432 Class 5 Slides

github.com/THOMASELOVE/2019-432

2019-02-07

Today's Materials

- Ohio County Health Rankings Data
- Variable Selection via Best Subsets
    - Adjusted R^2
    - Mallows' C_p
    - AIC after Correction for Bias
    - BIC
- Cross-Validating to Compare Two Model-Building Approaches
- Assessing Residual Diagnostic Plots

Setup

```
library(skimr); library(broom); library(car)
library(modelr); library(leaps)
library(tidyverse)

oh_count <- read.csv("data/counties2017a.csv") %>% tbl_df()
```

Ohio County Health Rankings Data http://www.countyhealthrankings.org/rankings/data/oh

Codebook (2017 County Health Rankings), I

Variable	Description
`fips`	FIPS code for county (an ID)
`state`	Ohio in all cases
`county`	County Name (88 counties in Ohio)
`years_lost`	Years of potential life lost before age 75 per 100,000 population (age-adjusted, 2012-14)
`population`	County population, Census Population Estimates, 2015
`female`	% female (Census Population Estimates, 2015)
`rural`	3 categories from % rural (0-20: Urban, 20.1-50: Suburban, 50.1+: Rural; Census 2015)
`non_white`	4 categories from 100 - % white non-hispanic: (> 20: High, 10.1-20: Medium, 5.1-10: Low, <=5: Very Low, Census 2015)

Codebook (2017 County Health Rankings), II

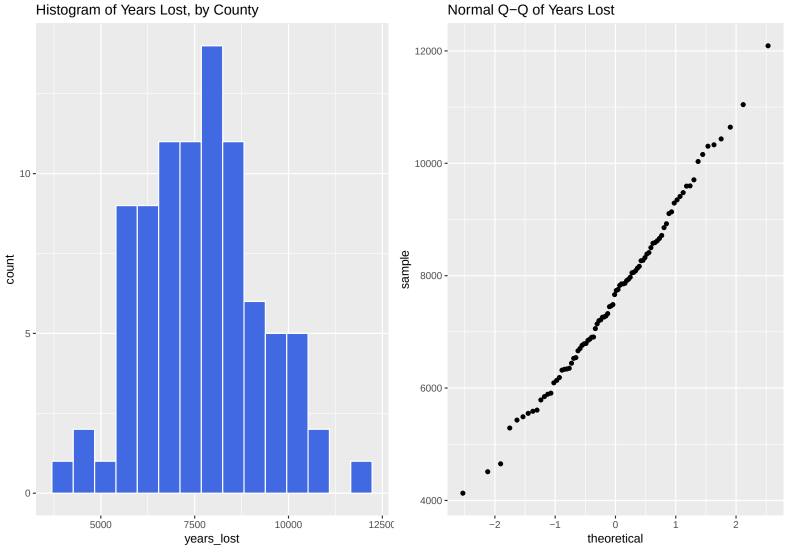
Variable	Description
`sroh_fairpoor`	% of adults reporting fair or poor health (age-adjusted via 2015 BRFSS)
`smoker_pct`	% of adults who currently smoke (2015 BRFSS)
`food_envir`	Food environment index (0 = worst, 10 = best) (via USDA Map the Meal 2014)
`exer_access`	% of population with adequate access to locations for physical activity (several sources)
`income_ratio`	Ratio of household income at the 80th percentile to income at the 20th percentile (ACS 2011-15)
`air_pollution`	Mean daily density of fine particulate matter in micrograms per cubic meter (PM2.5)
`health_costs`	Health Care Costs (from Dartmouth Atlas, 2014)

