Model Selection

Andrew Zieffler

Import Data and Fit Model

```
# Import data
usa = read_csv("~/Desktop/state-2019.csv")

  state      life_expectancy population income illiteracy murder_rate hs_grad frost  area
  <chr>                <dbl>      <dbl>  <dbl>      <dbl>       <dbl>   <dbl> <dbl> <dbl>
1 Alabama               75.4       49.0   23.6       14.8         7.8    85.3  42.8  5.08
2 Alaska                78.8        7.32  33.1        9.2         8.2    92.4 200.  57.1
3 Arizona               79.9       72.8   25.7       13.1         5.1    86.5  90.8 11.4
4 Arkansas              75.9       30.2   22.9       13.7         7.2    85.6  62.5  5.21
5 California            81.6      395.    30.4       23.1         4.4    82.5  27.5 15.6
6 Colorado              80.5       57.6   32.4        9.9         3.7    91.1 168.  10.4


# Create data frame that includes all rows/columns except the state names
usa2 = usa[ , -1])

# Create data frame of standardized variables (no state names)
z_usa = usa[ , -1]) %>%
  data.frame()
```

Our modeling goal is to explore the predictors of life expectancy. We have no a priori hypotheses about which predictors should be included in the model nor about the importance of these predictors.

Predict Life Expectancy

```
# Use all variables as predictors
lm.all = lm(life_expectancy ~ ., data = usa2)


# Examine output
tidy(lm.all)

  term          estimate   std.error statistic   p.value
  <chr>            <dbl>       <dbl>     <dbl>     <dbl>
1 (Intercept) 79.8876     12.0844       6.61078  4.26233e-8
2 population   0.0806344   0.0390966    2.06244  4.51001e-2
3 income       0.160090    0.0561521    2.85101  6.61109e-3
4 illiteracy  -0.157523    0.0828696   -1.90086  6.38833e-2
5 murder      -0.129908    0.105719    -1.22880  2.25678e-1
6 hs_grad     -0.0364175   0.137427    -0.264995 7.92251e-1
7 frost       -0.00622470  0.00693942  -0.897006 3.74598e-1
8 area         0.0189268   0.0289705    0.653311 5.16955e-1
```

Variables that are positively related to life expectancy are population, area, and income.

Variables that are negatively related to life expectancy are illiteracy rate, murder rate, days with a temperature below freezing, and graduation rate.

Only population, income, and maybe illiteracy rate are statistically significant.

Model Evaluation and Predictor Selection

Evaluating and Selecting Models

- Predictors have, so far, been selected a priori (based on substantive work)
- Most of our analytic work has focused on:
  - Identifying functional form of predictors
  - Examining and fixing problems with assumptions
- When theory does not specify the predictor set, variable (model) selection is an important analytical problem

When we evaluate models (or predictors within a model) we do so by examining some **criterion/metric of the model's/predictor's performance**. For example, one criterion we use is the *p*-value. Another criterion we use at the model-level is the R^2 value.

Once we have identified the criterion/metric to measure performance, we then need to determine **how to select a model using this metric**. For example, with a *p*-value, we might retain a predictor in the model if the *p*-value is less than some a priori defined threshold (e.g., $p < .05$).

The threshold level will be different for exploratory and confirmatory analyses.

A third thing that we need to consider is the **model-building strategy** that we are going to employ as we use our metric to select a model(s). Will we add one predictor at a time into the model? In which order? Maybe we will include all the predictors and drop those that don't meet our criteria. Should we drop them all at once or one-at-a-time? Once we drop them should we reconsider including them at other stages of the model-building process. When should we check collinearity? Assumptions?

Model Purpose

<u>Description</u>

- Purpose of the model is to describe the data, or to understand a complex system
- *Goal* is to choose the smallest number of predictors that account for a substantial amount of variation in the outcome
- Our goal has two competing requirements as explaining more variation in the outcome generally requires more predictors

<u>Inference/Prediction</u>

- Purpose of the model is to predict the outcome/mean outcome for new cases or make inferences about the effects of predictors
- *Goal* is prediction/inferential accuracy.
- Performance in our sample is not as important as performance in future (out-of-sample) cases

The model’s purpose determines how we will measure “model success”. Each purpose points to a different criteria to use in the model evaluation process.

Model Evaluation Criteria

There are several criteria that have been proposed to evaluate model performance when the purpose is description.

Sum of Squared Residuals

$$\text{SSE} = \sum (Y_i - \hat{Y}_i)^2$$

Residual Mean Square

$$\text{RMS} = \frac{\text{SSE}}{\text{df}_{\text{residual}}}$$

When using criteria that measure the residual error (SSE, RMS), we want to select a model that minimize these values .

R2

$$R^2 = \frac{\text{SST} - \text{SSE}}{\text{SST}}$$

$$R^2 = 1 - (\text{df}_{\text{Residual}})\frac{\text{RMS}}{\text{SST}}$$

Adjusted R2

$$R^2_{\text{Adj}} = \frac{\text{SST} - \text{SSE}}{\text{SST}} \times \frac{\text{df}_{\text{Total}}}{\text{df}_{\text{Residual}}}$$

$$R^2_{\text{Adj}} = 1 - (\text{df}_{\text{Total}})\frac{\text{RMS}}{\text{SST}}$$

When using an R2 value to evaluate model performance, we want to select a model that maximizes these values.

