

HW 1 Solutions

1) As derived in class, the likelihood function is:

$$f(y_1, y_2, \dots, y_n; \mu, \sigma) = \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right\}$$

Setting the partial derivative with respect to σ equal to zero gives:

$$\begin{aligned} \frac{\partial f(y_1, y_2, \dots, y_n; \mu, \sigma)}{\partial \sigma} &= \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right\} \frac{1}{\sigma^3} \sum_{i=1}^n (y_i - \mu)^2 \\ &\quad - \frac{n}{(2\pi)^{n/2} \sigma^{n+1}} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right\} = 0 \end{aligned}$$

Canceling out common factors gives

$$\frac{1}{\sigma^3} \sum_{i=1}^n (y_i - \mu)^2 - \frac{n}{\sigma} = 0.$$

Solving for σ and plugging in the MLE for μ ($\hat{\mu} = \bar{y}$) gives

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

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: $y \sim p[1] * x / (p[2] + x)$

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 13.10 of KNN.

- (a) Calculate the observed Fisher information matrix and the covariance matrix of the estimated parameter vector $\hat{\gamma} = [\hat{\gamma}_0, \hat{\gamma}_1]^T$ 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 $\theta = \gamma$ (see notes),

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

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

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

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

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

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

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 $\hat{\gamma}$ 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,

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

from which the standard errors are

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

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

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
```

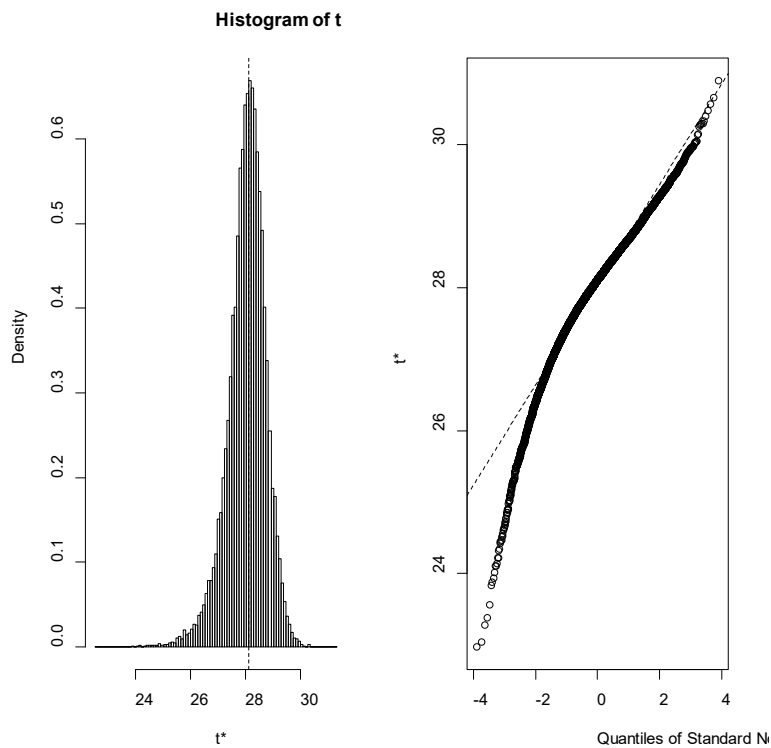
ORDINARY NONPARAMETRIC BOOTSTRAP

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 :

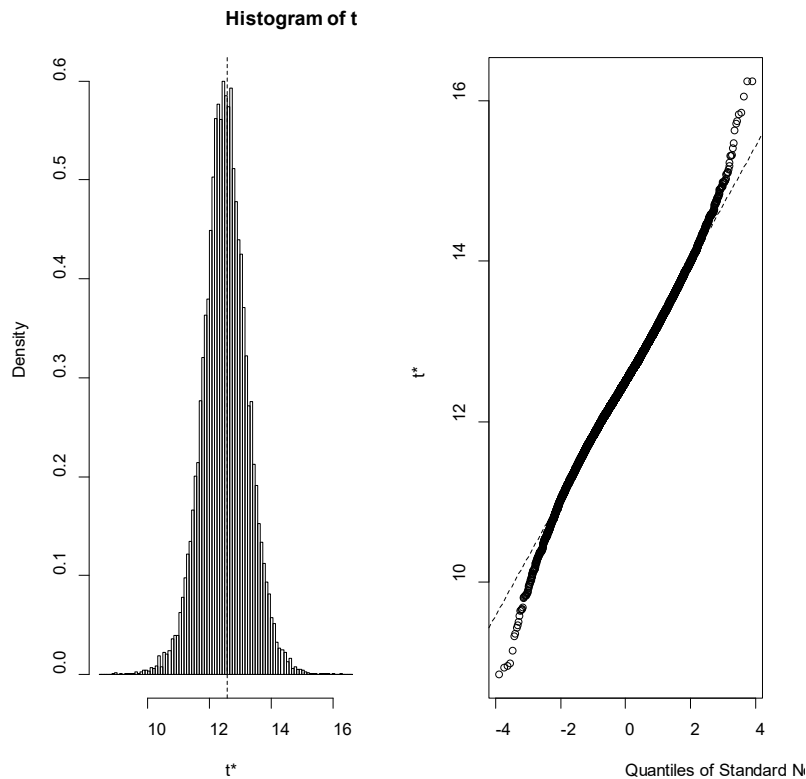
```
boot.ci(boot.out = Enzboot, conf = 0.95, type = c("norm", "basic"),  
index = 1)
```

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

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

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

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.

$$\text{For } \gamma_0: \hat{\gamma}_0 \pm z_{.025} SE(\hat{\gamma}_0) = 28.14 \pm 1.96 * 0.704 = 28.14 \pm 1.38 = [26.76, 29.51]$$

$$\text{For } \gamma_1: \hat{\gamma}_1 \pm z_{.025} SE(\hat{\gamma}_1) = 12.57 \pm 1.96 * 0.733 = 12.57 \pm 1.44 = [11.13, 14.01]$$

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:

$$\text{For } \gamma_0: [27.02, 29.84]$$

$$\text{For } \gamma_1: [11.18, 14.09]$$

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

- (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:


```

> Yhat0<-Enzboot$t0
> Yhatboot<-Enzboot$t
> e<-rnorm(nrow(Yhatboot), mean=0, sd=sqrt(MSE))
> Yboot<-Yhatboot+e
> SEY<-sqrt(var(Yboot))
> Yquant<-quantile(Yboot,prob=c(.025,.975))
> L<-2*Yhat0-Yquant[2]
> U<-2*Yhat0-Yquant[1]
> c(L,U)
    97.5%    2.5%
18.14644 20.33650
> SEY
    [,1]
[1,] 0.5570887

```

From the above, the reflected CI and PI (corresponding to type = “basic”) are:

CI on $g(\mathbf{x}^*, \theta)$: [18.91, 19.71]
 PI on Y^* : [18.15, 20.34]

- 6) Use the AIC criterion to compare the model that you fitted in Problem (2) with the alternative model $Y_i = \beta_0 + \beta_1 \sqrt{X_i} + \varepsilon_i$. Which model does AIC suggest is the better model?

From Problem 2, the log-likelihood is $\log\text{Lik}(\text{out2}) = -12.65983$. Hence

$$AIC = \frac{-2 \log f(\mathbf{y}; \hat{\theta})}{n} + \frac{2p}{n} = \frac{-2(-12.66)}{18} + \frac{2(2)}{18} = 1.629$$

For the new model, we can simply fit a linear regression model using the `lm()` command in R, as follows:

```

> out3<-lm(y~sqrt(x),data=Enz)
> summary(out3)

```

Call:

`lm(formula = y ~ sqrt(x), data = Enz)`

Residuals:

Min	1Q	Median	3Q	Max
-1.7997	-0.5451	0.2085	0.4413	1.6771

Coefficients:

Estimate	Std. Error	t value	Pr(> t)
----------	------------	---------	----------

