Escanaba Lake Walleye Assignments

1 Initial Preparation, Get Data, and Simple Summaries

- Load all necessary packages

```
> library(FSA)
> library(nlstools)
> library(AICcmodavg)
> library(dplyr)
> library(magrittr)
```

- Load the data in `WAE_Escanaba_2011_2014.csv` into an R data.frame. Examine the contents of the data.frame.

```
> wae <- read.csv("WAE_Escanaba_2011_14.csv")
> str(wae)
```

```
'data.frame':   2186 obs. of  7 variables:
 $ Lake      : Factor w/ 1 level "Escanaba": 1 1 1 1 1 1 1 1 1 1 ...
 $ Assessment: Factor w/ 1 level "Spring Fyke Net": 1 1 1 1 1 1 1 1 1 1 ...
 $ year      : int  2011 2011 2011 2011 2011 2011 2011 2011 2011 2011 ...
 $ inches    : num  22.3 17.5 22.5 18 19.4 16.1 15.9 22.8 15.1 17.2 ...
 $ sex       : int  2 1 2 2 2 2 1 2 2 1 ...
 $ age       : int  11 13 12 NA NA NA 7 11 6 7 ...
 $ pounds    : num  3.63 1.56 3.38 NA NA NA 1.25 3.69 1.13 1.44 ...
```

```
> headtail(wae)
```

```
         Lake      Assessment year inches sex age pounds
1    Escanaba Spring Fyke Net 2011   22.3   2  11   3.63
2    Escanaba Spring Fyke Net 2011   17.5   1  13   1.56
3    Escanaba Spring Fyke Net 2011   22.5   2  12   3.38
2184 Escanaba Spring Fyke Net 2014   13.9   1  NA     NA
2185 Escanaba Spring Fyke Net 2014   19.8   2   9   2.67
2186 Escanaba Spring Fyke Net 2014   16.5   1  NA     NA
```

- Modify the data.frame in the following ways:
  - Remove the `Lake` and `Assessment` variables (they do not vary and will not be used in any analyses … this simplifies the data.frame),
  - Rename the `inches` and `pounds` variables (to something better),
  - Change `sex` codes to words (note that 1=male, 2=female, 3=unknown),
  - Change the new `sex` variable to a factor variable (this is required for later analyses and can be done within `mutate()` as follows … `sex=factor(sex)`),
  - Add a 1-in length bins variable,
  - Add logs of the length and weight variables,
  - Sort individuals by year, then age, then length, and
  - Examine the resulting data.frame.

```
> wae %<>% select(-Lake,-Assessment) %>%
+   rename(len=inches,wt=pounds) %>%
+   mutate(sex=mapvalues(sex,from=1:3,to=c("male","female","unknown")),
+          sex=factor(sex),
+          lcat=lencat(len,w=1),
+          loglen=log(len),logwt=log(wt)) %>%
```

```
+   arrange(year,age,len)
> headtail(wae)
```

```
     year  len     sex age   wt lcat   loglen     logwt
1    2011  8.5 unknown   1 0.06    8 2.140066 -2.813411
2    2011 10.2 unknown   2 0.19   10 2.322388 -1.660731
3    2011 11.4    male   3 0.31   11 2.433613 -1.171183
2184 2014 21.2  female  NA   NA   21 3.054001        NA
2185 2014 21.3  female  NA   NA   21 3.058707        NA
2186 2014 22.7  female  NA   NA   22 3.122365        NA
```

- Produce some simple summaries that could be used to answer the following questions:
  - What is the mean length of all Walleye?
  - What is the standard deviation of Walleye lengths in each year?
  - How many fish were captured in each year?
  - How many fish of each sex were captured in each year?
  - [*Bonus*] What is the maximum length of Walleye for each sex in each year?

```
> Summarize(~len,data=wae,digits=1)
```

```
     n   mean     sd    min     Q1 median     Q3    max
2186.0   16.3    2.4    8.2   14.7   16.0   17.7   27.3
```

```
> Summarize(len~year,data=wae,digits=1)
```

```
  year   n mean  sd  min   Q1 median   Q3  max
1 2011 399 16.7 2.4  8.5 15.0   16.4 18.2 25.6
2 2012 664 16.1 2.4  9.3 14.5   15.7 17.3 26.0
3 2013 530 16.2 2.7  8.2 14.4   15.9 17.7 27.3
4 2014 593 16.5 2.2 10.2 15.0   16.3 17.6 24.3
```

```
> xtabs(~year,data=wae)
```

```
year
2011 2012 2013 2014
 399  664  530  593
```

```
> xtabs(~sex+year,data=wae)
```

```
         year
sex        2011 2012 2013 2014
  female    256  201  228  300
  male      140  424  256  266
  unknown     3   39   46   27
```

```
> Summarize(len~sex:year,data=wae,digits=1)
```

```
       sex year   n mean  sd  min   Q1 median   Q3  max
1   female 2011 256 17.7 2.0 12.9 16.2   17.4 19.0 25.6
2     male 2011 140 14.9 1.6 11.4 14.0   14.8 15.7 20.3
3  unknown 2011   3 10.1 1.5  8.5  9.3   10.2 10.8 11.5
4   female 2012 201 18.5 2.3 13.9 16.9   18.2 19.9 26.0
5     male 2012 424 15.1 1.4 11.3 14.2   15.0 15.9 19.5
6  unknown 2012  39 14.4 2.7  9.3 12.6   14.5 15.9 20.7
7   female 2013 228 18.2 2.3 13.8 16.6   17.8 19.6 27.3
8     male 2013 256 14.7 1.5 10.4 13.7   14.8 15.8 19.3
9  unknown 2013  46 14.4 2.9  8.2 13.0   14.1 16.1 22.7
10  female 2014 300 17.8 2.1 13.3 16.4   17.4 18.9 24.3
```

```
11    male 2014 266 15.2 1.3 12.2 14.3   15.1 16.1 19.7
12 unknown 2014  27 15.3 3.1 10.2 13.2   15.3 17.1 23.2
```

2 Create an Age-Length Key

- Create a new data.frame of aged female Walleye captured in 2014. [*Check your work*]

```
> wae14F.aged <- filterD(wae,sex=="female",year==2014,!is.na(age))
> headtail(wae14F.aged)
```

```
   year  len    sex age   wt lcat   loglen      logwt
1  2014 13.3 female   4 0.66   13 2.587764 -0.4155154
2  2014 13.3 female   4 0.64   13 2.587764 -0.4462871
3  2014 13.8 female   4 0.82   13 2.624669 -0.1984509
91 2014 22.7 female  13 3.95   22 3.122365  1.3737156
92 2014 22.7 female  13 4.43   22 3.122365  1.4883996
93 2014 24.3 female  15 5.50   24 3.190476  1.7047481
```

- Construct an age-length key (by 1-in length categories) for female Walleye captured in 2014.

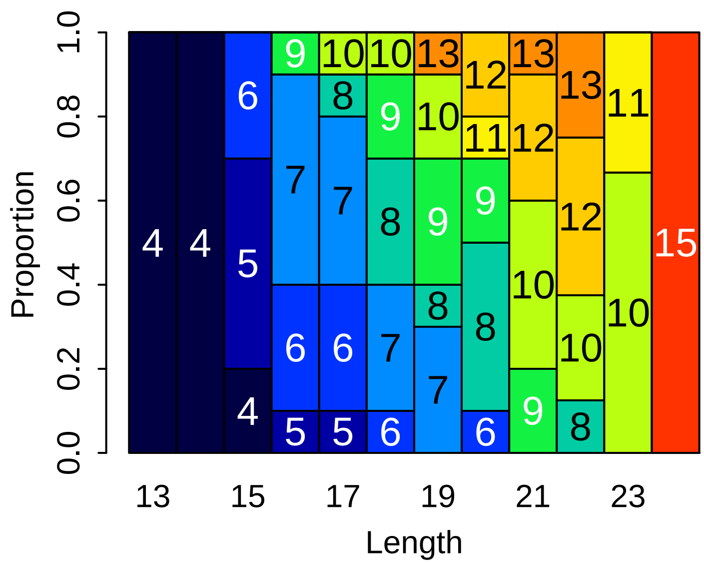
```
> wae14F.raw <- xtabs(~lcat+age,data=wae14F.aged)
> wae14F.alk <- prop.table(wae14F.raw,margin=1)
```