The adjusted R2 value penalizes the R2 value for model complexity, so when the number of predictors varies across the models, this is a better criterion.

The criteria that have been proposed to evaluate model performance when the purpose is prediction/inference focus on measuring out-of-sample performance.

In each of these formulae, *k* is the total number of parameters being estimated (including the residual variance).

Inference measures

t-value/*p*-value

These are at the predictor-level; not the model-level. Using the maximum *p*-value or minimum *t*-value can make this a model-level metric.

Mallow's Cp

$$C_p = \frac{\text{SSE}}{\hat{\sigma}^2_{\text{Residual (Full)}}} + 2k - n$$

Mallow's Cp is an estimate of the average mean squared error of prediction. The Cp value should be near to the number of predictors in the model.

AIC

$$\text{AIC} = n \times \ln(\frac{\text{SSE}}{n}) + 2k$$

Corrected AIC

$$\text{AICc} = \text{AIC} + \frac{2(k+2)(k+3)}{n-k-3}$$

AIC has a penalty for model complexity. It must be computed on the same set of observations; no missing data. Corrected AIC has been found to perform better.

BIC

$$\text{BIC} = n \times \ln(\frac{\text{SSE}}{n}) + k\ln(n)$$

BIC has a larger penalty term than AIC, which is based on sample size and model complexity. It performs best when the "true" model is among the candidate models.

Computing Model Evaluation Criteria

```r
# Compute minimum t-value
min(tidy(lm.all)$statistic)

[1] -1.90086
```

```r
# Compute maximum p-value
max(tidy(lm.all)$p.value)

[1] 0.7922506
```

```r
# Compute SSE and RMS
anova(lm.all)

# SSE = 31.668
# RMS = 0.7197
```

```r
# Compute R2 and adj. R2
glance(lm.all)

# R2 = 0.379
# Adj. R2 = 0.280
```

These metrics are useful for summarizing a single model and for comparing models.

```r
# Assign values for k and n
k = 8; n = 52; sse = 31.668; rms = 0.7197
```

```r
# Compute Mallow's Cp
sse / rms + 2 * k - n

[1] 8.001667
```

```r
# AIC
aic_mod = n * log(sse / n) + 2 * k
aic_mod

[1] -9.788725
```

```r
# AICc
aic_mod + (2 * (k + 2) * (k + 3)) / (n - k - 3)

[1] -4.422871
```

```r
# BIC
n * log(sse / n) + k * log(n)

[1] 5.821225
```

These metrics are not useful for summarizing a single model; only for comparing models.

Model Building Strategy

Model Building Strategy: Forward Selection

Determine the criteria/metric you will use to measure the performance of each model fitted *and* how to select a model using this metric.

Forward Selection

- *Step 1:* We fit each of the one-predictor models and measure the performance of each model using the criteria/metric chosen. The predictor from the model that has the best performance is retained.
- *Step 2:* We then fit each of the two-predictor models that can be fitted with the predictor retained in Step 1. The predictors from the model that has the best performance are retained.

We continue this process until we have either (a) fitted a model with all the predictors, or (b) hit some stopping/selection criteria that we have identified (e.g., stop once one of the p-values is greater than 0.05).

Once a predictor is retained in forward-selection, it is always included in all later stages.

In our example, we will employ forward selection to adopt a model using the following performance metric and selection criteria:

- **Metric of Performance:** Select the predictor with the highest t-value (absolute value).
- **Selection Criterion:** All t-values for predictors in the model need to be greater than 1.

```r
# Step 1: Fit all one-predictor models
tidy(lm(life_expectancy ~ -1 + population, data = z_usa))  #t =  1.63
tidy(lm(life_expectancy ~ -1 + income,     data = z_usa))  #t =  4.09
tidy(lm(life_expectancy ~ -1 + illiteracy, data = z_usa))  #t = -0.45
tidy(lm(life_expectancy ~ -1 + murder,     data = z_usa))  #t = -2.93
tidy(lm(life_expectancy ~ -1 + hs_grad,    data = z_usa))  #t =  2.47
tidy(lm(life_expectancy ~ -1 + frost,      data = z_usa))  #t =  1.19
tidy(lm(life_expectancy ~ -1 + area,       data = z_usa))  #t =  0.38
```

Note: Only the t-value for the added predictor is shown.

