Discussion Questions:

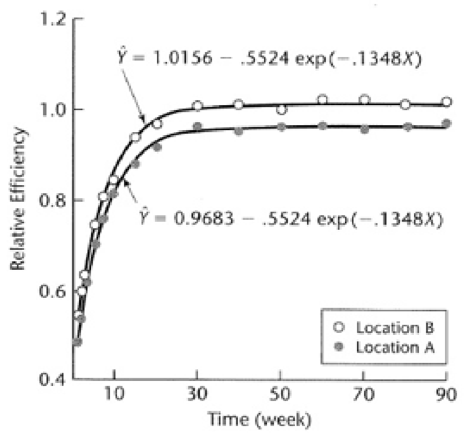
1. In the expression for the likelihood function, which rows have a pi, and which have a 1-pi?

$$p_i\!: y_i = 1 \text{ and } 1 - p_i\!: y_i = 0$$

2. Intuitively, why does it make sense to maximize the likelihood function when fitting the model?

Maximizing the likelihood function makes sense because we are finding the model that is the most likely to fit the data (Maximize the likelihood that you observed what you observed)

3. If facilities A and B had different asymptotic efficiencies as in Fig. 13.5, how would you modify the model?



I would add a dummy variable

4. If facilities A and B had different exponential rates, how would you modify the model?

I would capture it using an interaction term

5. If the objective was to determine if the two facilities had different asymptotic efficiencies, how could you?

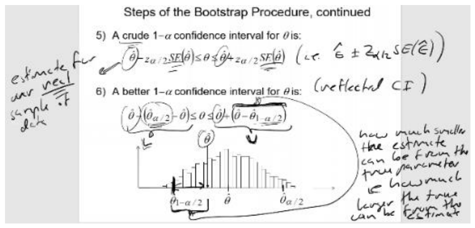
Look at the p-value for the interaction term (Prof said hypothesis test)

7. What is the difference between the two CIs (crude versus reflected) on the previous slide?

Crude: use $\pm z_{\frac{\alpha}{2}} * se(\hat{\epsilon})$

Reflected (Quantls): use $-(\widehat{\theta_{\frac{\alpha}{2}}} - \hat{\theta})$ or $+(\hat{\theta} - \widehat{\theta_{1-\frac{\alpha}{2}}})$

The reflected CI was sifted left a little on the prev.


Steps of the Bootstrap Procedure, continued

5) A crude 1−α confidence interval for θ is:

$\hat{\theta} - z_{\alpha/2}SE(\hat{\theta}) \le \theta \le \hat{\theta} + z_{\alpha/2}SE(\hat{\theta})$ (i.e. $\hat{\epsilon} \pm z_{\alpha/2} SE(\hat{\epsilon})$)

6) A better 1−α confidence interval for θ is: (reflected CI)

$\hat{\theta} - (\hat{\theta}_{\alpha/2} - \hat{\theta}) \le \theta \le \hat{\theta} + (\hat{\theta} - \hat{\theta}_{1-\alpha/2})$

estimate for our real sample of data
$\hat{\theta}_{1-\alpha/2}$
$\hat{\theta}$
$\hat{\theta}_{\alpha/2}$
how much smaller the estimate can be from the true parameter
is how much larger the true can be from the estimate

8. When would the two confidence intervals differ?

When the distribution of $\hat{\theta}$ is highly non-normal

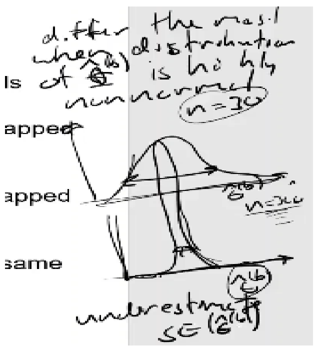
9. What are the effects of increasing B (Number of bootstrapped samples) on the bootstrapped histogram of a parameter estimate? Would the histogram become tighter?

A smoother/better estimate of the true parameter

10. What are the effects of increasing n on the bootstrapped histogram of a parameter estimate? Would the histogram become tighter?

