CSC465HW02

Kaiyuan Hu

2017/3/3

Question 1

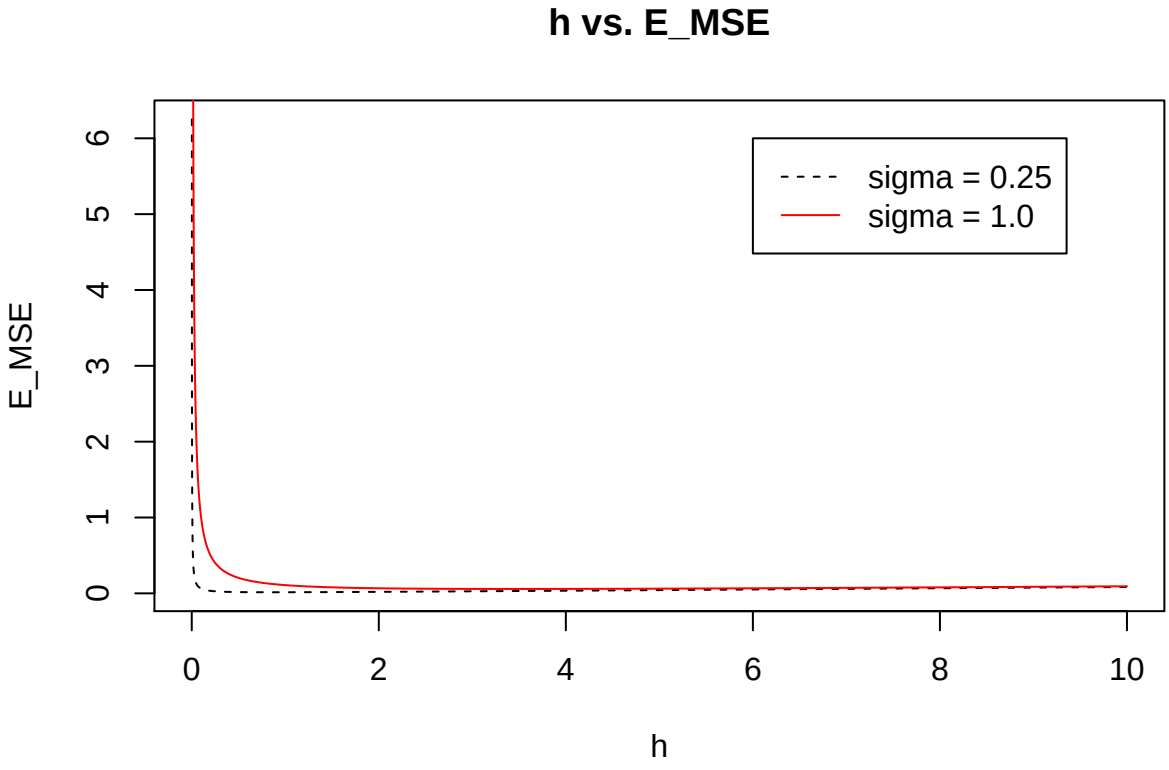
d)

```
h = seq(0.001, 10, 0.001)
MSE = function(Theta, Length){
  # set variables
  h = seq(0.001, 10, 0.001)
  lambda = 10
  beta_1 = 1
  n = lambda*h
  step = seq(1, Length)
  E = 0
  MSE = NULL

  for(i in 1:length(n)){
    p0 = dpois(0, n[i])
    p_step = dpois(step, n[i])
    p = sum(p_step)/(1-p0)
    E_MSE = Theta^2 / n[i] + beta_1^2 * h[i] / (12 * lambda)
    temp = E_MSE * p
    MSE[i] = temp
  }
  return(MSE)
}

# For theta = 0.25
E_MSE_025 = MSE(0.25, 200)
min_h_025 = h[which.min(E_MSE_025)]
# For theta = 1.0
E_MSE_1 = MSE(1, 200)
min_h_1 = h[which.min(E_MSE_1)]

# Draw the plot
plot(h, E_MSE_025, type = 'l', ylab = 'E_MSE', xlab = 'h', col = 'black', lty = 2, main = 'h vs. E_MSE')
lines(h, E_MSE_1, col = 'red')
legend(6, 6, c('sigma = 0.25', 'sigma = 1.0'), lty = c(2,1), col = c('black', 'red'))
```



