ST516 Homework 4

Sam Wozniak

April 20, 2018

Spring Failure Data

1. Estimate an AFT model with stress as a quantitative explanatory variable and specifying a weibull distribution. Report parameter estimates, and produce plots of survival, hazard, and probability density functions. Interpret the effect of stress on the AFT model, and on the hazard function.

Multiplicative effect of stress based on the AFT model:

For each 100 unit increase in stress, the number of thousands of cycles until failure is compressed by a factor of 0.1525.

Multiplicative effect of stress on hazard function:

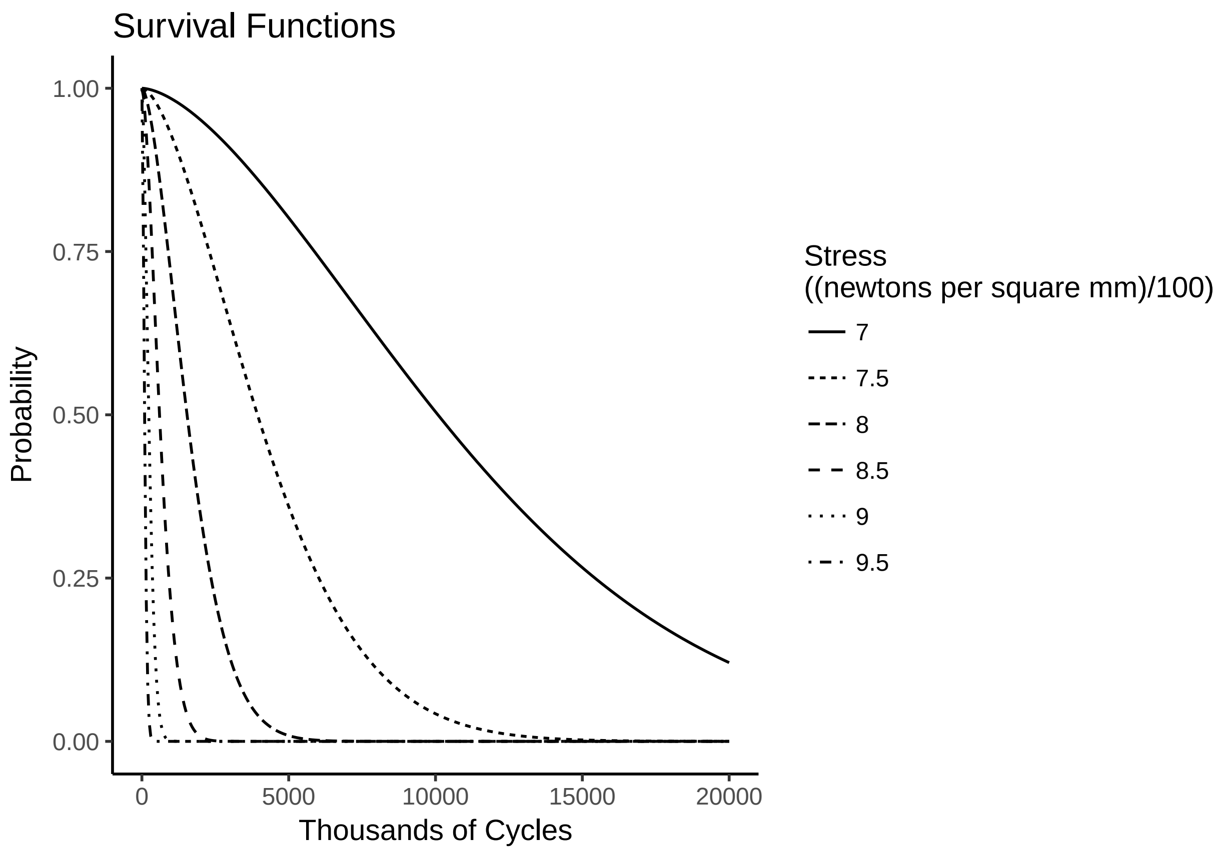
For each 100 unit increase in stress, the hazards increase by factor of 21.4.

```
springs$stress <- as.numeric(as.character(springs$stress))
springs$stressquant <- as.numeric(as.character(springs$stress))/100
m <- flexsurvreg(Surv(cycles, cens) ~ stressquant, dist = "weibull", data = springs)
print(m)
```