The standard error would be less. The distribution would be very tight. Would underestimate the standard error. That's why in bootstrapping, you should always use n where n is the size of your dataset

11. Why must n for each bootstrapped sample be the same as n for the real sample?



differ when the most distribution of $\hat{\theta}$ is highly nonnormal
Is
apped
apped
same
n=30
n=300
underestimate SE

12. In boot.ci, type = "norm (crude)" gives our crude CI based on the SE and the normal percentiles, but translated by subtracting out the estimated Bias (taken to be the bootstrap average minus the original parameter estimate); type = "basic (reflected)" interval gives the better CI obtained by reflecting the percentiles.

13. How can we determine if there is statistically significant evidence that the asymptotic relative efficiencies of the two manufacturing facilities differ?

Perform hypothesis test. Does the confidence interval for θ_0 contain 0?

14. What is a 95% CI on the asymptotic relative efficiency of the older facility (x1 = 0) & the newer facility (x1 = 1)?

Older: 95% CI on θ_0

Newer: 95% CI on $\theta_0 + \theta_1$

16. Which estimated coefficients (for l = 0 or for l = 0.2) make more sense?

0.2, Lambda always needs to be greater than 0 or else over-fitting will occur

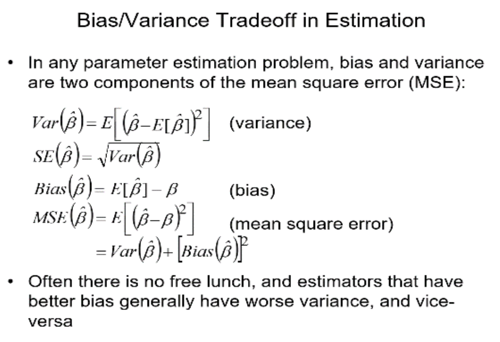
17. n-fold CV indicates that l = 0.001 is slightly better than l = 0.137 or l = 0.667, but this is probably not correct

18. In general, averaging across multiple replicates of K-fold CV is better than using n-fold CV

19. Regarding the bias/variance tradeoff, as λ increases:

$Bias(\widehat{\boldsymbol{\beta}})$ increases

$Var(\widehat{\boldsymbol{\beta}})$ decreases


Bias/Variance Tradeoff in Estimation

- In any parameter estimation problem, bias and variance are two components of the mean square error (MSE):

$$Var(\hat{\beta}) = E\left[\left(\hat{\beta} - E[\hat{\beta}]\right)^2\right] \quad \text{(variance)}$$
$$SE(\hat{\beta}) = \sqrt{Var(\hat{\beta})}$$
$$Bias(\hat{\beta}) = E[\hat{\beta}] - \beta \quad \text{(bias)}$$
$$MSE(\hat{\beta}) = E\left[\left(\hat{\beta} - \beta\right)^2\right] \quad \text{(mean square error)}$$
$$= Var(\hat{\beta}) + \left[Bias(\hat{\beta})\right]^2$$

- Often there is no free lunch, and estimators that have better bias generally have worse variance, and vice-versa

20. Why does the LASSO penalty tend to completely drop some terms from the model, as opposed to keeping the terms in the model but shrinking their coefficients (like ridge regression)?

When the value of lambda is large, the LASSO penalty will drive many of the coefficients towards zero, effectively dropping those terms from the model. Vice Versa for small lambda. LASSO penalty helps to reduce overfitting and improve the interpretability of the model by selecting a subset of the most important variables and setting the coefficients of the less important variables to zero.

21. What is the potential advantage of shrinking coefficients AND dropping some terms via LASSO, versus just shrinking coefficients via ridge regression?

Feature selection, model generalization by reducing the number of coefficients, and reducing overfitting, handles multicollinearity, sparse models, when coefficients are set to zero relationships are more easily

22. How can we get r^2 when R doesn't provide it?

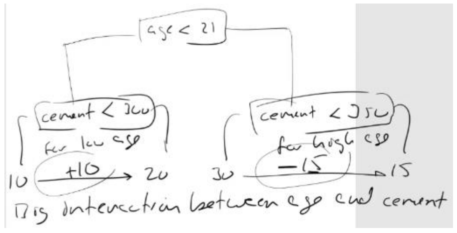
$$R^2 = 1 - \frac{var\big((y - \hat{y})\big)}{var(y)}$$

23. Provide an interpretation of the model and which predictor variables are most important

Which variable appears highest on the tree/the greatest number of times in the tree

24. Do there appear to be any interactions between Age and any other predictor?

Below or else ALE plots



age < 21
cement < 300 for low age
cement < 350 for high age
10
+10
20
30
−15
15
Is interaction between age and cement

