean   : 0.1277
## 3rd Qu.:15.0      3rd Qu.: 0.0000  3rd Qu.: 1.0000  3rd Qu.: 0.0000
## Max.   :18.0      Max.   : 1.0000  Max.   : 1.0000  Max.   : 1.0000
##      hs      smcol      col      age
## Min.   : 0.0000  Min.   : 0.0000  Min.   : 0.0000  Min.   : 25.00
## 1st Qu.: 0.0000  1st Qu.: 0.0000  1st Qu.: 0.0000  1st Qu.: 32.00
## Median : 0.0000  Median : 0.0000  Median : 0.0000  Median : 40.00
## Mean   : 0.3819  Mean   : 0.2422  Mean   : 0.2482  Mean   : 41.08
## 3rd Qu.: 1.0000  3rd Qu.: 0.0000  3rd Qu.: 0.0000  3rd Qu.: 48.00
## Max.   : 1.0000  Max.   : 1.0000  Max.   : 1.0000  Max.   : 64.00
##      fsize      marr
```

```
## Min.   : 1.00   Min.   :0.0000
## 1st Qu.: 2.00   1st Qu.:0.0000
## Median : 3.00   Median :1.0000
## Mean   : 2.87   Mean   :0.6075
## 3rd Qu.: 4.00   3rd Qu.:1.0000
## Max.   :13.00   Max.   :1.0000
```

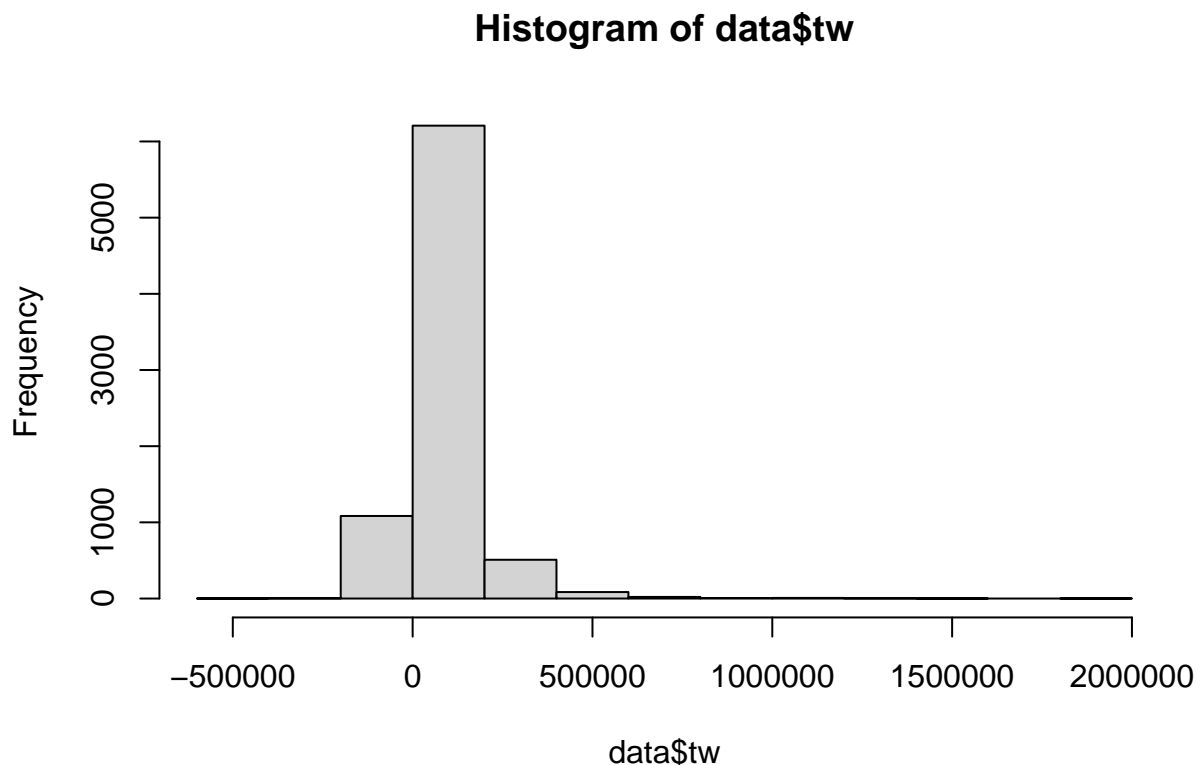
While there exist observations where total wealth is negative, it should be noted that the variable includes home equity, which can be negative, so it does not necessarily indicate that there are incorrect data entries.

The variables **ira**, **nohs**, **smcol**, **col**, and **male** exhibited a value of 0 at the 3rd quantile. They are probably a significant number of data points taking on the value of 0. Since **male** is on the list, it should also be noted that most observations are associated with female participants.

Also, the variable **tw**, **nifa**, **hmort**, and **hequity** have means that are much greater than medians, showing signs of large outliers.

In the histogram below, we can visualize the existence of outliers with enormous wealth.