Step 1: The best one-predictor model under this criterion includes income.

```r
# Step 2: Fit all two-predictor models that include income
tidy(lm(life_expectancy ~ -1 + income + population, data = z_usa))  #t =  1.71
tidy(lm(life_expectancy ~ -1 + income + illiteracy, data = z_usa))  #t = -0.58
tidy(lm(life_expectancy ~ -1 + income + murder,     data = z_usa))  #t = -1.37
tidy(lm(life_expectancy ~ -1 + income + hs_grad,    data = z_usa))  #t =  0.46
tidy(lm(life_expectancy ~ -1 + income + frost,      data = z_usa))  #t =  0.03
tidy(lm(life_expectancy ~ -1 + income + area,       data = z_usa))  #t =  0.51
```

Step 2: The best two-predictor model under this criterion includes income and population.

```r
# Step 3: Fit all three-predictor models that include income and population
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy, data = z_usa))  #t = -2.13
tidy(lm(life_expectancy ~ -1 + income + population + murder,     data = z_usa))  #t = -1.54
tidy(lm(life_expectancy ~ -1 + income + population + hs_grad,    data = z_usa))  #t =  1.50
tidy(lm(life_expectancy ~ -1 + income + population + frost,      data = z_usa))  #t =  0.90
tidy(lm(life_expectancy ~ -1 + income + population + area,       data = z_usa))  #t =  0.25
```

Step 3: The best three-predictor model under this criterion includes income, population, and illiteracy.

```r
# Step 4: Fit all four-predictor models that include income, population, and illiteracy
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + murder,  data = z_usa))  #t = -1.08
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + hs_grad, data = z_usa))  #t =  0.45
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + frost,   data = z_usa))  #t = -0.35
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + area,    data = z_usa))  #t =  0.13
```

Step 4: The best four-predictor model under this criterion includes income, population, illiteracy, and murder rate.

Continue this process to determine the best four-, five-, six- and seven-predictor models

```
# Step 5: Fit all five-predictor models that include income, population, illiteracy, and murder_rate
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + murder + hs_grad, data = z_usa))  #t = -0.31
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + murder + frost,   data = z_usa))  #t = -0.77
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + murder + area,    data = z_usa))  #t =  0.21

# Step 6: Fit all six-predictor models that include income, population, illiteracy, murder_rate, and frost
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + murder + frost + hs_grad, data = z_usa))  #t = -0.12
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + murder + frost + area,    data = z_usa))  #t =  0.62

# Step 7: Fit all seven-predictor models that include income, population, illiteracy, murder_rate, frost, and area
tidy(lm(life_expectancy ~ -1 + income + population + illiteracy + murder + frost + area + hs_grad, data = z_usa))  #t = -0.27
```

At each stage we could also check to see that all the predictors in the selected model meet our selection criterion that the *t*-value for all predictors is greater than 1. With this criteria we could have stopped after Stage 4 since not all of the *t*-values of the best model in Stage 5 were above 1.

Based on the forward-selection process and the criteria we adopted, we would adopt the best performing model from Stage 4.

```
  term        estimate std.error statistic    p.value
  <chr>          <dbl>     <dbl>     <dbl>      <dbl>
1 income      0.418421  0.131650   3.17827 0.00259272
2 population  0.378425  0.145513   2.60063 0.0123322
3 illiteracy -0.269648  0.149487  -1.80383 0.0775360
4 murder     -0.146428  0.135083  -1.08399 0.283785
```

$$\widehat{\text{Life Expectancy}}_i = 0.42(\text{Income}_i) + 0.38(\text{Population}_i) - 0.27(\text{Illiteracy Rate}_i) - 0.15(\text{Murder}_i)$$

where all the variables in the equation are standardized.

The *p*-values are irrelevant as that did not factor into our selection criteria.
(In fact, I would not even report them if I was using this selection criteria.)

Model Building Strategies: Backward Elimination

Backward Elimination

- *Step 0:* We begin with a model that includes all of the predictors.
- *Step 1:* We then fit each of the models that include all of the predictors except one, and measure the performance. The predictor that has the least decreases the performance is removed from the model.
- We continue this process, at each stage removing the predictor that has the least impact on performance until we get down to an intercept-only model.

Once a predictor is removed, it is removed in all later stages.

```
# Step 0: Fit model with all predictors
glance(lm(life_expectancy ~ . - 1,  data = z_usa))$r.squared  #R2 = 0.379

# Step 1: Fit all models with one predictor removed
glance(lm(life_expectancy ~ . -1 - population, data = z_usa))$r.squared  #R2 = 0.319
glance(lm(life_expectancy ~ . -1 - income,     data = z_usa))$r.squared  #R2 = 0.264
glance(lm(life_expectancy ~ . -1 - illiteracy, data = z_usa))$r.squared  #R2 = 0.328
glance(lm(life_expectancy ~ . -1 - murder,     data = z_usa))$r.squared  #R2 = 0.357
glance(lm(life_expectancy ~ . -1 - hs_grad,    data = z_usa))$r.squared  #R2 = 0.378
glance(lm(life_expectancy ~ . -1 - frost,      data = z_usa))$r.squared  #R2 = 0.367
glance(lm(life_expectancy ~ . -1 - area,       data = z_usa))$r.squared  #R2 = 0.373
```

Criteria: Select the model with the highest R2 value.

