# HW4_R

February 16, 2020

[3]:
```
set.seed(123)
library(margins)
library(repr)
library(ggplot2)
library(rsample)
library(dplyr)
library(purrr)
library(splines)
library(msmtools)
options(repr.plot.width=5, repr.plot.height=4)
```

```
Loading required package: tidyr

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

[80]:
```
gss_train = read.csv('data/gss_train.csv')
gss_test = read.csv('data/gss_test.csv')
```

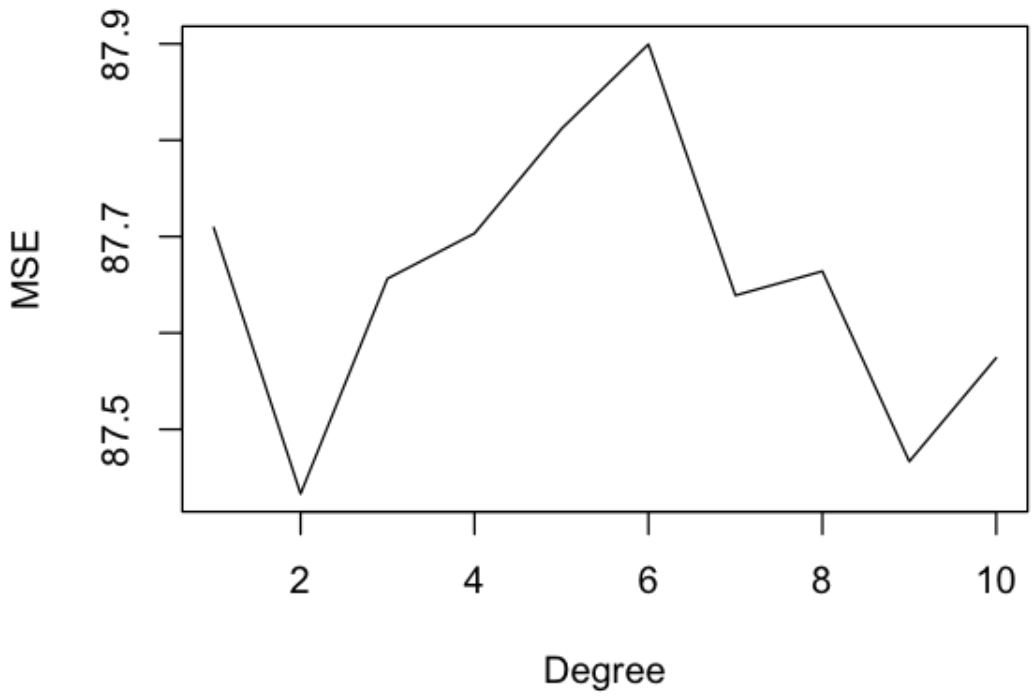
## 0.1 Egalitarianism and Income

### 0.1.1 1. Polynomial Regression

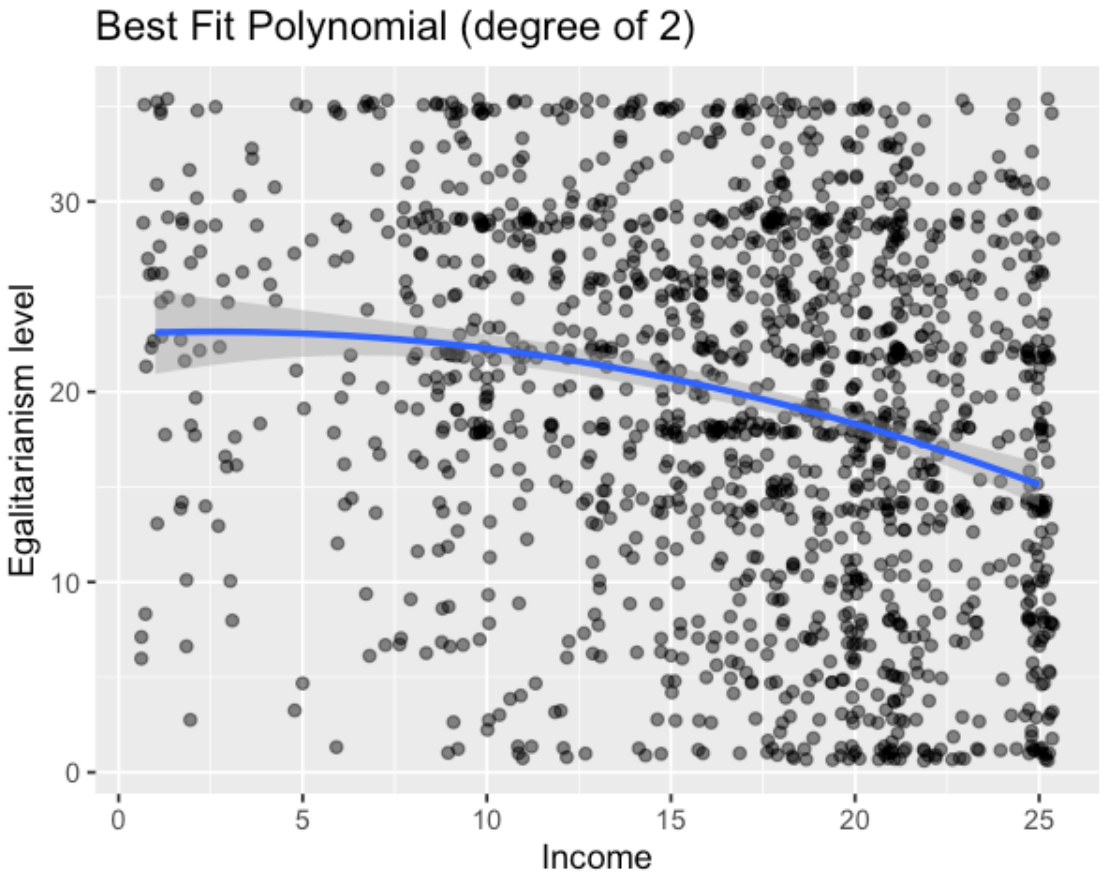
[4]:
```
k = 10
fold = sample(k, nrow(gss_train), replace = TRUE)