Basic Data Summaries

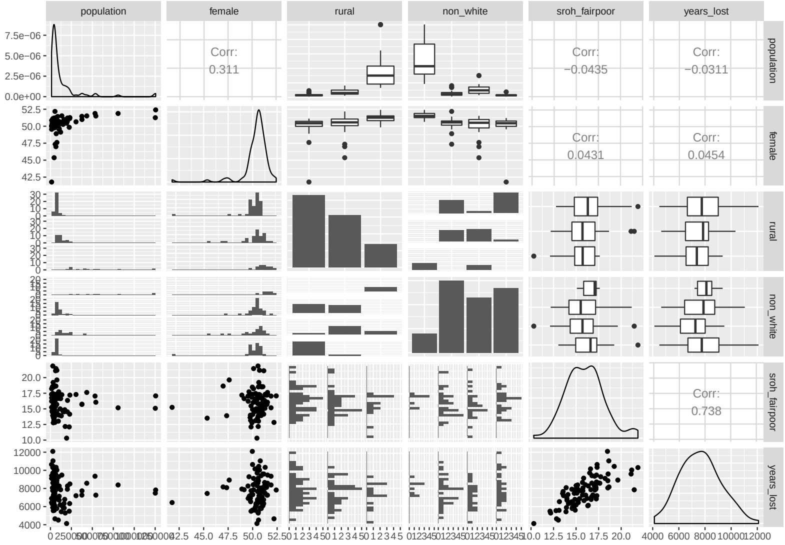
```
oh_count %>% select(-fips, -state, -county) %>% skim()
```

```
> oh_count %>% select(-fips, -state, -county) %>% skim()
Skim summary statistics
 n obs: 88
 n variables: 12

-- Variable type:character ------------------------------------------------------
  variable missing complete  n min max empty n_unique
 non_white       0       88 88   3   8     0        4
     rural       0       88 88   5   8     0        3

-- Variable type:numeric --------------------------------------------------------
      variable missing complete  n      mean        sd      p0      p25      p50       p75    p100  hist
 air_pollution       0       88 88     11.38      0.47   10.5     11.1     11.3      11.7     13
   exer_access       0       88 88     68.19     17.43   26.2     58.18    69.73     80.09    96.23
        female       0       88 88     50.34      1.38   41.78    50.05    50.58     50.96    52.41
    food_envir       0       88 88      7.4       0.67    5.3      7        7.45      7.8      8.9
  health_costs       0       88 88  10158.06    859.43 8274.48  9650.2  10093.36  10577.49 13702.91
  income_ratio       0       88 88      4.33      0.6     3.45     3.94     4.21      4.57     7.24
    population       0       88 88 131970.72 216261.12 13048   36982.25 57733.5  123712.75 1255921
    smoker_pct       0       88 88     19.33      2.05   13.82    18.23    19.28     20.61    24.53
  sroh_fairpoor      0       88 88     15.99      2.14   10.31    14.58    15.86     17.21    21.86
    years_lost       0       88 88   7659.12   1563.34 4129     6538.75  7700      8597.5  12091
```

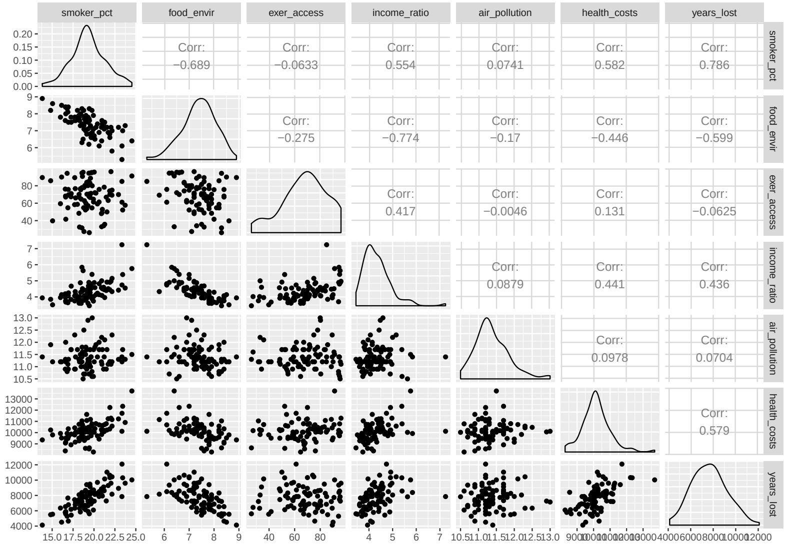
Our Outcome: Age-Adjusted Years Lost



Scatterplot Matrix with `GGally`, Part I



Scatterplot Matrix with `GGally`, Part II



The “Kitchen Sink” Model?

Predict years_lost using 11 predictors.

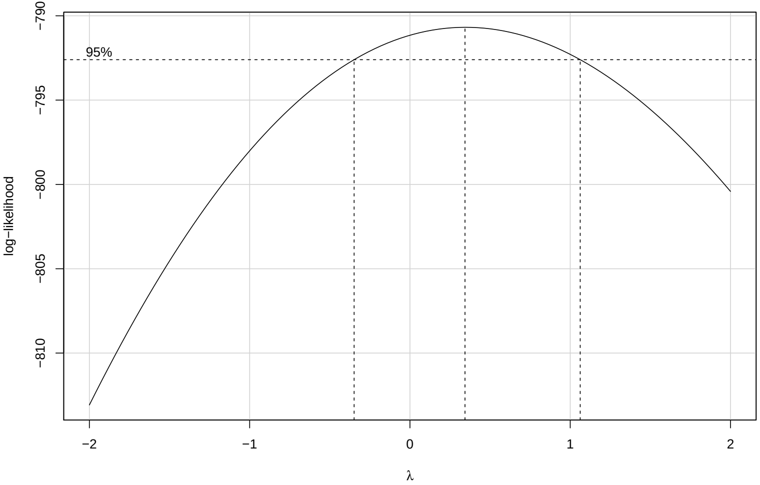
```
m_ks <- lm(years_lost ~ population + female + rural +
               non_white + sroh_fairpoor + smoker_pct +
               food_envir + exer_access + income_ratio +
               air_pollution + health_costs,
           data = oh_count)

glance(m_ks) %>% select(r.squared, adj.r.squared)
```

```
# A tibble: 1 x 2
  r.squared adj.r.squared
      <dbl>         <dbl>
1     0.689         0.630
```