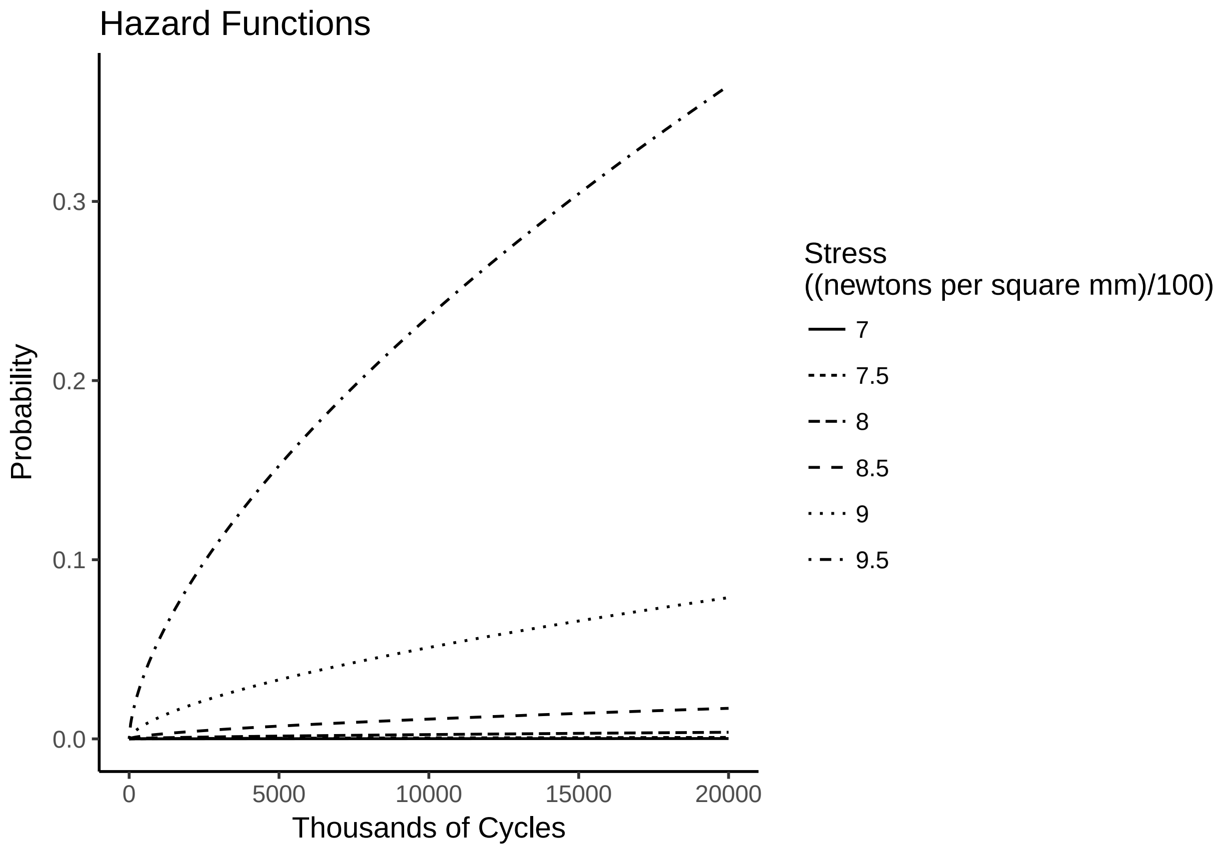
```
## Call:
## flexsurvreg(formula = Surv(cycles, cens) ~ stressquant, data = springs,
##     dist = "weibull")
##
## Estimates:
##              data mean        est              L95%
## shape                    NA           1.6290           1.3258
## scale                    NA  6569034315.1760  1592178577.5975
## stressquant          8.2500          -1.8803          -2.0480
##              U95%             se               exp(est)
## shape                2.0014           0.1711               NA
## scale        27102620548.4267  4750114387.3375               NA
## stressquant         -1.7127           0.0855           0.1525
##              L95%             U95%
## shape                    NA               NA
## scale                    NA               NA
## stressquant          0.1290           0.1804
##
## N = 60,  Events: 53,  Censored: 7
## Total time at risk: 169993
## Log-likelihood = -401.9, df = 3
## AIC = 809.8
```

```
#survival function plots
d <- data.frame(stressquant = seq(7, 9.5, by = 0.5))
d <- summary(m, newdata = d, t = seq(0, 20000, by = 50), type = "survival", tidy = TRUE)
```