25. To grow the initial tree larger (it is better to overgrow the initial tree, then prune it back) you can decrease minbucket and/or cp

26. What is the variable importance measure for trees?

Number assigned by R for variable importance. Branch length is proportional to reduction in SSE. If we want a numerical value for the branch length, we get that from the printout of the tree

27. If you want a summary measure of the predictive quality of the tree model, what would it be?

Use CV(misclass) or CV(deviance)

28. When fitting a tree model, never use misclass to split. Instead, use deviance (information) or Gini

29. How can you tell what impurity measure rpart used to fit the model? R help command lol (Gini default)

30. You choose the best size the same way – choosing the size with the lowest CV error measure (the best M was about 15—20)

31. If you want a summary measure of the predictive quality for the 6-class tree model, what would it be? Same as binary case

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Models (1: NNet, 2: Tree):

Pros and Cons of Neural Networks

- Pros:
  - very flexible; with enough nodes, can model almost any nonlinear relationship
  - can efficiently model linear behavior if the relationship is truly linear
  - often very good predictive power
  - a lot of ways to customize the architecture to tackle specific modeling problems (recurrent NNs; convolutional NNs; autoencoders; other "deep learning" architectures ...)
- Cons:
  - model fitting can be unstable and sensitive to initial guesses
  - for very large data sets, model fitting can be very slow relative to some methods like trees and linear models, which makes CV very computationally expensive
  - overfitting (but can avoid by using CV to choose λ)
  - sensitive to user-chosen "tuning parameters" (but can use CV to choose them wisely)
  - poor interpretability, but this can be improved with ALE or PD plots

Pros and Cons of CART Models

- Pros:
  - almost as flexible as neural networks
  - highly interpretable (at least with simple trees)
  - built-in variable importance measure for each predictor
  - automatically discards irrelevant predictors
  - insensitive to monotonic transformation or outliers in predictors
  - computationally efficient to fit
  - for quantitative responses or binary categorical responses, easy to handle and interpret nominal categorical predictors with many categories (even easier to handle ordinal categorical predictors)
  - can handle missing predictor values
- Cons:
  - poor at representing linear behavior; results in non-smooth response surface; and predictive power usually not as good as neural networks
  - high variance/instability of fitted tree

Formulas:

AIC (smaller is better | df(LogLik) = p+1)

$$AIC = \frac{-2\ln f(\boldsymbol{y};\widehat{\boldsymbol{\theta}})}{n} + \frac{2p}{n}$$

Homework 01

1) In class, we showed that the MLE of the mean of a random sample $\{y_1, y_2, \ldots, y_n\}$ from an $N(\mu, \sigma^2)$ population is exactly the sample average (i.e., $\hat{\mu} = \bar{y}$). Show that the MLE for the standard deviation is:

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Hint: Find the partial derivative of the likelihood function with respect to σ, then set this equal to zero and solve for σ. The result will be a function of the unknown μ, but you can plug in the MLE of μ from class. Hence, we are really finding the *joint* MLEs of μ and σ.

$$Joint\ PDF \Rightarrow\ f(y;\mu,\sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} e^{\frac{-1}{2\sigma^2}\Sigma_{i=1}^{n}(y_i-\mu)^2}$$

$$Likelihood\ function \Rightarrow\ ln(1) - (\frac{n}{2}ln(2\pi) + nln(\sigma)) - \frac{1}{2\sigma^2}\Sigma_{i=1}^{n}(y_i-\mu)^2$$

$$\frac{\partial(f(y;\mu,\sigma))}{\partial\sigma} = \frac{-n}{\sigma} + \frac{1}{\sigma^3}\Sigma_{i=1}^{n}(y_i-\mu)^2 \Rightarrow\ Set = 0$$

$$\Rightarrow\ \frac{\Sigma_{i=1}^{n}(y_i-\mu)^2}{\sigma^3} = \frac{n}{\sigma} \Rightarrow from\ class:\ \mu = \hat{y} \Rightarrow \sigma = \sqrt{\frac{\Sigma_{i=1}^{n}(y_i-\bar{y})^2}{n}}$$