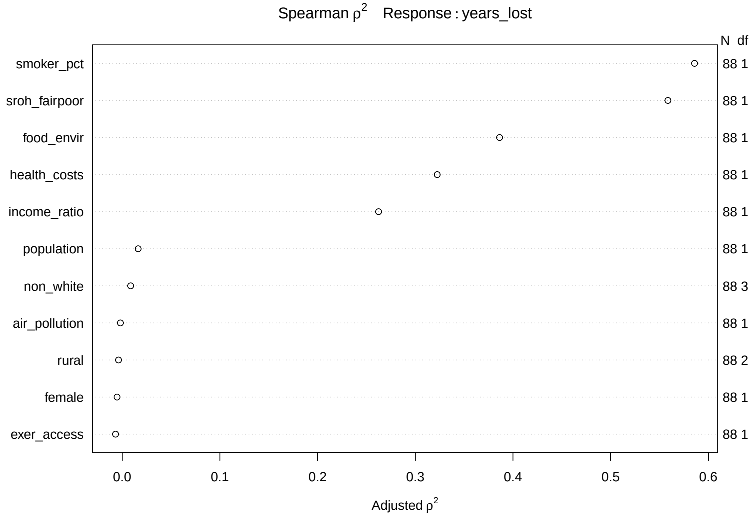
- 11 predictors with 88 observations?

Box-Cox plot: Outcome transformation?

```
boxCox(m_ks)
```



Spearman ρ^2 Plot (code in R Markdown)



Spearman ρ^2 Plot (code)

```
plot(Hmisc::spearman2(years_lost ~ population + female +
                          rural + non_white +
                          sroh_fairpoor + smoker_pct +
                          food_envir + exer_access +
                          income_ratio + air_pollution +
                          health_costs, data = oh_count))
```

Using "Best Subsets" to Select Variables

Using “Best Subsets” to Select Variables

We’ll consider models using some combination of the 11 available meaningful predictors.

```
bs_preds <- with(oh_count, cbind(population, female, rural,
                              non_white, sroh_fairpoor,
                              smoker_pct, food_envir,
                              exer_access, income_ratio,
                              air_pollution, health_costs))
```

We’ll look for models using up to 8 of those predictors.

```
bs_subs <- regsubsets(bs_preds,
                      y = oh_count$years_lost,
                      nvmax = 8)
bs_mods <- summary(bs_subs)
```

Looking at bs_mods

```
bs_mods
```

```
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         population female rural non_white sroh_fairpoor smoker_pct food_envir
1  ( 1 ) " "        " "    " "   " "       " "           "*"        " "
2  ( 1 ) " "        " "    " "   " "       " "           "*"        " "
3  ( 1 ) " "        " "    " "   " "       "*"           "*"        " "
4  ( 1 ) " "        " "    " "   " "       " "           "*"        "*"
5  ( 1 ) " "        "*"    " "   " "       " "           "*"        "*"
6  ( 1 ) " "        "*"    " "   " "       " "           "*"        "*"
7  ( 1 ) " "        "*"    " "   " "       "*"           "*"        "*"
8  ( 1 ) " "        "*"    " "   "*"       "*"           "*"        "*"
         exer_access income_ratio air_pollution health_costs
1  ( 1 ) " "         " "          " "           " "
2  ( 1 ) " "         " "          " "           "*"
3  ( 1 ) " "         " "          " "           "*"
4  ( 1 ) " "         "*"          " "           "*"
5  ( 1 ) " "         "*"          " "           "*"
6  ( 1 ) "*"         "*"          " "           "*"
7  ( 1 ) "*"         "*"          " "           "*"
8  ( 1 ) "*"         "*"          " "           "*"
```

