

# STAT 516 Lec 04

## Multiple linear regression (part 2/2)

Karl Gregory

2024-02-20

# Rental rates of commercial properties example

As in part 1/2, consider these data from Kutner et al. (2005).

```
link <- url("https://people.stat.sc.edu/gregorkb/data/KNLIcp.txt")
commprop <- read.table(link,col.names=c("rent","age","optx","vac","sqft"))
commprop$sqft <- commprop$sqft/10000 # rescale sqft
head(commprop)
```

	rent	age	optx	vac	sqft
1	13.5	1	5.02	0.14	12.3000
2	12.0	14	8.19	0.27	10.4079
3	10.5	16	3.00	0.00	3.9998
4	15.0	4	10.70	0.05	5.7112
5	14.0	11	8.97	0.07	6.0000
6	10.5	15	9.45	0.24	10.1385

```
n <- nrow(commprop)
p <- ncol(commprop) - 1
```

There are  $n = 81$  rows and  $p = 4$  predictors.

# Setup

Consider data  $(Y_1, \mathbf{x}_1), \dots, (Y_n, \mathbf{x}_n)$ , with each  $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ .

The multiple linear regression model is

$$Y_i = \beta_0 + x_{i1}\beta_1 + \dots + x_{ip}\beta_p + \varepsilon_i, \quad i = 1, \dots, n,$$

where

- ▶  $\mathbf{x}_1, \dots, \mathbf{x}_n$  are vectors in  $\mathbb{R}^p$  of covariate or predictor values.
- ▶  $Y_1, \dots, Y_n$  are the response values
- ▶  $\beta_0, \beta_1, \dots, \beta_p$  are the regression coefficients.
- ▶  $\varepsilon_1, \dots, \varepsilon_n$  are iid  $\text{Normal}(0, \sigma^2)$  error terms.
- ▶  $\sigma^2$  is the error term variance.

# Goals in multiple linear regression

In part 1/2, we addressed these goals:

1. Estimate the regression coefficients  $\beta_0$  and  $\beta_1, \dots, \beta_p$ .
2. Estimate the error term variance  $\sigma^2$ .
3. Perform inference on  $\beta_1, \dots, \beta_p$ .
4. Build a CI for  $\beta_0 + \beta_1 x_{\text{new},1} + \dots + \beta_p x_{\text{new},p}$  at any  $\mathbf{x}_{\text{new}}$ .
5. Build a prediction interval for  $Y$  at any  $\mathbf{x}_{\text{new}}$ .
6. Decompose the variation in  $Y$  into (sums of) sums of squares.
7. Check whether the model assumptions are satisfied.
8. Identify outliers and understand their effects.

In part 2/2 we focus on these:

8. Test for significance of a subset of covariates
9. Understand how correlations among the covariates affect inferences
10. Do variable selection

## Review of F distributions

For  $W_1 \sim \chi_{\nu_1}^2(\phi)$ ,  $W_2 \sim \chi_{\nu_2}^2$  independent,  $R = \frac{W_1/\nu_1}{W_2/\nu_2} \sim F_{\nu_1, \nu_2}(\phi)$ .

$F_{\nu_1, \nu_2}(\phi)$  denotes the  $F$  distribution with

- ▶ numerator degrees of freedom  $\nu_1$
- ▶ denominator degrees of freedom  $\nu_2$
- ▶ noncentrality parameter  $\phi \geq 0$

If  $\phi > 0$  the distribution is a non-central F distribution.

When  $\phi = 0$  we just write  $F_{\nu_1, \nu_2}$  to denote the “central” F distribution.

We will encounter ratios of sums of squares which have  $F$  distributions.

