

AIC vs BIC

Lieven Clement

Invalid Date

Table of contents

1	When large datasets are used, the BIC will favour smaller models than the AIC.	1
2	Illustration	2
2.1	Backward modelselection with AIC	2
2.2	Backward modelselection with BIC.	5
3	What happens when the variance increase?	8
3.1	Backward modelselection with AIC	8
3.2	Backward modelselection with BIC.	11
	Session info	14

1 When large datasets are used, the BIC will favour smaller models than the AIC.

Both the BIC and the AIC criterion are a trade-off between fit ($-2 \log L$) and model complexity ($2p$ for the AIC and $p \log(n)$ for BIC).

- When we use the BIC for model selection we will select the model with the lowest BIC.
- When we use the AIC for model selection we will select the model with the lowest AIC.

However, the trade-off between fit and model complexity is different for the AIC and the BIC.

- They use the same measure for the fit $-2 \log L$.
- Increasing the model complexity will result in a better fit and thus a decrease of $-2 \log L$.
- However, both criteria also penalise for model complexity: $2p$ vs $p \log(n)$.

In order to favour a more complex model,

- the decrease of $-2 \log L$ has to be larger than the increase in $2p$ for the AIC criterion,
- while the decrease of $-2 \log L$ has to be larger than the increase in $p \log(n)$ for the BIC criterion.

Because $\log(n) > 2$ as soon as $n \geq 8$, the BIC criterion will penalise more for the increase in model complexity and it will therefore favour smaller models than the AIC criterion, which punishes less for the increase in model complexity.

2 Illustration

```
set.seed(2)
x <- as.data.frame(matrix(rnorm(1000),ncol=10))
pred <- paste0("V",1:3)
```

model

$$Y = V1 + 2 * V2 + 4 * V3 + \epsilon$$

```
set.seed(19325)
y <- 10 + x[,1] + 2* x[,2] + 4*x[,3] + rnorm(100)
```

2.1 Backward modelselection with AIC

```
lm0 <- lm(y~.,data=x)
lmAIC <- step(lm0)
```

```
#> Start:  AIC=10.31
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V7   1     0.31  89.28  8.657
#> - V4   1     0.37  89.33  8.722
#> - V9   1     0.66  89.63  9.053
#> - V10  1     1.15  90.12  9.599
#> - V6   1     1.20  90.17  9.652
```

```

#> <none>                88.97  10.311
#> - V5      1          5.07   94.04  13.855
#> - V8      1          5.66   94.63  14.476
#> - V1      1       127.33  216.30  97.150
#> - V2      1       320.34  409.30 160.929
#> - V3      1     1518.09 1607.06 297.699
#>
#> Step:  AIC=8.66
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V4    1      0.56   89.83   7.278
#> - V9    1      0.63   89.91   7.359
#> - V10   1      1.05   90.32   7.824
#> - V6    1      1.15   90.43   7.942
#> <none>                89.28   8.657
#> - V5    1      5.01   94.28  12.113
#> - V8    1      5.59   94.87  12.731
#> - V1    1     127.24  216.52  95.251
#> - V2    1     325.09  414.37 160.158
#> - V3    1    1572.65 1661.92 299.056
#>
#> Step:  AIC=7.28
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V9    1      0.64   90.47   5.984
#> - V10   1      1.13   90.96   6.527
#> - V6    1      1.18   91.01   6.583
#> <none>                89.83   7.278
#> - V5    1      4.58   94.41  10.249
#> - V8    1      5.25   95.08  10.956
#> - V1    1     127.10  216.93  93.441
#> - V2    1     327.62  417.46 158.901
#> - V3    1    1584.28 1674.11 297.787
#>
#> Step:  AIC=5.98
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V10   1      1.49   91.96   5.620
#> - V6    1      1.52   91.99   5.654
#> <none>                90.47   5.984

```

```

#> - V5      1      4.44   94.91   8.776
#> - V8      1      5.83   96.30  10.228
#> - V1      1    130.29  220.76  93.190
#> - V2      1    327.02  417.49 156.909
#> - V3      1   1588.71 1679.18 296.089
#>
#> Step:  AIC=5.62
#> y ~ V1 + V2 + V3 + V5 + V6 + V8
#>
#>      Df Sum of Sq    RSS    AIC
#> - V6    1      1.33   93.29   5.058
#> <none>          91.96   5.620
#> - V5    1      4.59   96.55   8.491
#> - V8    1      4.88   96.84   8.788
#> - V1    1    128.80  220.76  91.190
#> - V2    1    336.71  428.67 157.553
#> - V3    1   1587.72 1679.68 294.119
#>
#> Step:  AIC=5.06
#> y ~ V1 + V2 + V3 + V5 + V8
#>
#>      Df Sum of Sq    RSS    AIC
#> <none>          93.29   5.058
#> - V8    1      3.94   97.23   7.193
#> - V5    1      4.11   97.41   7.371
#> - V1    1    127.53  220.83  89.221
#> - V2    1    348.90  442.19 158.658
#> - V3    1   1608.86 1702.16 293.448