mse = c()
for(j in 1:k){
```

```
    sub_mse = c()
    for(i in 1:k){
        test_set = gss_train[fold==i,]
        train_set = gss_train[fold!=i,]
        model = lm(egalit_scale ~ poly(income06, j), data = train_set)
        pred = predict(model, test_set)
        sub_mse = c(sub_mse, mean((pred - test_set$egalit_scale)^2))
    }
    mse = c(mse, mean(sub_mse))
}
```

[4]:
```
plot(mse, type = 'l', xlab = 'Degree', ylab = 'MSE')
```




[133]:
```
ggplot(gss_train, aes(income06, egalit_scale)) +
    geom_jitter(alpha = 0.5) +
    stat_smooth(method = 'lm', formula = y ~ poly(x, degree=2)) +
    labs(title = 'Best Fit Polynomial (degree of 2)',
         x = 'Income', y = 'Egalitarianism level')
```



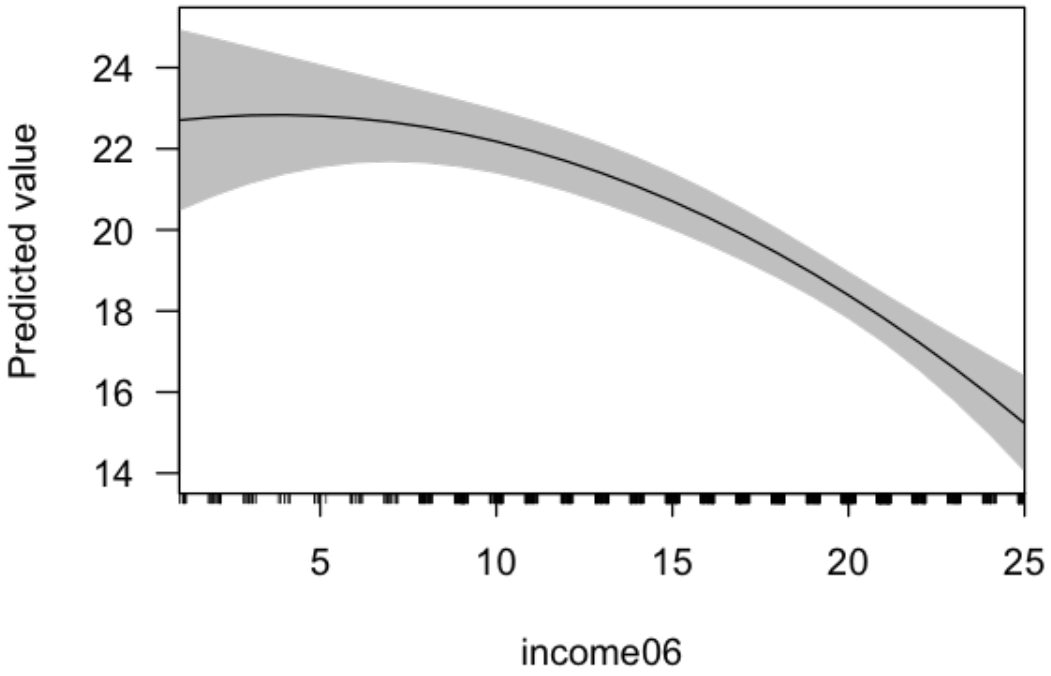

[134]:
```
best_model = lm(egalit_scale ~ stats::poly(income06, 2), data = train_set)
summary(margins(best_model))
```

| factor | AME | SE | z | p | lower | upper |
|---|---|---|---|---|---|---|
| income06 | -0.4422506 | 0.04976301 | -8.887136 | 6.270432e-19 | -0.5397843 | -0.3447169 |

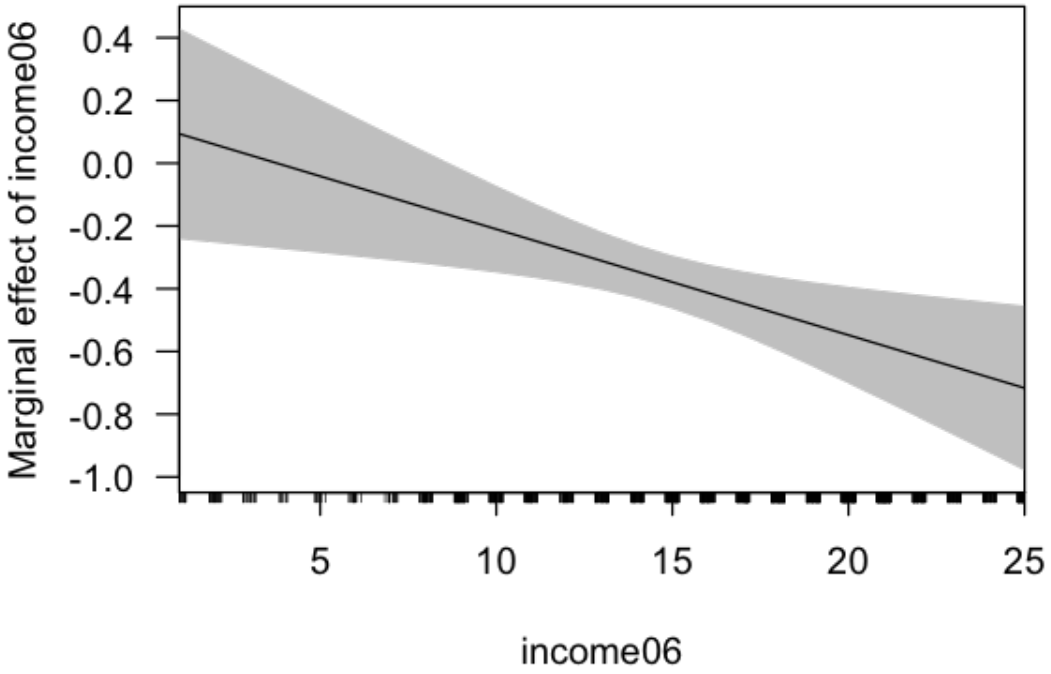
[57]:
```
cplot(best_model, 'income06', what = 'prediction')
```