Step 1: The model with the highest R2 removes high school graduation rate.

```
# Step 2: Fit all models with hs_grad and one other predictor removed
glance(lm(life_expectancy ~ . -1 - hs_grad - population, data = z_usa))$r.squared  #R2 = 0.309
glance(lm(life_expectancy ~ . -1 - hs_grad - income,     data = z_usa))$r.squared  #R2 = 0.232
glance(lm(life_expectancy ~ . -1 - hs_grad - illiteracy, data = z_usa))$r.squared  #R2 = 0.324
glance(lm(life_expectancy ~ . -1 - hs_grad - murder,     data = z_usa))$r.squared  #R2 = 0.352
glance(lm(life_expectancy ~ . -1 - hs_grad - frost,      data = z_usa))$r.squared  #R2 = 0.366
glance(lm(life_expectancy ~ . -1 - hs_grad - area,       data = z_usa))$r.squared  #R2 = 0.373
```

Step 2: The model with the highest R2 removes high school graduation rate and area.

Continue this process to determine the best four-, three-, two- and one-predictor models. At each stage we could also check to see that the selected model does not decrease the criteria beyond some threshold that is identified a priori. For example, we might stop when the R2-value is less than 0.3.

With this criteria we could have stopped after Stage 4 since the R2 value of the best model in Stage 5 has an R2 value that is less than 0.3 .

```
tidy(lm(life_expectancy ~ . -1 - hs_grad - area - frost - murder, data = z_usa))
```

```
  term        estimate std.error statistic     p.value
  <chr>          <dbl>     <dbl>     <dbl>       <dbl>
1 population  0.392355  0.145203   2.70212 0.00943906
2 income      0.487685  0.115309   4.22936 0.000102091
3 illiteracy -0.309651  0.145119  -2.13378 0.0378920
```

$$\widehat{\text{Life Expectancy}}_i = 0.39(\text{Population}_i)+0.49(\text{Income}_i)-0.31(\text{Illiteracy Rate}_i)$$

where all the variables in the equation are standardized.

Again, the *p*-values are irrelevant as that did not factor into our selection criteria. In fact, I would not even report them if I was using this selection criteria.

Model Building Strategy: Stepwise Regression

Determine the criteria/metric you will use to measure the performance of each model fitted *and* how to select a model using this metric.

Stepwise Regression

- Use backward elimination, but at each step, the predictors that were removed in earlier steps can be considered for re-rentry into the model.

We continue this process until we have hit some stopping/selection criteria that we have identified.

Once a predictor is removed, it might be re-included in later stages.

This is a combination of backward elimination and forward selection.

Model Building Strategy: Some Considerations

Different model strategies, metrics of model performance, and criteria for model selection lead to different “final” models.

Many statistical programs have functionality that can automate these fitting strategies.

Before automating the selection process, it is important to understand the purpose of your model (is it to describe the data? make predictions? inference?). This often guides the choice of performance metrics and model building strategy.

Although these packages can select a model based on some performance metric, there are several problems that automation does not solve:

- It does not address the functional form of the predictors.
- It does not address interactions.
- It does not address outliers.
- It does not address collinearity problems.

Most common software will require that you deal with these problems sequentially; first selecting the variables for the model and then determining their functional form, interactons, etc.

Automated Model Building

Functions from the **olsrr** package perform automated forward selection, backward elimination, and stepwise regression using either the AIC or *p*-value as a performance metric. All of the functions require a lm object that includes all possible predictors.

```r
# Load olsrr library
library(olsrr)

# Fit forward selection
fs_output = ols_step_forward_aic(lm.all, details = TRUE)

# Plot results
plot(fs_output)
```

The `ols_step_forward_aic()` and `ols_step_forward_p()` functions perform forward selection using the AIC and *p*-value, respectively, as a performance metric.

The `ols_step_backward_aic()` and `ols_step_backward_p()` functions from the **olsrr** package perform backward elimination using the AIC and *p*-value, respectively, as a performance metric.

The `ols_step_both_aic()` and `ols_step_both_p()` functions from the **olsrr** package perform stepwise regression using the AIC and *p*-value, respectively, as a performance metric.

```
Step 0: AIC = 209.8146
life_expectancy ~ 1
```

Variable	DF	AIC	Sum Sq	RSS	R-Sq	Adj. R-Sq
income	1	197.033	39.438	119.950	0.247	0.232
murder	1	203.710	23.001	136.386	0.144	0.127
hs_grad	1	205.929	17.057	142.331	0.107	0.089
population	1	209.174	7.892	151.495	0.050	0.031
frost	1	210.386	4.319	155.069	0.027	0.008
illiteracy	1	211.607	0.636	158.752	0.004	-0.016
area	1	211.668	0.449	158.939	0.003	-0.017

```
Step 1 : AIC = 197.0327
life_expectancy ~ income
```