Question 2

a)

```
library(ISLR)
auto = subset(Auto, Auto$origin == 1)
auto$Y = as.integer(auto$year >= 75)
```

b)

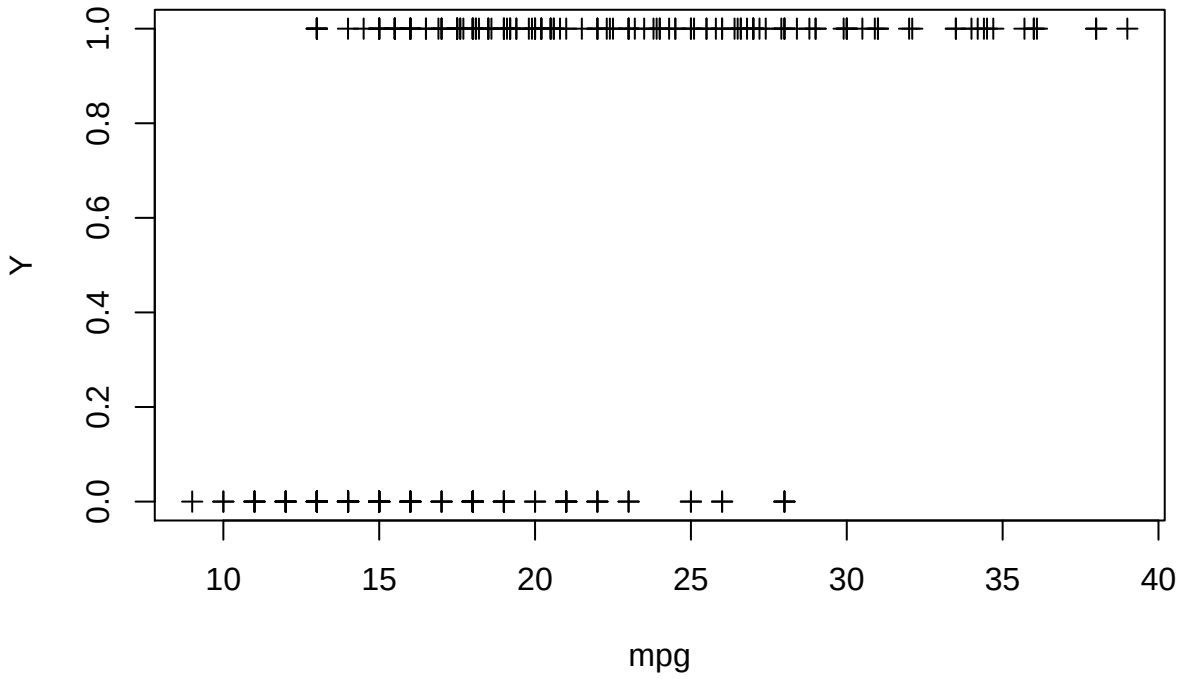
```
m2b = glm(auto$Y ~auto$mpg, family = binomial)
summary(m2b)$coeff
```

```
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -4.2712626 0.66622693 -6.411123 1.444519e-10
## auto$mpg     0.2439545 0.03617442  6.743839 1.542558e-11
```

c)

