A Three Day Course in Applied Regression Analysis

[3] ANOVA, F-testing, and Model Selection

Matt Taddy, University of Chicago Booth School of Business

faculty.chicagobooth.edu/matt.taddy/teaching

Variable Selection

So far, we've focused on building regression models, testing, evaluating model fit, and prediction.

Today, we look at how to choose from competing models. This comes down to choosing the regression variables.

Two steps:

1. Select the "Universe of Variables".
2. Choose the best model.

This is very important! And also difficult.

Variable Selection

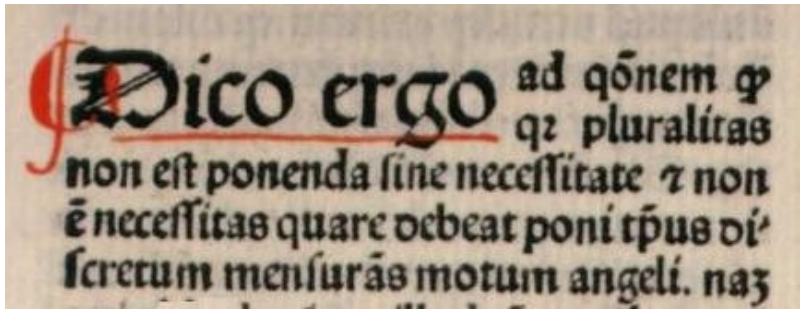
The universe of variables includes all possible covariates that you think might have a linear effect on the response.

You decide on this set through your experience and discipline based knowledge (and are limited by data availability).

- Consult subject matter research and experts.
- Consider carefully what variables have explanatory power, and how they should be transformed.
- If you can avoid it, don't just throw everything in.

Variable Selection

We need a method for selecting a final regression specification.
Why not just include all of the variables and be done with it?



Ockhams Razor: **Plurality ought not be posited without necessity.**

Overly complicated models lead to bad forecasts:

Over-fit $\Rightarrow$ less general model.

R^2 for multiple regression

Recall R^2 as a measure of "fit":

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

And the correlation interpretation:

$$R^2 = \text{cor}^2(\hat{Y}, Y) = r^2_{\hat{y}y}$$

($r_{\hat{y}y} = r_{xy}$ in SLR since $\text{cor}(X, \hat{Y}) = 1$.)

Goodness of Fit

This indicates how we can ask "big picture" questions:

Is this regression worthwhile? Does the model fit?

What is the effect of grouping covariates?

Do we need this *group* of covariates.

The F-test

Revisit the F test for your ANOVA results:

$$f = \frac{SSR/(p-1)}{SSE/(n-p)} = \frac{R^2/(p-1)}{(1-R^2)/(n-p)}$$

If f is big, then the regression is "working":

- Big SSR relative to SSE.
- R^2 close to one.

As usual, we need a statistical notion of "how big is big".

The F-test

Recall what we are testing:

$$H_0 : \beta_1 = \beta_2 = \ldots \beta_d = 0$$
$$H_1 : \text{at least one } \beta_j \neq 0.$$

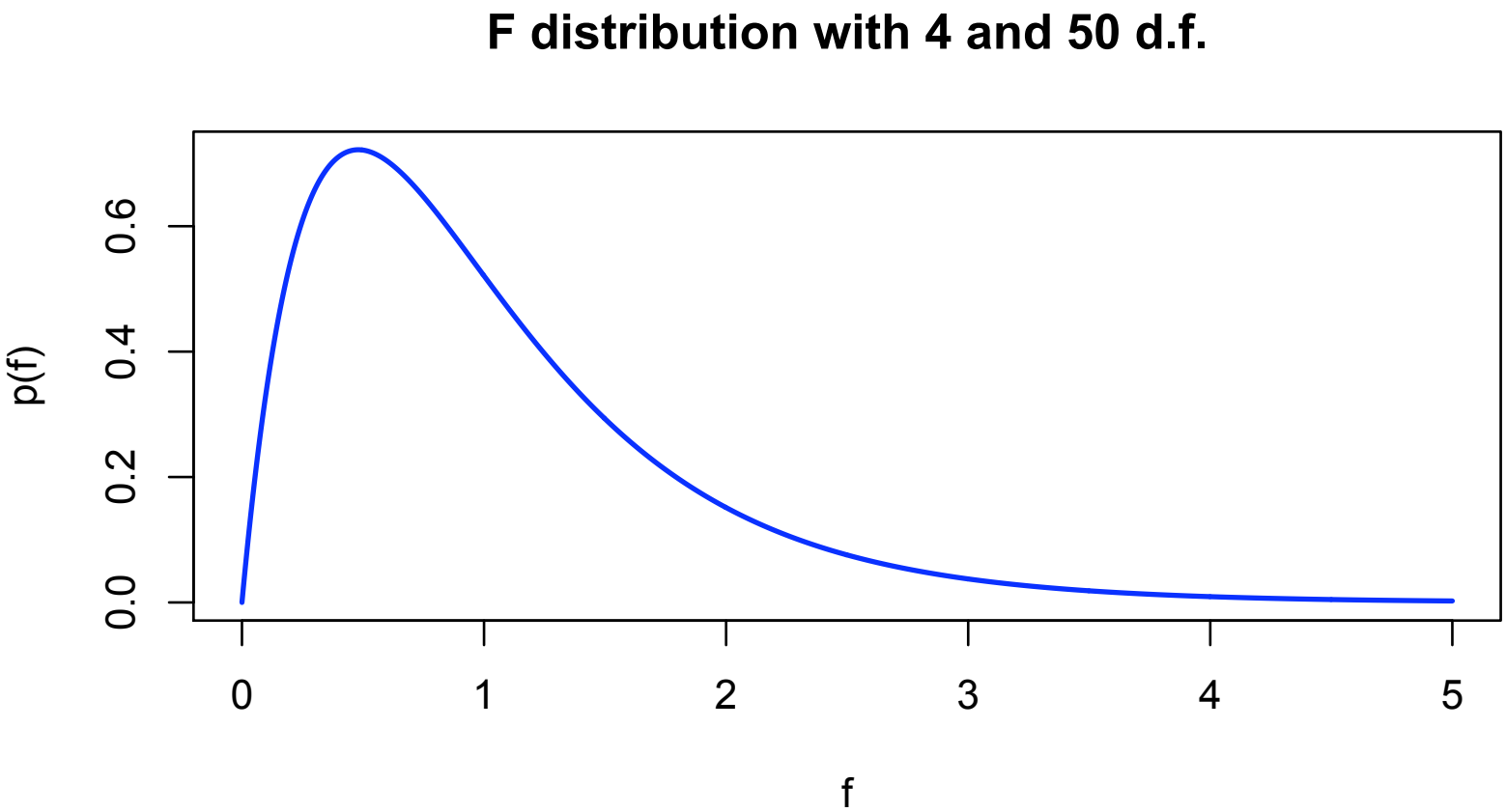
Under H_0, f has $F_{p-1,n-p}$ distribution with $p-1$ numerator and $n-p$ denominator degrees of freedom.