## Plot of some F distribution pdfs

```
nu1 <- c(1,2,3,5,5,5,50,50)
nu2 <- c(3,3,3,10,10,10,50,50)
phi <- c(0,0,0,0,4,8,0,4)
f <- seq(.01,4,length=200)
dfmat <- matrix(0,length(f),200)
for(j in 1:length(nu1)){

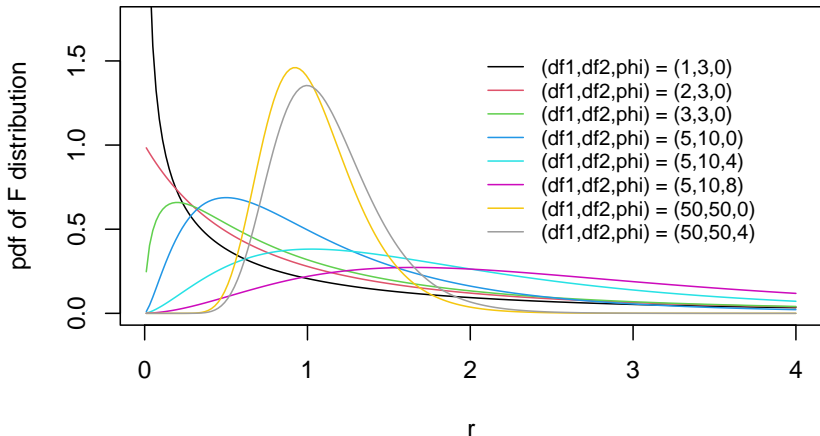
  dfmat[j,] <- df(f,df1 = nu1[j],df2=nu2[j],ncp=phi[j])

}
lab <- paste("(df1,df2,phi) = (",
  apply(cbind(nu1,nu2,phi),1,paste,collapse = ", "),
  ")",sep="")
```

```

plot(NA,xlim = range(f),ylim = c(0,1.2*max(dfmat[-1,])),
     xlab = "r",
     ylab = "pdf of F distribution")
for(j in 1:length(nu1)) lines(dfmat[j,]~f, col = j)
legend(x = .5*max(f),y = 1.1*max(dfmat[-1,]),legend = lab,
       col = 1:length(nu1), lty = 1,bty = "n", cex = .8)

```



## The overall F-test

We may wish to test whether *any* covariates are important, that is

$$H_0: \beta_j = 0 \text{ for all } j = 1, \dots, p.$$

The overall F-test of significance is carried out as follows:

1. Fit the model with all the covariates and obtain the value

$$F_{\text{stat}} = \frac{\text{MS}_{\text{Reg}}}{\text{MS}_{\text{Error}}} \left( = \frac{\text{SS}_{\text{Reg}}/p}{\text{SS}_{\text{Error}}/(n - (p + 1))} \right)$$

2. Reject  $H_0$  at  $\alpha$  if  $F_{\text{stat}} > F_{p, n-(p+1), \alpha}$ .
3. Obtain p-value is  $P(F > F_{\text{stat}})$ , where  $F \sim F_{p, n-(p+1)}$ .

This test statistic and p-value are reported by `summary()` on `lm()`.



**Exercise:** Show that the test statistic of the overall F test can be written

$$F_{\text{stat}} = \frac{\text{MS}_{\text{Reg}}}{\text{MS}_{\text{Error}}} = \frac{(n - (p + 1))}{p} \frac{R^2}{1 - R^2},$$

where  $R^2$  is the coefficient of determination.

**Exercise:** Suppose you fit a regression model with 3 predictors on a data set with 81 observations, and you obtain  $\hat{\sigma} = 1.132$  and  $R^2 = 0.583$ . Use this information to fill in the entire ANOVA table:

Source	Df	SS	MS	F value	p-value
Regression	$p$	$SS_{\text{Reg}}$	$MS_{\text{Reg}}$	$F_{\text{stat}}$	$P(F > F_{\text{stat}})$
Error	$n - (p + 1)$	$SS_{\text{Error}}$	$MS_{\text{Error}}$		
Total	$n - 1$	$SS_{\text{Tot}}$			

# Testing for significance of a subset of covariates

Consider testing the significance of a subset  $D \subset \{1, \dots, p\}$  of covariates:

$$H_0: \beta_j = 0 \text{ for all } j \in D.$$

Use the full-reduced model F-test:

1. Let  $s$  be the number of covariates in  $D$  and compute

$$F_{\text{stat}} = \frac{(\text{SS}_{\text{Error}}(\text{Reduced}) - \text{SS}_{\text{Error}}(\text{Full}))/s}{\text{SS}_{\text{Error}}(\text{Full})/(n - (p + 1))},$$

- ▶ “Full” is the model with all  $p$  covariates.
  - ▶ “Reduced” is the model after dropping the covariates in  $D$ .
2. Reject  $H_0$  at  $\alpha$  if  $F_{\text{stat}} > F_{s, n-(p+1), \alpha}$ .
  3. Obtain p-value as  $P(F > F_{\text{stat}})$ , where  $F \sim F_{s, n-(p+1)}$ .

## Rental rates of commercial properties example (cont)

Check whether `vac` and `optx` contribute significantly to the rent.