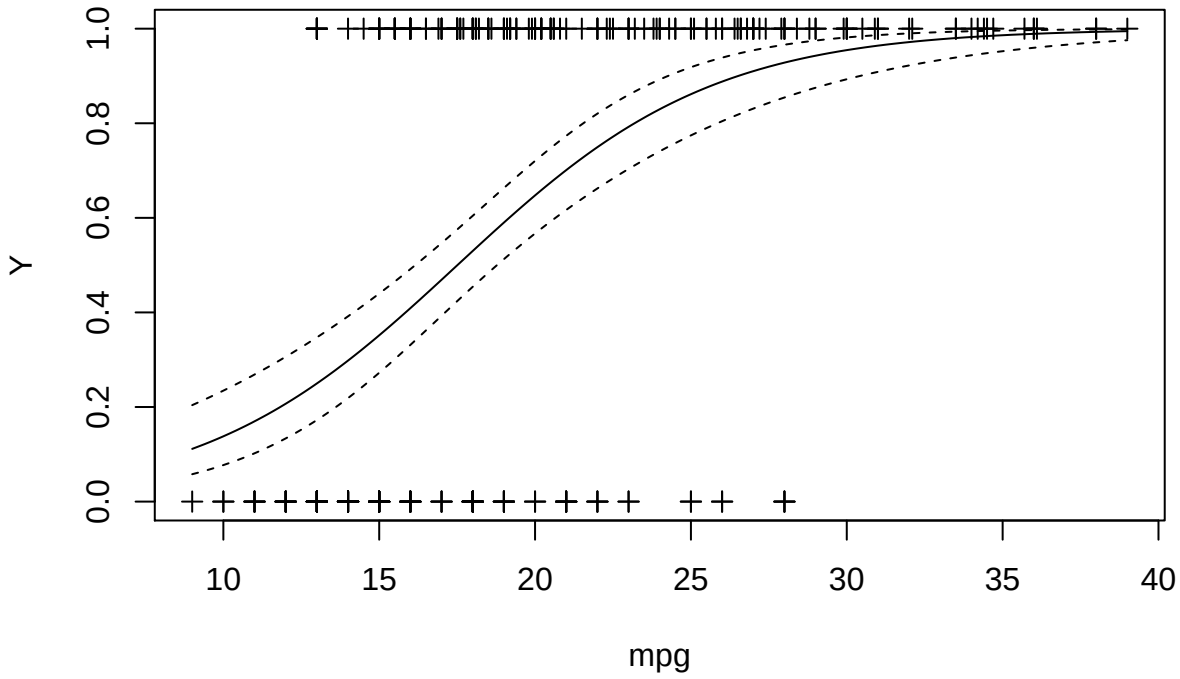
I)

```
plot(auto$Y~auto$mpg, pch = 3, ylab = 'Y', xlab = 'mpg')
```



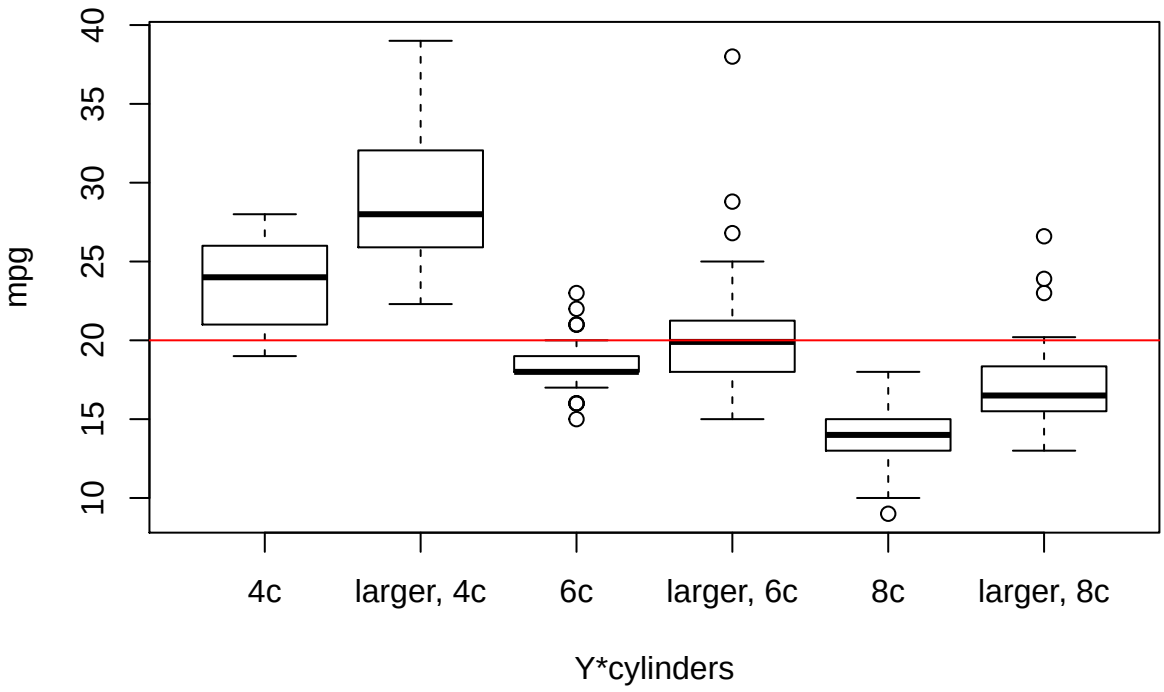
II)

```
logistic = function(x) {(1+exp(-x))^(-1)}
fit = glm(Y ~ mpg, family= binomial, data=auto)
mpg.range = seq(min(auto$mpg),max(auto$mpg),0.1)
pr = predict(fit,newdata=list(mpg=mpg.range),se=T)
plot(auto$Y~auto$mpg, pch = 3, ylab = 'Y', xlab = 'mpg')
lines(mpg.range,logistic(pr$fit))
lines(mpg.range,logistic(pr$fit-2*pr$se.fit),lty=2)
lines(mpg.range,logistic(pr$fit+2*pr$se.fit),lty=2)
```