```
hist(data$tw)
```



Using the graph, we can determine that removing the outliers with **tw** above \$1,000,000 would be appropriate.

```
data = subset(data, data$tw<1000000)
summary(data)
```

```
##      tw      ira      e401      nifa
```

```
## Min.    :-502302  Min.    :    0  Min.    :0.0000  Min.    :    0
## 1st Qu.:  3213   1st Qu.:    0  1st Qu.:0.0000  1st Qu.:   200
## Median : 25100   Median :    0  Median :0.0000  Median :  1649
## Mean   : 61545   Mean    : 3444  Mean    :0.3714  Mean    : 12132
## 3rd Qu.: 81774   3rd Qu.:    0  3rd Qu.:1.0000  3rd Qu.:  8771
## Max.    :967800   Max.    :100000  Max.    :1.0000  Max.    :898000
##      inc      hmort      hval      hequity
## Min.    :   -9   Min.    :    0   Min.    :    0   Min.    : -40000
## 1st Qu.: 19386   1st Qu.:    0   1st Qu.:    0   1st Qu.:    0
## Median : 31527   Median :   8000  Median : 50000  Median : 10000
## Mean   : 37043   Mean    : 30152  Mean    : 63743  Mean    : 33592
## 3rd Qu.: 48543   3rd Qu.: 51750  3rd Qu.: 95000  3rd Qu.: 48000
## Max.    :200997   Max.    :150000  Max.    :300000  Max.    :300000
##      educ      male      twoearn      nohs
## Min.    : 1.0   Min.    :0.0000  Min.    :0.000  Min.    :0.0000
## 1st Qu.:12.0   1st Qu.:0.0000  1st Qu.:0.000  1st Qu.:0.0000
## Median :12.0   Median :0.0000  Median :0.000  Median :0.0000
## Mean   :13.2   Mean    :0.2019  Mean    :0.381  Mean    :0.1278
## 3rd Qu.:15.0   3rd Qu.:0.0000  3rd Qu.:1.000  3rd Qu.:0.0000
## Max.    :18.0   Max.    :1.0000  Max.    :1.000  Max.    :1.0000
##      hs      smcol      col      age
## Min.    :0.0000  Min.    :0.0000  Min.    :0.0000  Min.    :25.00
## 1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:32.00
## Median :0.0000  Median :0.0000  Median :0.0000  Median :40.00
## Mean   :0.3822  Mean    :0.2422  Mean    :0.2478  Mean    :41.06
## 3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:0.0000  3rd Qu.:48.00
## Max.    :1.0000  Max.    :1.0000  Max.    :1.0000  Max.    :64.00
##      fsize      marr
## Min.    : 1.00   Min.    :0.0000
## 1st Qu.: 2.00   1st Qu.:0.0000
## Median : 3.00   Median :1.0000
## Mean   : 2.87   Mean    :0.6071
## 3rd Qu.: 4.00   3rd Qu.:1.0000
## Max.    :13.00   Max.    :1.0000
```

## Testing and Removing Multi-collinearity

Let's test whether removing different educational level predictors affect my model's performance, gauged by (MSPE). For simplicity sake, I did not use k-fold cross validation.

```
k <- 10
set.seed(123)
rand <- sample(nrow(data), floor(nrow(data)/k))
train <- setdiff(c(1:nrow(data)), rand)
y_rand <- data$tw[rand]

regnohs <- lm(tw ~ 1 + hs + smcol + col, data = data[train,])
reghs <- lm(tw ~ 1 + nohs + smcol + col, data = data[train,])
regsmcol <- lm(tw ~ 1 + nohs + hs + col, data = data[train,])
regcol <- lm(tw ~ 1 + nohs + hs + smcol, data = data[train,])

prnohs <- predict(regnohs, newdata = data[rand,])
prhs <- predict(reghs, newdata = data[rand,])
```

```
prsmcol <- predict(regsmcol, newdata = data[rand,])
prcol <- predict(regcol, newdata = data[rand,])

MSEnohs <- mean((y_rand-prnohs)^2)
MSEhs <- mean((y_rand-prhs)^2)
MSEsmcol <- mean((y_rand-prsmcol)^2)
MSEcol <- mean((y_rand-prcol)^2)

c(MSEnohs, MSEhs, MSEsmcol, MSEcol)
```

```
## [1] 9119936474 9119936474 9119936474 9119936474
```

No difference in performance is found between removing different terms for multi-collinearity. For interpretability, we choose to remove **hs** for education level.

## More Feature Selections

Since **hequity** represents home value minus home mortgage, it is intuitively a better predictor of total wealth than **hval** or **hmort** itself. Hence, choosing **hequity** over **hval** and **hmort** is the more sensible choice.

Including years of education (**educ**) along with education levels is redundant. Considering that diplomas are usually much more important than years of education, prioritizing education level over years of education is appropriate.