Variable	DF	AIC	Sum Sq	RSS	R-Sq	Adj. R-Sq
population	1	196.078	6.626	113.323	0.289	0.260
murder	1	197.128	4.314	115.636	0.274	0.245
illiteracy	1	198.680	0.811	119.139	0.253	0.222
area	1	198.762	0.622	119.328	0.251	0.221
hs_grad	1	198.813	0.505	119.445	0.251	0.220
frost	1	199.032	0.002	119.948	0.247	0.217

```
Step 2 : AIC = 196.0776
life_expectancy ~ income + population
```

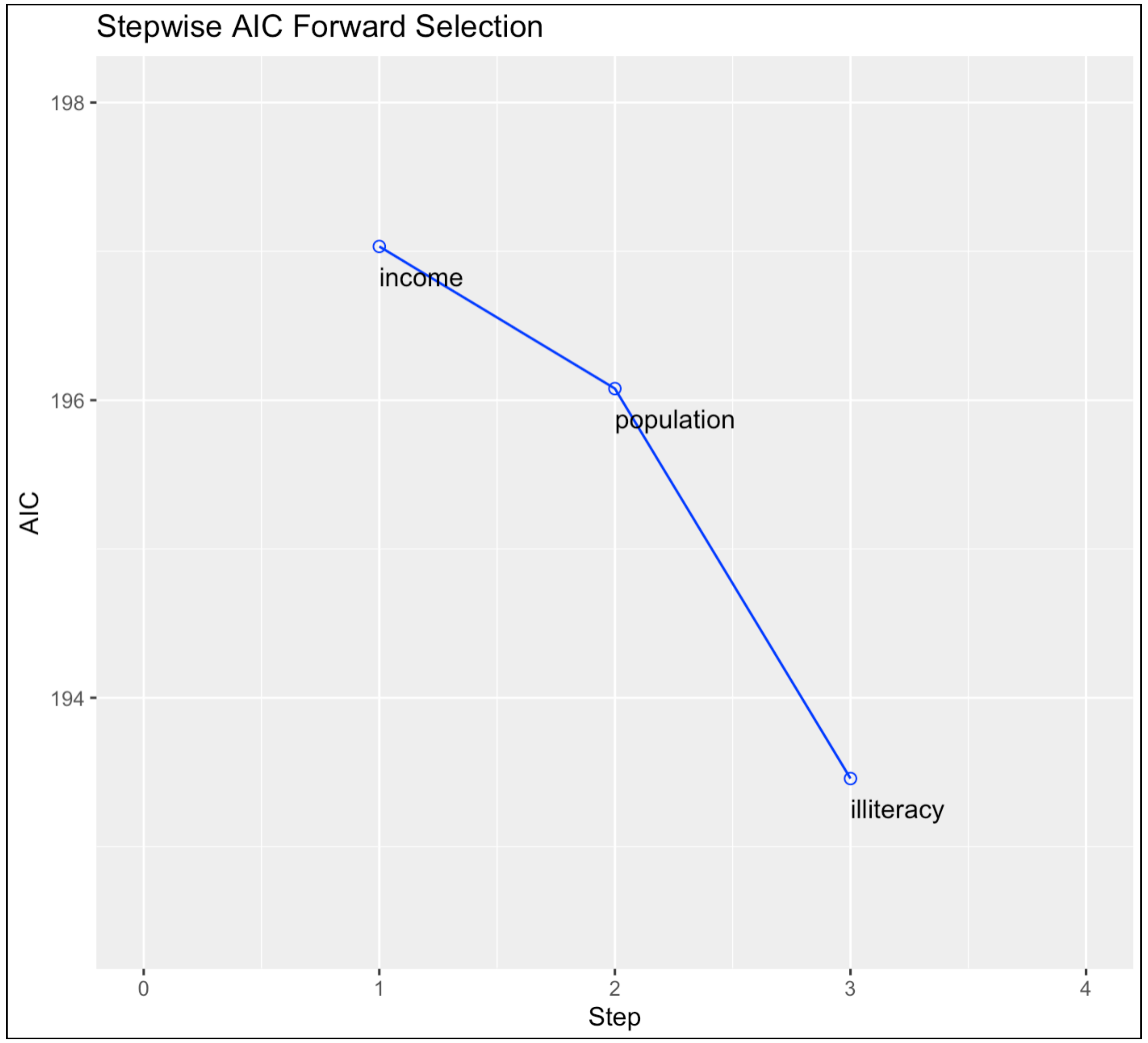
Variable	DF	AIC	Sum Sq	RSS	R-Sq	Adj. R-Sq
illiteracy	1	193.457	9.635	103.689	0.349	0.309
murder	1	195.610	5.251	108.072	0.322	0.280
hs_grad	1	195.747	4.966	108.357	0.320	0.278
frost	1	197.233	1.826	111.497	0.300	0.257
area	1	198.012	0.143	113.180	0.290	0.246

```
Step 3 : AIC = 193.4573
life_expectancy ~ income + population + illiteracy
```

Variable	DF	AIC	Sum Sq	RSS	R-Sq	Adj. R-Sq
murder	1	194.200	2.478	101.211	0.365	0.311
hs_grad	1	195.238	0.436	103.253	0.352	0.297
frost	1	195.324	0.265	103.424	0.351	0.296
area	1	195.439	0.036	103.652	0.350	0.294

```
No more variables to be added.
```