d)

```
boxplot(mpg~Y*cylinders, data = auto, ylab = 'mpg', xlab = 'Y*cylinders',
        names = c("4c", "larger, 4c", "6c", "larger, 6c", "8c", "larger, 8c"))
abline(h = 20, col = "red")
```



mpg cannot be strong evidence that model year larger than 1975 for both 4 cylinder and 8 cylinder.

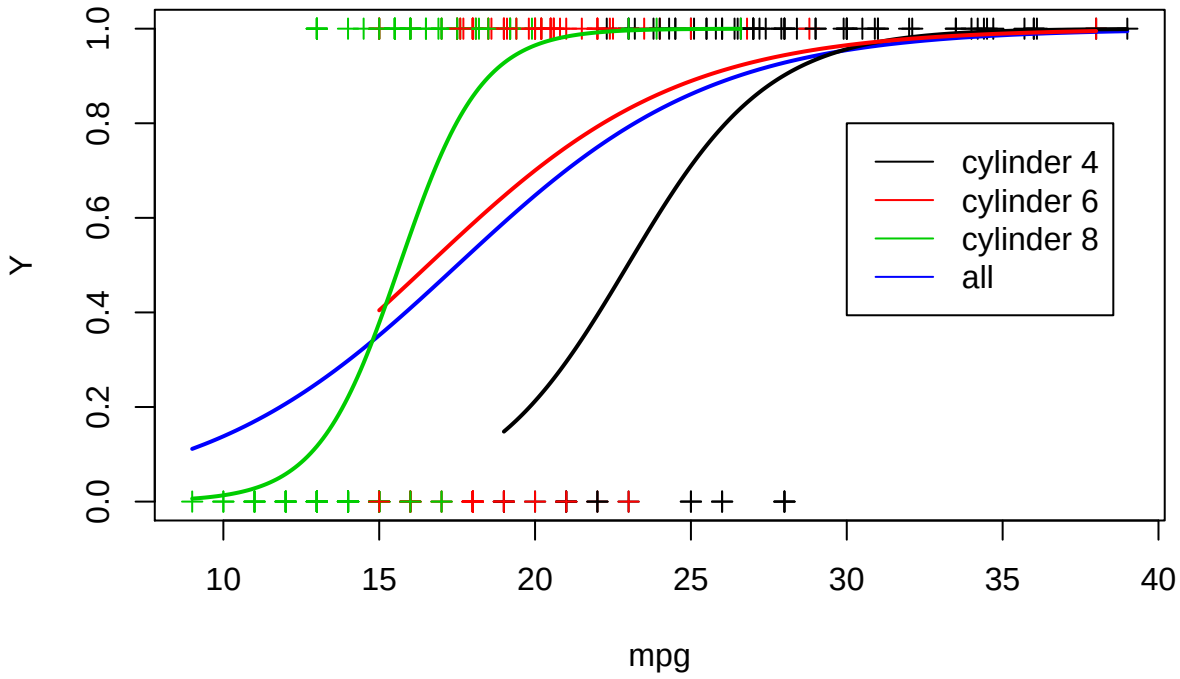
e)

```
fit2 = glm(Y~mpg*as.factor(cylinders), family = binomial, data = auto)
summary(fit2)$coef
```

```
##                                Estimate Std. Error    z value    Pr(>|z|)
## (Intercept)                -10.1238248  3.3717263 -3.0025643 0.002677154
## mpg                          0.4407129  0.1341488  3.2852547 0.001018902
## as.factor(cylinders)6        6.0314261  4.0327915  1.4955958 0.134759020
## as.factor(cylinders)8       -1.7997344  4.1374597 -0.4349854 0.663573046
## mpg:as.factor(cylinders)6   -0.1936746  0.1775849 -1.0906032 0.275447513
## mpg:as.factor(cylinders)8    0.3209773  0.2075835  1.5462561 0.122042724
```