```
data <- data[, !(names(data) %in% c("hs", "hval", "hmort", "educ"))]
```

## Creating a Linear Baseline Model

### Using Lasso and Forward/Backward Stepwise Selection

We will strive to create a linear baseline model. This will serve as a baseline to compare to when we later add nonlinear transformations and interaction terms.

For this approach, we are going to include all the features in the dataset. We will let the feature selection algorithms, Lasso and Stepwise Selection, to select the features for us.

For better accuracy, I employed 10-fold cross validation. Leave-one-out cross validation would yield a even more accurate result, but doing so on a dataset containing 7919 observations would take too much computational power.

```
library(MASS)
library(glmnet)

n <- nrow(data)
k <- 10
set.seed(123)
id <- sample(rep(1:k, length=n))
MSPE.stepwise_backward <- MSPE.stepwise_forward <- MSPE.lasso <- rep(NA, k)

for (f in 1:k){
  test <- (id == f)
```

```

train <- (id != f)

# Stepwise
full <- lm(tw ~ ., data=data[train,])
null <- lm(tw ~ 1, data=data[train,])

forward <- stepAIC(null, scope=list(lower=null, upper=full), trace = FALSE, direction='forward')
backward <- stepAIC(full, scope=list(lower=null, upper=full), trace = FALSE, direction='backward')

pr.stepwise_forward <- predict(forward, newdata=data[test,])
pr.stepwise_backward <- predict(backward, newdata=data[test,])

# Lasso
response_var <- "tw"
y <- data[[response_var]]
X <- as.matrix(data[, !(names(data) %in% response_var)])
X.train <- as.matrix(X[train,])
y.train <- y[train]
X.test <- X[test,]

lasso.cv <- cv.glmnet(x = X.train, y = y.train, nfolds = k, alpha = 1)
lasso <- glmnet(x = X.train, y = y.train, lambda = lasso.cv$lambda.min, alpha = 1)

pr.lasso <- predict(lasso, newx=X.test)

# MSPE
MSPE.stepwise_forward[f] <- mean((y[test] - pr.stepwise_forward)^2)
MSPE.stepwise_backward[f] <- mean((y[test] - pr.stepwise_backward)^2)
MSPE.lasso[f] <- mean((y[test] - pr.lasso)^2)
}

# Check MSPE
c(mean(MSPE.lasso), mean(MSPE.stepwise_forward), mean(MSPE.stepwise_backward))

## [1] 1427662119 1427849850 1427849850

```

As shown in the results above, Lasso yielded a lower MSPE than forward or backward stepwise selection.

Let's now inspect the coefficients that Lasso chose and their associated p-values.

Since Lasso performs both variable selection and shrinkage, leading to biased coefficient estimates, traditional significance tests for coefficients (like p-values) are not straightforwardly available. Therefore, we use the *hdi* (High Dimensional Inference) package to approximate the p-values.

```

library(hdi)

response_var <- "tw"
y <- data[[response_var]]
X <- as.matrix(data[, !(names(data) %in% response_var)])

lasso_cv <- cv.glmnet(X, y, alpha = 1)
best_lambda <- lasso_cv$lambda.min

lasso_model <- glmnet(X, y, lambda = best_lambda, alpha = 1)

```

```
lasso_inference <- hdi::lasso.proj(X, y)
```

```
print(lasso_inference$pval)
```

```
##          ira          e401          nifa          inc          hequity
## 7.802417e-203  6.990767e-18  0.000000e+00  6.008738e-24  0.000000e+00
##          male          twoearn          nohs          smcol          col
## 1.497555e-03  1.286912e-07  8.743909e-01  2.139569e-01  8.645575e-01
##          age          fsize          marr
## 9.360306e-08  7.926808e-01  5.224558e-02
```

```
print(coef(lasso_model))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##          s0
## (Intercept) -1.576371e+04
## ira          1.606418e+00
## e401          7.935739e+03
## nifa          1.102051e+00
## inc          2.500733e-01
## hequity       1.081333e+00
## male         3.169643e+03
## twoearn      -5.386380e+03
## nohs         .
## smcol        1.006324e+03
## col          .
## age         2.470886e+02
## fsize        .
## marr        1.473061e+03
```