2) The linear regression approach in part (a) gives starting values $\hat{\gamma}_0 = 29.6$ and $\hat{\gamma}_1 = 13.4$. The nlm() script and output is (including results we will use for the next problem):

```
> Enz<-read.table("Enz.csv",sep=",",header=TRUE)
> x=Enz$x
> y=Enz$y
> out<-nlm(function(p) sum((y-p[1]*x/(p[2]+x))^2),p=c(29.6,13.4),hessian=TRUE)
> theta<-out$estimate  #parameter estimates
> MSE<-out$minimum/(length(y) - length(theta))  #estimate of the error variance
> InfoMat<-out$hessian/2/MSE  #observed information matrix
> CovTheta<-solve(InfoMat)
> SE<-sqrt(diag(CovTheta))  #standard errors of parameter estimates
> theta
[1] 28.13688 12.57428
> MSE
[1] 0.2688919
> InfoMat
          [,1]      [,2]
[1,]  15.37455 -13.73842
[2,] -13.73842  13.92197
> CovTheta
          [,1]      [,2]
[1,] 0.5502797 0.5430248
[2,] 0.5430248 0.6076944
> SE
[1] 0.7418084 0.7795476
```

Hence, the parameter estimates are $\hat{\gamma}_0 = 28.14$ and $\hat{\gamma}_1 = 12.57$. The nls() script and output is as below, for which the estimates are the same.

```
> out2<-nls(y~p[1]*x/(p[2]+x),start=list(p=c(29.6,13.4)),data=Enz)
> summary(out2)

Formula: y ~ p[1] * x/(p[2] + x)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
p1  28.1370     0.7280   38.65  < 2e-16 ***
p2  12.5745     0.7631   16.48 1.85e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5185 on 16 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 4.479e-07

> logLik(out2)
'log Lik.' -12.65983 (df=3)
> vcov(out2)
          p1        p2
p1 0.5299534 0.5202828
p2 0.5202828 0.5822505
> confint.default(out2)
      2.5 %   97.5 %
p1 26.71024 29.56386
p2 11.07890 14.07001
```

3) Refer to the same data from Problem (2).

(a) Calculate the observed Fisher information matrix and the covariance matrix of the estimated parameter vector $\hat{\boldsymbol{\gamma}} = [\hat{\gamma}_0, \hat{\gamma}_1]^T$ using the Hessian produced by nlm(). Based on this, calculate the standard errors of the estimated parameters.
(b) Calculate the covariance matrix of $\hat{\boldsymbol{\gamma}}$ using the vcov() function applied to the output of nls(), and based on this calculate the standard errors of the estimated parameters. Do the results agree with Part (a)?
(c) Using the results of Part (a), calculate two-sided 95% CIs on the parameters γ_0 and γ_1. Compare this with the results of the confint.default() function applied to the output of nls().

3) Refer to the same data from Problem 13.10 of KNN.
(a) Calculate the observed Fisher information matrix and the covariance matrix of the estimated parameter vector $\hat{\boldsymbol{\gamma}} = [\hat{\gamma}_0, \hat{\gamma}_1]^T$ using the Hessian produced by nlm(). Based on this, calculate the standard errors of the estimated parameters.

See the R output from Problem 2 for calculations. For nonlinear LS with $\boldsymbol{\theta} = \boldsymbol{\gamma}$ (see notes),

$$log\ f(\mathbf{Y};\boldsymbol{\theta}) = \text{constant} - \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^{n}[y_i - g(\mathbf{x}_i,\boldsymbol{\theta})]^2$$

$$\hat{\mathbf{I}}(\hat{\boldsymbol{\theta}}) = -\frac{\partial^2 log\ f(\mathbf{Y};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \frac{1}{2\hat{\sigma}^2}\frac{\partial^2\sum_{i=1}^{n}[y_i - g(\mathbf{x}_i,\boldsymbol{\theta})]^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \frac{1}{2\hat{\sigma}^2}Hessian$$

$$= \begin{bmatrix} 15.37 & -13.74 \\ -13.74 & 13.92 \end{bmatrix}$$

where $\hat{\sigma}^2 = \text{MSE} = 0.268$

The covariance matrix is the inverse of the information matrix, i.e.,

$$Cov(\hat{\boldsymbol{\theta}}) = \hat{\mathbf{I}}^{-1}(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} 0.550 & 0.543 \\ 0.543 & 0.608 \end{bmatrix}$$

and the standard errors are the square roots of the diagonal elements of the covariance matrix:

$\text{SE}(\hat{\gamma}_0) = 0.742$
$\text{SE}(\hat{\gamma}_1) = 0.780$

(b) Calculate the covariance matrix of $\hat{\boldsymbol{\gamma}}$ using the vcov() function applied to the output of nls(), and based on this calculate the standard errors of the estimated parameters. Do the results agree with Part (a)?