f)

```
# I)
plot(auto$Y~auto$mpg, pch = 3, ylab = 'Y', xlab = 'mpg', col = as.factor(auto$cylinders))
# II)
fit = glm(Y ~ mpg, family= binomial, data=auto)
pr = predict(fit,newdata=list(mpg=mpg.range),se=T)
lines(mpg.range,logistic(pr$fit), col = 4, lwd = 2)
# III)
range.by.cylinder = tapply(auto$mpg, auto$cylinders, function(x) range(x))
# IV) V)
fit2 = glm(Y~mpg*as.factor(cylinders), family = binomial, data = auto)
for(i in 1:3){
  mpg.col = i*2
  mpg.type = 2+i*2
  mpg.range = seq(range.by.cylinder[[i]][1], range.by.cylinder[[i]][2], 0.1)
  ngrid = length(mpg.range)
  pr = predict(fit2, newdata = list(mpg = mpg.range, cylinders = rep(mpg.type, ngrid)), se = F)
  lines(mpg.range, logistic(pr), col=i, lwd = 2)
}
legend(30, 0.8, c('cylinder 4', 'cylinder 6', 'cylinder 8', 'all'), lty = c(1,1,1,1), col = c(1,2,3,4))
```



$P(1975 + |mpg = 20) = 0.64$

$P(1975 + |mpg = 20, cylinder = 4) = 0.21$

$P(1975 + |mpg = 20, cylinder = 6) = 0.70$

$P(1975 + |mpg = 20, cylinder = 8) = 0.96$

Question 3

```
#install.packages("MASS")
library(MASS)
```

a)

```
data = subset(fgl, fgl$type == 'WinF'| fgl$type =='WinNF' | fgl$type =='Veh'| fgl$type =='Head')
```

d)

```
library(class)
library(MASS)

CE = function(classes, features){
  # LDA
  lda.fit = lda(classes ~., data = features, CV = T)
  lda.table = table(lda.fit$class, classes)
  lda.ce = 1 - round(sum(diag(lda.table))/sum(lda.table), 4)
```

```
  # QDA
  qda.fit = qda(classes ~., data = features, CV = T)
  qda.table = table(qda.fit$class, classes)
  qda.ce = 1 - round(sum(diag(qda.table))/sum(qda.table), 4)

  # KNN
  knn.data = features
  knn.data$RI = scale(knn.data$RI)
  RI.scale = c(1:150)/10
  knn.ce = data.frame(CE = numeric(0), K = integer(0), alpha = numeric(0))
  for (i in 1:length(RI.scale)) {
    knn.data$RI = knn.data$RI * RI.scale[i]
    knn.ce.i = NULL
    for (ki in seq(1, 25)) {
      knn.fit = knn.cv(knn.data, classes, k = ki)
      knn.table = table(knn.fit, classes)
      knn.ce.i[ki] = 1 - round(sum(diag(knn.table))/sum(knn.table),4)
    }
    knn.ce[i,] = c(min(knn.ce.i), which.min(knn.ce.i), RI.scale[i])
    knn.data$RI = knn.data$RI / RI.scale[i]
  }
  knn.ce.min = knn.ce[which.min(knn.ce$CE),]
  df = rbind(c(lda.ce,0,0), c(qda.ce,0,0), knn.ce.min)
  rownames(df) <- c('lda', 'qda', 'knn')
  return(df)
}
```

e)

```
# load data
data$type = factor(data$type)
class = data$type
predictors = data[, -10]

ce.compare = CE(class, predictors)
lst = c(1:9)

Blist = list(lst)
CElist = list(ce.compare[3,])
j = 2
while (length(lst) > 2) {
  ce.df = data.frame()
  for (i in 2:length(lst)) {
    features = predictors[, lst]
    features = features[, -i]
    ce.i = CE(class, features)
    ce.i = cbind(i_rm = c(i,i,i), ce.i)
    ce.df = rbind(ce.df, ce.i)
  }
  ce.min = ce.df[which.min(ce.df[,2]),]
  i_rm = ce.min[1,1]
```

```
  lst = lst[-i_rm]
  Blist[[j]] = lst
  CElist[[j]] = ce.min
  j = j + 1
}
ce.value = NULL
for (i in 1:length(CElist)) {
  ce.value[i] = CElist[[i]][[2]]
}
ce.min.index = tail(which(ce.value %in% min(ce.value)), 1)
```

f)

```
Blist[[ce.min.index]]
```

```
## [1] 1 3 6 7 9
```

```
CElist[[ce.min.index]]
```

