The bootstrap approach can be used to assess the variability of the estimates and predictions from any statistical method. Consider the `Auto` data set in library `ISLR` and use the bootstrap approach in order to assess the variability of the estimates b_0 and b_1 (of β_0 and β_1) for the linear regression model that uses `horsepower` to predict `mpg`.

a) Compare the estimates obtained using the bootstrap to those obtained using standard linear regression.

b) Use the `boot()` function to find the standard errors of 1000 bootstrap estimates for the intercept and slope.

c) Use the `boot.ci()` function to find confidence intervals for the intercept and slope.

d) Find bootstrap estimate for the mileage of a car with horsepower equal to 140.

```
# bootf.r       James p195

library(ISLR)      # Auto dataframe
library(boot)      # boot(),  boot.ci()

# Simple linear regression
#===========================================================

m1=lm(mpg~horsepower,Auto)
summary(m1)
coef(m1)
# (Intercept)  horsepower
#  39.9358610  -0.1578447

# function estimates b0 and b1
bfunction1=function(data,index) coef(lm(mpg~horsepower,data=data,subset=index))

# index is set of rows to be used to fit the model


# fit model with all rows
bfunction1(Auto,1:392)
# (Intercept)  horsepower
#  39.9358610  -0.1578447

set.seed(1)
# fit model with a bootstrap sample (with replacement) of 392 rows
index1=sample(392,392,replace=T)
bfunction1(Auto,index1)
# (Intercept)  horsepower
#  38.7387134  -0.1481952

index2=sample(392,392,replace=T)
bfunction1(Auto,index2)
# (Intercept)  horsepower
# 40.0383086  -0.1596104

# repeat this process N times,
# keep all bo and b1 values
# find average and std deviation of these values
```

```
# Bootstraping the regression coeffs
#============================================================
set.seed(1)
B=1000
boot(Auto,bfunction1,B)

# ORDINARY NONPARAMETRIC BOOTSTRAP
# Bootstrap Statistics :
#       original        bias    std. error
# t1* 39.9358610  0.0296667441 0.860440524
# t2* -0.1578447 -0.0003113047 0.007411218

# compare to the full model

m2=lm(mpg~horsepower,data=Auto)
summary(m2)$coef
#               Estimate  Std. Error   t value      Pr(>|t|)
# (Intercept) 39.9358610 0.717498656  55.65984 1.220362e-187
# horsepower  -0.1578447 0.006445501 -24.48914  7.031989e-81

# Not difference between Estimates
# bias is small
# bootstrap std errors are larger

#----------------------------------------------------------------
confint(m2)
#                2.5 %     97.5 %
#(Intercept) 38.525212 41.3465103
#horsepower  -0.170517 -0.1451725

b2=boot(Auto,bfunction1,B)
boot.ci(b2,type="basic",index=1)
#BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
#Based on 1000 bootstrap replicates
#Intervals :
#Level      Basic
#95%   (38.18, 41.65 )

boot.ci(b2,type="basic",index=2)
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 1000 bootstrap replicates
# Intervals :
# Level      Basic
# 95%   (-0.1724, -0.1422 )
```

```
# Bootstraping prediction
#======================================================================
newdata = data.frame(horsepower=140)
bfunction2=function(data,index) predict(lm(mpg~horsepower,data=data,subset=index),newdata)
bfunction2(Auto,1:392)
# 17.8376

set.seed(1)
boot(Auto,bfunction2,1000)

# ORDINARY NONPARAMETRIC BOOTSTRAP
# Bootstrap Statistics :
#     original      bias    std. error
# t1*  17.8376 -0.01373409   0.3243347

predict(m2,newdata,se.fit=T)
# $fit
#       1
# 17.8376

# $se.fit
# [1] 0.337403

# $df
# [1] 390

# $residual.scale
# [1] 4.905757
```

```
# confidence intervals
#----------------------------------------------------------------------
predict(m2,newdata,interval="conf")
#       fit      lwr      upr
# 1 17.8376 17.17424 18.50095

set.seed(1)
b1=boot(Auto,bfunction2,1000)
boot.ci(b1,type="basic")

# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Intervals :
# Level      Basic
# 95%   (17.25, 18.52)

 boot.ci(b1,type="basic",conf=0.99)
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 1000 bootstrap replicates
# Intervals :
# Level      Basic
# 99%   (17.10, 18.82 )

predict(m2,newdata,interval="conf",level=0.99)
#      fit      lwr      upr
#1 17.8376 16.96423 18.71096
```