Look at the models that “win”

```r
bs_mods$which
```

```
> bs_mods$which
  (Intercept) population female rural non_white sroh_fairpoor smoker_pct food_envir exer_access income_ratio air_pollution health_costs
1        TRUE      FALSE  FALSE FALSE     FALSE         FALSE       TRUE      FALSE       FALSE        FALSE         FALSE        FALSE
2        TRUE      FALSE  FALSE FALSE     FALSE         FALSE       TRUE      FALSE       FALSE        FALSE         FALSE         TRUE
3        TRUE      FALSE  FALSE FALSE     FALSE          TRUE       TRUE      FALSE       FALSE        FALSE         FALSE         TRUE
4        TRUE      FALSE  FALSE FALSE     FALSE         FALSE       TRUE       TRUE       FALSE         TRUE         FALSE         TRUE
5        TRUE      FALSE   TRUE FALSE     FALSE         FALSE       TRUE       TRUE       FALSE         TRUE         FALSE         TRUE
6        TRUE      FALSE   TRUE FALSE     FALSE         FALSE       TRUE       TRUE        TRUE         TRUE         FALSE         TRUE
7        TRUE      FALSE   TRUE FALSE     FALSE          TRUE       TRUE       TRUE        TRUE         TRUE         FALSE         TRUE
8        TRUE      FALSE   TRUE FALSE      TRUE          TRUE       TRUE       TRUE        TRUE         TRUE         FALSE         TRUE
```

Sometimes easier to transpose this. . .

```r
t(bs_mods$which)
```

```
> t(bs_mods$which)
                  1     2     3     4     5     6     7     8
(Intercept)    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
population    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
female        FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
rural         FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
non_white     FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
sroh_fairpoor FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
smoker_pct     TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
food_envir    FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
exer_access   FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
income_ratio  FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
air_pollution FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
health_costs  FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

Look at the R-square values for each “winning” model

```r
bs_mods$rsq
```

```
[1] 0.6172471 0.6397030 0.6460605 0.6530869 0.6649312
[6] 0.6730306 0.6783975 0.6802613
```

```r
bs_mods$adjr2
```

```
[1] 0.6127964 0.6312255 0.6334198 0.6363682 0.6445001
[6] 0.6488107 0.6502573 0.6478827
```

Place winning results in bs_winners

```
bs_winners <- tbl_df(bs_mods$which)
bs_winners$k <- 2:9 ## in general, this is 2:(nvmax + 1)
bs_winners$r2 <- bs_mods$rsq
bs_winners$adjr2 <- bs_mods$adjr2
bs_winners$cp <- bs_mods$cp
bs_winners$bic <- bs_mods$bic
```

Calculate Bias-Corrected AIC from Residual Sum of Squares

This requires specifying the sample size (`temp.n`) and the number of inputs that you'll look at in your largest subset (here, we limited the number of variables to 8 with `nvmax` and so that's 9 inputs, including the intercept term.)

```
temp.n <- nrow(oh_count)
temp.inputs <- 9 ## nvmax + 1

bs_mods$aic.corr <- temp.n*log(bs_mods$rss / temp.n) +
    2*(2:temp.inputs) +
    (2 * (2:temp.inputs) * ((2:temp.inputs)+1) /
        (temp.n - (2:temp.inputs) - 1))

bs_winners$aic.corr <- bs_mods$aic.corr
```

Detailed Breakdown: `bs_winners`

Inputs	Predictors	Raw r^2	Adj. r^2	C_p	BIC	AIC_c
2	`smoker_pct`	.617	.613	8.0	-75.6	1213.0
3	+ `health_costs`	.640	.631	4.6	**-76.4**	**1209.9**
4	+ `sroh_fairpoor`	.646	.633	5.1	-73.5	1210.5
5	(*see below*)	.653	.636	**5.4**	-70.8	1211.0
6	+ `female`	.665	.645	4.5	-69.4	1210.2
7	+ `exer_access`	.673	.649	4.6	-67.0	1210.4
8	+ `sroh_fairpoor`	.678	**.650**	5.3	-64.0	1211.4
9	+ `non_white`	.680	.648	6.9	-60.0	1213.4

- The "best" model with 5 inputs includes `smoker_pct`, `health_costs`, `food_envir` and `income_ratio`.
- That model forms the basis for the "best" models with 6-9 inputs.

Resulting `bs_winners` tibble

```
head(bs_winners, 2)
```

```
# A tibble: 2 x 18
  `(Intercept)` population female rural non_white
  <lgl>         <lgl>      <lgl>  <lgl> <lgl>
1 TRUE          FALSE      FALSE  FALSE FALSE
2 TRUE          FALSE      FALSE  FALSE FALSE
# ... with 13 more variables: sroh_fairpoor <lgl>,
#   smoker_pct <lgl>, food_envir <lgl>, exer_access <lgl>,
#   income_ratio <lgl>, air_pollution <lgl>,
#   health_costs <lgl>, k <int>, r2 <dbl>, adjr2 <dbl>,
#   cp <dbl>, bic <dbl>, aic.corr <dbl>
```

str(bs_winners)