```
(Intercept) -0.4566  0.5154 -0.886  0.389
sqrt(x)      3.7720  0.1401 26.918 9.41e-15 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9483 on 16 degrees of freedom
Multiple R-squared: 0.9784, Adjusted R-squared: 0.977
F-statistic: 724.6 on 1 and 16 DF, p-value: 9.414e-15

```
> logLik(out3)
'log Lik.' -23.52533 (df=3)
```

Thus,

$$AIC = \frac{-2 \log f(\mathbf{y}; \hat{\boldsymbol{\theta}})}{n} + \frac{2p}{n} = \frac{-2(-23.53)}{18} + \frac{2(2)}{18} = 2.836$$

Since this is much larger than for the Problem 2 model, we conclude the model from Problem 2 is better.

- 7) Use n-fold cross-validation to compare the model that you fitted in Problem (2) with the alternative model $Y_i = \beta_0 + \beta_1 \sqrt{X_i} + \varepsilon_i$. Which model does n-fold cross-validation suggest is the better model?

The R code and results for n-fold CV for the Problem 2 model is:

```
> Enz<-read.table("Enz.csv",sep="," ,header=TRUE)
> n<-nrow(Enz)
> K<-n
> m<-floor(n/K) #approximate size of each part
> r<-n-m*K
> I<-sample(n,n) #random reordering of the indices
> FitFun <- function(x,p) p[1]*x/(p[2]+x)
> SSE<-0
> for (k in 1:K) {
+   if (k <= r) kpart <- ((m+1)*(k-1)+1):((m+1)*k)
+   else kpart<-((m+1)*r+m*(k-r-1)+1):((m+1)*r+m*(k-r))
+   Itest <- I[kpart] #indices for kth part of data
+   Itrain <- I[-kpart] #indices for everything but kth part of data
+   Xtrain<-Enz[Itrain,] #training data for kth part
+   Xtest<-Enz[Ittest,] #test data for kth part
+   out<-nls(y~FitFun(x,p),data=Xtrain,start=list(p=c(29.6,13.4)))
+   Yhat<-predict(out,Xtest)
+   SSE<-SSE+sum((Xtest$y-Yhat)^2)
+ }
> SSE
```

[1] 5.297425

Analogous R code and results for n-fold CV for the new model is:

```
> Enz<-read.table("Enz.csv",sep="," ,header=TRUE)
> n<-nrow(Enz)
> K<-n
> m<-floor(n/K) #approximate size of each part
> r<-n-m*K
> I<-sample(n,n) #random reordering of the indices
> SSE<-0
> for (k in 1:K) {
+   if (k <= r) kpart <- ((m+1)*(k-1)+1):((m+1)*k)
+   else kpart<-((m+1)*r+m*(k-r-1)+1):((m+1)*r+m*(k-r))
+   Itest <- I[kpart] #indices for kth part of data
+   Itrain <- I[-kpart] #indices for everything but kth part of data
+   Xtrain<-Enz[Itrain,] #training data for kth part
+   Xtest<-Enz[Itest,] #test data for kth part
+   out<-lm(y~sqrt(x),data=Xtrain)
+   Yhat<-predict(out,Xtest)
+   SSE<-SSE+sum((Xtest$y-Yhat)^2)
+ }
> SSE
[1] 19.99179
```

The CV SSEs are 5.297 for the Problem 2 model versus 19.992 for the new model. Hence, the Problem 2 model is clearly much better.

- 8) For the two models that you compared in Problems 6 and 7, construct plots of the residuals versus X . Based on the residual plots, does one model appear more appropriate than the other, and does this agree with your conclusions from Problems 6 and 7?

The residuals for the new model of problems 6 and 7 (left plot) and for the model of Problem 2 (right plot) are below. Clearly the residuals for the Problem 2 model are more random and have smaller variance. Hence the conclusion is the same (Problem 2 model is better).