```
   xvals    yvals    upper    lower
1      1 22.70467 24.93877 20.47057
2      2 22.78141 24.72831 20.83451
3      3 22.82438 24.51346 21.13530
4      4 22.83360 24.29562 21.37159
5      5 22.80906 24.07630 21.54182
6      6 22.75076 23.85695 21.64457
7      7 22.65870 23.63828 21.67911
8      8 22.53288 23.41943 21.64632
9      9 22.37329 23.19701 21.54957
10    10 22.17995 22.96501 21.39489
11    11 21.95285 22.71570 21.19000
```

```
12     12 21.69199 22.44122 20.94276
13     13 21.39737 22.13490 20.65984
14     14 21.06899 21.79188 20.34610
15     15 20.70685 21.40932 20.00439
16     16 20.31095 20.98645 19.63546
17     17 19.88130 20.52486 19.23773
18     18 19.41788 20.02914 18.80661
19     19 18.92070 19.50777 18.33363
20     20 18.38976 18.97336 17.80616
```




[56]:
```
cplot(best_model, 'income06', what = 'effect')
```



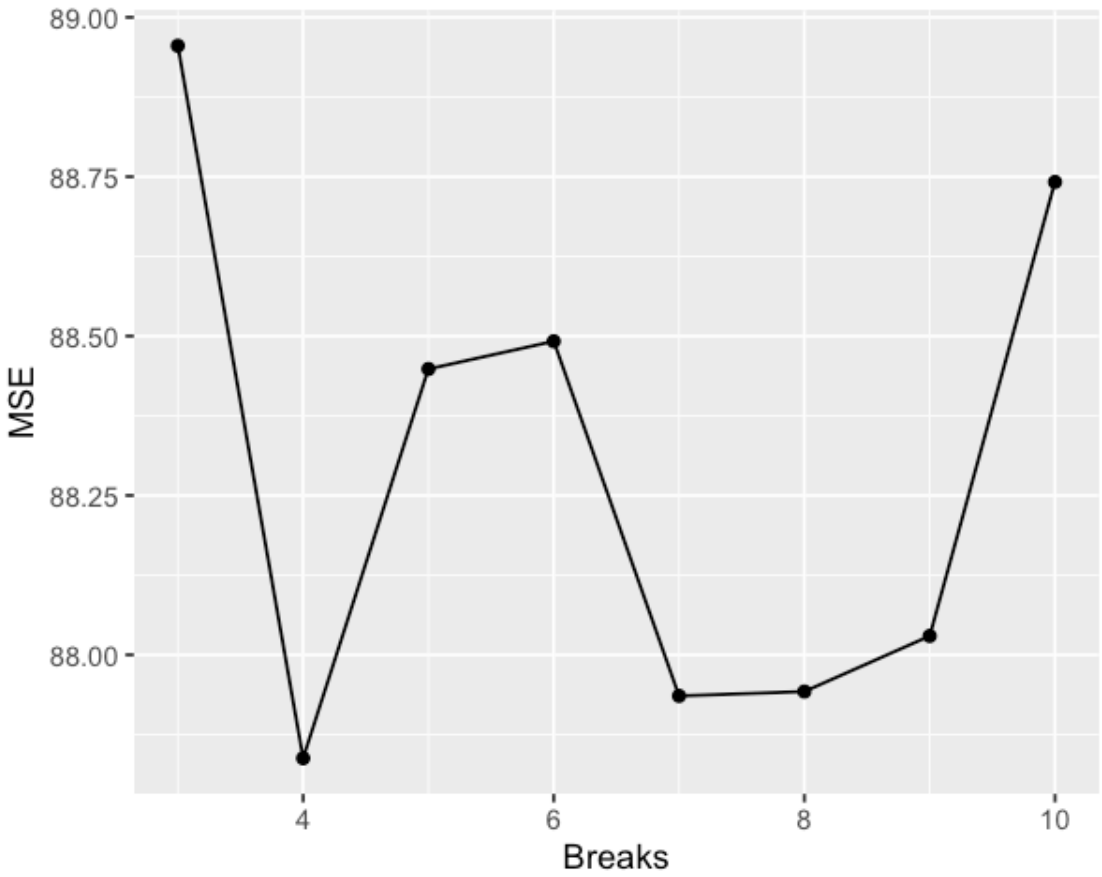

As a conclusion, the best polynomial model has 2 degrees, although the increase of MSE is not monotonic as the degree increases. Also, as `income06` increases, its marginal effect decreases from positive to negative. The AME for the model of 2 degrees is -0.4369. This suggests that on average, one unit increase in income decreases the level of egalitarianism by about 0.4 unit. In another word, the richer people tend to be care less about equal distribution of the wealth.

### 0.1.2 2. Step Function

[14]:
```
mse = c()
for(j in 3:k){
    sub_mse = c()
    for(i in 1:k){
        test_set = gss_train[fold==i,]
        train_set = gss_train[fold!=i,]