```
> str(bs_winners)
Classes 'tbl_df', 'tbl' and 'data.frame':	8 obs. of  18 variables:
 $ (Intercept)  : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ population   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ female       : logi  FALSE FALSE FALSE FALSE TRUE TRUE ...
 $ rural        : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ non_white    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ sroh_fairpoor: logi  FALSE FALSE TRUE FALSE FALSE FALSE ...
 $ smoker_pct   : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ food_envir   : logi  FALSE FALSE FALSE TRUE TRUE TRUE ...
 $ exer_access  : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
 $ income_ratio : logi  FALSE FALSE FALSE TRUE TRUE TRUE ...
 $ air_pollution: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ health_costs : logi  FALSE TRUE TRUE TRUE TRUE TRUE ...
 $ k            : int  2 3 4 5 6 7 8 9
 $ r2           : num  0.617 0.64 0.646 0.653 0.665 ...
 $ adjr2        : num  0.613 0.631 0.633 0.636 0.645 ...
 $ cp           : num  8 4.6 5.07 5.38 4.54 ...
 $ bic          : num  -75.6 -76.4 -73.5 -70.8 -69.4 ...
 $ aic.corr     : num  1213 1210 1210 1211 1210 ...
```

If You're Curious: A Stepwise Fit

```
step(lm(years_lost ~ population + female + rural +
            non_white + sroh_fairpoor + smoker_pct +
            food_envir + exer_access + income_ratio +
            air_pollution + health_costs, data = oh_count))
```

using backwards elimination produces the model containing:

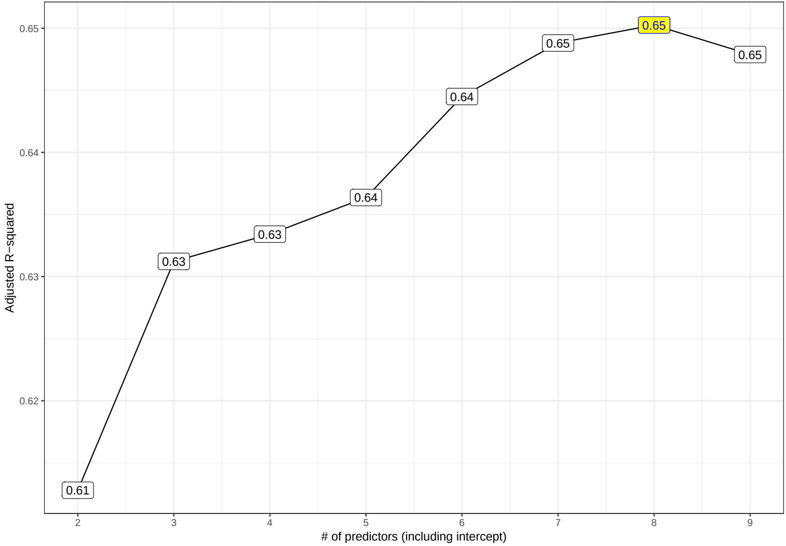
- `smoker_pct`, `health_costs`, `food_envir`, `income_ratio`, `female`, and `exer_access`
- also known as what "best subsets" chose for its model 7.

Building the “Best Subsets” Plots

Adjusted R-square plot using ggplot2

```
p1 <- ggplot(bs_winners, aes(x = k, y = adjr2,
                       label = round(adjr2,2))) +
    geom_line() +
    geom_label() +
    geom_label(data = subset(bs_winners,
                             adjr2 == max(adjr2)),
               aes(x = k, y = adjr2, label = round(adjr2,2)),
               fill = "yellow", col = "blue") +
    theme_bw() +
    scale_x_continuous(breaks = 2:9) +
    labs(x = "# of predictors (including intercept)",
         y = "Adjusted R-squared")
```

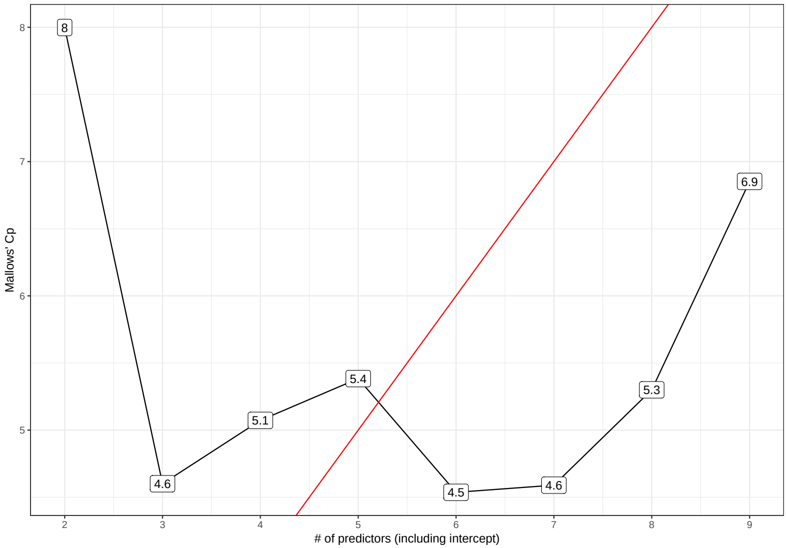
Adjusted R-square plot using `ggplot2`



Mallows' C_p plot using ggplot2