- Examine the age-length key (both as a table and as a plot). Do you see any potential issues with this age-length key.

```
> round(wae14F.alk*100,1)
```

```
    age
lcat     4     5     6     7     8     9    10    11    12    13    15
  13 100.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  14 100.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  15  20.0  50.0  30.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  16   0.0  10.0  30.0  50.0   0.0  10.0   0.0   0.0   0.0   0.0   0.0
  17   0.0  10.0  30.0  40.0  10.0   0.0  10.0   0.0   0.0   0.0   0.0
  18   0.0   0.0  10.0  30.0  30.0  20.0  10.0   0.0   0.0   0.0   0.0
  19   0.0   0.0   0.0  30.0  10.0  30.0  20.0   0.0   0.0  10.0   0.0
  20   0.0   0.0  10.0   0.0  40.0  20.0   0.0  10.0  20.0   0.0   0.0
  21   0.0   0.0   0.0   0.0   0.0  20.0  40.0   0.0  30.0  10.0   0.0
  22   0.0   0.0   0.0   0.0  12.5   0.0  25.0   0.0  37.5  25.0   0.0
  23   0.0   0.0   0.0   0.0   0.0   0.0  66.7  33.3   0.0   0.0   0.0
  24   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 100.0
```

```
> alkPlot(wae14F.alk)
```

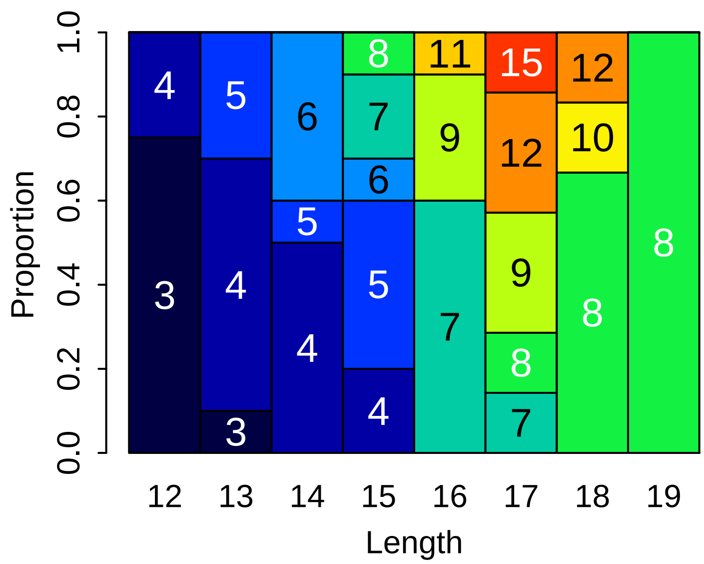


- Repeat the three previous steps for aged male Walleye captured in 2014.

```
> wae14M.aged <- filterD(wae,sex=="male",year==2014,!is.na(age))
> wae14M.raw <- xtabs(~lcat+age,data=wae14M.aged)
> wae14M.alk <- prop.table(wae14M.raw,margin=1)
> round(wae14M.alk*100,1)
```

```
    age
lcat     3     4     5     6     7     8     9    10    11    12    15
  12  75.0  25.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  13  10.0  60.0  30.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  14   0.0  50.0  10.0  40.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  15   0.0  20.0  40.0  10.0  20.0  10.0   0.0   0.0   0.0   0.0   0.0
  16   0.0   0.0   0.0   0.0  60.0   0.0  30.0   0.0  10.0   0.0   0.0
  17   0.0   0.0   0.0   0.0  14.3  14.3  28.6   0.0   0.0  28.6  14.3
  18   0.0   0.0   0.0   0.0   0.0  66.7   0.0  16.7   0.0  16.7   0.0
  19   0.0   0.0   0.0   0.0   0.0 100.0   0.0   0.0   0.0   0.0   0.0
```

```
> alkPlot(wae14M.alk)
```



3 Apply Age-Length Key (assign ages to unaged fish)

- Create a new data.frame of unaged female Walleye captured in 2014. [*Check your work*]

```
> wae14F.unaged <- filterD(wae,sex=="female",year==2014,is.na(age))
> headtail(wae14F.unaged)
```

```
    year  len    sex age wt lcat   loglen logwt
1   2014 14.5 female  NA NA   14 2.674149    NA
2   2014 14.5 female  NA NA   14 2.674149    NA
3   2014 14.7 female  NA NA   14 2.687847    NA
205 2014 21.2 female  NA NA   21 3.054001    NA
206 2014 21.3 female  NA NA   21 3.058707    NA
207 2014 22.7 female  NA NA   22 3.122365    NA
```

- Use the age-length key for female Walleye captured in 2014 (from above) to assign ages to all fish in this new data.frame.

```
> wae14F.unaged <- alkIndivAge(wae14F.alk,age~len,data=wae14F.unaged)
> headtail(wae14F.unaged)
```

```
    year  len    sex age wt lcat   loglen logwt
1   2014 14.5 female   4 NA   14 2.674149    NA
2   2014 14.5 female   4 NA   14 2.674149    NA
3   2014 14.7 female   4 NA   14 2.687847    NA
205 2014 21.2 female  10 NA   21 3.054001    NA
206 2014 21.3 female  10 NA   21 3.058707    NA
207 2014 22.7 female  10 NA   22 3.122365    NA
```

- Create a data.frame that contains ALL (now with ages) female Walleye captured in 2014.

```
> wae14F <- rbind(wae14F.aged,wae14F.unaged)
> headtail(wae14F)
```

```
    year  len    sex age   wt lcat   loglen      logwt
1   2014 13.3 female   4 0.66   13 2.587764 -0.4155154
```

```
2   2014 13.3 female   4 0.64   13 2.587764 -0.4462871
3   2014 13.8 female   4 0.82   13 2.624669 -0.1984509
298 2014 21.2 female  10   NA   21 3.054001         NA
299 2014 21.3 female  10   NA   21 3.058707         NA
300 2014 22.7 female  10   NA   22 3.122365         NA
```

- Repeat all of the steps above for male Walleye captured in 2014.

```
> wae14M.unaged <- filterD(wae,sex=="male",year==2014,is.na(age))
> wae14M.unaged <- alkIndivAge(wae14M.alk,age~len,data=wae14M.unaged)
> wae14M <- rbind(wae14M.aged,wae14M.unaged)
```

- Combine the female and male data.frames from above into one data.frame that contains all (sexed) Walleye captured in 2014 (now with ages).

```
> wae14 <- rbind(wae14F,wae14M)
```

4 Estimate Mortality Rate

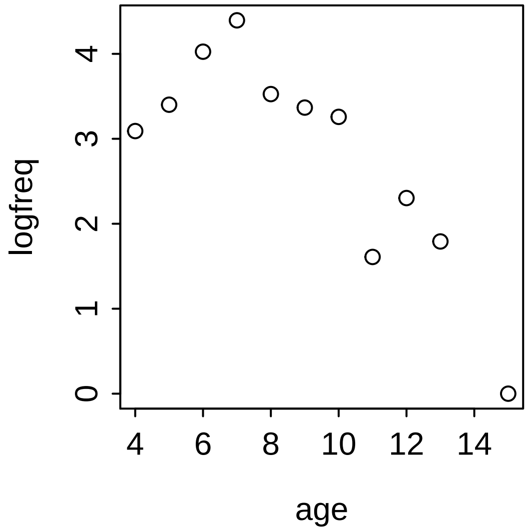
- Create a data.frame that contains the frequency (and log frequency) at age of female Walleye captured in 2014.

```
> wae14F.af <- group_by(wae14F,age) %>%
+   summarise(freq=n()) %>%
+   mutate(logfreq=log(freq)) %>%
+   as.data.frame()
> wae14F.af
```

```
   age freq  logfreq
1    4   22 3.091042
2    5   30 3.401197
3    6   56 4.025352
4    7   81 4.394449
5    8   34 3.526361
6    9   29 3.367296
7   10   26 3.258097
8   11    5 1.609438
9   12   10 2.302585
10  13    6 1.791759
11  15    1 0.000000
```

- Construct a plot and determine which ages define the "descending limb."

```
> plot(logfreq~age,data=wae14F.af)
```



- Estimate (point and 95% confidence interval) Z (and A) using a weighted linear regression, but not using `catchCurve()`.

```
> wae14F.af.rec <- filterD(wae14F.af,age>=7,age<15)
> wae14F.cc1 <- lm(logfreq~age,data=wae14F.af.rec)
> wae14F.af.rec %<>% mutate(wts=predict(wae14F.cc1))
> wae14F.cc2 <- lm(logfreq~age,data=wae14F.af.rec,weights=wts)
> cbind(Est=coef(wae14F.cc2),confint(wae14F.cc2))
```

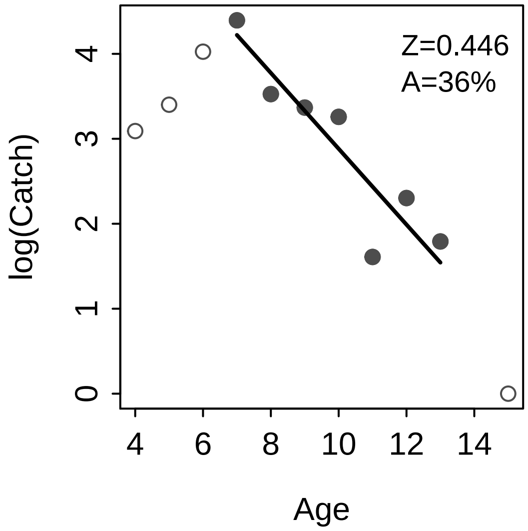
```
                   Est      2.5 %     97.5 %
(Intercept)  7.3455621  5.2156786  9.4754457
age         -0.4462935 -0.6681867 -0.2244004
```