- The F has decreasing variance as the df's increase.
- Generally, $f > 4$ is very significant (reject the null).

The p-value for this test is $\varphi = \Pr(F_{p-1,n-p} > f)$.

The F-test

What kind of distribution is this?



It is a right skewed, positive valued family of distributions indexed by two parameters (the two df values).

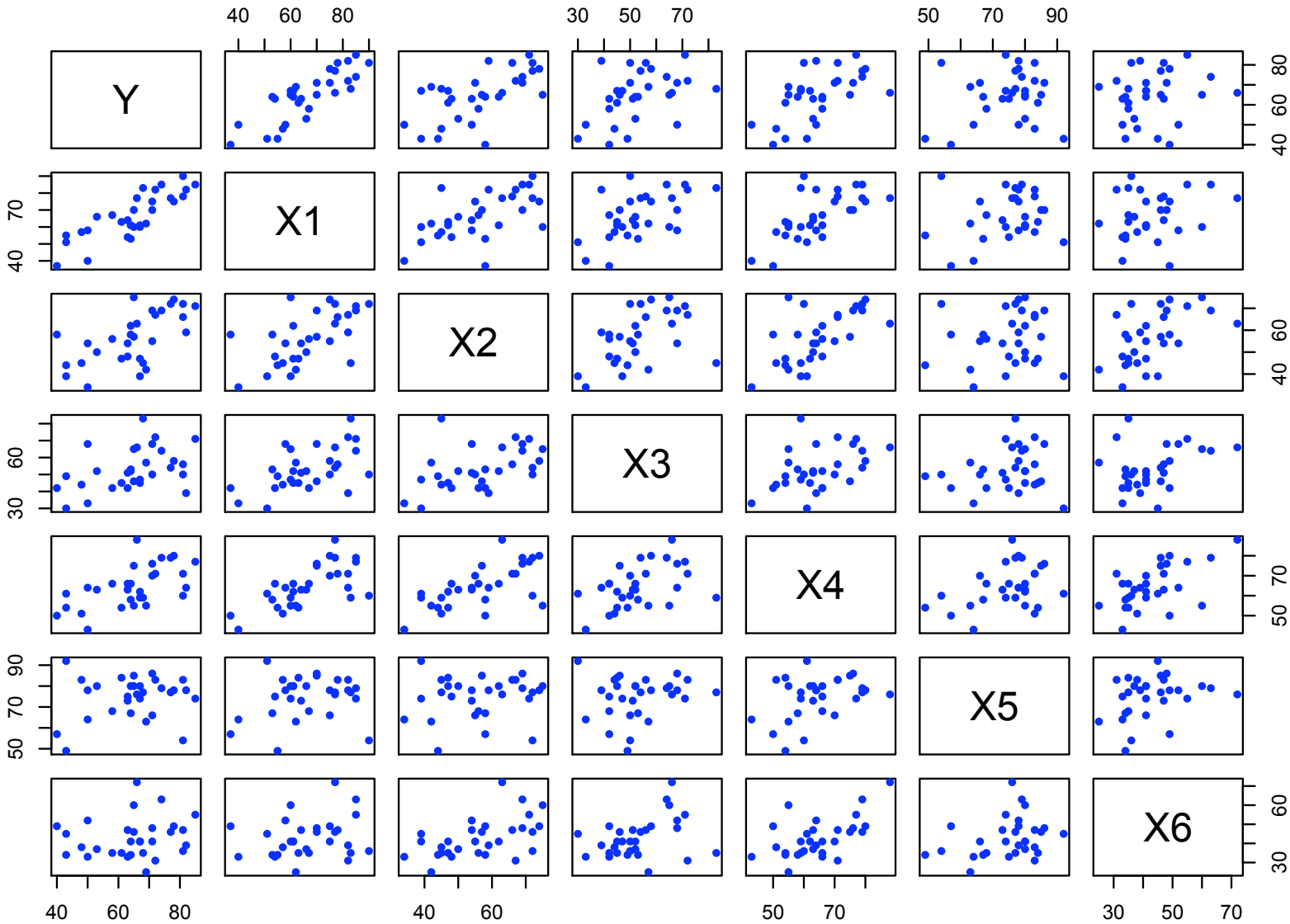
Supervisor Performance Data

This example looks at employee ratings of their supervisor and investigates how they relate to performance metrics.

The Data:

- Y: Overall rating of supervisor
- X1: Handles employee complaints
- X2: Opportunity to learn new things
- X3: Does not allow special privileges
- X4: Raises based on performance
- X5: Overly critical of performance
- X6: Rate of advancing to better jobs

Supervisor Performance Data



Supervisor Performance Data

```
> summary( bosslm <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 ) )

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6)

Residuals:
     Min       1Q   Median       3Q      Max
-10.9418  -4.3555   0.3158   5.5425  11.5990

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.78708   11.58926   0.931 0.361634
X1           0.61319    0.16098   3.809 0.000903 ***
X2           0.32033    0.16852   1.901 0.069925 .
X3          -0.07305    0.13572  -0.538 0.595594
X4           0.08173    0.22148   0.369 0.715480
X5           0.03838    0.14700   0.261 0.796334
X6          -0.21706    0.17821  -1.218 0.235577
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.068 on 23 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.7326, Adjusted R-squared: 0.6628
F-statistic:  10.5 on 6 and 23 DF,  p-value: 1.240e-05
```

F value of 10.5 is very significant (p-value $= 1.24 * 10^{-5}$).