        labs = levels(cut(gss_train$income06, j))
        breaks = unique(c(as.numeric(sub("\\((.+),.*", "\\1", labs)),
            as.numeric(sub("[^,]*,([^]]*)\\]", "\\1", labs))))
```

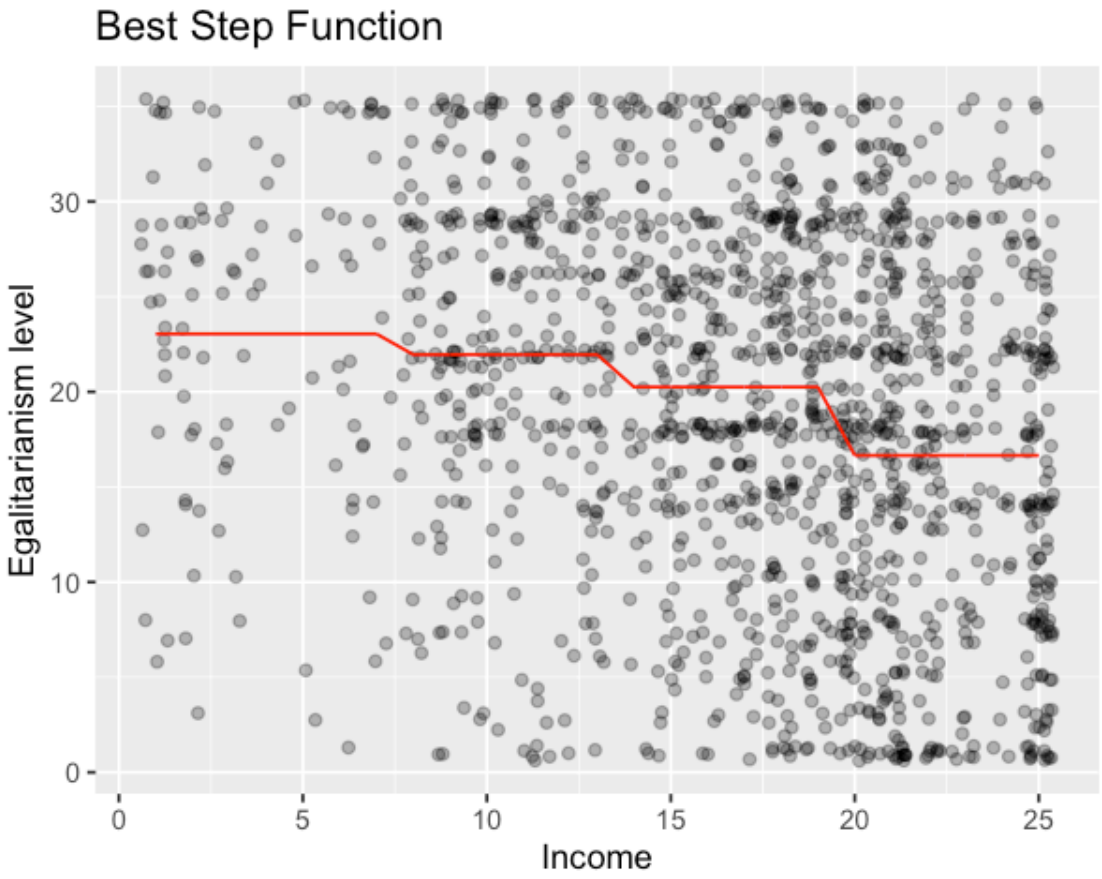
```
        model = lm(egalit_scale~cut(income06,unique(breaks)), data = train_set)
        pred = predict(model, test_set)