```
p2 <- ggplot(bs_winners, aes(x = k, y = cp,
                             label = round(cp,1))) +
    geom_line() +
    geom_label() +
    geom_abline(intercept = 0, slope = 1,
                col = "red") +
    theme_bw() +
    scale_x_continuous(breaks = 2:9) +
    labs(x = "# of predictors (including intercept)",
         y = "Mallows' Cp")
```

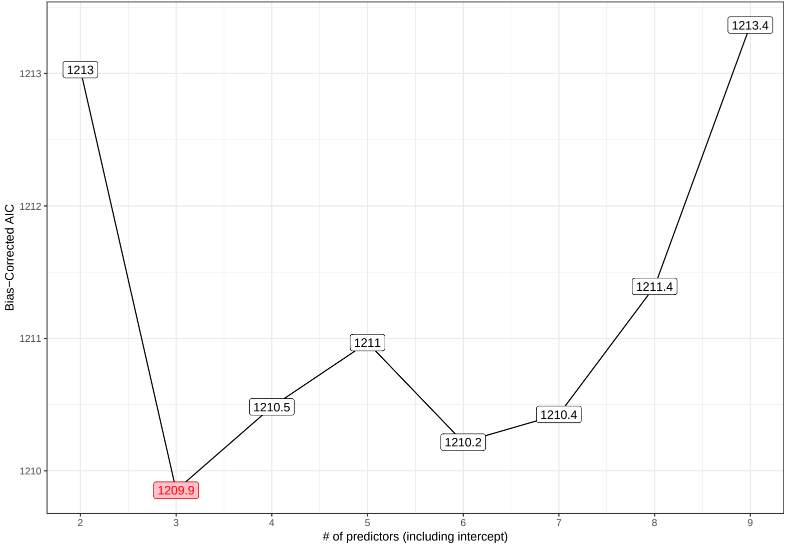
Mallows' C_p plot using `ggplot2`



Corrected AIC plot using ggplot2