From the R output above,

$$Cov(\hat{\boldsymbol{\theta}}) = \hat{\mathbf{I}}^{-1}(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} 0.530 & 0.520 \\ 0.520 & 0.582 \end{bmatrix},$$

from which the standard errors are

$\text{SE}(\hat{\gamma}_0) = 0.727$
$\text{SE}(\hat{\gamma}_1) = 0.763$

which is very close to part (a).

(c) Using the results of Part (a), calculate two-sided 95% CIs on the parameters γ_0 and γ_1. Compare this with the results of the confint.default() function applied to the output of nls().

For γ_0: $\hat{\gamma}_0 \pm z_{.025}SE(\hat{\gamma}_0) = 28.14 \pm 1.96*0.742 = 28.14 \pm 1.454 = [26.69, 29.59]$
For γ_1: $\hat{\gamma}_1 \pm z_{.025}SE(\hat{\gamma}_1) = 12.57 \pm 1.96*0.780 = 12.57 \pm 1.529 = [11.04, 14.10]$

For the nls() function, the CIs are:

```
> confint.default(out2)
      2.5 %   97.5 %
p1 26.71024 29.56386
p2 11.07890 14.07001
```

which are very close. The difference is probably due to the SEs being a little smaller for the nls() results.

4) This is a repeat of Problem (3), but using bootstrapping to calculate the standard errors and confidence intervals. You can use the boot() command in R (requires the boot package to be loaded with the library(boot) command). Use at least 20,000 bootstrap replicates.

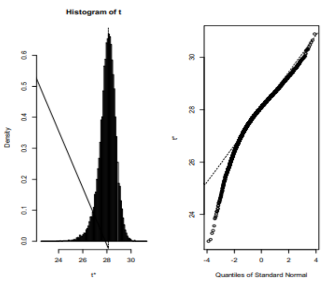
The R code and results for parts (a)—(c) are:

```
> library(boot)   #need to load the boot package
> Enzfit<-function(Z,i,theta0) {
+   Zboot<-Z[i,]
+   x<-Zboot[[2]];
+   y<-Zboot[[1]]
+   fn <- function(p) {yhat<-p[1]*x/(p[2]+x); sum((y-yhat)^2)}
+   out<-nlm(fn,p=theta0)
+   theta<-out$estimate}  #parameter estimates
> Enzboot<-boot(Enz, Enzfit, R=20000, theta0=c(29.6,13.4))
> CovTheta<-cov(Enzboot$t)
> SE<-sqrt(diag(CovTheta))
> Enzboot

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Enz, statistic = Enzfit, R = 20000, theta0 = c(29.6,
   13.4))


Bootstrap Statistics :
   original      bias    std. error
t1* 28.13688 -0.08920329   0.7040837
t2* 12.57428 -0.06325487   0.7331663
> CovTheta
          [,1]      [,2]
[1,] 0.4957338 0.4795740
[2,] 0.4795740 0.5375328
> SE
[1] 0.7040837 0.7331663
> plot(Enzboot,index=1)  #index=i calculates results for ith parameter
```

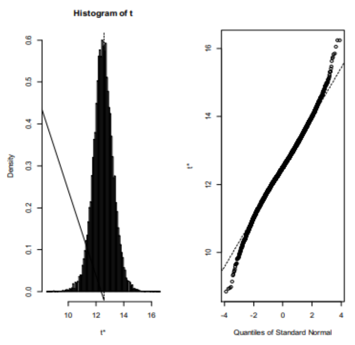


```
> boot.ci(Enzboot,conf=.95,index=1,type=c("norm","basic"))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 20000 bootstrap replicates

CALL :
boot.ci(boot.out = Enzboot, conf = 0.95, type = c("norm", "basic"),
   index = 1)

Intervals :
Level      Normal              Basic
95%   (26.85, 29.61 )   (27.02, 29.84 )
Calculations and Intervals on Original Scale

> plot(Enzboot,index=2)  #index=i calculates results for ith parameter
```



```
> boot.ci(Enzboot,conf=.95,index=2,type=c("norm","basic"))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 20000 bootstrap replicates

CALL :
boot.ci(boot.out = Enzboot, conf = 0.95, type = c("norm", "basic"),
   index = 2)

Intervals :
Level      Normal              Basic
95%   (11.20, 14.07 )   (11.18, 14.09 )
Calculations and Intervals on Original Scale
```

(a) Calculate and plot bootstrapped histograms of $\hat{\gamma}_0$ and $\hat{\gamma}_1$, and calculate the corresponding bootstrapped standard errors.