        sub_mse = c(sub_mse, mean((pred - test_set$egalit_scale)^2))
    }
    mse = c(mse, mean(sub_mse))
}
```

[15]:
```
step_sum = data.frame('Breaks' = 3:k, "MSE"=mse)
ggplot(step_sum, aes(Breaks, MSE))+ geom_line()+geom_point()
```




[50]:
```
labs = levels(cut(gss_train$income06, 4))
breaks = unique(c(as.numeric(sub("\\((.+),.*", "\\1", labs)), as.
 ↪numeric(sub("[^,]*,([^]]*)\\]", "\\1", labs))))
best_model = lm(egalit_scale~cut(income06,unique(breaks)), data = gss_train)
pred = predict(best_model, gss_train)
df_pred = data.frame('income06' = gss_train$income06, 'pred' = pred)
```

[54]:
```
ggplot(gss_train, aes(income06, egalit_scale)) +
    geom_jitter(alpha = 0.3) +
    geom_line(data = df_pred, aes(income06, pred), color = 'red') +
    labs(title = 'Best Step Function',
         x = 'Income', y = 'Egalitarianism level')
```



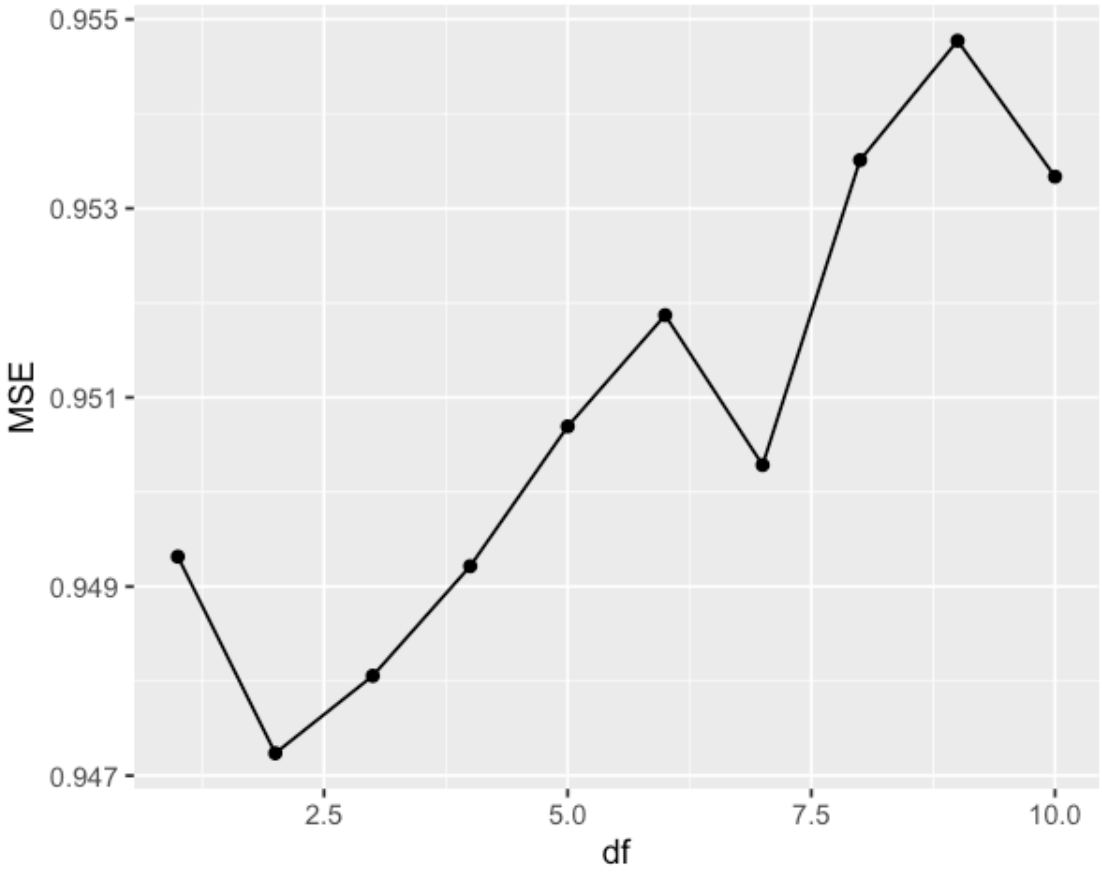

As a summary, the best step function has 4 breaks (5 intervals). Still, as income increases, the predicted level of egalitarianism decreases step by step.

### 0.1.3 Natural Regression Spline

[59]:
```
mse = c()
for(j in 1:k){
    sub_mse = c()
    for(i in 1:k){
        test_set = gss_train[fold==i,]
        train_set = gss_train[fold!=i,]
        model = lm(egalit_scale ~ ns(income06, df = j), data = train_set)
```