- Estimate (point and 95% confidence interval) Z (and A) using a weighted linear regression, using `catchCurve()`.

```
> wae14F.cc1 <- catchCurve(freq~age,data=wae14F.af,
+                          ages2use=7:13,weighted=TRUE)
> cbind(Est=coef(wae14F.cc1),confint(wae14F.cc1))
```

```
         Est    95% LCI    95% UCI
Z  0.4462935  0.2244004  0.6681867
A 36.0004122 20.1004825 48.7362707
```

```
> plot(wae14F.cc1)
```



- [*Bonus*] What impact does the low catch of age-11 fish have on the estimate of Z (and A)?

```
> wae14F.cc2 <- catchCurve(freq~age,data=wae14F.af,
+                          ages2use=c(7:10,12),weighted=TRUE)
> cbind(Est=coef(wae14F.cc2),confint(wae14F.cc2))
```

```
        Est     95% LCI     95% UCI
Z  0.377174  0.1588109  0.5955371
A 31.420326 14.6842290 44.8733611
```

- Estimate (point and 95% confidence interval) Z (and A) for male Walleye captured in 2014 using a weighted linear regression.

```
> wae14M.af <- group_by(wae14M,age) %>%
+   summarise(freq=n()) %>%
+   mutate(logfreq=log(freq)) %>%
+   as.data.frame()
> wae14M.af
```

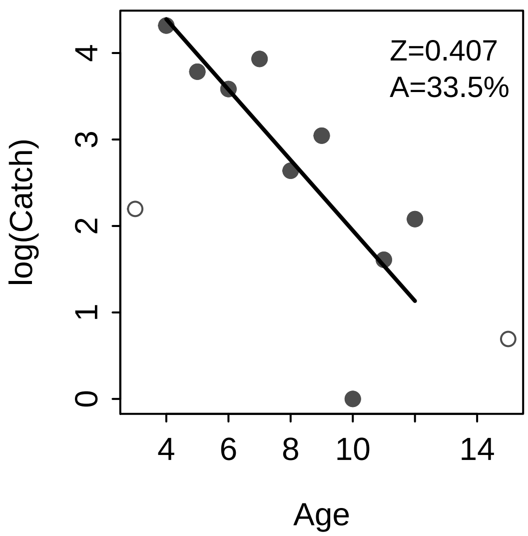
```
   age freq   logfreq
1    3    9 2.1972246
2    4   75 4.3174881
3    5   44 3.7841896
4    6   36 3.5835189
5    7   51 3.9318256
6    8   14 2.6390573
7    9   21 3.0445224
8   10    1 0.0000000
9   11    5 1.6094379
10  12    8 2.0794415
11  15    2 0.6931472
```

```
> wae14M.cc1 <- catchCurve(freq~age,data=wae14M.af,
+                          ages2use=4:12,weighted=TRUE)
> cbind(Est=coef(wae14M.cc1),confint(wae14M.cc1))
```

```
         Est     95% LCI     95% UCI
Z  0.4074049  0.1514543  0.6633555
```

```
A 33.4625244 14.0542803 48.4880029
```

```
> plot(wae14M.cc1)
```



5 Compare Mortality Rates

- Create a data.frame that contains the frequency (and log frequency) by age of Walleye captured in 2014 separated by sex.

```
> ALL.af <- group_by(wae14,sex,age) %>%
+   summarise(freq=n()) %>%
+   mutate(logfreq=log(freq)) %>%
+   as.data.frame()
> ALL.af
```

```
      sex age freq   logfreq
1  female   4   22 3.0910425
2  female   5   30 3.4011974
3  female   6   56 4.0253517
4  female   7   81 4.3944492
5  female   8   34 3.5263605
6  female   9   29 3.3672958
7  female  10   26 3.2580965
8  female  11    5 1.6094379
9  female  12   10 2.3025851
10 female  13    6 1.7917595
11 female  15    1 0.0000000
12   male   3    9 2.1972246
13   male   4   75 4.3174881
14   male   5   44 3.7841896
15   male   6   36 3.5835189
16   male   7   51 3.9318256
17   male   8   14 2.6390573
18   male   9   21 3.0445224
19   male  10    1 0.0000000
```

```
20   male  11    5 1.6094379
21   male  12    8 2.0794415
22   male  15    2 0.6931472
```

- Fit a weighted indicator variable regression to the descending limbs so that Z (i.e., the slopes) can be statistically compared between sexes. [*Note that this will require a careful filtering of the summaries produced above to isolate both descending limbs.*]

```
> ALL.af.rec <- filterD(ALL.af,(age>=4 & age<14 & sex=="male") |
+                              (age>=7 & age<13 & sex=="female"))
> ALL.af.rec
```

```
      sex age freq  logfreq
1  female   7   81 4.394449
2  female   8   34 3.526361
3  female   9   29 3.367296
4  female  10   26 3.258097
5  female  11    5 1.609438
6  female  12   10 2.302585
7    male   4   75 4.317488
8    male   5   44 3.784190
9    male   6   36 3.583519
10   male   7   51 3.931826
11   male   8   14 2.639057
12   male   9   21 3.044522
13   male  10    1 0.000000
14   male  11    5 1.609438
15   male  12    8 2.079442
```

```
> ALL.cc1 <- lm(logfreq~age*sex,data=ALL.af.rec)
> ALL.af.rec %<>% mutate(wts=predict(ALL.cc1))
> ALL.cc2 <- lm(logfreq~age*sex,data=ALL.af.rec,weights=wts)
> cbind(Est=coef(ALL.cc2),confint(ALL.cc2))
```

```
                    Est      2.5 %     97.5 %
(Intercept)  7.60947001  4.3417958 10.8771443
age         -0.47770044 -0.8326152 -0.1227857
sexmale     -1.58795854 -5.2091828  2.0332657
age:sexmale  0.07029558 -0.3417350  0.4823261
```

- Statistically test if the slopes (i.e., Z) differ between the sexes.

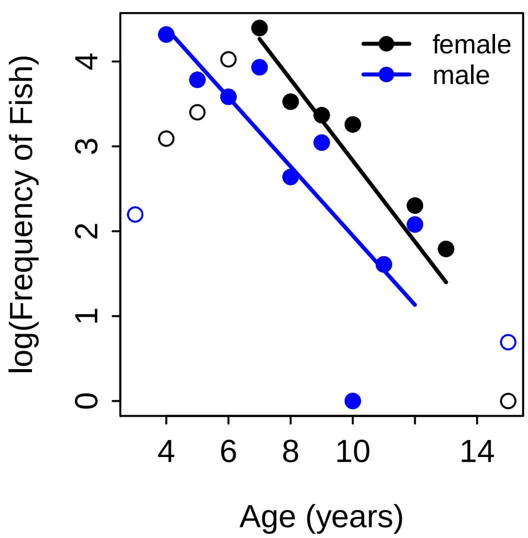
```
> anova(ALL.cc2)
```

```
Analysis of Variance Table

Response: logfreq
          Df  Sum Sq Mean Sq F value    Pr(>F)
age        1 26.9507 26.9507 20.6347 0.0008404
sex        1  8.4993  8.4993  6.5075 0.0269484
age:sex    1  0.1842  0.1842  0.1410 0.7144252
Residuals 11 14.3670  1.3061
```

- Construct a "fancy" plot that demonstrates the catch-curves both sexes.

```
> clrs <- c("black","blue")
> plot(logfreq~age,data=ALL.af,col=clrs[as.numeric(sex)],
+      xlab="Age (years)",ylab="log(Frequency of Fish)")
```

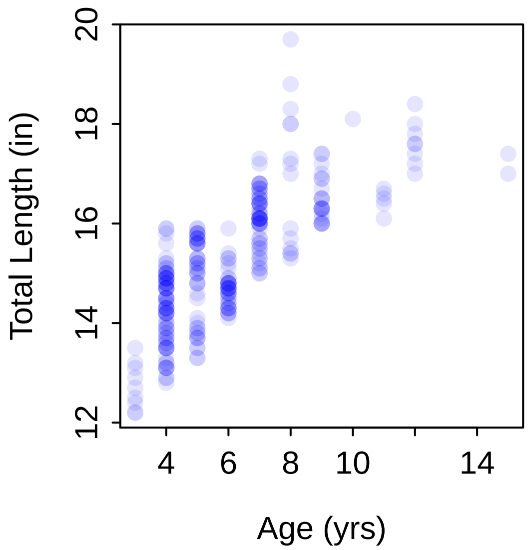
```
> points(logfreq~age,data=filterD(ALL.af,sex=="female",age>=7,age<14),
+        pch=19,col=clrs[1])
> tmp <- c(7,13)
> lines(tmp,predict(ALL.cc2,data.frame(age=tmp,sex="female")),
+       col=clrs[1],lwd=2)
> points(logfreq~age,data=filterD(ALL.af,sex=="male",age>=4,age<13),
+        pch=19,col=clrs[2])
> tmp <- c(4,12)
> lines(tmp,predict(ALL.cc2,data.frame(age=tmp,sex="male")),
+       col=clrs[2],lwd=2)
> legend("topright",levels(wae14$sex),col=clrs,pch=19,lwd=2,
+        bty="n",cex=0.8)
```