All of the AIC values are higher when any new variables are added to the model.



```
Final Model Output
------------------

                        Model Summary
--------------------------------------------------------------
R                       0.591       RMSE                1.470
R-Squared               0.349       Coef. Var           1.867
Adj. R-Squared          0.309       MSE                 2.160
Pred R-Squared          0.181       MAE                 1.120
--------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                              ANOVA
-------------------------------------------------------------------
              Sum of
             Squares        DF    Mean Square      F         Sig.
-------------------------------------------------------------------
Regression    55.699         3         18.566    8.595      1e-04
Residual     103.689        48          2.160
Total        159.388        51
-------------------------------------------------------------------

                                  Parameter Estimates
-----------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta      t        Sig      lower     upper
-----------------------------------------------------------------------------------------
(Intercept)      74.975         1.269                 59.104    0.000    72.424    77.525
     income       0.162         0.039        0.488     4.186    0.000     0.084     0.240
 population       0.095         0.036        0.392     2.674    0.010     0.024     0.166
 illiteracy      -0.127         0.060       -0.310    -2.112    0.040    -0.247    -0.006
-----------------------------------------------------------------------------------------
>
```

The selected model includes income, population, and illiteracy rate.

$$\widehat{\text{Life Expectancy}}_i = 0.49(\text{Income}_i) + 0.39(\text{Population}_i) - 0.31(\text{Illiteracy Rate}_i)$$

where all the variables in the equation are standardized.

All Subsets Regression

Model Building Strategy: All Subsets Regression

Determine the criteria/metric you will use to measure the performance of each model fitted *and* how to select a model using this metric.

All Subsets Regression

- Fit all possible *k*-predictor models.

Use the selection criteria to select frm among all the possible models.

$$\text{Number of Models} = 2^p - 1$$

The `ols_step_all_possible()` function from the **olsrr** package can be used to exhaustively fit a set of models.

```r
# Fit all subsets of predictors
all_output = ols_step_all_possible(lm.all) %>%
    data.frame()
```

The function takes an lm object with all potential predictors. We also coerce the output to a data frame.

$$\text{Number of Models} = 2^7 - 1 = 127$$

There are 127 models outputted.

```r
# Output
head(all_output)
```

```
  mindex n predictors     rsquare         adjr     predrsq        cp      aic     sbic      sbc     msep      fpe       apc        hsp
2      1 1     income 0.247434318  0.232383004  0.11556421  5.326316 197.0327 49.36238 202.8864 124.7515 2.491263 0.8127709 0.04895906
4      2 1     murder 0.144310731  0.127196945 -0.15879139 12.633586 203.7105 55.53078 209.5642 141.8461 2.832639 0.9241444 0.05566789
5      3 1    hs_grad 0.107015741  0.089156056 -0.05245937 15.276285 205.9289 57.58295 211.7826 148.0284 2.956099 0.9644230 0.05809416
1      4 1 population 0.049515821  0.030506137 -0.01697460 19.350692 209.1738 60.58864 215.0276 157.5601 3.146445 1.0265229 0.06183489
6      5 1      frost 0.027098563  0.007640534 -0.06480510 20.939164 210.3860 61.71282 216.2398 161.2761 3.220654 1.0507336 0.06329327
3      6 1 illiteracy 0.003990589 -0.015929599 -0.09500651 22.576580 211.6067 62.84565 217.4604 165.1067 3.297149 1.0756902 0.06479659
```

```r
# Add AICc to output
all_output = all_output %>%
    mutate(
     aic_c = aic + (2 * (n + 2) * (n + 3)) / (nrow(z_usa) - n - 3)
    )
```

```
  mindex n predictors     rsquare         adjr     predrsq        cp      aic     sbic      sbc     msep      fpe       apc        hsp    aic_c
1      1 1     income 0.247434318  0.232383004  0.11556421  5.326316 197.0327 49.36238 202.8864 124.7515 2.491263 0.8127709 0.04895906 197.5327
2      2 1     murder 0.144310731  0.127196945 -0.15879139 12.633586 203.7105 55.53078 209.5642 141.8461 2.832639 0.9241444 0.05566789 204.2105
3      3 1    hs_grad 0.107015741  0.089156056 -0.05245937 15.276285 205.9289 57.58295 211.7826 148.0284 2.956099 0.9644230 0.05809416 206.4289
4      4 1 population 0.049515821  0.030506137 -0.01697460 19.350692 209.1738 60.58864 215.0276 157.5601 3.146445 1.0265229 0.06183489 209.6738
5      5 1      frost 0.027098563  0.007640534 -0.06480510 20.939164 210.3860 61.71282 216.2398 161.2761 3.220654 1.0507336 0.06329327 210.8860
6      6 1 illiteracy 0.003990589 -0.015929599 -0.09500651 22.576580 211.6067 62.84565 217.4604 165.1067 3.297149 1.0756902 0.06479659 212.1067
```

Select model(s) with the lowest AICc

```r
# Get best model
all_output %>%
  filter(aic_c == min(aic_c))
```

	mindex	n	predictors	rsquare	adjr	predrsq	cp	aic	sbic	sbc	msep	fpe	apc	hsp	aic_c
1	29	3	population income illiteracy	0.3494565	0.3087975	0.1814021	2.097091	193.4573	46.87937	203.2136	112.4283	2.326347	0.7589674	0.04596127	194.7617

$$\widehat{\text{Life Expectancy}}_i = \hat{\beta}_0 + \hat{\beta}_1(\text{Population}_i) + \hat{\beta}_2(\text{Illiteracy Rate}_i) + \hat{\beta}_3(\text{Area}_i)$$

To obtain the coefficients, fit the model using population, income, and illiteracy to predict variation in life expectancy.

```r
# Get best k-predictor models
all_output %>%
  group_by(n) %>%
  filter(aic_c == min(aic_c)) %>%
  ungroup() %>%
  arrange(aic_c)
```

Several model have an AICc within 4 from the minimum AICc.

	mindex	n	predictors	rsquare	adjr	predrsq	cp	aic	sbic	sbc	msep	fpe	apc	hsp	aic_c
	<int>	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	29	3	population income illiteracy	0.349456	0.308798	0.181402	2.09709	193.457	46.8794	203.214	112.428	2.32635	0.758967	0.0459613	194.762
2	64	4	population income illiteracy murder	0.365001	0.310959	-0.218784	2.99561	194.200	48.1383	205.907	112.128	2.36049	0.770105	0.0468136	196.066
3	8	2	population income	0.289009	0.259989	0.141506	4.38037	196.078	48.6988	203.883	120.315	2.44615	0.798051	0.0481816	196.929
4	1	1	income	0.247434	0.232383	0.115564	5.32632	197.033	49.3624	202.886	124.751	2.49126	0.812771	0.0489591	197.533
5	99	5	population income illiteracy murder frost	0.372834	0.304664	-0.290870	4.44058	195.554	49.9681	209.213	113.205	2.42384	0.790775	0.0482911	198.100
6	120	6	population income illiteracy murder frost area	0.378061	0.295135	-0.392393	6.07022	197.119	52.0039	212.729	114.813	2.49942	0.815432	0.0500654	200.468
7	127	7	population income illiteracy murder hs_grad fro…	0.379052	0.280264	-0.793953	8	199.036	54.3096	216.597	117.296	2.59541	0.846748	0.0523105	203.322

Remember that models that have similar AICc values are all viable candidates.

```
# Get models with 20 lowest AIC values
all_output %>%
    arrange(aic_c)
```

	mindex	n	predictors	aic_c
1	29	3	population income illiteracy	194.7617
2	64	4	population income illiteracy murder	196.0664
3	30	3	population income murder	196.9147
4	8	2	population income	196.9287
5	31	3	population income hs_grad	197.0518
6	65	4	population income illiteracy hs_grad	197.1051
7	66	4	population income illiteracy frost	197.1910
8	67	4	population income illiteracy area	197.3059
9	1	1	income	197.5327
10	9	2	income murder	197.9792
11	99	5	population income illiteracy murder frost	198.0998
12	32	3	population income frost	198.5373
13	100	5	population income illiteracy murder hs_grad	198.6384
14	101	5	population income illiteracy murder area	198.6978
15	68	4	population income murder hs_grad	199.0321
16	33	3	population income area	199.3164
17	69	4	population income murder area	199.3552
18	70	4	population income murder frost	199.3807
19	102	5	population income illiteracy hs_grad frost	199.4163
20	10	2	income illiteracy	199.5312