```
        pred = predict(model, test_set)
        sub_mse = c(sub_mse, mean((pred - test_set$egalit_scale)^2))
    }
    mse = c(mse, mean(sub_mse))
}
```

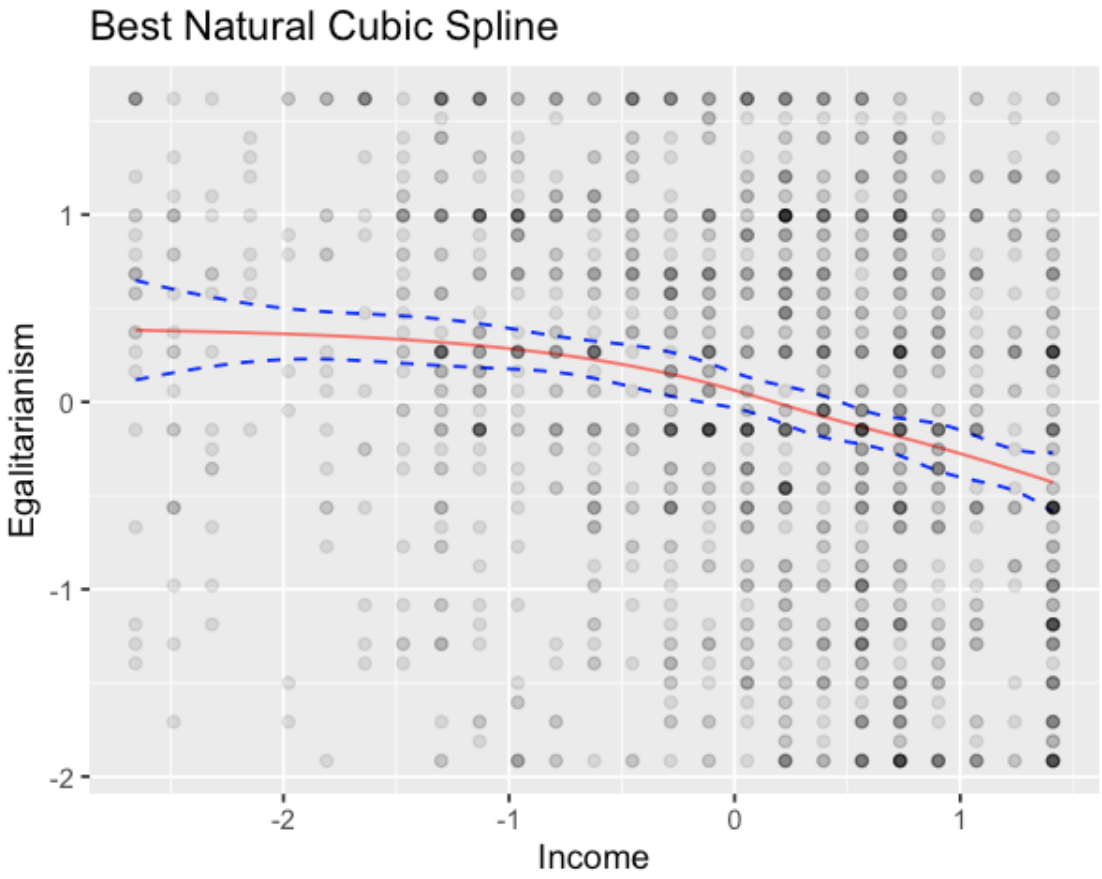
[60]:
```
step_sum = data.frame('df' = 1:k, "MSE"=mse)
ggplot(step_sum, aes(df, MSE))+ geom_line()+geom_point()
## best knots is 2 (df = 5)
```




[62]:
```
model = glm(egalit_scale ~ ns(income06, df = 5), data = gss_train) %>%
        cplot("income06", what = "prediction", n = 100, draw = FALSE)
model %>% ggplot(aes(x = xvals)) +
        geom_line(aes(y = yvals, color = 'red')) +
        geom_line(aes(y = upper), linetype = 2, color = 'blue') +
        geom_line(aes(y = lower), linetype = 2, color = 'blue') +
        geom_point(data = gss_train, aes(income06, egalit_scale), alpha = 0.1) +
        labs(x = "Income", y = "Egalitarianism", title = "Best Natural Cubic␣
 ↪Spline") +
```

```
        theme(legend.position = "none")
```

```
       xvals     yvals     upper     lower
1  -2.655726 0.3829889 0.6488579 0.1171198
2  -2.614648 0.3819858 0.6370112 0.1269603
3  -2.573571 0.3809774 0.6253548 0.1366000
4  -2.532494 0.3799584 0.6139188 0.1459980
5  -2.491416 0.3789235 0.6027349 0.1551122
6  -2.450339 0.3778675 0.5918363 0.1638987
7  -2.409262 0.3767850 0.5812575 0.1723125
8  -2.368185 0.3756707 0.5710341 0.1803073
9  -2.327107 0.3745193 0.5612028 0.1878359
10 -2.286030 0.3733256 0.5518001 0.1948512
11 -2.244953 0.3720843 0.5428621 0.2013064
12 -2.203875 0.3707900 0.5344232 0.2071567
13 -2.162798 0.3694374 0.5265143 0.2123605
14 -2.121721 0.3680213 0.5191614 0.2168813
15 -2.080644 0.3665364 0.5123831 0.2206897
16 -2.039566 0.3649774 0.5061890 0.2237658
17 -1.998489 0.3633389 0.5005770 0.2261009
18 -1.957412 0.3616158 0.4955323 0.2276992
19 -1.916334 0.3598026 0.4910261 0.2285790
20 -1.875257 0.3578941 0.4870157 0.2287725
```




As shown from the two figures above, the best natural regression spline has 2 knots (df = 3). This model is smoother compared to the previous two models and basically demonstrates the same information that, as income increases, people tend to be less egalitarian.

### 0.1.4 4. Egalitarianism and Everything

[98]:
```
options(warn=-1)
```

**Data Pre-Processing**

[175]:
```
standardize = function(data){
    df = data %>%
    mutate_if(is.numeric, c) %>%
    mutate_if(is.numeric, scale)
    df$relig = data$relig %in% c('CATHOLIC', 'PROTESTANT', 'CHRISTIAN')
    df$marital = data$marital  == 'Married'