6 Fit Growth Model

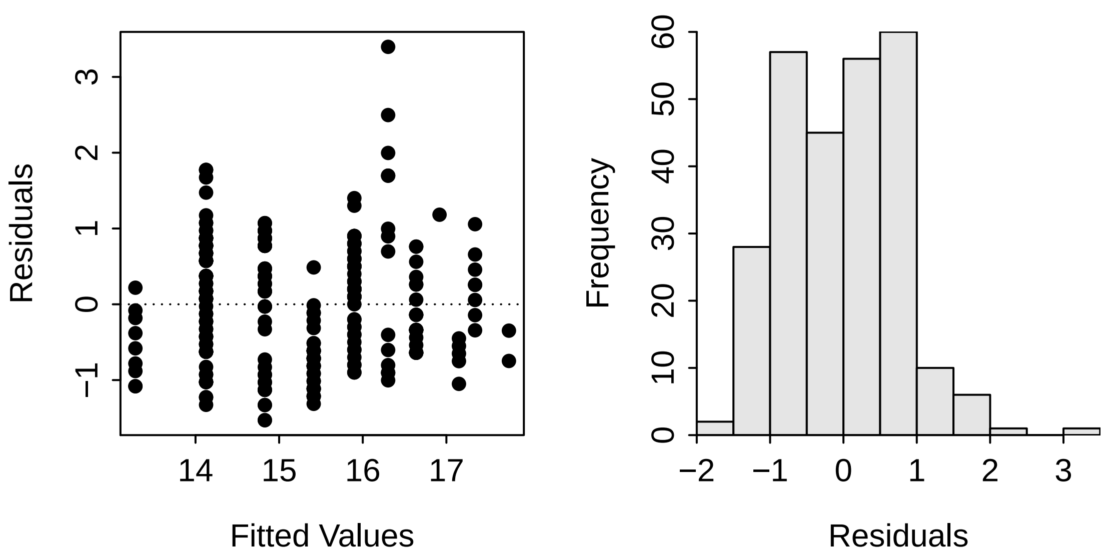
- Create a plot of length versus age for all male Walleye captured in 2014. Comment on whether you think there will be any "problems" with fitting the von Bertalanffy growth function (VBGF).

```
> xlbl <- "Age (yrs)"
> ylbl <- "Total Length (in)"
> clrs2 <- col2rgbt(clrs,1/10)
> plot(len~age,data=wae14M,pch=19,col=clrs2[2],xlab=xlbl,ylab=ylbl)
```



- Fit the VBGF for male Walleye captured in 2014. Comment on assumptions from the residual plot.

```
> vb <- vbFuns("Typical")
> wae14M.vbs <- vbStarts(len~age,data=wae14M,type="Typical")
> wae14M.vbf <- nls(len~vb(age,Linf,K,t0),data=wae14M,
+                   start=wae14M.vbs)
> residPlot(wae14M.vbf)
```



- Comment on the correlations among parameter estimates.

```
> summary(wae14M.vbf,correlation=TRUE)
```

```
Formula: len ~ vb(age, Linf, K, t0)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
Linf 18.29457    0.57973  31.557  < 2e-16
```

```
K      0.18464    0.03988   4.630 5.76e-06
t0    -4.01077    1.21378  -3.304  0.00108

Residual standard error: 0.7926 on 263 degrees of freedom

Correlation of Parameter Estimates:
   Linf  K
K  -0.97
t0 -0.91  0.98

Number of iterations to convergence: 7
Achieved convergence tolerance: 9.67e-07
```

- Construct profile likelihood confidence intervals for each parameter.

```
> ( wae14M.vbc <- coef(wae14M.vbf) )
```

```
      Linf          K         t0
18.2945733  0.1846424 -4.0107704
```

```
> cbind(Est=wae14M.vbc,confint(wae14M.vbf))
```

```
Waiting for profiling to be done...

            Est       2.5%      97.5%
Linf 18.2945733 17.4357491 19.9345415
K     0.1846424  0.1132037  0.2673795
t0   -4.0107704 -7.0628076 -2.1186219
```

- Construct bootstrapped confidence intervals for each parameter.

```
> wae14M.vbb <- nlsBoot(wae14M.vbf,niter=999)
> cbind(EST=wae14M.vbc,confint(wae14M.vbb))
```

```
            EST    95% LCI    95% UCI
Linf 18.2945733 17.4539440 20.2826885
K     0.1846424  0.1051315  0.2671615
t0   -4.0107704 -7.5429452 -2.1355998
```

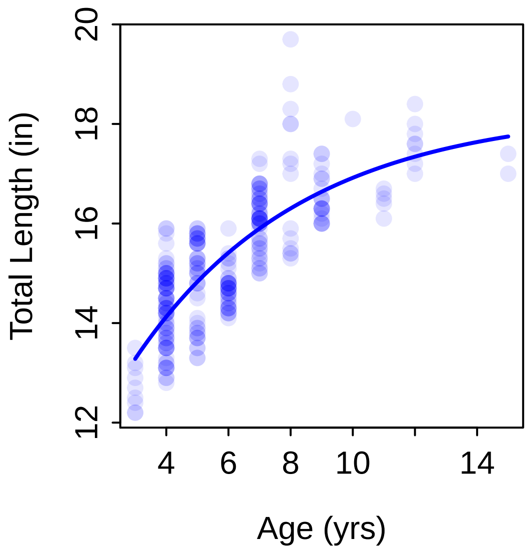
- Predict, with a bootstrapped confidence interval, the mean length for a chosen age (you choose the age).

```
> ageX <- 9
> wae14M.vbbp <- apply(wae14M.vbb$coefboot,MARGIN=1,FUN=vb,t=ageX)
> c(pred=predict(wae14M.vbf,data.frame(age=ageX)),
+   quantile(wae14M.vbbp,c(0.025,0.975)))
```

```
    pred     2.5%    97.5%
16.63881 16.48740 16.79173
```

- Plot the best-fit VBGF over the observed data.

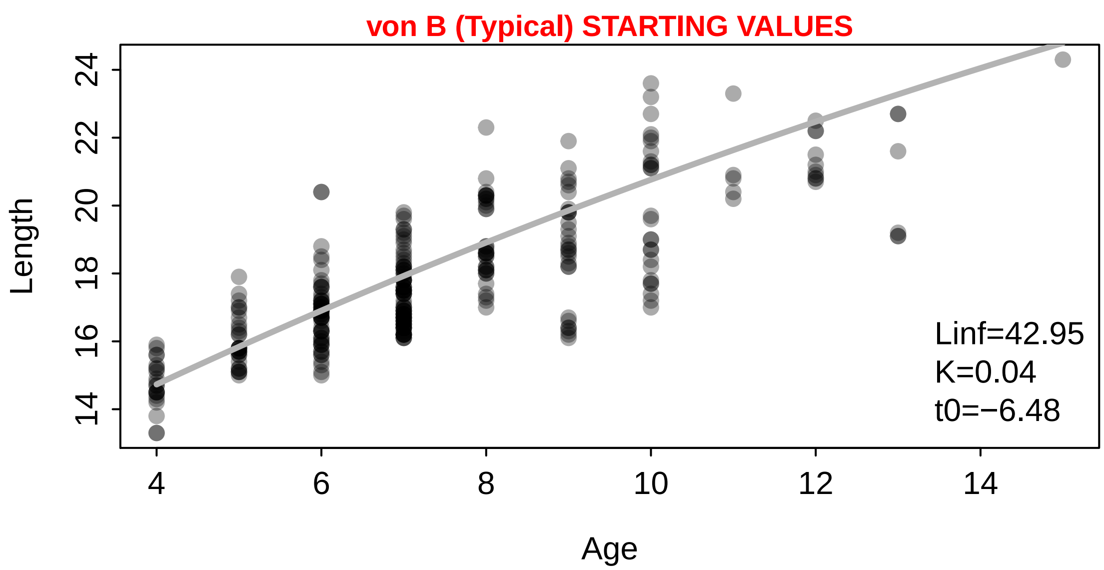
```
> plot(len~age,data=wae14M,xlab=xlbl,ylab=ylbl,
+      pch=19,col=clrs2[2])
> curve(vb(x,wae14M.vbc),from=3,to=15,n=500,
+       lwd=2,col=clrs[2],add=TRUE)
```



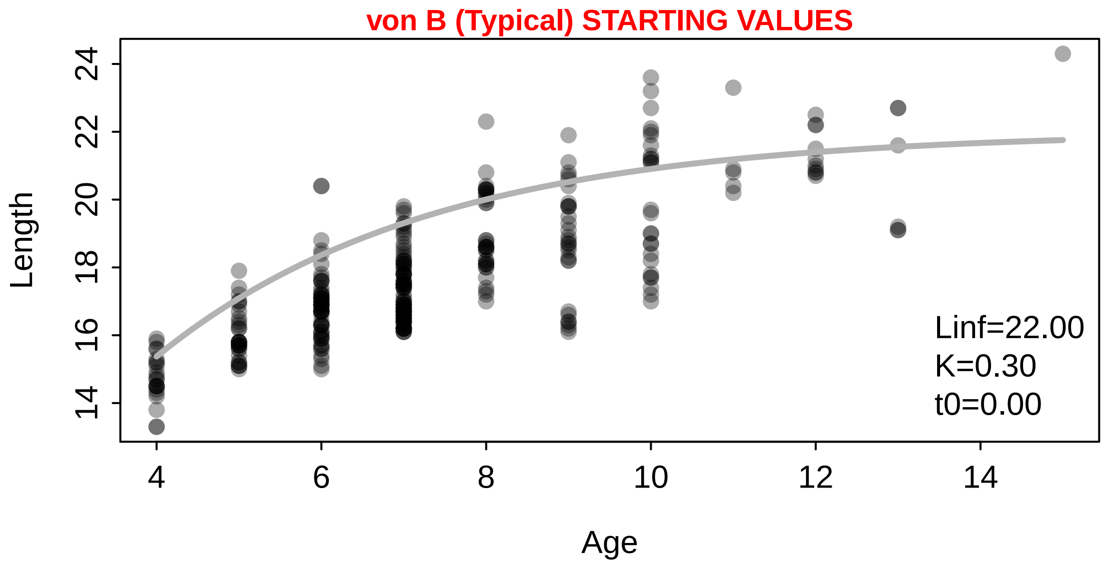
- Repeat the above for female Walleye captured in 2014.

```
> wae14F.vbs <- vbStarts(len~age,data=wae14F,type="Typical",plot=TRUE)
```

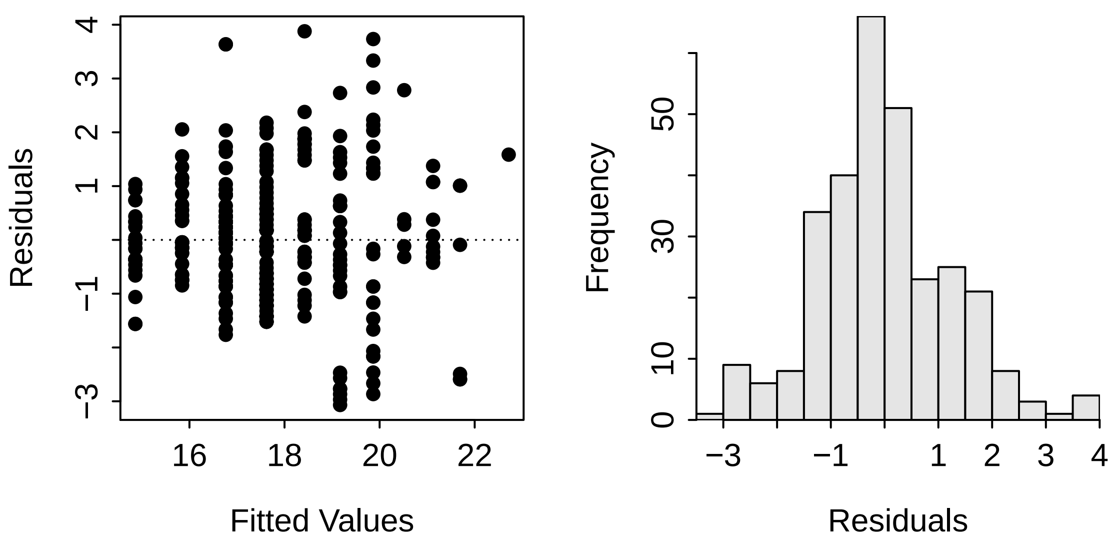
```
Warning: Starting value for Linf is very different from the observed maximum
length, which suggests a model fitting problem. See a Walford or
Chapman plot to examine the problem. Consider either using the mean
length for several of the largest fish (i.e., use 'oldAge' in
'methLinf=') or manually setting Linf in the starting value list
to the maximum observed length.
```




```
> wae14F.vbs <- vbStarts(len~age,data=wae14F,type="Typical",
+                        fixed=list(Linf=22,K=0.3,t0=0),plot=TRUE)
```




```
> wae14F.vbf <- nls(len~vb(age,Linf,K,t0),data=wae14F,
+                  start=wae14F.vbs)
> residPlot(wae14F.vbf)
```




```
> summary(wae14F.vbf,correlation=TRUE)
```

```
Formula: len ~ vb(age, Linf, K, t0)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
Linf 29.64217    5.19201   5.709 2.75e-08
K     0.06889    0.03137   2.196  0.02884
t0   -6.10133    2.22913  -2.737  0.00657

Residual standard error: 1.24 on 297 degrees of freedom
```

```
Correlation of Parameter Estimates:
   Linf  K
K  -0.99
t0 -0.96  0.99

Number of iterations to convergence: 9
Achieved convergence tolerance: 7.024e-08
```

```
> ( wae14F.vbc <- coef(wae14F.vbf) )
```

```
       Linf           K          t0
29.64216578  0.06889361 -6.10133125
```

```
> wae14F.vbb <- nlsBoot(wae14F.vbf,niter=999)
```

```
Warning in nlsBoot(wae14F.vbf, niter = 999): The fit did not converge 69
times during bootstrapping
```

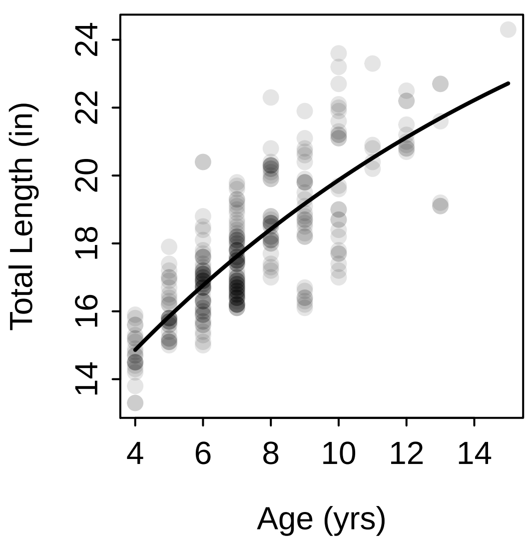
```
> cbind(Est=wae14F.vbc,confint(wae14F.vbb))
```

```
             Est      95% LCI    95% UCI
Linf 29.64216578  24.01484778 44.8885225
K     0.06889361   0.02892965  0.1337408
t0   -6.10133125 -10.26991059 -3.0009723
```

```
> wae14F.vbbp <- apply(wae14F.vbb$coefboot,MARGIN=1,FUN=vb,t=ageX)
> c(pred=predict(wae14F.vbf,data.frame(age=ageX)),
+   quantile(wae14F.vbbp,c(0.025,0.975)))
```

```
    pred     2.5%    97.5%
19.16904 18.98578 19.36571
```

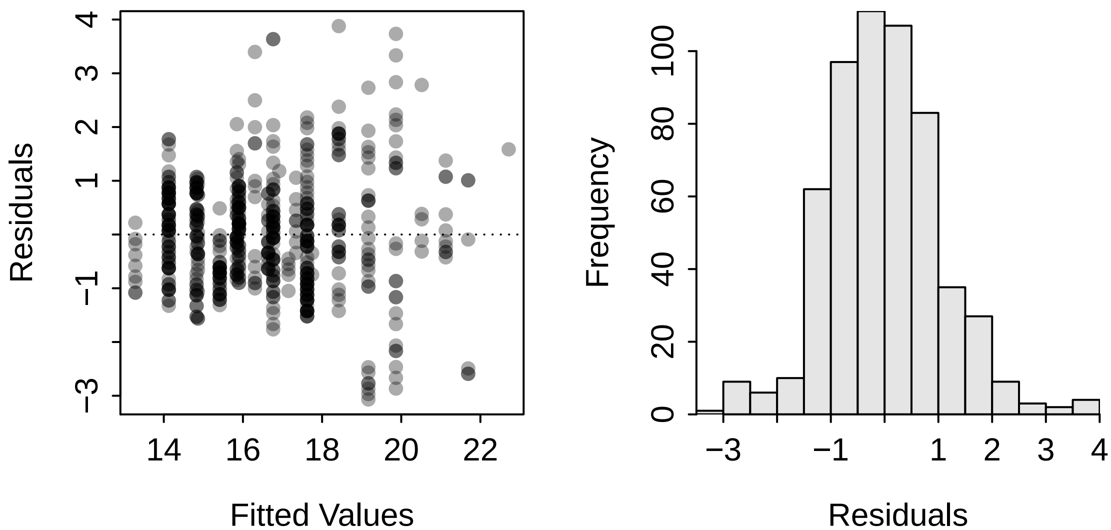
```
> plot(len~age,data=wae14F,xlab=xlbl,ylab=ylbl,
+      pch=19,col=clrs2[1])
> curve(vb(x,wae14F.vbc),from=4,to=15,n=500,
+       lwd=2,col=clrs[1],add=TRUE)
```



7 Compare Growth Model Parameters

- Fit the ulimated full model to all male and female Walleye captured in 2014. Visually assess the assumptions.

```
> svOm <- vbStarts(len~age,data=wae14)
> svLKt <- Map(rep,svOm,c(2,2,2))
> vbLKt <- len~Linf[sex]*(1-exp(-K[sex]*(age-t0[sex])))
> fitLKt <- nls(vbLKt,data=wae14,start=svLKt)
> residPlot(fitLKt,col=col2rgbt("black",1/3))
```



- Fit the ultimate simple model and statistically compare it to the ultimate full model to determine if at least some of the VBGF parameters differ.

```
> vbOm <- len~Linf*(1-exp(-K*(age-t0)))
> fitOm <- nls(vbOm,data=wae14,start=svOm)
> extraSS(fitOm,sim.name="{Omega}",
+         com=fitLKt,com.name="{Linf,K,t0}")

Model 1: {Omega}
Model A: {Linf,K,t0}

    DfO    RSSO DfA    RSSA Df      SS      F    Pr(>F)
1vA 563 1071.12 560  622.09  3  449.03 134.74 < 2.2e-16
```

```
> lrt(fitOm,sim.name="{Omega}",
+     com=fitLKt,com.name="{Linf,K,t0}")

Loading required namespace: lmtest

Model 1: {Omega}
Model A: {Linf,K,t0}

    DfO logLikO DfA logLikA Df  logLik  Chisq Pr(>Chisq)
1vA 563 -983.63 560 -829.86  3 -153.78 307.55  < 2.2e-16
```

- Use model reduction methods to find the most parsimonious model. Interpret from this model which parameters differ between the sexes.

```
> vbLK <- len~Linf[sex]*(1-exp(-K[sex]*(age-t0)))
> svLK <- Map(rep,svOm,c(2,2,1))
> fitLK <- nls(vbLK,data=wae14,start=svLK)
> vbLt <- len~Linf[sex]*(1-exp(-K*(age-t0[sex])))
> svLt <- Map(rep,svOm,c(2,1,2))
> fitLt <- nls(vbLt,data=wae14,start=svLt)
> vbKt <- len~Linf*(1-exp(-K[sex]*(age-t0[sex])))
> svKt <- Map(rep,svOm,c(1,2,2))
> fitKt <- nls(vbKt,data=wae14,start=svKt)
> extraSS(fitLK,fitLt,fitKt,
+         sim.names=c("{Linf,K}","{Linf,t0}","{K,t0}"),
+         com=fitLKt,com.name="{Linf,K,t0}")
```

```
Model 1: {Linf,K}
Model 2: {Linf,t0}
Model 3: {K,t0}
Model A: {Linf,K,t0}

    DfO     RSSO DfA     RSSA Df      SS       F   Pr(>F)
1vA 561 622.8911 560 622.0900  1  0.8011  0.7211 0.396133
2vA 561 627.0491 560 622.0900  1  4.9591  4.4641 0.035056
3vA 561 633.8964 560 622.0900  1 11.8064 10.6280 0.001182
```

```
> vbL <- len~Linf[sex]*(1-exp(-K*(age-t0)))
> ( svL <- Map(rep,svOm,c(2,1,1)) )
```

```
$Linf
[1] 19.74559 19.74559

$K
[1] 0.2980978

$t0
[1] -0.4782422
```

```
> fitL <- nls(vbL,data=wae14,start=svL)
> vbK <- len~Linf*(1-exp(-K[sex]*(age-t0)))
> svK <- Map(rep,svOm,c(1,2,1))
> fitK <- nls(vbK,data=wae14,start=svK,trace=TRUE)
```

```
1229.69 :  19.7455897  0.2980978  0.2980978 -0.4782422
945.4668 :  20.1804884  0.2338042  0.2133049 -1.4285985
935.7601 :  20.9067017  0.1783919  0.1560752 -2.5334872
926.4284 :  21.4209380  0.1569097  0.1363291 -3.1455223
923.8196 :  22.0010534  0.1379553  0.1193259 -3.7828538
910.465 :  22.3283253  0.1296014  0.1119662 -4.1130337
898.6174 :  22.6740284  0.1217393  0.1050843 -4.4483261
888.149 :  23.03950952  0.11434126  0.09864397 -4.78837451
878.9448 :  23.42621716  0.10738071  0.09261261 -5.13280809
870.9024 :  23.83571013  0.10083225  0.08696036 -5.48124231
863.93 :  24.26966490  0.09467179  0.08166004 -5.83327661
857.9461 :  24.72988594  0.08887650  0.07668682 -6.18849523
852.8786 :  25.21831532  0.08342484  0.07201803 -6.54646616
848.6635 :  25.73704203  0.07829652  0.06763293 -6.90674056
845.2449 :  26.28831466  0.07347243  0.06351254 -7.26885351
```

```
842.5737 :  26.87455400  0.06893463  0.05963944 -7.63232500
840.6073 :  27.49836568  0.06466623  0.05599763 -7.99666020
839.3087 :  28.16255140  0.06065144  0.05257239 -8.36134885
838.6462 :  28.87012783  0.05687540  0.04935016 -8.72586964
838.5931 :  29.62434190  0.05332418  0.04631840 -9.08969079
834.2069 :  30.02651196  0.05165444  0.04489198 -9.27098042
830.0807 :  30.44136845  0.05003478  0.04350784 -9.45179744
826.2011 :  30.86938280  0.04846373  0.04216469 -9.63207591
822.5558 :  31.31103907  0.04693987  0.04086132 -9.81174824
819.1329 :  31.76684343  0.04546184  0.03959654 -9.99074889
815.9215 :   32.23731546   0.04402831   0.03836921 -10.16901082
812.9111 :   32.72299693   0.04263798   0.03717823 -10.34646870
810.0919 :   33.22444655   0.04128960   0.03602253 -10.52305679
807.4548 :   33.74224042   0.03998196   0.03490109 -10.69870907
804.991 :   34.27698039   0.03871387   0.03381291 -10.87336195
802.6923 :   34.82928361   0.03748420   0.03275703 -11.04695066
800.5509 :   35.39979140   0.03629184   0.03173252 -11.21941208
798.5597 :   35.98916505   0.03513571   0.03073849 -11.39068325
796.7117 :   36.59808818   0.03401478   0.02977406 -11.56070214
795.0004 :   37.22727550   0.03292802   0.02883839 -11.72940976
793.4198 :   37.87745385   0.03187447   0.02793066 -11.89674515
791.964 :   38.54938639   0.03085314   0.02705009 -12.06265145
790.6276 :   39.24385895   0.02986313   0.02619589 -12.22707233
789.4054 :   39.96167082   0.02890353   0.02536735 -12.38995015
788.2925 :   40.70366850   0.02797347   0.02456372 -12.55123319
787.2844 :   41.47071714   0.02707209   0.02378431 -12.71086936
786.3766 :   42.26370959   0.02619857   0.02302843 -12.86880809
785.5648 :   43.08356499   0.02535211   0.02229543 -13.02500013
784.8452 :   43.93123640   0.02453193   0.02158467 -13.17939888
784.2139 :   44.80770156   0.02373728   0.02089553 -13.33195877
783.6673 :   45.71397928   0.02296742   0.02022740 -13.48263778
783.2022 :   46.65112811   0.02222162   0.01957969 -13.63139711
782.8149 :   47.62021679   0.02149919   0.01895182 -13.77819658
782.5027 :   48.62238732   0.02079944   0.01834323 -13.92300379
782.2621 :   49.65874326   0.02012174   0.01775339 -14.06577875
782.0906 :   50.73051664   0.01946543   0.01718177 -14.20649601

Error in nls(vbK, data = wae14, start = svK, trace = TRUE): number of iterations exceeded maximum of 50
```

```
> extraSS(fitL,sim.names=c("{Linf}"),
+         com=fitLK,com.name="{Linf,K}")
```

```
Model 1: {Linf}
Model A: {Linf,K}

    DfO   RSSO DfA   RSSA Df     SS     F    Pr(>F)
1vA 562 683.58 561 622.89  1  60.69 54.66 5.242e-13
```

```
> vbt <- len~Linf*(1-exp(-K*(age-t0[sex])))
> svt <- Map(rep,svOm,c(1,1,2))
> fitt <- nls(vbt,data=wae14,start=svt,trace=TRUE)
```

```
1229.69 :  19.7455897  0.2980978 -0.4782422 -0.4782422
1029.133 :  20.2269703  0.2196821 -1.6388506 -1.3098412
993.8685 :  20.6122695  0.1913749 -2.2935842 -1.7978244
971.0064 :  21.0659482  0.1664461 -2.9888158 -2.3274716
```

```
959.8716 :  21.599396  0.144470 -3.724142 -2.900053
947.5562 :  21.913643  0.134764 -4.112013 -3.208635
936.7345 :  22.2531504  0.1256021 -4.5102105 -3.5287345
927.4081 :  22.6207573  0.1169519 -4.9188180 -3.8605523
919.5943 :  23.0197544  0.1087824 -5.3379555 -4.2043099
913.3312 :  23.4539780  0.1010641 -5.7677789 -4.5602481
908.6831 :  23.92792671  0.09376929 -6.20848288 -4.92863073
905.7478 :  24.4469060  0.0868717 -6.6602978 -5.3097439
904.667 :  25.01721880  0.08034673 -7.12349324 -5.70390004
899.6869 :  25.331815  0.077259 -7.360936 -5.907670
895.0953 :  25.66218659  0.07425363 -7.60127749 -6.11477039
890.8922 :  26.00953721  0.07132816 -7.84455445 -6.32524077
887.0793 :  26.3751917  0.0684802 -8.0908059 -6.5391219
883.6613 :  26.76061582  0.06570742 -8.34007401 -6.75645733
880.6454 :  27.16743275  0.06300758 -8.59240274 -6.97729177
878.042 :  27.59744547  0.06037849 -8.84783829 -7.20167174
875.8652 :  28.05266401  0.05781804 -9.10643002 -7.42964642
874.1335 :  28.53533498  0.05532417 -9.36822979 -7.66126705
872.8703 :  29.04797792  0.05289489 -9.63329245 -7.89658743
872.1051 :  29.59342996  0.05052827 -9.90167658 -8.13566463
871.8744 :   30.17489532   0.04822243 -10.17344380  -8.37855842
869.6911 :   30.48545060   0.04709899 -10.31105116  -8.50194469
867.6187 :   30.80649608   0.04599004 -10.44950104  -8.62628307
865.6579 :   31.13857050   0.04489538 -10.58880071  -8.75158053
863.8098 :   31.48225062   0.04381479 -10.72895803  -8.87784455
862.0755 :   31.83815263   0.04274806 -10.86998054  -9.00508236
860.4567 :   32.206938   0.041695 -11.011876  -9.133302
858.9552 :   32.58931528   0.04065538 -11.15465377  -9.26251035
857.5736 :   32.98604740   0.03962903 -11.29832128  -9.39271652
856.3145 :   33.39795321   0.03861574 -11.44288739  -9.52392825
855.1812 :   33.82591766   0.03761531 -11.58836162  -9.65615457
854.1776 :   34.27089343   0.03662756 -11.73475312  -9.78940418
853.3081 :   34.7339082   0.0356523 -11.8820708  -9.9236855
852.5779 :   35.21607759   0.03468935 -12.03032483 -10.05900824
851.9927 :   35.71860864   0.03373852 -12.17952515 -10.19538178
851.5594 :   36.24281351   0.03279962 -12.32968244 -10.33281625
851.2856 :   36.79011830   0.03187249 -12.48080739 -10.47132179
851.1802 :   37.36207582   0.03095695 -12.63291077 -10.61090862
850.1848 :   37.66122984   0.03050488 -12.70945757 -10.68124827
849.2215 :   37.96725915   0.03005568 -12.78624509 -10.75185470
848.2907 :   38.28040187   0.02960931 -12.86327432 -10.82272884
847.3925 :   38.60090960   0.02916576 -12.94054679 -10.89387213
846.5275 :   38.929044   0.028725 -13.018064 -10.965286
845.6961 :   39.26508255   0.02828702 -13.09582664 -11.03697105
844.8987 :   39.60931203   0.02785179 -13.17383685 -11.10892937
844.1357 :   39.96203498   0.02741929 -13.25209557 -11.18116181
843.4079 :   40.32356825   0.02698951 -13.33060403 -11.25366954
```

```
Error in nls(vbt, data = wae14, start = svt, trace = TRUE): number of iterations exceeded maximum of 50
```

- Use AIC to identify the most supported model(s).

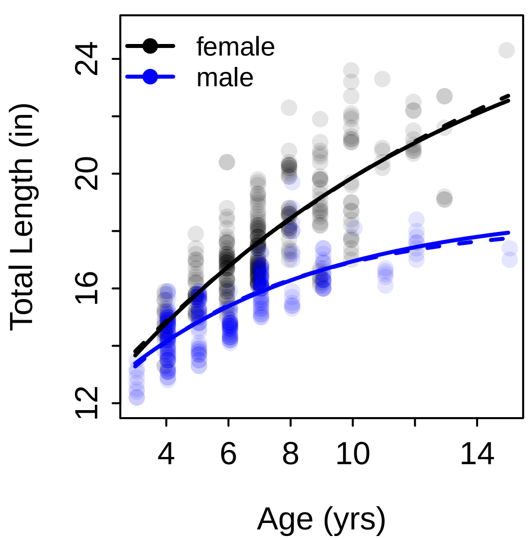
```
> ms <- list(fitOm,fitL,fitLK,fitLt,fitKt,fitLKt)
> mnames <- c("{Omega}","{Linf}","{Linf,K}",
+             "{Linf,t0}","{K,t0}","{Linf,K,t0}")
```

```
> aictab(ms,mnames)


Model selection based on AICc:

            K    AICc Delta_AICc AICcWt Cum.Wt      LL
{Linf,K}    6 1672.60       0.00   0.60    0.6 -830.22
{Linf,K,t0} 7 1673.92       1.32   0.31    0.9 -829.86
{Linf,t0}   6 1676.36       3.77   0.09    1.0 -832.11
{K,t0}      6 1682.51       9.91   0.00    1.0 -835.18
{Linf}      5 1723.18      50.58   0.00    1.0 -856.54
{Omega}     4 1975.34     302.74   0.00    1.0 -983.63
```

- Plot the best-fit VBGFs (according to the most parsimonious model or most supported models).

```
> ( cfLK <- coef(fitLK) )

      Linf1       Linf2          K1          K2          t0
27.73215835 18.81689943  0.08311313  0.15217990 -5.16650196
```

```
> jit <- 0.05
> plot(len~I(age-jit),data=filterD(wae14,sex=="female"),
+      pch=19,col=clrs2[1],xlab=xlbl,ylab=ylbl,
+      ylim=c(12,25),xlim=c(3,15))
> points(len~I(age+jit),data=filterD(wae14,sex=="male"),
+        pch=19,col=clrs2[2])
> curve(vb(x,cfLK[c("Linf1","K1","t0")]),from=3,to=15,
+       add=TRUE,col=clrs[1],lwd=2)
> curve(vb(x,cfLK[c("Linf2","K2","t0")]),from=3,to=15,
+       add=TRUE,col=clrs[2],lwd=2)
> legend("topleft",levels(wae14$sex),col=clrs,pch=19,
+        lwd=2,bty="n",cex=0.8)
> cfLKt <- coef(fitLKt)
> curve(vb(x,cfLKt[c("Linf1","K1","t01")]),from=3,to=15,
+       add=TRUE,col=clrs[1],lwd=2,lty=2)
> curve(vb(x,cfLKt[c("Linf2","K2","t02")]),from=3,to=15,
+       add=TRUE,col=clrs[2],lwd=2,lty=2)
```



8 Compute Weight-Length Relationship

- Fit the weight-length relationship to male Walleye captured in 2014.

```
> wae14M.lw <- lm(logwt~loglen,data=wae14M)
> cbind(Est=coef(wae14M.lw),confint(wae14M.lw))
```

```
                  Est      2.5 %    97.5 %
(Intercept) -8.353681 -8.755571 -7.951791
loglen       3.082247  2.934621  3.229873
```

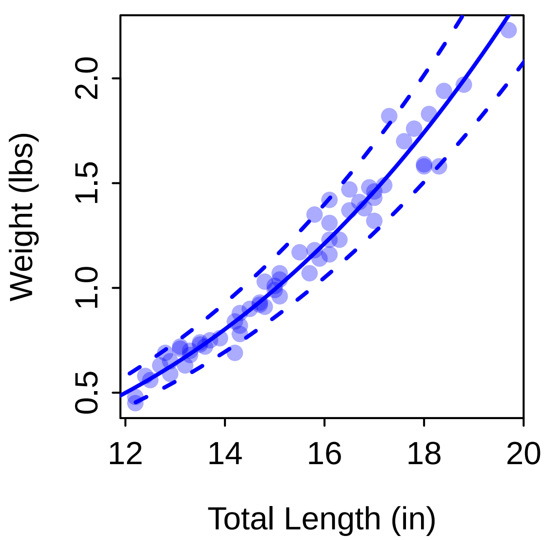
- Predict the weight for a fish with a chosen length (you choose the length).

```
> chLen <- 20
> exp(predict(wae14M.lw,data.frame(loglen=log(chLen)),
+             interval="prediction"))
```

```
       fit     lwr      upr
1 2.410674 2.07612 2.799139
```

- Construct a plot (with a prediction band) that demonstrates the model fit.

```
> clrs2 <- col2rgbt(clrs,1/3)
> lens <- seq(8,25,length.out=100)
> wae14M.pwt <- exp(predict(wae14M.lw,data.frame(loglen=log(lens)),
+                            interval="prediction"))
> plot(wt~len,data=wae14M,pch=19,col=clrs2[2],
+      xlab="Total Length (in)",ylab="Weight (lbs)")
> lines(wae14M.pwt[,"fit"]~lens,lwd=2,col=clrs[2])
> lines(wae14M.pwt[,"lwr"]~lens,lwd=2,lty=2,col=clrs[2])
> lines(wae14M.pwt[,"upr"]~lens,lwd=2,lty=2,col=clrs[2])
```



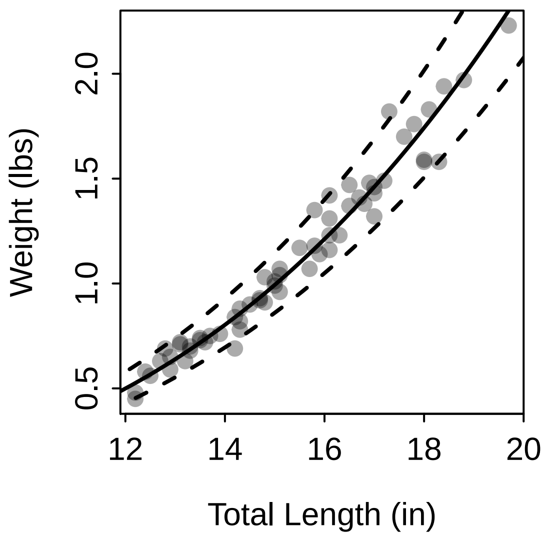
- Repeat the above analysis for female Walleye captured in 2014.

```
> wae14M.lw <- lm(logwt~loglen,data=wae14M)
> cbind(Est=coef(wae14M.lw),confint(wae14M.lw))
```

```
                  Est      2.5 %    97.5 %
(Intercept) -8.353681 -8.755571 -7.951791
```

```
loglen       3.082247  2.934621  3.229873
```

```
> wae14M.pwt <- exp(predict(wae14M.lw,data.frame(loglen=log(lens)),
+                            interval="prediction"))
> plot(wt~len,data=wae14M,pch=19,col=clrs2[1],
+      xlab="Total Length (in)",ylab="Weight (lbs)")
> lines(wae14M.pwt[,"fit"]~lens,lwd=2,col=clrs[1])
> lines(wae14M.pwt[,"lwr"]~lens,lwd=2,lty=2,col=clrs[1])
> lines(wae14M.pwt[,"upr"]~lens,lwd=2,lty=2,col=clrs[1])
```



9 Compare Weight-Length Model Parameters

- Statistically compare the weight-length relationships between male and female Walleye captured in 2014.

```
> ALL.lw <- lm(logwt~loglen*sex,data=wae14)
> anova(ALL.lw)
```

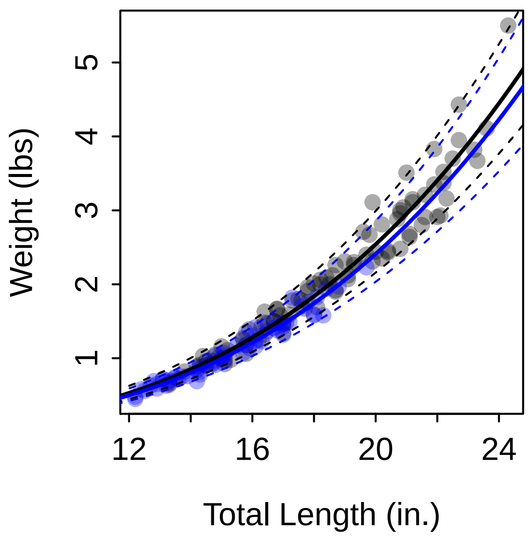
```
Analysis of Variance Table

Response: logwt
            Df Sum Sq Mean Sq   F value    Pr(>F)
loglen       1 43.775  43.775 6500.8302 < 2.2e-16
sex          1  0.078   0.078   11.6561 0.0008219
loglen:sex   1  0.000   0.000    0.0116 0.9145158
Residuals  151  1.017   0.007
```

```
> cbind(Est=coef(ALL.lw),confint(ALL.lw))
```

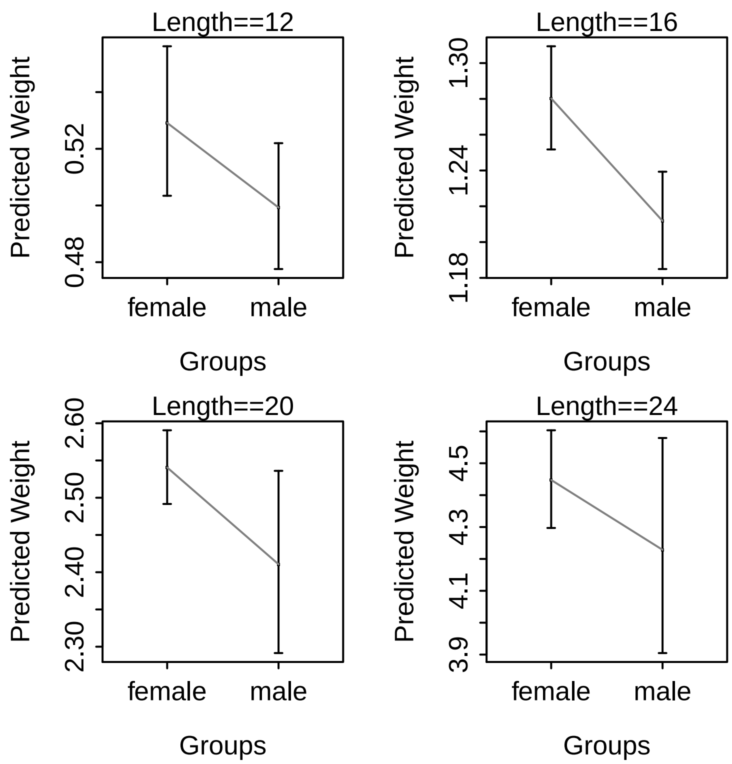
```
                        Est      2.5 %     97.5 %
(Intercept)     -8.26838395 -8.5915657 -7.9452022
loglen           3.07129518  2.9602861  3.1823043
sexmale         -0.08529665 -0.6449996  0.4744063
loglen:sexmale   0.01095185 -0.1902924  0.2121961
```

- Construct a plot that demonstrates the model fit.

```
> plot(wt~len,data=wae14,pch=19,col=clrs2[sex],
+      xlab="Total Length (in.)",ylab="Weight (lbs)")
> f.pwt <- exp(predict(ALL.lw,data.frame(loglen=log(lens),sex="female"),
+                      interval="prediction"))
> lines(f.pwt[,"fit"]~lens,lwd=2,col=clrs[1])
> lines(f.pwt[,"lwr"]~lens,lwd=1,lty=2,col=clrs[1])
> lines(f.pwt[,"upr"]~lens,lwd=1,lty=2,col=clrs[1])
> m.pwt <- exp(predict(ALL.lw,data.frame(loglen=log(lens),sex="male"),
+                      interval="prediction"))
> lines(m.pwt[,"fit"]~lens,lwd=2,col=clrs[2])
> lines(m.pwt[,"lwr"]~lens,lwd=1,lty=2,col=clrs[2])
> lines(m.pwt[,"upr"]~lens,lwd=1,lty=2,col=clrs[2])
```




```
> lwCompPreds(ALL.lw,lens=c(12,16,20,24))
```


Length==12
Length==16
Length==20
Length==24
Predicted Weight
0.48
0.52
1.18
1.24
1.30
2.30
2.40
2.50
2.60
3.9
4.1
4.3
4.5
female
male
Groups