Supervisor Performance Data

It looks (from the t-statistics and p-values) as though only X_1 and X_2 have a significant effect on Y.

What about fitting a reduced model with only these two X's?

```
summary( bosslm2 <- lm(Y ~ X1 + X2) )
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.8709     7.0612   1.398    0.174
X1            0.6435     0.1185   5.432 9.57e-06 ***
X2            0.2112     0.1344   1.571    0.128
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.817 on 27 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.708,  Adjusted R-squared: 0.6864
F-statistic: 32.74 on 2 and 27 DF,  p-value: 6.058e-08
```

Supervisor Performance Data

The full model (6 covariates) has $R^2_{full} = 0.733$, while the second model (2 covariates) has $R^2_{base} = 0.708$.

Is this difference worth 4 extra covariates?

The R^2 will always increase as more variables are added

- If you have more b's to tune, you can get a smaller SSE.
- Least squares is content fit "noise" in the data.
- This is known as overfitting.

More parameters will always result in a better fit to the sample data, but will not necessarily lead to better predictions.

Adjusted R^2

This is the reason some people like to look at adjusted R^2

$$R_a^2 = 1 - s^2/s_y^2$$

Since s^2/s_y^2 is a ratio of variance estimates, R_a^2 will not necessarily increase when new variables are added.

Unfortunately, the adjusted r-square is practically useless!

Problem: There is no theory for inference about R_a^2, so we will not be able to tell "how big is big".

Partial F-test

We're asking

"Is is useful to add these extra covariates to the regression?"

You **always** want to use the simplest model possible.

- Only add covariates if they are truly informative.

Partial F-test

Consider the regression model

$$Y = \beta_0 + \beta_1 X_1 \ldots + \beta_{d_{base}} X_{d_{base}} + \beta_{d_{base}+1} X_{d_{base}+1} \ldots \beta_{d_{full}} X_{d_{full}} + \varepsilon$$

Such that d_{base} is the number of covariates in the base (small) model and $d_{full} > d_{base}$ is the number in the full (larger) model.

The Partial F-test is concerned with the hypotheses

$$H_0 : \beta_{d_{base}+1} = \beta_{d_{base}+2} = \ldots = \beta_{d_{full}} = 0$$

H_1 : at least one $\beta_j \neq 0$ for $j > d_{base}$.

Partial F-test

It turns out that under the null H_0 (i.e. base model is true),

$$f = \frac{(R^2_{full} - R^2_{base})/(p_{full} - p_{base})}{(1 - R^2_{full})/(n - p_{full})} \sim F_{p_{full}-p_{base}, n-p_{full}}$$

- Big f means that $R^2_{full} - R^2_{base}$ is statistically significant.
- Big f means that at least one of the added X's is useful.

Supervisor Performance: Partial F-test

Back to our supervisor data; we want to test

$$H_0 : \beta_3 = \beta_4 = \beta_5 = \beta_6 = 0$$
$$H_1 : \text{at least one } \beta_j \neq 0 \text{ for } j \in 3 \ldots 6.$$

The F-stat is $f = \dfrac{(0.733 - .708)/(6-2)}{(1-.733)/(30-6-1)} = \dfrac{0.00625}{0.0116} = 0.54$

```
> ((0.733-.708)/(6-2))/((1-.733)/(30-7))
[1] 0.5383895
> 1- pf(.54, df1=4, df2=23)
[1] 0.707876
```

A p-value of 0.71 is not at all significant, so we stick with the null hypothesis and assume the base (2 covariate) model.

Supervisor Performance: Partial F-test

As always, your software should do this for you.

Just do "ANOVA" for the full model against the base model:

```
> anova(bosslm2, bosslm)
Analysis of Variance Table

Model 1:  Y ~ X1 + X2
Model 2:  Y ~ X1 + X2 + X3 + X4 + X5 + X6

   Res.Df   RSS   Df   Sum of Sq     F      Pr(>F)
1     27   1254.7
2     23   1149.0  4      105.65   0.5287   0.7158
```

This partial ANOVA just compares SSE's for each model.

The F-test vs t-test

Suppose that you have d covariates, $X_1 \ldots X_d$, and you are considering adding a new covariate X_{d+1} into your model.

The t-test is $H_0 : \beta_{d+1} = 0$ vs $H_1 : \beta_{d+1} \neq 0$, with test statistic $z = b_{d+1}/s_{b_{d+1}}$ and p-value $P(Z_{n-d-2} > |z|)$.

Partial F-test is $H_0 : \beta_{d+1} = 0$ vs $H_1 : \beta_{d+1} \neq 0$, with test stat $f = (SSE_{full} - SSE_{base})/MSE_{full}$ and p-value $P(F_{1,n-d-2} > f)$.

The test hypotheses are exactly the same!

The F-test vs t-test

It turns out that the tests also lead to the same p-values.

Consider testing the two covariate supervisor regression against an even more simple model that includes only X_1.

The t-test asks: is b_2 far enough from zero to be significant?

```
summary(bosslm2)
```

	Estimate	Std.Error	t value	Pr(>|t|)
X2	0.2112	0.1344	1.571	0.128

The F-test asks: is the increase in R^2 significantly big?