```
p3 <- ggplot(bs_winners, aes(x = k, y = aic.corr,
                             label = round(aic.corr,1))) +
    geom_line() +
    geom_label() +
    geom_label(data = subset(bs_winners,
                             aic.corr == min(aic.corr)),
               aes(x = k, y = aic.corr),
               fill = "pink", col = "red") +
    theme_bw() +
    scale_x_continuous(breaks = 2:9) +
    labs(x = "# of predictors (including intercept)",
         y = "Bias-Corrected AIC")
```

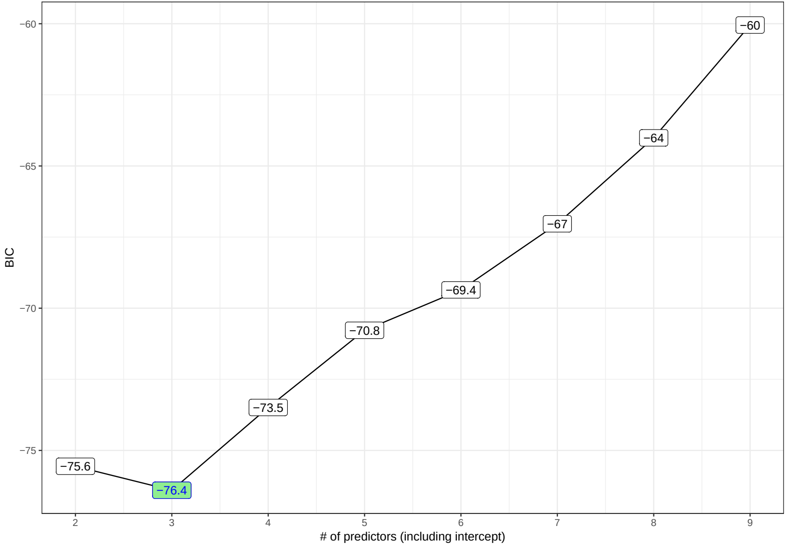
Corrected AIC plot using `ggplot2`



BIC plot using ggplot2

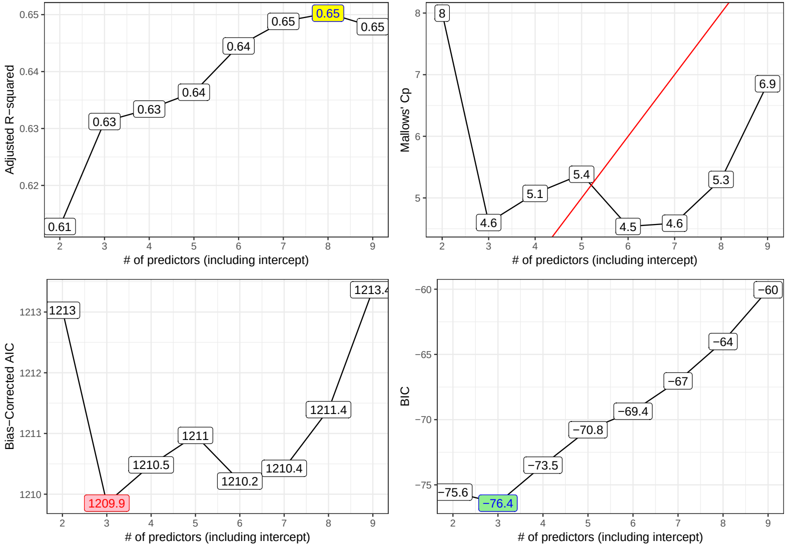
```
p4 <- ggplot(bs_winners, aes(x = k, y = bic,
                             label = round(bic,1))) +
    geom_line() +
    geom_label() +
    geom_label(data = subset(bs_winners, bic == min(bic)),
               aes(x = k, y = bic),
               fill = "lightgreen", col = "blue") +
    theme_bw() +
    scale_x_continuous(breaks = 2:9) +
    labs(x = "# of predictors (including intercept)",
         y = "BIC")
```

BIC plot using `ggplot2`



All Four Plots Together

```r
gridExtra::grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Candidate Models include

Inputs	Raw r^2	Adj. r^2	C_p	BIC	AIC_c
3	.640	.631	4.6	**-76.4**	**1209.9**
5	.653	.636	**5.4**	-70.8	1211.0
8	.678	**.650**	5.3	-64.0	1211.4

- 3: `smoker_pct` + `health_costs`
- 5: Model 3 + `food_envir` + `income_ratio`
- 8: Model 5 + `female` + `exer_access` + `sroh_fairpoor`

Comparing our Candidate Models in our Training Sample

In-Sample Comparisons of our Candidate Models

```
m3 <- lm(years_lost ~ smoker_pct + health_costs,
         data = oh_count)
m5 <- lm(years_lost ~ smoker_pct + health_costs +
             food_envir + income_ratio, data=oh_count)
m8 <- lm(years_lost ~ smoker_pct + health_costs +
              food_envir + income_ratio + female +
             exer_access + sroh_fairpoor, data=oh_count)
```

Models are **nested** so comparisons within samples are straightforward.

Comparisons in-sample with anova

```
anova(m3, m5, m8)
```

```
Analysis of Variance Table

Model 1: years_lost ~ smoker_pct + health_costs
Model 2: years_lost ~ smoker_pct + health_costs + food_envir +
Model 3: years_lost ~ smoker_pct + health_costs + food_envir +
    female + exer_access + sroh_fairpoor
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1     85 76610187
2     83 73764357  2   2845831 1.6647 0.1957
3     80 68382551  3   5381806 2.0987 0.1069
```

Comparisons in-sample with AIC

```
a <- AIC(m3, m5, m8)
b <- BIC(m3, m5, m8); b$model <- row.names(b)
left_join(a, b)
```

```
Joining, by = "df"
```

```
  df      AIC      BIC model
1  4 1461.301 1471.210    m3
2  6 1461.970 1476.834    m5
3  9 1461.303 1483.599    m8
```

What if the models you're comparing aren't nested?

What if you're comparing:

- Model A: `lm(y = x1 + x2 + x3, data = dataset)`
- Model B: `lm(y = x1 + x4 + x5, data = dataset)`

Then ...