That is test  $H_0: \beta_{\text{vac}} = 0$  and  $\beta_{\text{optx}} = 0$ .

```
lm_red <- lm(rent ~ age + sqft, data = commprop)
lm_full <- lm(rent ~ age + optx + vac + sqft, data = commprop)
SSE_red <- sum(lm_red$residuals^2)
SSE_full <- sum(lm_full$residuals^2)
s <- 2 # significance of two covariates being tested
Fstat <- (SSE_red - SSE_full)/s / ( SSE_full / (n - (p + 1)))
alpha <- 0.05
F_crit <- qf(1 - alpha,s,n-(p+1))
pval <- 1 - pf(Fstat, 2, n - (p+1))
```

We obtain  $F_{\text{stat}} = 10.939$  and  $F_{s,n-(p+1),0.05} = 3.117$ , and the p-value is 0.

## Full-reduced model F test for a single covariate

If we test  $H_0: \beta_j = 0$  for a single covariate using the full-reduced model F test, the test statistic  $F_{\text{stat}}$  will be equal to the square of the test statistic  $T_{\text{stat}}$  for testing  $H_0: \beta_j = 0$  in the full model.

```
lm_red <- lm(rent ~ age + optx + sqft, data = commprop)
lm_full <- lm(rent ~ age + optx + vac + sqft, data = commprop)
SSE_red <- sum(lm_red$residuals^2)
SSE_full <- sum(lm_full$residuals^2)
s <- 1 # significance of a single covariate being tested
Fstat <- (SSE_red - SSE_full)/s / ( SSE_full / (n - (p + 1)))
sqrt(Fstat) # absolute value of the t-statistic from the full model
```

```
[1] 0.5698714
```

## Effect of correlations among the covariates

From before  $\text{Var } \hat{\beta}_j = \sigma^2 \Omega_{jj} / n$ . An alternate expression gives

$$\text{Var } \hat{\beta}_j = \frac{1}{1 - R_j^2} \frac{\sigma^2}{\sum_{i=1}^n (x_{ji} - \bar{x}_j)^2},$$

where  $R_j^2$  is the  $R^2$  from regressing  $x_j$  on the other covariates.

So multicollinearity of  $x_j$  with the other covariates “inflates”  $\text{Var } \hat{\beta}_j$ :

- ▶ Makes confidence intervals for  $\beta_j$  wider.
- ▶ Makes tests of  $H_0: \beta_j = 0$  vs  $H_1: \beta_j \neq 0$  less powerful.

Call  $\frac{1}{1 - R_j^2}$  the variance inflation factor (VIF) for  $x_j$ ,  $j = 1, \dots, p$ .

## VIFs in commercial properties example

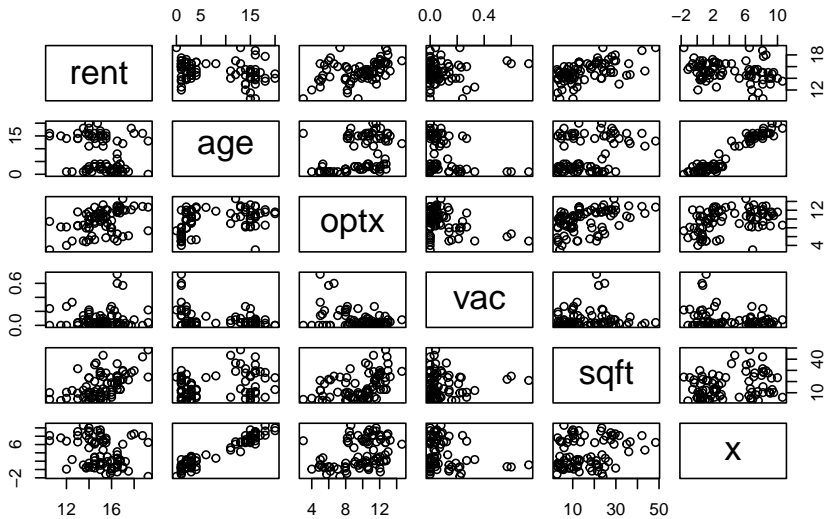
Add to the data set a spurious predictor highly correlated with age.