```

```
lmAIC
```

```

#>
#> Call:
#> lm(formula = y ~ V1 + V2 + V3 + V5 + V8, data = x)
#>
#> Coefficients:
#> (Intercept)          V1          V2          V3          V5          V8
#>      9.9709      0.9868      1.9685      3.8627     -0.2056      0.2121

```

```

realPredAIC <- sum(names(lmAIC$coefficients) %in% pred)
falsePredAIC <- length(lmAIC$coefficients) - realPredAIC - 1

```

The AIC criterion selects the model with the lowest AIC.

This model correctly selects 3 out of 3 real predictors. However, it also selects 2 predictors that are not associated with the response!

2.2 Backward modelselection with BIC.

We can do this in the step function by specifying k. By default k=2. If we define k=log(n) than we use the BIC.

```
lmBIC <- step(lm0, k=log(nrow(x)))
```

```
#> Start:  AIC=38.97
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V7    1      0.31   89.28  34.71
#> - V4    1      0.37   89.33  34.77
#> - V9    1      0.66   89.63  35.10
#> - V10   1      1.15   90.12  35.65
#> - V6    1      1.20   90.17  35.70
#> <none>                88.97  38.97
#> - V5    1      5.07   94.04  39.91
#> - V8    1      5.66   94.63  40.53
#> - V1    1     127.33  216.30 123.20
#> - V2    1     320.34  409.30 186.98
#> - V3    1    1518.09 1607.06 323.75
#>
#> Step:  AIC=34.71
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V4    1      0.56   89.83  30.72
#> - V9    1      0.63   89.91  30.81
#> - V10   1      1.05   90.32  31.27
#> - V6    1      1.15   90.43  31.39
#> <none>                89.28  34.71
#> - V5    1      5.01   94.28  35.56
#> - V8    1      5.59   94.87  36.18
#> - V1    1     127.24  216.52 118.70
#> - V2    1     325.09  414.37 183.60
#> - V3    1    1572.65 1661.92 322.50
```

```

#>
#> Step: AIC=30.72
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V9    1      0.64   90.47  26.83
#> - V10   1      1.13   90.96  27.37
#> - V6    1      1.18   91.01  27.42
#> <none>                89.83  30.72
#> - V5    1      4.58   94.41  31.09
#> - V8    1      5.25   95.08  31.80
#> - V1    1     127.10  216.93 114.28
#> - V2    1     327.62  417.46 179.74
#> - V3    1    1584.28 1674.11 318.63
#>
#> Step: AIC=26.83
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V10   1      1.49   91.96  23.856
#> - V6    1      1.52   91.99  23.891
#> <none>                90.47  26.825
#> - V5    1      4.44   94.91  27.013
#> - V8    1      5.83   96.30  28.465
#> - V1    1     130.29  220.76 111.426
#> - V2    1     327.02  417.49 175.145
#> - V3    1    1588.71 1679.18 314.325
#>
#> Step: AIC=23.86
#> y ~ V1 + V2 + V3 + V5 + V6 + V8
#>
#>      Df Sum of Sq    RSS    AIC
#> - V6    1      1.33   93.29  20.689
#> <none>                91.96  23.856
#> - V5    1      4.59   96.55  24.122
#> - V8    1      4.88   96.84  24.419
#> - V1    1     128.80  220.76 106.821
#> - V2    1     336.71  428.67 173.184
#> - V3    1    1587.72 1679.68 309.750
#>
#> Step: AIC=20.69
#> y ~ V1 + V2 + V3 + V5 + V8
#>