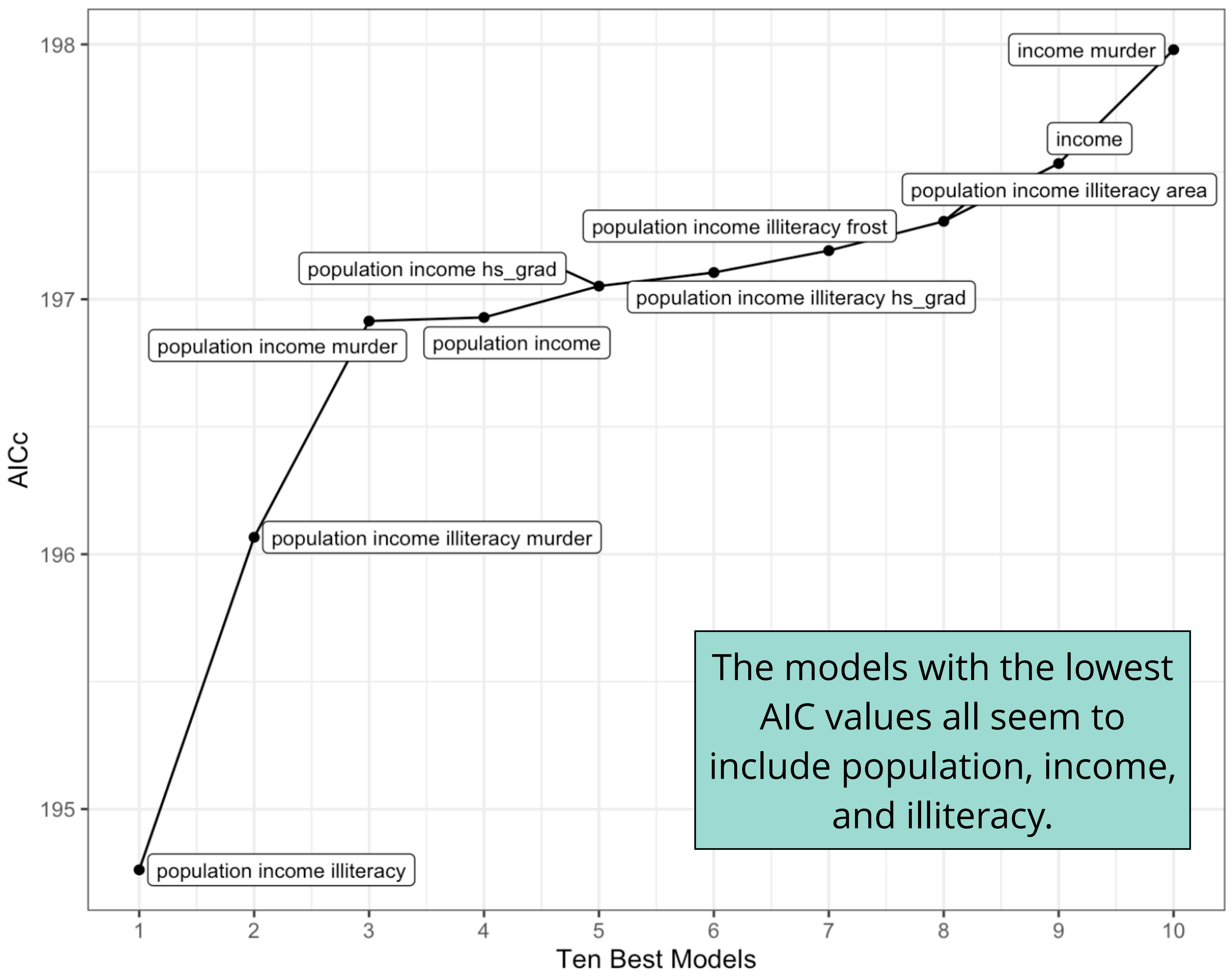
Several model have an AICc within 4 from the minimum AICc.

It can be useful to examine the predictors from the best models. (Here I do that in a plot, but it could also be done in a table.) This can help identify substantively important predictors.

```
# Get the 10 best models
ten_best = all_output %>%
  arrange(aic_c) %>%
  filter(row_number() <= 10)


# Load library for labeling
library(ggrepel)


# Plot the models
ggplot(data = ten_best, aes(x = as.numeric(rownames(ten_best)), y = aic_c)) +
  geom_line(group = 1) +
  geom_point() +
  geom_label_repel(aes(label = predictors), size = 3) +
  theme_bw() +
  scale_x_continuous(name = "Ten Best Models", breaks = 1:10) +
  ylab("AICc")
```



The models with the lowest AIC values all seem to include population, income, and illiteracy.

99 Problems . . .

In simulation studies, even when the true set of predictors are included in the subset of regression models, automated strategies may not identify such these predictors in the best models (Miller, 2018).

The essential problems with these methods have been summarized by Harrell (2001):

- R^2 values are biased high
- The *F*-statistics are not *F*-distributed.
- The standard errors of the parameter estimates are too small.
    - Consequently, the confidence intervals around the parameter estimates are too narrow.
- *p*-values are too low, due to multiple comparisons, and are difficult to correct.
- Parameter estimates are biased away from 0.
- Collinearity problems are exacerbated.

In sum, the parameter estimates are likely to be too far away from zero; the variance estimates for those parameter estimates are not correct; which implies that the confidence intervals and hypothesis tests will be wrong; and there are no reasonable ways of correcting these problems!

As Flom (2018) writes, “Most devastatingly, it allows the analyst not to think. Put in another way, for a data analyst to use stepwise methods is equivalent to telling his or her boss that his or her salary should be cut.”

In general the evidence around using automated selection methods is that these methods are subpar. If you need to use these methods backward elimination seems to be the best method to use. (Stepwise regression consistently performs the worst.) Also, information criteria (AIC, BIC) seems to be the best criterion when using these methods.

Flom, P. (2018). Stopping stepwise: Why stepwise selection is bad and what you should use instead. *Towards data science.* https://towardsdatascience.com/stopping-stepwise-why-stepwise-selection-is-bad-and-what-you-should-use-instead-90818b3f52df
Harrell, F. E. (2001). *Regression modeling strategies: With applications to linear models, logistic regression, and survival analysis.* Springer.
Miller, A. J. (2002). *Subset selection in regression,* Chapman & Hall.