Check the effect of this on our inferences for  $\beta_{\text{age}}$ .

```
# make new x correlated with age
x <- .5*commprop$age + rnorm(n)
commpropx <- cbind(commprop,x)
round(cor(commpropx),4)
```

	rent	age	optx	vac	sqft	x
rent	1.0000	-0.2503	0.4138	0.0665	0.5353	-0.2766
age	-0.2503	1.0000	0.3888	-0.2527	0.2886	0.9615
optx	0.4138	0.3888	1.0000	-0.3798	0.4407	0.3874
vac	0.0665	-0.2527	-0.3798	1.0000	0.0806	-0.2040
sqft	0.5353	0.2886	0.4407	0.0806	1.0000	0.2585
x	-0.2766	0.9615	0.3874	-0.2040	0.2585	1.0000

```
plot(commpropx)
```





```
lm_out <- lm(rent ~ age + optx + vac + sqft, data = commpropx)
confint(lm_out)
```

	2.5 %	97.5 %
(Intercept)	11.04948640	13.35168536
age	-0.18454113	-0.09952615
optx	0.15619789	0.40783517
vac	-1.54523184	2.78391885
sqft	0.05166283	0.10682321

```
lmx_out <- lm(rent ~ age + optx + vac + sqft + x, data = commpropx)
confint(lmx_out)
```

	2.5 %	97.5 %
(Intercept)	10.88533335	13.18802394
age	-0.16512237	0.12179843
optx	0.17483677	0.42687873
vac	-1.15248447	3.22450020
sqft	0.04761752	0.10282506
x	-0.51038291	0.03332522

The width of the CI for  $\beta_{\text{age}}$  was 0.085.

With the new covariate the width of the CI for  $\beta_{\text{age}}$  becomes 0.287.

So including x in the model makes our estimation of  $\beta_{\text{age}}$  less accurate.

```
summary(lm_out)
```

Call:

```
lm(formula = rent ~ age + optx + vac + sqft, data = commpropx)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.1872	-0.5911	-0.0910	0.5579	2.9441

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	12.20059	0.57796	21.110	< 2e-16	***
age	-0.14203	0.02134	-6.655	3.89e-09	***
optx	0.28202	0.06317	4.464	2.75e-05	***
vac	0.61934	1.08681	0.570	0.57	
sqft	0.07924	0.01385	5.722	1.98e-07	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.137 on 76 degrees of freedom

Multiple R-squared: 0.5847, Adjusted R-squared: 0.5629

F-statistic: 26.76 on 4 and 76 DF, p-value: 7.272e-14

The p-value for age is very small.

```
summary(lmx_out)
```

Call:

```
lm(formula = rent ~ age + optx + vac + sqft + x, data = commpropx)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-3.1139	-0.6925	0.0010	0.5243	2.5780

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	12.03668	0.57795	20.826	< 2e-16 ***
age	-0.02166	0.07201	-0.301	0.7644
optx	0.30086	0.06326	4.756	9.35e-06 ***
vac	1.03601	1.09858	0.943	0.3487
sqft	0.07522	0.01386	5.429	6.72e-07 ***
x	-0.23853	0.13647	-1.748	0.0846 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.122 on 75 degrees of freedom

Multiple R-squared: 0.601, Adjusted R-squared: 0.5744

F-statistic: 22.59 on 5 and 75 DF, p-value: 9.432e-14

The p-value for age is not nearly as small when x is included!

## Getting VIFs with `vif()` from the `car` package

We can use the R package `car` from Fox and Weisberg (2019).

First time must install the package with `install.package("car")`.

```
library(car)
vif(lm_out)
```

```
      age      optx      vac      sqft
1.240348 1.648225 1.323552 1.412722
```

```
vif(lmx_out)
```

```
      age      optx      vac      sqft      x
14.503654 1.697512 1.388950 1.452779 14.064941
```

Note the change in VIF for `age` due to including `x` in the model!

## Variable selection

Sometimes the number of potentially important predictors is quite large.

Large  $p$  tends to increase the VIFs, leading to low power.

So we may wish to discard some predictors. We briefly discuss:

- ▶ Forward and backward stepwise selection with AIC
- ▶ Best subset selection with Mallows's  $C(p)$
- ▶ LASSO selection

And most importantly:

- ▶ The dangers of naïve post-selection inference!!

# 赤池弘次



Introduced Akaike's Information Criterion (AIC).

# Akaike's Information Criterion (AIC) for comparing models

For a given model, i.e. set of covariates, AIC is defined as

$$\text{AIC} = 2(p + 1) - 2 \underbrace{\ell(\hat{\sigma}^2, \hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)}_{\text{log-likelihood}}.$$

The log-likelihood is the log of the joint pdf of the data (STAT 512).

AIC can be used to compare several models for the same data.

The “best” model is the one which minimizes AIC.