```

```

#>           Df Sum of Sq      RSS      AIC
#> - V8       1         3.94   97.23  20.219
#> - V5       1         4.11   97.41  20.397
#> <none>                93.29  20.689
#> - V1       1       127.53  220.83 102.247
#> - V2       1       348.90  442.19 171.683
#> - V3       1      1608.86 1702.16 306.474
#>
#> Step:   AIC=20.22
#> y ~ V1 + V2 + V3 + V5
#>
#>           Df Sum of Sq      RSS      AIC
#> - V5       1         3.54  100.77  19.187
#> <none>                97.23  20.219
#> - V1       1       124.15  221.38  97.891
#> - V2       1       389.45  486.68 176.664
#> - V3       1      1620.25 1717.48 302.765
#>
#> Step:   AIC=19.19
#> y ~ V1 + V2 + V3
#>
#>           Df Sum of Sq      RSS      AIC
#> <none>                100.77  19.187
#> - V1       1       123.13  223.90  94.419
#> - V2       1       393.16  493.93 173.539
#> - V3       1      1618.02 1718.79 298.236

```

lmBIC

```

#>
#> Call:
#> lm(formula = y ~ V1 + V2 + V3, data = x)
#>
#> Coefficients:
#> (Intercept)          V1          V2          V3
#>    10.0389      0.9633      2.0267      3.8315

```

```

realPredBIC <- sum(names(lmBIC$coefficients) %in% pred)
falsePredBIC <- length(lmBIC$coefficients) - realPredBIC - 1

```

The BIC criterion selects the model with the lowest BIC.

This model correctly selects 3 out of 3 real predictors. However, it selects 0 predictors that are not associated with the response!

3 What happens when the variance increase?

If the noise increases it will be harder to select the correct model and we still can expect the AIC to result in more complex models than the BIC.

We will use the same seed so that the difference in the response is not induced by the random generator but only by the difference in variance.

```
set.seed(19325)
y <- 10 + x[,1] + 2* x[,2] + 4*x[,3] + rnorm(100,sd=10)
lm0 <- lm(y~., data=x)
```

3.1 Backward modelselection with AIC

```
lm0 <- lm(y~.,data=x)
lmAIC10 <- step(lm0)
```

```
#> Start:  AIC=470.83
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10
#>
#>           Df Sum of Sq    RSS    AIC
#> - V7       1      30.79 8927.6 469.17
#> - V4       1      36.64 8933.5 469.24
#> - V9       1      66.24 8963.1 469.57
#> - V10      1     115.29 9012.1 470.12
#> - V6       1     120.08 9016.9 470.17
#> - V1       1     127.61 9024.5 470.25
#> - V2       1     144.13 9041.0 470.44
#> <none>                8896.8 470.83
#> - V5       1     507.13 9404.0 474.37
#> - V8       1     565.69 9462.5 474.99
#> - V3       1     834.14 9731.0 477.79
#>
#> Step:  AIC=469.17
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10
#>
```

```

#>      Df Sum of Sq    RSS    AIC
#> - V4    1      55.60 8983.2 467.79
#> - V9    1      62.93 8990.6 467.88
#> - V10   1     104.79 9032.4 468.34
#> - V6    1     115.47 9043.1 468.46
#> - V1    1     126.64 9054.3 468.58
#> - V2    1     158.35 9086.0 468.93
#> <none>                8927.6 469.17
#> - V5    1     500.69 9428.3 472.63
#> - V8    1     559.12 9486.8 473.25
#> - V3    1     803.42 9731.1 475.79
#>
#> Step:  AIC=467.79
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V9    1      63.68 9046.9 466.50
#> - V10   1     112.93 9096.2 467.04
#> - V6    1     118.00 9101.2 467.10
#> - V1    1     125.12 9108.4 467.18
#> - V2    1     169.56 9152.8 467.66
#> <none>                8983.2 467.79
#> - V5    1     457.91 9441.1 470.77
#> - V8    1     524.92 9508.2 471.47
#> - V3    1     836.85 9820.1 474.70
#>
#> Step:  AIC=466.5
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V1    1     145.47 9192.4 466.10
#> - V10   1     149.20 9196.1 466.14
#> - V6    1     152.40 9199.3 466.17
#> - V2    1     162.80 9209.7 466.28
#> <none>                9046.9 466.50
#> - V5    1     444.13 9491.0 469.29
#> - V8    1     582.94 9629.9 470.75
#> - V3    1     854.90 9901.8 473.53
#>
#> Step:  AIC=466.1
#> y ~ V2 + V3 + V5 + V6 + V8 + V10
#>
#>      Df Sum of Sq    RSS    AIC