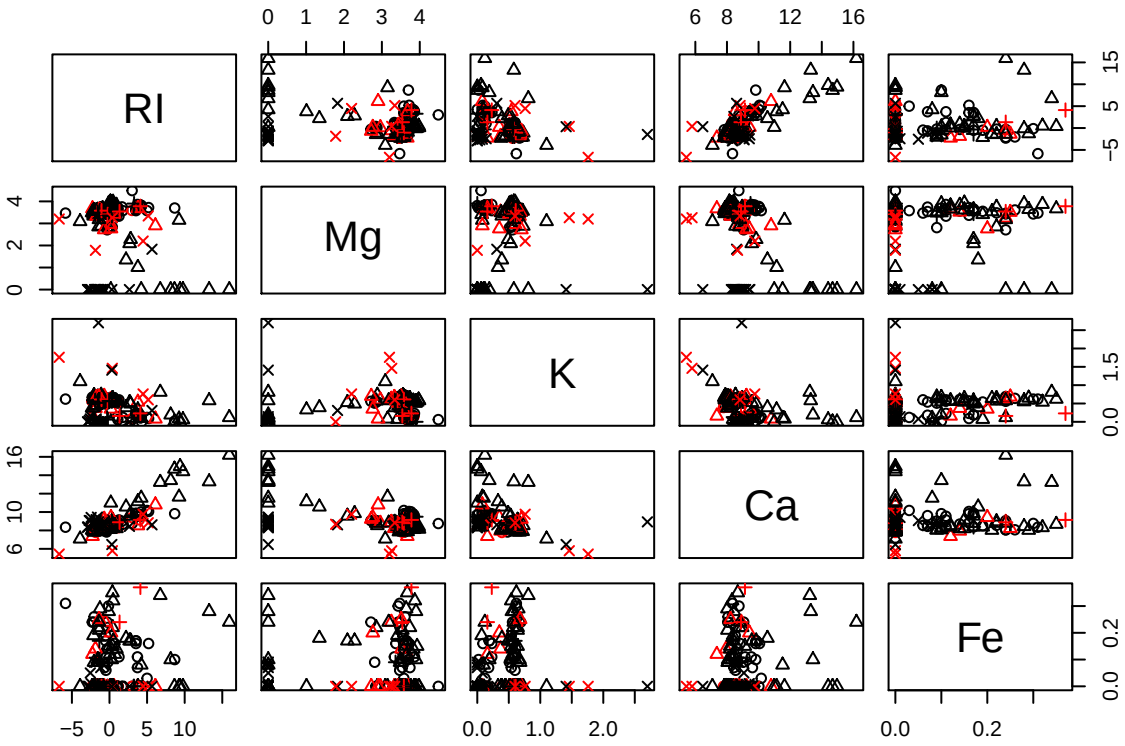
```
##      i_rm     CE K alpha
## knn1    3 0.1354 1   1.3
```

```
predictors$RI <- scale(predictors$RI) * CElist[[ce.min.index]][1,4]
knn.fit = knn.cv(predictors[,Blist[[ce.min.index]]], class, k = CElist[[ce.min.index]][1,3])
knn.table = table(knn.fit, class)
print(knn.table)
```

```
##        class
## knn.fit WinF WinNF Veh Head
##   WinF    65     8   4    0
##   WinNF    2    64   0    5
##   Veh      3     3  13    0
##   Head     0     1   0   24
```

g)

```
pairs(data[, Blist[[ce.min.index]]], col = 1 + (knn.fit != class), pch = as.integer(class))
```


RI
Mg
K
Ca
Fe
0 1 2 3 4
6 8 12 16
−5 5 15
0 2 4
0.0 1.5
6 10 16
0.0 0.2
−5 0 5 10
0.0 1.0 2.0
0.0 0.2