## The extractAIC() function

The extractAIC function in R returns a modified version of AIC:

$$\text{AIC}^* = 2(p + 1) + n \log(\text{SS}_{\text{Error}} / n)$$

```
lm_out <- lm(rent ~ age + optx + vac + sqft, data = commprop)
extractAIC(lm_out) # gives value p + 1 as well as AIC value
```

```
[1] 5.00000 25.62235
```

```
# compute it "manually"
```

```
p <- 4
```

```
2*(p+1) + n * log(sum(lm_out$residuals^2)/n)
```

```
[1] 25.62235
```



## Comparing models using AIC

Continuing with the rental property data, compare the model with all the covariates to the model with only the covariate age.

```
lm_all <- lm(rent ~ age + optx + vac + sqft, data = commprop)
extractAIC(lm_all)
```

```
[1] 5.00000 25.62235
```

```
lm_age <- lm(rent ~ age, data = commprop)
extractAIC(lm_age)
```

```
[1] 2.00000 85.57119
```

The model with all the covariates has AIC value 25.622, while the model with only the covariate age has AIC value 85.571.

So according to AIC the model with all the covariates is superior.

# Stepwise selection based on AIC

Stepwise selection:

- ▶ Backward: Begin with all the predictors and remove one at a time.
- ▶ Forward: Begin with no predictors and add one at a time.

In each step remove/add predictor to get largest decrease in AIC.

If a decrease in AIC is not possible, stop.

# Stepwise selection with commercial property data

```
# BACKWARD: fit model with all covariates
lm0 <- lm(rent ~ age + optx + vac + sqft, data = commprop)
extractAIC(lm0)[2]
```

```
[1] 25.62235
```

```
# find which variable to remove
lm1_1 <- lm(rent ~ optx + vac + sqft, data = commprop)
lm1_2 <- lm(rent ~ age + vac + sqft, data = commprop)
lm1_3 <- lm(rent ~ age + optx + sqft, data = commprop)
lm1_4 <- lm(rent ~ age + optx + vac, data = commprop)

c(extractAIC(lm1_1)[2], extractAIC(lm1_2)[2], extractAIC(lm1_3)[2], extractAIC(lm1_4)[2])
```

```
[1] 60.81404 42.48560 23.96773 52.64345
```

```
# removing vac decreases AIC the most. Now find next variable to remove:
```

```
lm2_1 <- lm(rent ~ optx + sqft, data = commprop)
lm2_2 <- lm(rent ~ age + sqft, data = commprop)
lm2_3 <- lm(rent ~ age + optx, data = commprop)

c(extractAIC(lm2_1)[2], extractAIC(lm2_2)[2], extractAIC(lm2_3)[2])
```

```
[1] 60.88082 42.11426 55.33500
```

```
# could not decrease the AIC, so stick with the model: rent ~ age + optx + sqft
```

```
# FORWARD: fit model with no covariates (intercept only)
lm0 <- lm(rent ~ 1, data = commprop)
extractAIC(lm0)[2]
```

```
[1] 88.81111
```

```
# find which variable to add
```

```
lm1_1 <- lm(rent ~ age, data = commprop)
lm1_2 <- lm(rent ~ optx, data = commprop)
lm1_3 <- lm(rent ~ vac, data = commprop)
lm1_4 <- lm(rent ~ sqft, data = commprop)
```

```
c(extractAIC(lm1_1)[2], extractAIC(lm1_2)[2], extractAIC(lm1_3)[2], extractAIC(lm1_4)[2])
```

```
[1] 85.57119 75.59929 90.45183 63.46705
```

```
# adding sqft decreases AIC the most. Now find next variable to add:
```

```
lm2_1 <- lm(rent ~ sqft + age, data = commprop)
lm2_2 <- lm(rent ~ sqft + optx, data = commprop)
lm2_3 <- lm(rent ~ sqft + vac, data = commprop)
```

```
c(extractAIC(lm2_1)[2], extractAIC(lm2_2)[2], extractAIC(lm2_3)[2])
```

```
[1] 42.11426 60.88082 65.40458
```

```
# adding age decreases AIC the most.
```

```
lm3_1 <- lm(rent ~ sqft + age + optx, data = commprop)
lm3_2 <- lm(rent ~ sqft + age + vac, data = commprop)
```

```
c(extractAIC(lm3_1)[2], extractAIC(lm3_2)[2])
```

```
[1] 23.96773 42.48560
```

```
# adding optx decreased AIC by the most. now add last variable
```

```
lm4_1 <- lm(rent ~ sqft + age + optx + vac, data = commprop)
extractAIC(lm4_1)[2]
```

```
[1] 25.62235
```

```
# this increased AIC, so stick with the model: rent ~ sqft + age + optx
```

