

Test

Eric Brist

March 28, 2019

```
library(ISLR) # Hitters #
library(leaps) # regsubsets #
library(CombMSC) # Cp #

## Warning: package 'CombMSC' was built under R version 3.5.3
##
## Attaching package: 'CombMSC'
## The following object is masked from 'package:stats':
##
##      BIC
library(car) # vif #

## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:CombMSC':
##
##      subsets
library(glmnet) # glmnet, cv.glmnet #

## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-16
library(knitr)
library(kableExtra)
### Data ###
dat = Hitters # p 14, 244 #
sum(is.na(dat$Salary)) # 59 #

## [1] 59
dat = na.omit(dat) # omit missing values #
y = dat$Salary
X = dat[,names(dat)!='Salary']
n = nrow(dat); p = ncol(X) # 263 x 20 #
### Best Subsets ###
# Functions #
fit.crit = function(fit) { # model fit criteria #
  sfit = summary(fit); # sse = deviance(fit) #
  c(r2=sfit$r.squared,adjr2=sfit$adj.r.squared,
    aic=AIC(fit,k=2),bic=AIC(fit,k=log(n))) }
predict.regsubsets = function(object,newdata,id) {
  form = as.formula(object$call[[2]]) # regsubset predict function #
  mat = model.matrix(form,newdata) # p 249 #
```

```

coefi = coef(object,id=id)
xvars = names(coefi)
mat[,xvars] %*% coefi }
mse.f = function(x,y) mean((y-x)^2) # mse #
# Model Fit #
fit.fo = lm(Salary~.,data=dat) # FO model #
sfit.fo = summary(fit.fo)
sfit.fo$coefficients # ch6nts, p 1 #

```

```

##              Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) 163.1035878 90.77853560  1.7967197 0.0736219581
## AtBat       -1.9798729  0.63397803 -3.1229361 0.0020076676
## Hits        7.5007675  2.37753415  3.1548517 0.0018082442
## HmRun        4.3308829  6.20144741  0.6983665 0.4856158216
## Runs       -2.3762100  2.98075530 -0.7971839 0.4261224977
## RBI        -1.0449620  2.60087649 -0.4017730 0.6882042223
## Walks        6.2312863  1.82850381  3.4078607 0.0007662281
## Years      -3.4890543 12.41218587 -0.2810991 0.7788735944
## CAtBat     -0.1713405  0.13523684 -1.2669660 0.2063803985
## CHits       0.1339910  0.67455233  0.1986369 0.8427129019
## CHmRun     -0.1728611  1.61723721 -0.1068867 0.9149670939
## CRuns       1.4543049  0.75045769  1.9378906 0.0537950850
## CRBI        0.8077088  0.69261896  1.1661662 0.2446905326
## CWalks     -0.8115709  0.32808251 -2.4736793 0.0140574151
## LeagueN     62.5994230 79.26140140  0.7897845 0.4304236438
## DivisionW  -116.8492456 40.36695165 -2.8946760 0.0041407553
## PutOuts      0.2818925  0.07744057  3.6401141 0.0003329325
## Assists      0.3710692  0.22119878  1.6775373 0.0947231832
## Errors     -3.3607605  4.39163222 -0.7652646 0.4448566263
## NewLeagueN  -24.7623251 79.00262945 -0.3134367 0.7542177622

```

```
fit.crit(fit.fo) # ch6nts, p 1 #
```

```

##              r2      adjr2      aic      bic
## 0.5461159 0.5106270 3794.3827797 3869.3980144

```

```
vif(fit.fo)
```

```

##      AtBat      Hits      HmRun      Runs      RBI      Walks
## 22.944366 30.281255  7.758668 15.246418 11.921715  4.148712
##      Years  CAtBat      CHits      CHmRun      CRuns      CRBI
##  9.313280 251.561160 502.954289 46.488462 162.520810 131.965858
##      CWalks  League  Division  PutOuts  Assists  Errors
## 19.744105  4.134115  1.075398  1.236317  2.709341  2.214543
## NewLeague
##  4.099063

```

```
# Variable Selection #
```

```

regfit.full = regsubsets(Salary ~ ., dat, nvmax = 19, nbest = 1)
# regfit.fwd = regsubsets(Salary~.,dat,numax=19,method="forward")
# regfit.bwd = regsubsets(Salary~.,dat,numax=19,method="backward")
reg.sum = summary(regfit.full) # id variables #
var.id = apply(reg.sum$which,2,as.numeric)[,-1]
k = apply(var.id,1,sum)
best.out1 = cbind(k,var.id) %>% data.frame()

```

```

# aic #
reg.sum$aic = reg.sum$bic-(k+1)*log(n)+2*(k+1)
crit.out = data.frame(reg.sum[c("rsq", "adjr2", "cp", "aic", "bic")])
crit.out$rsq <- ifelse(crit.out$rsq == max(crit.out$rsq),
                      cell_spec(round(crit.out$rsq, 4), bold = T),
                      round(crit.out$rsq, 4))
crit.out$adjr2 <- ifelse(crit.out$adjr2 == max(crit.out$adjr2),
                      cell_spec(round(crit.out$adjr2, 4), bold = T),
                      round(crit.out$adjr2, 4))
crit.out$cp <- ifelse(crit.out$cp == min(crit.out$cp),
                    cell_spec(round(crit.out$cp, 4), bold = T),
                    round(crit.out$cp, 4))
crit.out$aic <- ifelse(crit.out$aic == min(crit.out$aic),
                    cell_spec(round(crit.out$aic, 4), bold = T),
                    round(crit.out$aic, 4))
crit.out$bic <- ifelse(crit.out$bic == min(crit.out$bic),
                    cell_spec(round(crit.out$bic, 4), bold = T),
                    round(crit.out$bic, 4))

names(best.out1) <- c(names(best.out1)[1],
                    cell_spec(names(best.out1)[-1], angle = 90))
x <- cbind(best.out1, crit.out) # ch6nts, Appendix #

x %>% knitr::kable(format = "latex", escape = F) %>%
  kable_styling("striped", full_width = F) %>%
  row_spec(0, bold = T) %>%
  kableExtra::landscape()

```

k	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	LeagueN	DivisionW	PutOuts	Assists	Errors	NewLeagueN	rsq	adjr2	cp	aic	bic
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.3215	0.3189	104.2813	-97.9907	-90.8464
2	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.4252	0.4208	50.7231	-139.6427	-128.9262
3	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0.4514	0.4451	38.6931	-149.9155	-135.6269
4	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0.4754	0.4673	27.8562	-159.6697	-141.8089
5	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0.4908	0.4809	21.613	-165.5044	-144.0714
6	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0.5087	0.4972	14.0239	-172.922	-147.9169
7	0	1	0	0	0	1	0	1	1	1	0	0	0	0	1	1	0	0	0	0.5141	0.5008	13.1285	-173.8332	-145.2559
8	1	1	0	0	0	1	0	0	0	1	1	0	1	0	1	1	0	0	0	0.5286	0.5137	7.4007	-179.7646	-147.6152
9	1	1	0	0	0	1	0	1	0	0	1	1	1	0	1	1	0	0	0	0.5346	0.5181	6.1587	-181.1647	-145.4432
10	1	1	0	0	0	1	0	1	0	0	1	1	1	0	1	1	1	0	0	0.5405	0.5223	5.0093	-182.5102	-143.2165
11	1	1	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	0	0.5426	0.5226	5.8741	-181.7266	-138.8608
12	1	1	0	1	0	1	0	1	0	0	1	1	1	1	1	1	1	0	0	0.5436	0.5217	7.3308	-180.3108	-133.8728
13	1	1	0	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	0	0.5445	0.5207	8.8881	-178.7877	-128.7776
14	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	0	0.5452	0.5195	10.4816	-177.2265	-123.6442
15	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0.5455	0.5179	12.3462	-175.3728	-118.2183
16	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0.5458	0.5162	14.1875	-173.5443	-112.8177
17	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0.546	0.5144	16.0878	-171.6522	-107.3534
18	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0.5461	0.5126	18.0114	-169.7348	-101.8639
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.5461	0.5106	20	-167.7472	-96.3041