```

```

#> - V10    1    120.54  9312.9 465.40
#> - V6      1    128.99  9321.4 465.49
#> - V2      1    161.82  9354.2 465.84
#> <none>                9192.4 466.10
#> - V5      1    424.91  9617.3 468.61
#> - V8      1    512.28  9704.7 469.52
#> - V3      1    812.18 10004.6 472.56
#>
#> Step:  AIC=465.4
#> y ~ V2 + V3 + V5 + V6 + V8
#>
#>          Df Sum of Sq    RSS    AIC
#> - V6      1    115.07  9428.0 464.63
#> <none>                9312.9 465.40
#> - V2      1    198.94  9511.9 465.51
#> - V8      1    436.36  9749.3 467.98
#> - V5      1    440.21  9753.1 468.02
#> - V3      1    787.70 10100.6 471.52
#>
#> Step:  AIC=464.63
#> y ~ V2 + V3 + V5 + V8
#>
#>          Df Sum of Sq    RSS    AIC
#> <none>                9428.0 464.63
#> - V2      1    247.31  9675.3 465.22
#> - V8      1    355.83  9783.8 466.33
#> - V5      1    397.89  9825.9 466.76
#> - V3      1    718.83 10146.8 469.97

```

```
lmAIC10
```

```

#>
#> Call:
#> lm(formula = y ~ V2 + V3 + V5 + V8, data = x)
#>
#> Coefficients:
#> (Intercept)          V2          V3          V5          V8
#>      9.715       1.657       2.579      -2.022       2.004

```

```

realPredAIC10 <- sum(names(lmAIC10$coefficients) %in% pred)
falsePredAIC10 <- length(lmAIC10$coefficients) - realPredAIC10 - 1

```

The AIC criterion selects the model with the lowest AIC.

This model correctly selects 2 out of 3 real predictors. However, it also selects 2 predictors that are not associated with the response!

3.2 Backward modelselection with BIC.

We can do this in the step function by specifying k. By default k=2. If we define k=log(n) than we use the BIC.

```
lmBIC10 <- step(lm0, k=log(nrow(x)))
```

```
#> Start: AIC=499.49
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10
#>
#>      Df Sum of Sq  RSS   AIC
#> - V7    1    30.79 8927.6 495.23
#> - V4    1    36.64 8933.5 495.29
#> - V9    1    66.24 8963.1 495.62
#> - V10   1   115.29 9012.1 496.17
#> - V6    1   120.08 9016.9 496.22
#> - V1    1   127.61 9024.5 496.30
#> - V2    1   144.13 9041.0 496.49
#> <none>                8896.8 499.49
#> - V5    1   507.13 9404.0 500.42
#> - V8    1   565.69 9462.5 501.04
#> - V3    1   834.14 9731.0 503.84
#>
#> Step: AIC=495.23
#> y ~ V1 + V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10
#>
#>      Df Sum of Sq  RSS   AIC
#> - V4    1    55.60 8983.2 491.24
#> - V9    1    62.93 8990.6 491.32
#> - V10   1   104.79 9032.4 491.79
#> - V6    1   115.47 9043.1 491.91
#> - V1    1   126.64 9054.3 492.03
#> - V2    1   158.35 9086.0 492.38
#> <none>                8927.6 495.23
#> - V5    1   500.69 9428.3 496.08
#> - V8    1   559.12 9486.8 496.69
#> - V3    1   803.42 9731.1 499.24
```

```

#>
#> Step: AIC=491.24
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V9 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V9   1     63.68 9046.9 487.34
#> - V10  1    112.93 9096.2 487.89
#> - V6   1    118.00 9101.2 487.94
#> - V1   1    125.12 9108.4 488.02
#> - V2   1    169.56 9152.8 488.51
#> <none>      8983.2 491.24
#> - V5   1    457.91 9441.1 491.61
#> - V8   1    524.92 9508.2 492.31
#> - V3   1    836.85 9820.1 495.54
#>
#> Step: AIC=487.34
#> y ~ V1 + V2 + V3 + V5 + V6 + V8 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V1   1    145.47 9192.4 484.33
#> - V10  1    149.20 9196.1 484.37
#> - V6   1    152.40 9199.3 484.41
#> - V2   1    162.80 9209.7 484.52
#> <none>      9046.9 487.34
#> - V5   1    444.13 9491.0 487.53
#> - V8   1    582.94 9629.9 488.98
#> - V3   1    854.90 9901.8 491.77
#>
#> Step: AIC=484.33
#> y ~ V2 + V3 + V5 + V6 + V8 + V10
#>
#>      Df Sum of Sq    RSS    AIC
#> - V10  1    120.54 9312.9 481.03
#> - V6   1    128.99 9321.4 481.12
#> - V2   1    161.82 9354.2 481.47
#> - V5   1    424.91 9617.3 484.25
#> <none>      9192.4 484.33
#> - V8   1    512.28 9704.7 485.15
#> - V3   1    812.18 10004.6 488.19
#>
#> Step: AIC=481.03
#> y ~ V2 + V3 + V5 + V6 + V8
#>