# The step() function

Use step() function to do forward and backward stepwise selection.

```
# first fit two models: intercept-only and full
lm_intercept <- lm(rent ~ 1, data = commprop)
lm_all <- lm(rent ~ vac + age + optx + sqft, data = commprop)

# backward selection
step_back <- step(lm_all,
                  direction = "backward",
                  scope = formula(lm_all),
                  trace = 0) # suppress printed output

# forward selection
step_forw <- step(lm_intercept,
                  direction = "forward",
                  scope = formula(lm_all),
                  trace = 0) # suppress printed output
```

Set trace = 1 to see output for each step.

Forward and backward stepwise chose the same model for these data:

```
step_back
```

```
Call:
```

```
lm(formula = rent ~ age + optx + sqft, data = commprop)
```

```
Coefficients:
```

(Intercept)	age	optx	sqft
12.37058	-0.14416	0.26717	0.08178

```
step_forw
```

```
Call:
```

```
lm(formula = rent ~ sqft + age + optx, data = commprop)
```

```
Coefficients:
```

(Intercept)	sqft	age	optx
12.37058	0.08178	-0.14416	0.26717

But this need not be the case.

## Best subset selection with Mallows's $C_p$

Given  $q$  available covariates, there are  $2^q$  possible subset models (why?).

Mallows's  $C_p$  can be used to compare subset models: Let

$$C_p = (n - (p + 1)) \left[ \frac{\text{MS}_{\text{Error}}(\text{subset})}{\text{MS}_{\text{Error}}(\text{all})} - 1 \right] + (p + 1),$$

where

- ▶  $p$  is the number of predictors *in the subset model*.
- ▶  $\text{MS}_{\text{Error}}(\text{subset})$  is the  $\text{MS}_{\text{Error}}$  of the subset model.
- ▶  $\text{MS}_{\text{Error}}(\text{all})$  is the  $\text{MS}_{\text{Error}}$  of the model with all the covariates.

If the subset model is adequate,  $\text{MS}_{\text{Error}}(\text{subset})$  estimates the same target as  $\text{MS}_{\text{Error}}(\text{all})$ , so the first term should be small and  $C_p \approx p + 1$ .

Can look at  $C_p$  values for all subset models of each size  $p = 0, 1, 2, \dots, q$

Want smallest model such that  $C_p \approx p + 1$ .

## Mallow's $C_p$ on the rental properties data

Compute Mallow's  $C_p$  for a single subset model:

```
lm_all <- lm(rent ~ vac + age + optx + sqft, data = commprop)
lm_sub <- lm(rent ~ age + sqft, data = commprop)
MSE_sub <- sum(lm_sub$residuals^2) / (n - 3)
MSE_all <- sum(lm_all$residuals^2) / (n - 5)
Csub <- (MSE_sub / MSE_all - 1)*(n - 3) + 3
Csub
```

```
[1] 22.87781
```

This value is too large; the subset is not a good one.



# The regsubsets() function from the R package leaps

```
library(leaps) # first time run install.packages("leaps")
regsubsets_out <- regsubsets(rent ~ vac + age + optx + sqft, data = commprop)
summary(regsubsets_out)
```

Subset selection object

Call: regsubsets.formula(rent ~ vac + age + optx + sqft, data = commprop)

4 Variables (and intercept)

Forced in Forced out

vac FALSE FALSE

age FALSE FALSE

optx FALSE FALSE

sqft FALSE FALSE

1 subsets of each size up to 4

Selection Algorithm: exhaustive

vac age optx sqft

1 ( 1 ) " " " " " " "\*"

2 ( 1 ) " " "\*" " " " "\*"

3 ( 1 ) " " "\*" "\*" " "\*"

4 ( 1 ) "\*" "\*" "\*" "\*" "

```
summary(regsubsets_out)$cp
```

```
[1] 53.585208 22.877809 3.324753 5.000000
```

## LASSO selection

For  $\lambda > 0$ , define the LASSO objective function as

$$Q_\lambda(b_0, b_1, \dots, b_p) = \sum_{i=1}^n (Y_i - (b_0 + b_1 x_{i1} + \dots + b_p x_{ip}))^2 + \lambda \sum_{j=1}^p |b_j|$$

Denote by  $\tilde{\beta}_0, \tilde{\beta}_1, \dots, \tilde{\beta}_p$  the values of  $b_0, b_1, \dots, b_p$  which minimize  $Q_\lambda$ .