    a = as.character(df$attend)
```

```
    a[a == 'Never'] = 0
    a[(a == '<Once/yr')|(a == 'Once/yr')] = 1
    a[a == 'Sev times/yr'] = 2
    a[a == 'Once/mo'] = 3
    a[a == '2-3 times /mo'] = 4
    a[(a == 'Every wk')|(a =='>Once/wk')|(a == 'Nrly evry wk')] = 5
    df$attend = as.numeric(a)

    a = as.character(df$polviews)
    a[a == 'ExtrmLib'] = 0
    a[a == 'Liberal'] = 1
    a[a == 'SlghtLib'] = 2
    a[a == 'Moderate'] = 3
    a[a == 'SlghtCons'] = 4
    a[a == 'Conserv'] = 5
    a[a == 'ExtrmCons'] = 6
    df$polviews = as.numeric(a)


    a = as.character(df$degree)
    a[a == '<HS'] = 0
    a[a == 'HS'] = 1
    a[a == 'Junior Coll'] = 2
    a[a == 'Bachelor deg'] = 3
    a[a == 'Graduate deg'] = 4
    df$degree = as.numeric(a)

    a = as.character(df$news)
    a[a == 'NEVER'] = 0
    a[a == 'LESS THAN ONCE WK'] = 1
    a[a == 'ONCE A WEEK'] = 2
    a[a == 'FEW TIMES A WEEK'] = 3
    a[a == 'EVERYDAY'] = 4
    df$news = as.numeric(a)

    a = as.character(df$pray)
    a[a == 'NEVER'] = 0
    a[a == 'ONCE A WEEK'] = 1
    a[a == 'LT ONCE A WEEK'] = 2
    a[a == 'SEVERAL TIMES A WEEK'] = 3
    a[a == 'ONCE A DAY'] = 4
    a[a == 'SEVERAL TIMES A DAY'] = 5
    df$pray = as.numeric(a)

    df
}
```

```
gss_train = standardize(gss_train)
gss_test = standardize(gss_test)
```

### 0.1.5 Linear Regression

There is no need to tune anything for linear regression. The average of 10-fold CV MSE is 0.68, and the whole model MSE is 0.69. When all predictors are considered, egalitarianism is siginificantly related to political views: the more liberal, the more egalitarinian. Younger and poorer people, aside from political views, are also more egalitarinian.

[176]:
```
k = 10
fold = sample(k, nrow(gss_train), replace = TRUE)

mse = c()
for(i in 1:k){
    train_set = gss_train[fold!=i,]
    test_set = gss_train[fold==i,]

    model = lm(egalit_scale ~ ., data = train_set)
    pred = predict(model, test_set)
    mse = c(mse, mean((pred - test_set$egalit_scale)^2))
}

print(mean(mse))
```

```
[1] 0.6582933
```

[177]:
```
model = lm(egalit_scale ~ ., data = gss_train)
pred = predict(model, gss_test)
print(mean((pred - gss_test$egalit_scale)^2))
summary(model)
```

```
[1] 0.7034419
```

```
Call:
lm(formula = egalit_scale ~ ., data = gss_train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.98707 -0.51728 -0.00522  0.53809  2.10679

Coefficients: (1 not defined because of singularities)
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.053176   0.276427   0.192 0.847481
age                       -0.109088   0.027782  -3.927 9.03e-05 ***
attend                    -0.003953   0.027399  -0.144 0.885312
authoritarianism           0.005560   0.024779   0.224 0.822475
```