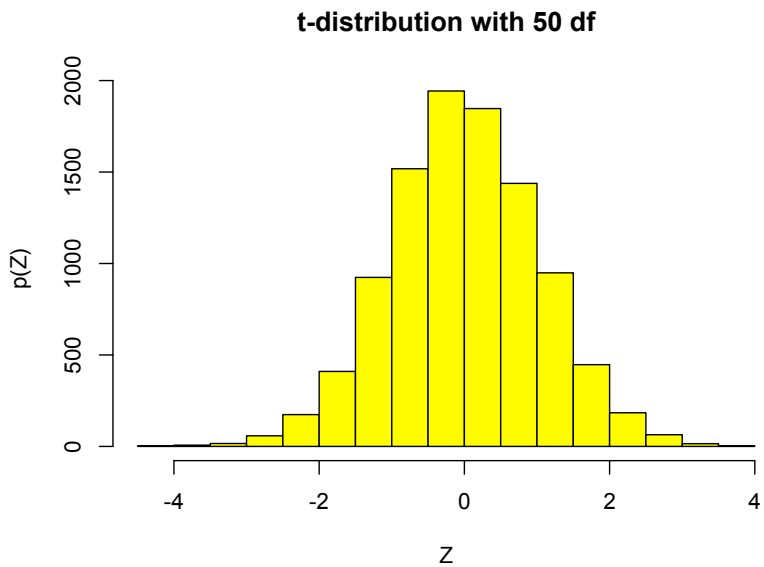
```
anova(lm(Y ~ X1), bosslm2)
```

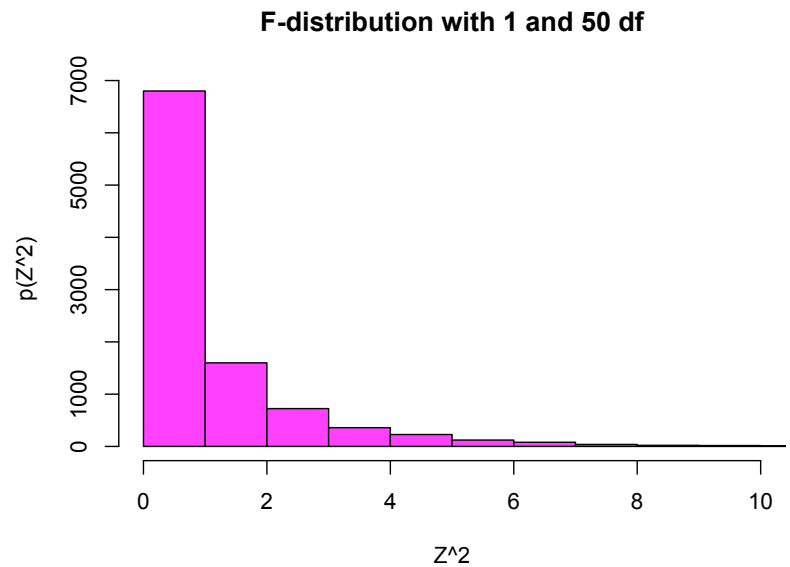
	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
2	27	1254.7	1	114.73	2.4691	0.1278

The F-test vs t-test

Ask the same question, F and t tests give the same answer!

In fact, $f = z^2$ and $P(Z_{n-d-2} > |z|) = P(F_{1,n-d-2} > z^2)$,
so the f stat for one extra variable is just a squared t stat.





The F-test vs t-test

Why not always decide what to include
by just looking at individual t-stats?

Multiple Testing: If you use $\alpha = 0.05 = 1/20$ to judge significance, then you expect to reject a true null $\beta = 0$ about once every 20 tests.

Multicollinearity: If the X's are highly correlated with each other, then s_{b_j}'s will be very big (since you don't know which X_j to regress onto), and you will fail to reject $\beta_j = 0$ for all of the X_j's even if they do have a strong effect on Y.

F-test for Variable Selection

A first method for variable selection

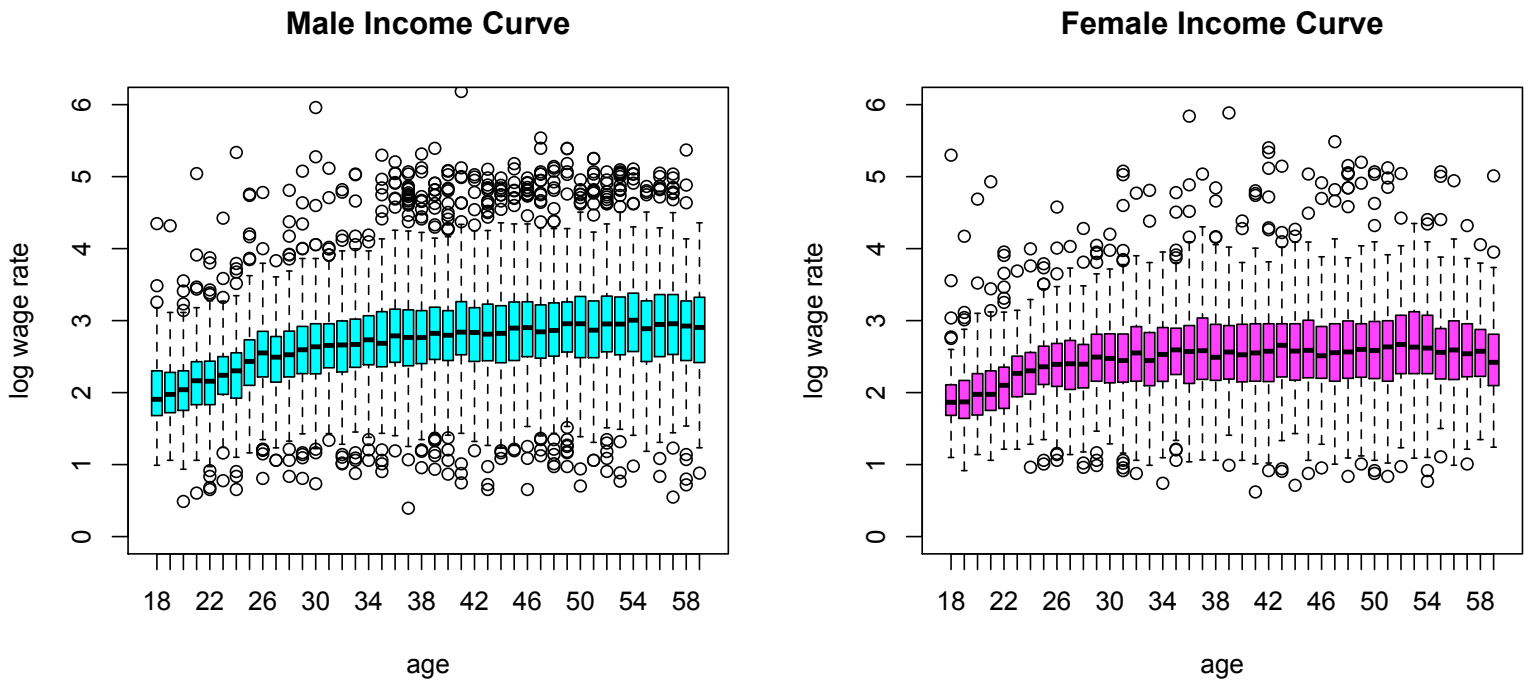
- Do your regression with covariates you think should be in the model.
- Kick out the variables that don't seem significant.
- Use a partial F-test to see if the simple model is good enough.