Then  $\tilde{\beta}_0, \tilde{\beta}_1, \dots, \tilde{\beta}_p$  are the LASSO estimators of  $\beta_0, \beta_1, \dots, \beta_p$ .

The penalty  $\lambda \sum_{j=1}^p |b_j|$  can cause  $\tilde{\beta}_j = 0$  for some  $j$ .

So LASSO performs variable selection and estimation simultaneously.

Nice, but trouble is: Hard to build CIs based on  $\tilde{\beta}_0, \tilde{\beta}_1, \dots, \tilde{\beta}_p$ .

# LASSO on the commercial properties data

Use `cv.ncvreg()` function from R package `ncvreg`.

Runs crossvalidation to choose the best value of  $\lambda$ .

```
library(ncvreg) # first time run install.packages("ncvreg")

# prepare response vector and design matrix
y <- commprop$rent
X <- cbind(commprop$age, commprop$optx, commprop$vac, commprop$sqft)

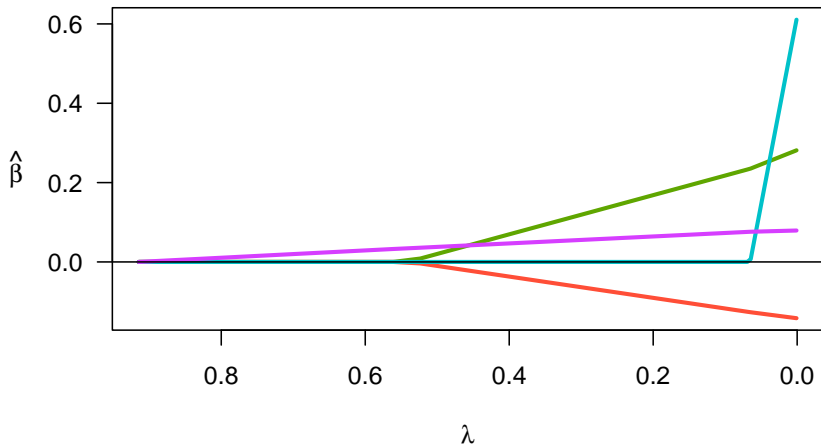
# crossvalidation to choose lambda
lasso <- cv.ncvreg(X, y, penalty = "lasso")
lasso$lambda[lasso$min] # the chosen value of lambda
```

```
[1] 0.0009147293
```

```
lasso$fit$beta[,lasso$min] # estimates under the "best" lambda
```

(Intercept)	V1	V2	V3	V4
12.2067767	-0.1418103	0.2813426	0.6106835	0.0791984

```
plot(lasso$fit)
```



```
lasso$lambda[25] # a larger value of lambda
```

```
[1] 0.1714036
```

```
lasso$fit$beta[,25] # the estimates under the larger value of lambda
```

```
(Intercept)          V1          V2          V3          V4  
13.07091918 -0.09820832  0.18250805  0.00000000  0.06674422
```

# The dangers of post-selection inference

It is dangerous to:

1. Ask the data what hypotheses to test (what model to build).
2. Use afterwards the same data to perform inference (get p values).

## Illustration:

Add 50 spurious predictors to the commercial properties data.

See how many we find to be significant.

```
X <- matrix(rnorm(n*50),n,50)
colnames(X) <- paste("x",1:50,sep="")
commpropX <- cbind(commprop,X)

lmX_out <- lm(rent ~ ., data = commpropX)
```

Call:

lm(formula = rent ~ ., data = commpropX)

Residuals:

	Min	1Q	Median	3Q	Max
	-1.59062	-0.28456	0.05265	0.37467	1.32721

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	12.240240	0.804196	15.220	1.83e-14	***
age	-0.159799	0.032199	-4.963	3.71e-05	***
optx	0.306053	0.088151	3.472	0.001821	**
vac	-1.019504	1.626735	-0.627	0.536309	
sqft	0.075606	0.019462	3.885	0.000631	***
x1	0.081065	0.202076	0.401	0.691580	
x2	-0.013859	0.215785	-0.064	0.949282	
x3	0.353769	0.156937	2.254	0.032835	*
x4	-0.246326	0.214493	-1.148	0.261255	
x5	0.094743	0.217846	0.435	0.667219	
x6	-0.255471	0.179398	-1.424	0.166325	
x7	0.044972	0.249455	0.180	0.858330	
x8	-0.093089	0.173642	-0.536	0.596451	
x9	-0.173610	0.200781	-0.865	0.395125	
x10	-0.456824	0.171506	-2.664	0.013095	*
x11	-0.198532	0.182312	-1.089	0.286157	
x12	0.167599	0.225407	0.744	0.463821	
x13	-0.042958	0.158114	-0.272	0.788006	
x14	-0.149825	0.157784	-0.950	0.351082	
x15	0.044798	0.206143	0.217	0.829660	
x16	-0.085366	0.179704	-0.475	0.638728	
x17	0.409642	0.198006	2.069	0.048639	*
x18	-0.014995	0.168287	-0.089	0.929681	
x19	0.310235	0.233058	1.331	0.194696	
x20	-0.095293	0.177272	-0.538	0.595460	
x21	-0.241792	0.201412	-1.200	0.240774	
x22	-0.187829	0.178686	-1.051	0.302854	
x23	-0.057653	0.150114	-0.384	0.704054	
x24	0.015410	0.160248	0.096	0.924129	
x25	0.045967	0.212807	0.216	0.830669	
x26	-0.163131	0.206006	-0.792	0.435601	
x27	0.298277	0.166657	1.790	0.085146	.

We reject  $H_0: \beta_j = 0$  at  $\alpha = 0.05$  for 3 of the spurious predictors.

So the Type I error rate was  $3/50 = 0.06$ .

Now do backwards stepwise selection to throw some variables away.

Then see how many of the spurious predictors we find “significant”.



```
stepX_out <- step(lmX_out, data = commpropX, trace = 0)
summary(stepX_out)
```

Call:

```
lm(formula = rent ~ age + optx + vac + sqft + x3 + x4 + x6 +
    x9 + x10 + x11 + x14 + x17 + x19 + x21 + x22 + x27 + x30 +
    x33 + x35 + x38 + x39 + x40 + x43 + x45, data = commpropX)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-1.56212 -0.38254  0.00396  0.45158  1.45098
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	12.30833	0.45134	27.271	< 2e-16 ***
age	-0.14311	0.01677	-8.535	1.03e-11 ***
optx	0.28698	0.04891	5.868	2.49e-07 ***
vac	-1.20706	0.88517	-1.364	0.178134
sqft	0.07511	0.01105	6.800	7.41e-09 ***
x3	0.34486	0.10129	3.405	0.001231 **
x4	-0.15437	0.10145	-1.522	0.133739
x6	-0.19038	0.09232	-2.062	0.043839 *
x9	-0.22542	0.11113	-2.029	0.047268 *
x10	-0.38397	0.09581	-4.008	0.000183 ***
x11	-0.29839	0.10528	-2.834	0.006377 **
x14	-0.17005	0.09444	-1.801	0.077133 .
x17	0.41548	0.10696	3.885	0.000273 ***
x19	0.32996	0.11166	2.955	0.004567 **
x21	-0.15718	0.10393	-1.512	0.136084
x22	-0.15554	0.10417	-1.493	0.141020
x27	0.26847	0.08775	3.060	0.003398 **
x30	-0.31647	0.10519	-3.009	0.003927 **
x33	-0.15086	0.09853	-1.531	0.131348
x35	-0.20031	0.10194	-1.965	0.054391 .
x38	0.16086	0.10969	1.466	0.148108
x39	0.12000	0.09388	1.278	0.206457
x40	0.20019	0.10926	1.832	0.072230 .
x43	0.16494	0.10597	1.557	0.125213
x45	0.20619	0.08264	2.495	0.015574 *

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 0.7825 on 56 degrees of freedom

Backwards stepwise selection keeps 20 of the 50 spurious predictors.

Among these 20, we reject  $H_0: \beta_j = 0$  at  $\alpha = 0.05$  for 10 of them.

So the post-selection Type I error rate was  $10/20 = 0.5$  😱.

**WARNING:** Selecting variables and then getting p-values in the selected model often leads to astonishingly inflated Type I error rates.

## References

- Fox, John, and Sanford Weisberg. 2019. *An R Companion to Applied Regression*. Third. Thousand Oaks CA: Sage.  
<https://socialsciences.mcmaster.ca/jfox/Books/Companion/>.
- Kutner, Michael H, Christopher J Nachtsheim, John Neter, and William Li. 2005. *Applied Linear Statistical Models*. McGraw-hill.