```
blackYes                    0.110447   0.071937   1.535 0.124928
bornYES                    -0.015159   0.070753  -0.214 0.830384
childs                      0.048020   0.024931   1.926 0.054295 .
colathNOT ALLOWED           0.050517   0.061864   0.817 0.414301
colracNOT ALLOWED          -0.002744   0.055899  -0.049 0.960849
colcomNOT FIRED            -0.003790   0.056887  -0.067 0.946891
colmilNOT ALLOWED          -0.127958   0.058364  -2.192 0.028513 *
colhomoNOT ALLOWED          0.095543   0.080219   1.191 0.233841
colmslmYes, allowed        -0.016024   0.057970  -0.276 0.782271
con_govt                   -0.031098   0.022631  -1.374 0.169621
degree                     -0.053523   0.028019  -1.910 0.056301 .
evangelicalLow              0.099281   0.115251   0.861 0.389149
evangelicalMod              0.081066   0.081474   0.995 0.319912
grassNOT LEGAL             -0.256908   0.056780  -4.525 6.56e-06 ***
happyPRETTY HAPPY          -0.061620   0.063865  -0.965 0.334793
happyVERY HAPPY            -0.126962   0.071737  -1.770 0.076972 .
hispanic_2Yes               0.006849   0.072819   0.094 0.925078
homosexALWAYS WRONG         0.009700   0.158882   0.061 0.951328
homosexNOT WRONG AT ALL    -0.021470   0.159768  -0.134 0.893118
homosexSOMETIMES WRONG      0.010654   0.179605   0.059 0.952707
income06                   -0.080075   0.026917  -2.975 0.002981 **
maritalTRUE                 0.087581   0.050698   1.728 0.084296 .
modeOVER THE PHONE          0.035322   0.064713   0.546 0.585274
news                       -0.024851   0.022857  -1.087 0.277105
owngunREFUSED              -0.068371   0.191685  -0.357 0.721383
owngunYES                  -0.093609   0.051789  -1.807 0.070897 .
partyid_3Ind               -0.226527   0.053794  -4.211 2.70e-05 ***
partyid_3Rep               -0.376205   0.078977  -4.763 2.10e-06 ***
polviews                   -0.248724   0.025681  -9.685  < 2e-16 ***
pornlaw2Not illegal to all -0.065707   0.060931  -1.078 0.281043
pray                       -0.018983   0.027359  -0.694 0.487910
pres08Obama                 0.420867   0.067121   6.270 4.78e-10 ***
reborn_rYes                 0.057367   0.095652   0.600 0.548772
religTRUE                         NA         NA      NA       NA
science_quiz               -0.008975   0.028491  -0.315 0.752797
sexMale                    -0.140215   0.049865  -2.812 0.004993 **
sibs                        0.046377   0.023256   1.994 0.046322 *
social_connect              0.006305   0.023081   0.273 0.784774
social_cons3Liberal        -0.028398   0.080271  -0.354 0.723555
social_cons3Mod             0.036945   0.063176   0.585 0.558771
southSouth                 -0.038128   0.047539  -0.802 0.422674
spend3Liberal               0.177557   0.051434   3.452 0.000573 ***
spend3Mod                   0.077658   0.056824   1.367 0.171950
teensexALWAYS WRONG         0.054632   0.068509   0.797 0.425325
teensexNOT WRONG AT ALL     0.008988   0.114279   0.079 0.937322
teensexSOMETIMES WRONG      0.045572   0.089005   0.512 0.608720
tolerance                  -0.092893   0.037595  -2.471 0.013594 *
tvhours                     0.052055   0.022957   2.268 0.023508 *
```

```
vetyearsLESS THAN 2 YRS     0.089385   0.159210   0.561 0.574595
vetyearsMORE THAN 4 YRS     0.056080   0.144789   0.387 0.698579
vetyearsNONE                0.070798   0.098646   0.718 0.473058
wordsum                     0.011214   0.026217   0.428 0.668899
zodiacARIES                -0.182145   0.105577  -1.725 0.084705 .
zodiacCANCER               -0.043771   0.105318  -0.416 0.677763
zodiacCAPRICORN            -0.055798   0.103847  -0.537 0.591137
zodiacGEMINI               -0.153646   0.100733  -1.525 0.127414
zodiacLEO                  -0.088359   0.097377  -0.907 0.364353
zodiacLIBRA                -0.127554   0.099621  -1.280 0.200615
zodiacPISCES               -0.150056   0.104707  -1.433 0.152047
zodiacSAGITTARIUS          -0.121600   0.103519  -1.175 0.240324
zodiacSCORPIO              -0.177538   0.104357  -1.701 0.089114 .
zodiacTAURUS               -0.019748   0.100745  -0.196 0.844621
zodiacVIRGO                 0.014054   0.103855   0.135 0.892378
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7939 on 1415 degrees of freedom
Multiple R-squared:  0.3975,Adjusted R-squared:  0.3698
F-statistic: 14.36 on 65 and 1415 DF,  p-value: < 2.2e-16
```

### 0.1.6 Elastic Net Regression

With 10 fold elastic net, the best alpha is 0.6 and the best lambda is 0.03345934. The train CV MSE is 0.6739 and the final MSE is 0.7457. This is almost the same–even a bit worse than the linear regression.

[188]:
```
library(caret)
```

[186]:
```
folds = trainControl(method = "cv", number = 10)
gss_elnet = train(
    egalit_scale ~ ., data = gss_train, method = "glmnet",
    trControl = folds,
    tuneLength = 10
)
best_alpha = gss_elnet$bestTune$alpha
best_lambda = gss_elnet$bestTune$lambda
```

[187]:
```
elnet_best = train(egalit_scale ~ ., data = gss_test,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = best_alpha,
                              lambda = best_lambda))
elnet_best
```

```
glmnet

493 samples
 44 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 493, 493, 493, 493, 493, 493, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.8635278  0.2678851  0.6900269

Tuning parameter 'alpha' was held constant at a value of 0.6
Tuning
 parameter 'lambda' was held constant at a value of 0.03345934
```

[202]:
```
mean(gss_elnet$results$RMSE^2)
elnet_best$results$RMSE^2
```

0.673869032287275

0.745680303330922

**PCR**

[ ]:
```

```