This only works for a small number of variables, which you've chosen intelligently. Use your head! Nothing is automatic.

Be aware of multicollinearity: b's have small t, but lead to big F.

Male and Female Wage Rates

Consider the relationship between log wage rate (`log(income/hours)`) and age, which is a proxy for experience.

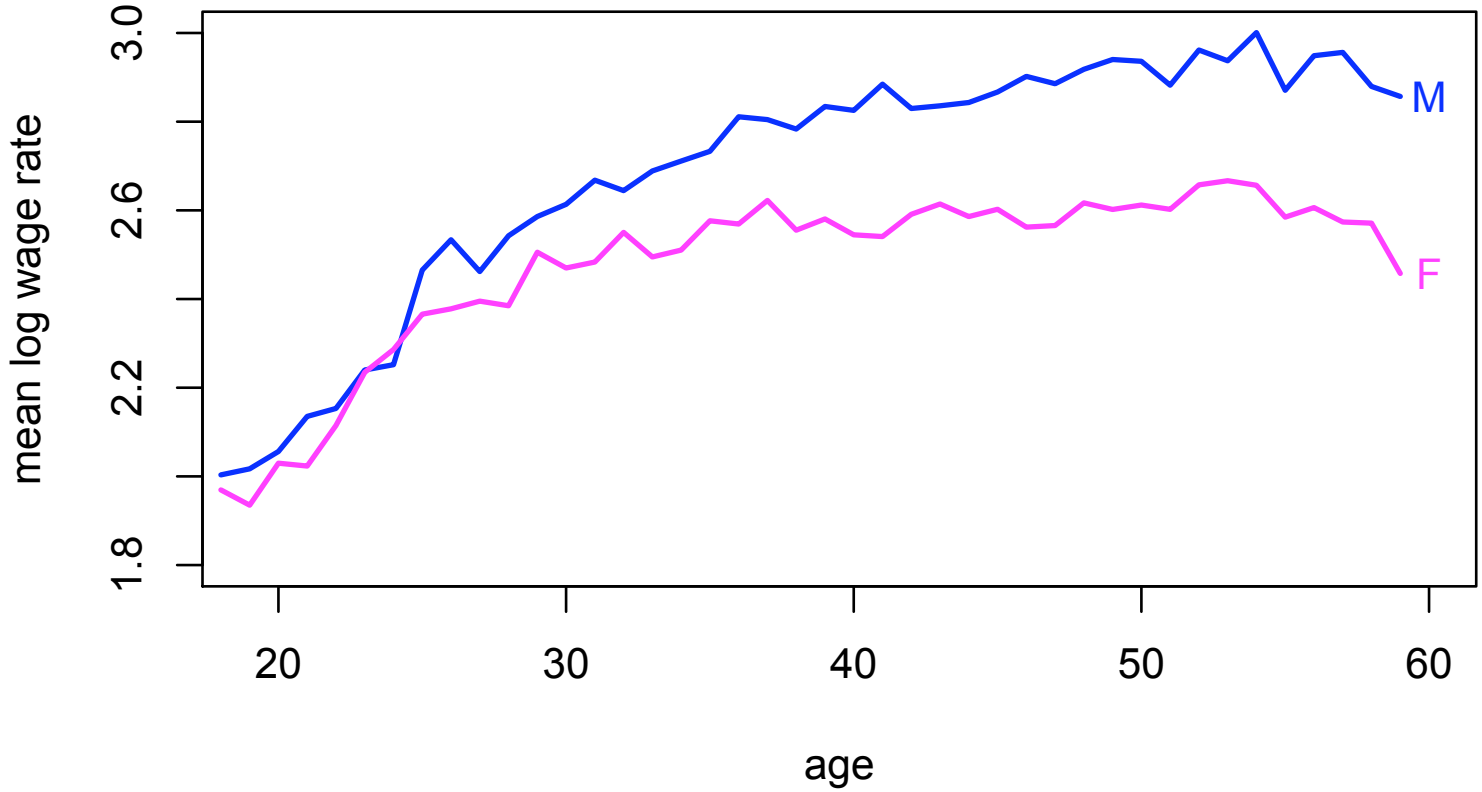


We look at people earning >\$5000, working >500 hrs, and <60 years old.

Male and Female Wage Rates

We see a discrepancy between mean logWR for men and women.

- The female wages flatten at about 30, while men's keep rising.