- default *p* values from the ANOVA table comparing Model A to Model B aren't reasonable
- AIC and BIC are OK, can also used adjusted R^2 to help make a decision within the model building sample
- Still useful to think about out-of-sample prediction and cross-validation

Comparing out-of-sample predictive ability of our Candidate Models with cross-validation

10-fold Cross-Validation for Model 3

```
set.seed(432012)

cv_3 <- oh_count %>%
  crossv_kfold(k = 10) %>%
  mutate(model = map(train, ~ lm(years_lost ~
                   smoker_pct + health_costs, data = .)))

cv3_pred <- cv_3 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv3_res <- cv3_pred %>%
  summarize(Model = "3",
            RMSE = sqrt(mean((years_lost - .fitted) ^2)),
            MAE = mean(abs(years_lost - .fitted)))
```

10-fold Cross-Validation for Model 5

```
set.seed(432013)

cv_5 <- oh_count %>%
  crossv_kfold(k = 10) %>%
  mutate(model = map(train, ~ lm(years_lost ~
                          smoker_pct + health_costs +
                          food_envir + income_ratio, data = .)))

cv5_pred <- cv_5 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv5_res <- cv5_pred %>%
  summarize(Model = "5",
            RMSE = sqrt(mean((years_lost - .fitted) ^2)),
            MAE = mean(abs(years_lost - .fitted)))
```

10-fold Cross-Validation for Model 8

```
set.seed(432014)

cv_8 <- oh_count %>%
  crossv_kfold(k = 10) %>%
  mutate(model = map(train, ~ lm(years_lost ~
                           smoker_pct + health_costs +
                           food_envir + income_ratio +
                           female + exer_access +
                           sroh_fairpoor, data = .)))

cv8_pred <- cv_8 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv8_res <- cv8_pred %>%
  summarize(Model = "8",
            RMSE = sqrt(mean((years_lost - .fitted) ^2)),
            MAE = mean(abs(years_lost - .fitted)))
```

Cross-Validation Results

```
bind_rows(cv3_res, cv5_res, cv8_res)
```

```
# A tibble: 3 x 3
  Model  RMSE   MAE
  <chr> <dbl> <dbl>
1 3      975.  785.
2 5      976.  797.
3 8     1004.  809.
```

Fitting the Chosen Model

Fitting the Chosen Model

```
m3 <- lm(years_lost ~ smoker_pct + health_costs,
         data = oh_count)

arm::display(m3)
```

```
lm(formula = years_lost ~ smoker_pct + health_costs, data = oh
             coef.est coef.se
(Intercept)  -5749.51  1248.81
smoker_pct     517.62    61.10
health_costs     0.34     0.15
---
n = 88, k = 3
residual sd = 949.37, R-Squared = 0.64
```

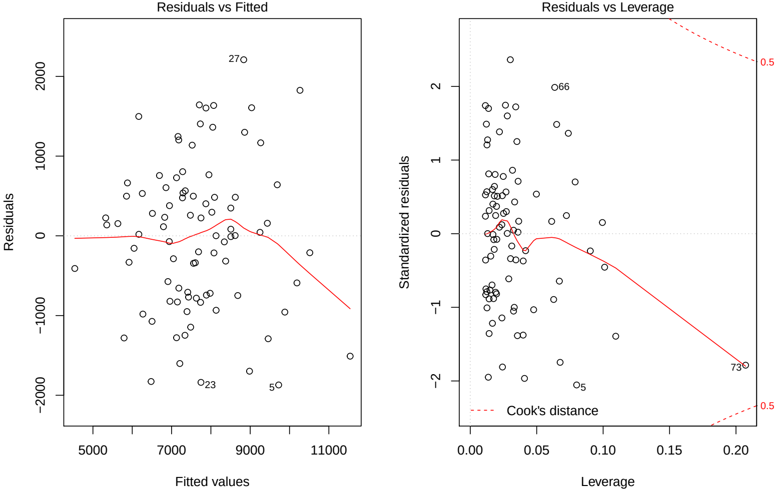
Fitting the Chosen Model

```
glance(m3) %>% print.data.frame
```

```
  r.squared adj.r.squared    sigma statistic      p.value
1  0.639703     0.6312255 949.3663  75.45825 1.439049e-19
  df    logLik      AIC     BIC deviance df.residual
1  3 -726.6504 1461.301 1471.21 76610187          85
```

Residual Plots for the Chosen Model

```
par(mfrow = c(1,2)); plot(m3, which = c(1, 5))
```



Coming Up

- Another Example: Low Birth Weight
- More on Cross-Validation of Linear Regression Models
- Limitations of Best Subsets
- More on the Spearman ρ^2 Plot
  - Spending Degrees of Freedom on Non-Linearity
- Building Non-Linear Predictors with
  - Polynomial Functions
  - Product Terms
  - Splines, specifically Restricted Cubic Splines
- Building a Nomogram for a Linear Regression

not to mention . . .

- Logistic Regression