```

```

#>      Df Sum of Sq    RSS    AIC
#> - V6    1    115.07  9428.0 477.65
#> - V2    1    198.94  9511.9 478.54
#> - V8    1    436.36  9749.3 481.00
#> <none>                9312.9 481.03
#> - V5    1    440.21  9753.1 481.04
#> - V3    1    787.70 10100.6 484.54
#>
#> Step: AIC=477.65
#> y ~ V2 + V3 + V5 + V8
#>
#>      Df Sum of Sq    RSS    AIC
#> - V2    1    247.31  9675.3 475.64
#> - V8    1    355.83  9783.8 476.75
#> - V5    1    397.89  9825.9 477.18
#> <none>                9428.0 477.65
#> - V3    1    718.83 10146.8 480.40
#>
#> Step: AIC=475.64
#> y ~ V3 + V5 + V8
#>
#>      Df Sum of Sq    RSS    AIC
#> - V5    1    437.44 10112.7 475.45
#> <none>                9675.3 475.64
#> - V8    1    548.90 10224.2 476.55
#> - V3    1    723.85 10399.1 478.25
#>
#> Step: AIC=475.45
#> y ~ V3 + V8
#>
#>      Df Sum of Sq    RSS    AIC
#> <none>                10113 475.45
#> - V8    1    492.56 10605 475.60
#> - V3    1    693.98 10807 477.49

```

```
lmBIC10
```

```

#>
#> Call:
#> lm(formula = y ~ V3 + V8, data = x)
#>
#> Coefficients:

```

```
#> (Intercept)          V3          V8
#>      10.020       2.533       2.281
```

```
realPredBIC10 <- sum(names(lmBIC10$coefficients) %in% pred)
falsePredBIC10 <- length(lmBIC10$coefficients) - realPredBIC10 - 1
```

The BIC criterion selects the model with the lowest BIC.

This model correctly selects 1 out of 3 real predictors. However, it selects 1 predictor that is not associated with the response!

Note, that AIC and BIC are a good estimate of the insample error, however, when building prediction models we are interested in using the model for predictor patterns that are not observed in the training set. So it is better to build a model based on an estimate of the outsample error.

Session info

Session info

```
#> [1] "2026-09-15 16:05:16 CEST"
```

```
#> - Session info -----
#> setting    value
#> version    R version 4.6.1 (2026-06-24)
#> os          macOS Tahoe 26.1
#> system      aarch64, darwin23
#> ui          X11
#> language    (EN)
#> collate     nl_BE.UTF-8
#> ctype       nl_BE.UTF-8
#> tz          Europe/Brussels
#> date        2026-09-15
#> pandoc      3.10.2 @ /opt/homebrew/bin/ (via rmarkdown)
#> quarto      1.9.37 @ /usr/local/bin/quarto
#>
#> - Packages -----
#> package      * version date (UTC) lib source
#> cli           3.6.6   2026-04-09 [1] CRAN (R 4.6.0)
#> digest        0.6.39  2025-11-19 [1] CRAN (R 4.6.0)
#> evaluate      1.0.5   2025-08-27 [1] CRAN (R 4.6.0)
```

```

#> fastmap      1.2.0    2024-05-15 [1] CRAN (R 4.6.0)
#> htmltools    0.5.9    2025-12-04 [1] CRAN (R 4.6.0)
#> jsonlite     2.0.0    2025-03-27 [1] CRAN (R 4.6.0)
#> knitr        1.51     2025-12-20 [1] CRAN (R 4.6.0)
#> otel         0.2.0    2025-08-29 [1] CRAN (R 4.6.0)
#> rlang        1.2.0    2026-04-06 [1] CRAN (R 4.6.0)
#> rmarkdown    2.31     2026-03-26 [1] CRAN (R 4.6.0)
#> sessioninfo  1.2.4    2026-06-04 [1] CRAN (R 4.6.0)
#> xfun         0.59     2026-06-19 [1] CRAN (R 4.6.0)
#> yaml         2.3.12   2025-12-10 [1] CRAN (R 4.6.0)
#>
#> [1] /Library/Frameworks/R.framework/Versions/4.6/Resources/library
#>
#> -----

```