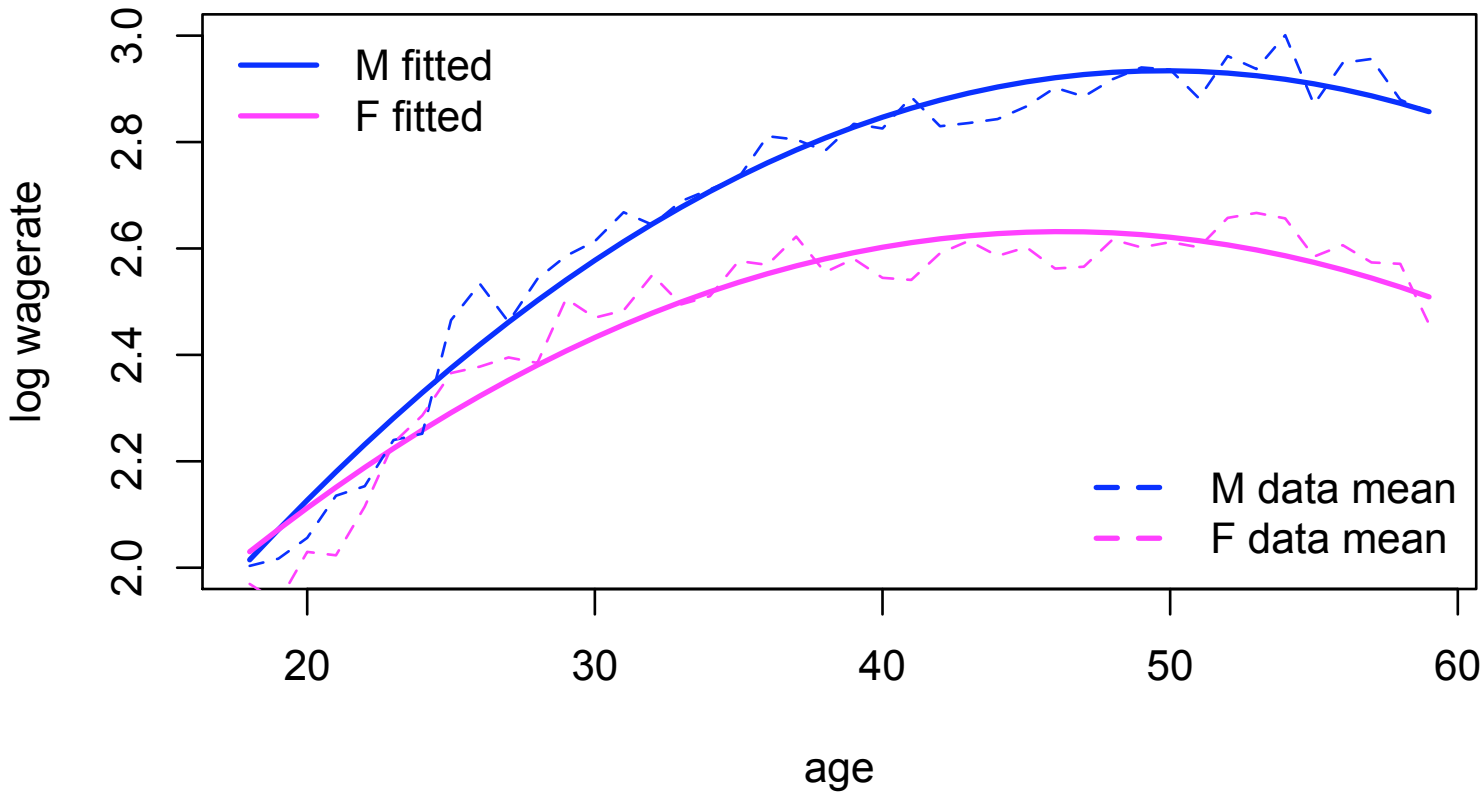


(`tapply(log.WR, age, mean)` gives mean logWR at each age level.)

Male and Female Wage Rates

$$\mathbb{E}[\log(WR)] = 1 + .07age - .0008age^2 + (.02age - .00015age^2 - .34)\mathbb{1}_{[M]}.$$

`lm(log.WR ~ age*sex + age2*sex)`



The age^2 term allows us to capture a nonlinear wage curve.

The interaction terms give us different curves for each sex.

Census Income Data

The null model is then

$$\mathbb{E}[\log(WR)] = 1 + .07age - .0008age^2 + (.02age - .00015age^2 - .34)\mathbb{1}_{[M]}.$$

But there were other possible variables:

- Education: 9 levels from none to phd.
- Marital status: married, divorced, separated, or single.
- Race: white, black, asian, other.

We also need to consider possible interactions.

Census Income Data

A good habit: build a dataframe with your relevant variables.

```
YX <- data.frame(log.WR = log(census$income/census$hours))
```

Build it up with our covariates from last week.

```
YX$age <- census$age
YX$age2 <- census$age^2
YX$sex <- census$sex
```

Use relevel to make "White" and "Married" the intercept

```
YX$race <- relevel(census$race, "White")
YX$marital <- relevel(census$marital, "Married")
```

Census Income Data

Create a bunch of education indicator variables

```
YX$hs <- census$edu=="3.hsgrad"
YX$assoc <- census$edu=="4.assoc"
YX$coll <- census$edu=="5.bachs"
YX$grad <- as.numeric(census$edu)>6
```

And only include "workers"

```
YX <- YX[census$hours > 500 &
                census$income > 5000 &
                       census$age < 60, ]
```

Census Income Data

Consider our base model from last week,
plus main effects for the new covariates.

```
> summary( reg1 <- lm(log.WR ~ age*sex + age2*sex + ., data=YX) )
```

```
Coefficients:
(Intercept)         1.196e+00  6.744e-02  17.737  < 2e-16 ***
age                 4.657e-02  3.549e-03  13.123  < 2e-16 ***
sexM               -2.133e-01  8.594e-02  -2.482  0.01306 *
age2               -4.832e-04  4.510e-05 -10.715  < 2e-16 ***
raceAsian           1.397e-02  1.860e-02   0.751  0.45267
raceBlack          -3.165e-02  1.134e-02  -2.791  0.00525 **
raceNativeAmerican -7.479e-02  3.824e-02  -1.956  0.05048 .
raceOther          -8.112e-02  1.338e-02  -6.063 1.36e-09 ***
maritalDivorced    -6.981e-02  1.066e-02  -6.549 5.91e-11 ***
maritalSeparated   -1.381e-01  1.612e-02  -8.563  < 2e-16 ***
maritalSingle      -1.065e-01  9.413e-03 -11.316  < 2e-16 ***
maritalWidow       -1.502e-01  3.213e-02  -4.674 2.98e-06 ***
hsTRUE              1.499e-01  1.157e-02  12.947  < 2e-16 ***
assocTRUE           3.111e-01  1.146e-02  27.157  < 2e-16 ***
collTRUE            6.082e-01  1.278e-02  47.602  < 2e-16 ***
gradTRUE            7.970e-01  1.498e-02  53.203  < 2e-16 ***
age:sexM            1.876e-02  4.631e-03   4.051 5.12e-05 ***
sexM:age2          -1.721e-04  5.927e-05  -2.903  0.00369 **
```