The histograms and SEs are shown above. The SEs are

$\text{SE}(\hat{\gamma}_0) = 0.704$
$\text{SE}(\hat{\gamma}_1) = 0.733$

which are quite close to the results from Problem 3

(b) Calculate approximate two-sided 95% CIs on γ_0 and γ_1 using the normal approximation to their bootstrapped distributions.

For γ_0: $\hat{\gamma}_0 \pm z_{.025}SE(\hat{\gamma}_0) = 28.14 \pm 1.96*0.704 = 28.14 \pm 1.38 = [26.76, 29.51]$
For γ_1: $\hat{\gamma}_1 \pm z_{.025}SE(\hat{\gamma}_1) = 12.57 \pm 1.96*0.733 = 12.57 \pm 1.44 = [11.13, 14.01]$

These differ from the type = "normal" CIs calculated by boot.ci() only in that the latter subtracts the bias and calculates the CIs as:

$$\hat{\gamma}_j \pm z_{.025}SE(\hat{\gamma}_j) - BIAS(\hat{\gamma}_j)$$

(c) Calculate the reflected two-sided 95% CIs on γ_0 and γ_1 (this corresponds to the type = "basic" option of the boot.ci() function).

From the above results, the CIs are:

For γ_0: [27.02, 29.84]
For γ_1: [11.18, 14.09]

These are of the form $2\hat{\theta} - \hat{\theta}_{\alpha/2} \le \theta \le 2\hat{\theta} - \hat{\theta}_{1-\alpha/2}$

(d) Do the CIs in part (c) agree with those in part (b)? Relate this to the histograms you see in part (a).

They are somewhat different, mainly because the bootstrapped distributions are not centered about $\hat{\theta}$, especially for the γ_0 parameter.

5) Use bootstrapping to calculate a two-sided 95% prediction interval on a "future" response Y^* at $X^* = 27$. Compare this to a two-sided 95% confidence interval on the predictable part $g(\mathbf{x}^*,\boldsymbol{\theta})$ of Y^* at $X^* = 27$. Which interval do you think better represents an interval that you would expect to contain the future response with roughly 95% chance? Explain

The R code and results for the CI on $g(\mathbf{x}^*,\boldsymbol{\theta})$ are:

```
> Enzfit<-function(Z,i,theta0,x_pred) {
+   Zboot<-Z[i,]
+   x<-Zboot[[2]];
+   y<-Zboot[[1]]
+   fn <- function(p) {yhat<-p[1]*x/(p[2]+x); sum((y-yhat)^2)}
+   out<-nlm(fn,p=theta0)
+   theta<-out$estimate
+   y_pred<- theta[1]*x_pred/(theta[2]+x_pred)} #predicted response
> Enzboot<-boot(Enz, Enzfit, R=20000, theta0=c(29.6,13.4), x_pred=27)
> VarYhat<-var(Enzboot$t)
> SEYhat<-sqrt(VarYhat)
> Enzboot

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Enz, statistic = Enzfit, R = 20000, theta0 = c(29.6,
   13.4), x_pred = 27)


Bootstrap Statistics :
   original      bias    std. error
t1* 19.19671 -0.03116383   0.2049908
> VarYhat
          [,1]
[1,] 0.04202122
> SEYhat
          [,1]
[1,] 0.2049908
> plot(Enzboot)
> boot.ci(Enzboot,conf=.95,type=c("norm","basic"))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 20000 bootstrap replicates

CALL :
boot.ci(boot.out = Enzboot, conf = 0.95, type = c("norm", "basic"))

Intervals :
Level      Normal              Basic
95%   (18.83, 19.63 )   (18.91, 19.71 )
Calculations and Intervals on Original Scale
```

Additional R code and results for the PI on are: