Consider the data frame `Auto` from library `ISLR`. It is of interest to predict the car's mileage (`mpg`) using predictor `horsepower`. Use cross validation to find the best polynomial model (up to degree 6). Use function `cv.glm` from library `boot`. Use the following methods

a) Validation set approach with roughly 50% of available data as training set (use `set.seed(1)`).

b) Leave-One-Out cross validation.

c) 10-fold cross validation

```
library(ISLR)   # Auto dataframe
d0 = Auto
plot(mpg~horsepower,d0,cex=0.7)
grid()

# validation set
#============================================================
train=sample(x=1:10,size=6)   # 4  8  5 10  9  7

set.seed(1)
train=sample(x=1:392,size=196)  # default is no replacement
head(train)                     # 105 146 224 354  79 348
m1=lm(mpg~horsepower,Auto,subset=train)

mpg = Auto$mpg
res1=(mpg-predict(m1,Auto))[-train]^2
head(res1)
#        1         2         3         6         7         8
# 1.740320  1.796492  3.666511 44.575314 85.264958 70.987302
mspe1=mean(res1)    # [1] 26.14142   squared-miles

# nonlinear
m2=lm(mpg~poly(horsepower,2),Auto,subset=train)
res2=(mpg-predict(m2,Auto))[-train]^2
mspe2=mean(res2)    # [1] 19.82259

m3=lm(mpg~poly(horsepower,3),Auto,subset=train)
res3=(mpg-predict(m3,Auto))[-train]^2
mean(res3)          # [1] 19.78252

# different seeds - different results
set.seed(2)
train=sample(392,196)
m1=lm(mpg~horsepower,Auto,subset=train)
mean((mpg-predict(m1,Auto))[-train]^2)
# [1] 23.29559
m2=lm(mpg~poly(horsepower,2),Auto,subset=train)
mean((mpg-predict(m2,Auto))[-train]^2)
# [1] 18.90124
m3=lm(mpg~poly(horsepower,3),Auto,subset=train)
mean((mpg-predict(m3,Auto))[-train]^2)
# [1] 19.2574
```

```
# Leave-One-Out Cross-Validation
#============================================================
library(boot)        # cv.glm()
# cv.glm() requires glm() function
# cv.glm() requires argument K, the defaul is K=1 (LOOCV)

# glm() with no family argument same as lm()
m1=lm(mpg~horsepower,Auto)
coef(m1)
# (Intercept)  horsepower
#  39.9358610  -0.1578447

glm1=glm(mpg~horsepower,data=Auto)  # 'data' required
coef(glm1)
# (Intercept)  horsepower
#  39.9358610  -0.1578447

# MSPE from glm1
cverr=cv.glm(Auto,glm1)
summary(cverr)
#       Length Class  Mode
# call    3    -none- call
# K       1    -none- numeric
# delta   2    -none- numeric
# seed  626    -none- numeric

cverr$delta     #[1] 24.23151 24.23114   #MSPE or CV(1)

# MSPE for polynomial fittings
cverror=rep(0,6)   # create vector of zeros

cverror            #[1] 0 0 0 0 0
for (i in 1:6)
{
 models=glm(mpg~poly(horsepower,i),data=Auto)   # word 'data' is required
 cverror[i]=cv.glm(Auto,models)$delta[1]
 }
# wait

cverror  #[1] 24.23151 19.24821 19.33498 19.42443 19.03321 18.97864

plot(cverrors,type="l")
grid()
```

```
# k-Fold Cross-Validation
#============================================================
# Leave 10-out models

set.seed(17)
cverrors=rep(0,6)   # initialize vector

for (i in 1:6)
{
 models=glm(mpg~poly(horsepower,i),data=Auto)
 cverrors[i]=cv.glm(Auto,models,K=10)$delta[1]
}

cverrors
#  [1] 24.20 19.18 19.30 19.337 18.87 19.02

plot(cverrors,type="l")
grid()
```

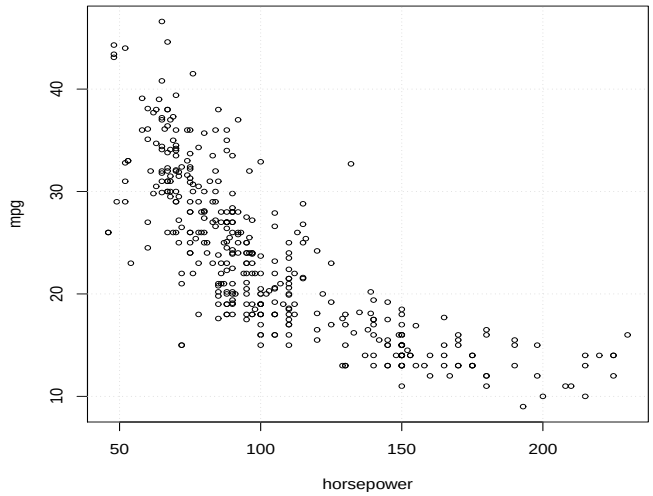


Figure 1: Mileage and horsepower from cars in the Auto dataset

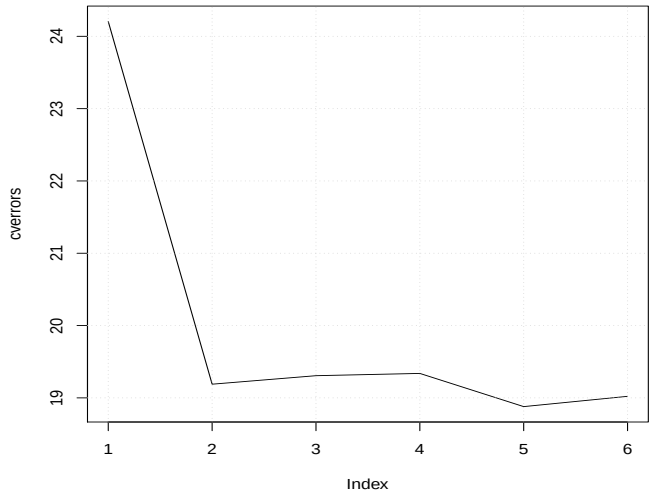


Figure 2: MSPE using LOOCV for polynomial models

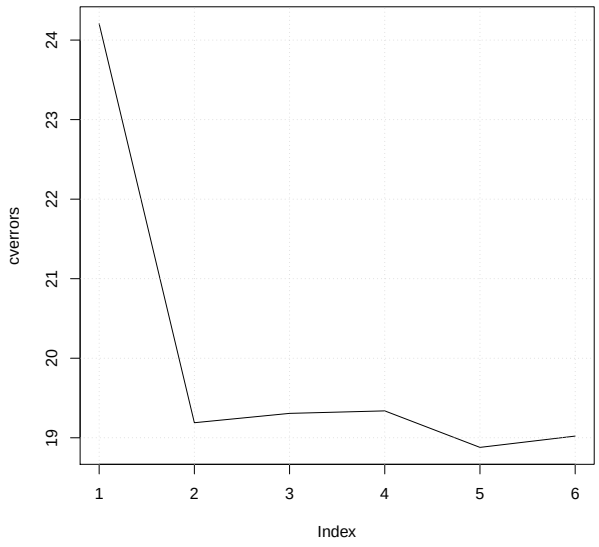


Figure 3: MSPE using k-fold cross validation, for polynomial models