Census Income Data

Bring in some interactions with race and education:

```
> summary( reg2 <- lm(log.WR ~ age*sex + age2*sex +
                                marital + (hs+assoc+coll+grad)*age +
                                race*age,  data=YX) )
> anova(reg1, reg2)

Analysis of Variance Table

Model 1: log.WR ~ age * sex + age2 * sex +
              (age + age2 + sex + race + marital + hs + assoc + coll + grad)
Model 2: log.WR ~ age * sex + age2 * sex + marital +
                (hs + assoc + coll + grad) * age + race * age

  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1  25385 7187.4
2  25377 7163.7  8    23.656 10.475 8.891e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The new variables are significant.

Census Income Data

Three way interaction!

```
> summary(reg3 <- lm(log.WR ~ race*age*sex + age2*sex + marital +
                                (hs+assoc+coll+grad)*age, data=YX) )
> anova(reg2, reg3)

Analysis of Variance Table

Model 1: log.WR ~ age * sex + age2 * sex + marital +
         (hs + assoc + coll + grad) * age + race * age
Model 2: log.WR ~ age * sex + age2 * sex + marital +
         (hs + assoc + coll + grad) * age + race * age * sex

  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1  25377 7163.7
2  25369 7145.8  8    17.957 7.9688 8.804e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These additions also appear usefull.

Census Income Data

Do we get away without race main effects? (-race)

```
> summary(reg4 <- lm(log.WR ~ race*age*sex + age2*sex + marital +
                                  (hs+assoc+coll+grad)*age - race, data=YX) )
> anova(reg3, reg4)

Analysis of Variance Table

Model 1: log.WR ~ race * age * sex + age2 * sex + marital +
                   (hs + assoc +  coll + grad) * age
Model 2: log.WR ~ race * age * sex + age2 * sex + marital +
                  (hs +  assoc + coll + grad) * age - race
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1  25369 7145.8
2  25373 7146.0 -4  -0.20565 0.1825 0.9476
```

Reduced model is best.

Census Income Data

Testing is a difficult and imperfect way to compare models

- You need to have a good prior sense of what model you want.
- H_0 vs H_1 is not designed to answer these types of questions.
- A p-value doesn't really measure how much better a model is.

Multiple Testing

A big problem with using tests (t or F) for comparing models is the false discovery rate associated with multiple testing:

- If you do 20 tests of true H_0, with $\alpha = .05$, you expect to see 1 positive (i.e. you expect to reject at true null).

Suppose that 10% of your regression coefficients are actually influential, and that you manage to find all of them significant.

Since you reject H_0 for 5% of the useless 90% variables, $4.5/14.5 \approx 1/3$ of your significant b_j's will be false positives!

In some online marketing data, $<1\%$ of variables are influential.

BIC for Model Selection

Instead of testing, there are Information Criteria metrics which attempt to quantify how well our model would have predicted the data (regardless of what you've estimated for the β_j's).

The best of these is the BIC: Bayes Information Criterion, which is based on a "Bayesian" philosophy of statistics.

$$BIC = n \log(s^2) + p \log(n)$$

You want to choose the model that leads to minimum BIC.

BIC for Model Selection

Another popular metric is the Akaike Information Criterion:

$$AIC = n \log(s^2) + 2p$$

A general form for these criterion is $n \log(s^2) + kp$,
where $k = 2$ for AIC and $k = log(n)$ for BIC.

In R, we can use the `extractAIC()` function to get the BIC.

`extractAIC(reg)` $\Rightarrow$ `AIC`
`extractAIC(reg, k=log(n))` $\Rightarrow$ `BIC`

AIC prefers more complicated models than BIC,
and it is not as easily interpretable.

BIC for Model Selection

Consider our wage regressions; we can compare the full model to the first (too small) and second (just right) reduced models.

AIC

```
> extractAIC(reg1)
    18.0   -32036.23
> extractAIC(reg2)
    26.0   -32103.98
> extractAIC(reg3)
    34.0   -32151.74
> extractAIC(reg3)
    30.0   -32159.00
```

BIC

```
> extractAIC(reg1,k=log(n))
    18.0   -31889.67
> extractAIC(reg2,k=log(n))
    26.0   -31892.27
> extractAIC(reg3,k=log(n))
    34.0    -31874.89
> extractAIC(reg4,k=log(n))
    30.0    -31914.73
```

BIC and AIC both agree with our F-testing selection (`reg4`).

BIC for Model Selection

One (very!) nice thing about the BIC is that you can interpret it in terms of model probabilities.

Given a list of possible models $\{M_1, M_2, \ldots, M_R\}$, the probability that model i is correct is

$$P(M_i) \approx \frac{e^{-\frac{1}{2}BIC(M_i)}}{\sum_{r=1}^{R} e^{-\frac{1}{2}BIC(M_r)}} = \frac{e^{-\frac{1}{2}[BIC(M_i)-BIC_{min}]}}{\sum_{r=1}^{R} e^{-\frac{1}{2}[BIC(M_r)-BIC_{min}]}}$$

(Subtract $BIC_{min} = \min\{BIC(M_1) \ldots BIC(M_R)\}$ for numerical stability.)

BIC for Model Selection

BIC indicates that we are practicaly 100% sure `reg4` is best.

```
> BIC <- c(reg1=extractAIC(reg1, k=log(n))[2],
           reg2=extractAIC(reg2, k=log(n))[2],
           reg3=extractAIC(reg3, k=log(n))[2],
           reg4=extractAIC(reg4, k=log(n))[2])

> print(eBIC <- exp(-.5*(BIC-min(BIC))))

        reg1          reg2          reg3          reg4
3.615035e-06 1.330649e-05 2.233478e-09 1.000000e+00

> round(probs <- eBIC/sum(eBIC), 2)

reg1 reg2 reg3 reg4
   0    0    0    1
```

BIC for Model Selection

Thus BIC is an alternative to testing for comparing models.

- It is easy to calculate.
- You are able to evaluate model probabilities.
- There are no "multiple testing" type worries.
- It generally leads to more simple models than F-tests.

I prefer it to F-tests, but many others love hypothesis testing so you need to know both. Again, there is no silver bullet!

As with testing, you need to narrow down your options before comparing models. What if there are too many possibilities?

Forward Stepwise Regression

One approach is to build your regression model step-by-step, adding one variable at a time:

- Run $Y \sim X_i$ for each covariate, then choose the one leading to the smallest BIC to include in your model.
- Given you chose covariate $X^\star$, now run $Y \sim X^\star + X_i$ for each i and again select the model with smallest BIC.
- Repeat this process until none of the expanded models lead to smaller BIC than the previous model.

This is called "forward stepwise regression".

There is a backwards version (put in all X's, then eliminate them step-by-step), but it is not as practically useful.

Forward Stepwise Regression

R has a `step()` function to do forward stepwise regression.

- This is nice, since doing the steps is time intensive.

Easiest way to use this function: run null and full regressions.

```
null <- lm(Y ~ 1, data=YX)
full <- lm(Y ~ . + .^2, data=YX)
```

" $\sim$. " says "Give me everything".

This is one of the reasons that building `YX` is a good idea.

Forward Stepwise Regression

Given null (most simple) and full (most complicated) models,

```
fwd <- step(null, scope=formula(full),
            direction="forward", k=log(n))
```

- `scope` is the largest possible model that we will consider.
- `scope=formula(full)` makes this our "full" model
- `k=log(n)` uses the BIC metric.

Community Crime Data

We have violent crime rates for 1994 communities across the US, as well as 25 descriptive variables. For example,

- householdsize: mean people per household
- PctUnemployed: % of people 16 and over unemployed
- PctFam2Par: % of families (with kids) that have two parents
- PctRecentImmig: % immigrated within 3 years
- PctHousOccup: % of housing occupied
- RentMedian: rental housing - median rent
- PctUsePubTrans: % of people who use public transit

Community Crime Data

We'll build a regression model for log crime rate (`logCR`).

Make `XY` from only your response and covariates.

```
> XY <- crime[,-(1:2)]
> XY$logCR <- log(crime$ViolentCR)
```

Community Crime Data

First step through possible main effects.

```
> null <- lm(logCR ~ 1, data=crime)
> full <- lm(logCR ~ ., data=crime)
> fwdBIC <- step(null, scope=formula(full),
                 direction="forward", k=log(1000))
```

⋮

```
Step:  AIC=-756.45
logCR ~ PctFam2Par + PctForeignBorn + TotalPctDiv +
        PctUrban + PersPerFam + PctHousOccup
```

Community Crime Data

Then tell stepwise regression to look for interactions.

```
> fwdBICinter <- step(fwdBIC, scope= ~ . + .^2,
                          direction="forward", k=log(1000))
```

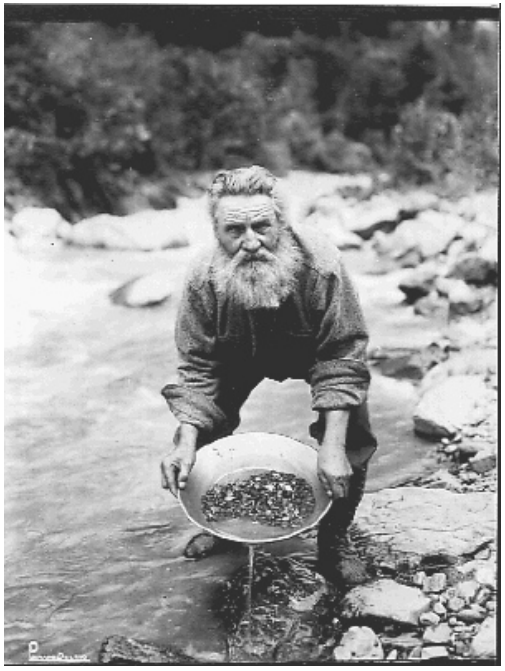
⋮

```
Step:  AIC=-757.79
logCR ~ PctFam2Par + PctForeignBorn + TotalPctDiv + PctUrban +
    PersPerFam + PctHousOccup + PctForeignBorn:PctHousOccup
```

Add one interaction: the algorithm stopped here because none of the 1-step expanded models led to a lower BIC.

Data Mining

"Data Mining" refers to tools that seek to uncover a small number of influential variables within large, high-dimensional datasets.



Forward stepwise regression is a data mining technique.
Unfortunately, it is more like the guy on the left.

Data Mining

As mentioned before, this is a very hard problem:

- Since very few variables are influential, testing is useless.
- You cannot consider all transformations and interactions.
- It is easy to overfit, which leads to bad predictions.

Stepwise regression will only help you out in relatively small datasets, when you already have a sense of where to look.

For industrial mining, you need tools on a totally different scale.

Putting it all together...

Regardless: Remember your discipline based knowledge and don't get lost in fancy regression techniques.

A Strategy for Building Regression Models:

- Select the Universe of Variables and put them all in.
  - If you have too many, use forward stepwise regression.
- Plot diagnostics and take remedies (transformations, etc).
- Reduce set of X variables via BIC or F-testing.
- Repeat the residual diagnostics and remedies.
- Evaluate your model predictions.