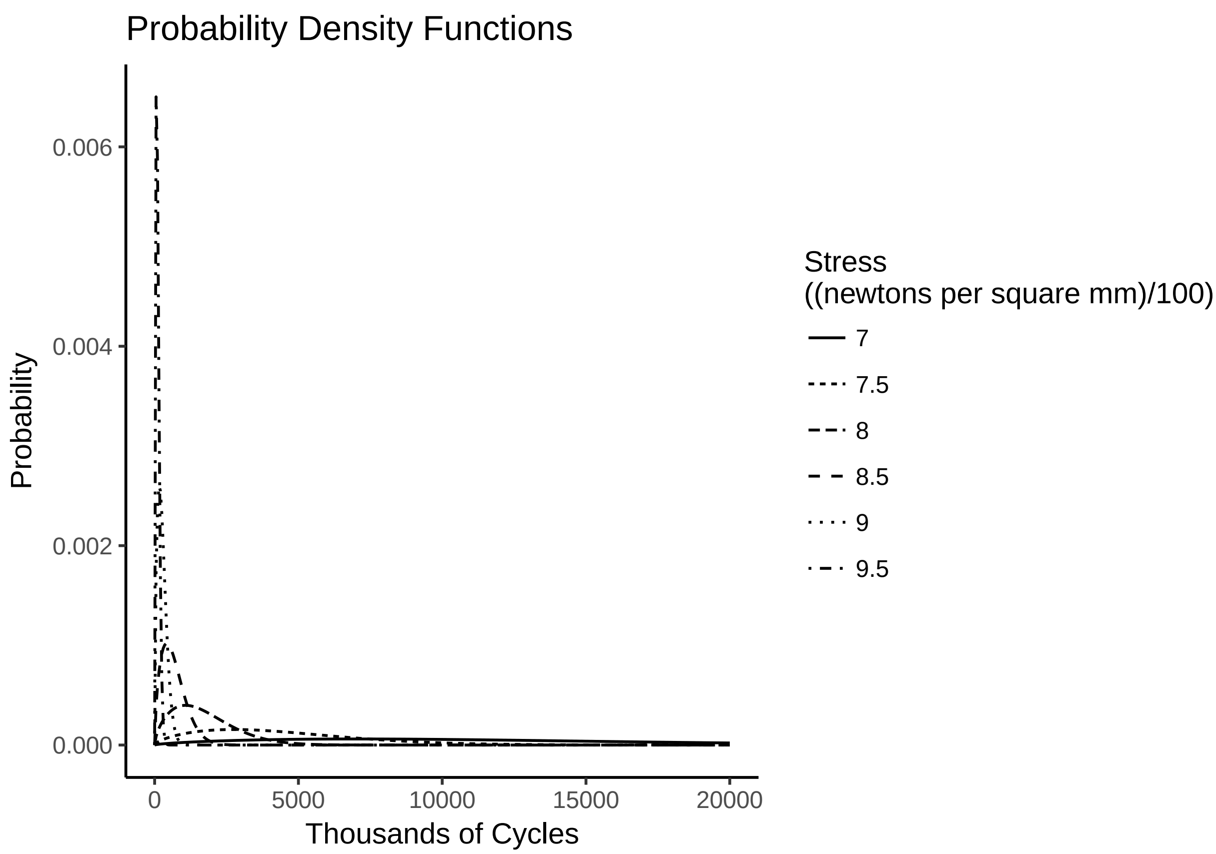
```r
p <- ggplot(d, aes(x = time, y = est))
p <- p + geom_line(aes(linetype = factor(stressquant)))
p <- p + labs(x = "Thousands of Cycles",
              y = "Probability",
              title = 'Survival Functions',
              linetype = 'Stress \n((newtons per square mm)/100)')
p <- p + theme(legend.position = 'none')
p <- p + theme_classic()
plot(p)
```




```r
#plot hazard function
d <- data.frame(stressquant = seq(7, 9.5, by = 0.5))
d <- summary(m, newdata = d, t = seq(0, 20000, by = 50), type = "hazard", tidy = TRUE)

p <- ggplot(d, aes(x = time, y = est))
p <- p + geom_line(aes(linetype = factor(stressquant)))
p <- p + labs(x = "Thousands of Cycles",
              y = "Probability",
              title = 'Hazard Functions',
              linetype = 'Stress \n((newtons per square mm)/100)')
p <- p + theme_classic()
plot(p)
```




```r
#plot probability density
d <- data.frame(stressquant = seq(7, 9.5, by = 0.5))
d <- summary(m, newdata = d,
            t = seq(0, 20000, by = 50),
            fn = function(t, ...) dweibull(t, ...), tidy = TRUE)

p <- ggplot(d, aes(x = time, y = est))
p <- p + geom_line(aes(linetype = factor(stressquant)))
p <- p + labs(x = "Thousands of Cycles",
             y = "Probability",
             title = 'Probability Density Functions',
             linetype = 'Stress \n((newtons per square mm)/100)')
p <- p + theme_classic()
plot(p)
```




```r
#estimate the expected number of thousands of cycles for each of these six stress levels
d <- data.frame(stressquant = seq(7, 9.5, by = 0.5))
summary(m, newdata = d, type = "mean", tidy = TRUE)
```

```
##       est     lcl     ucl stressquant
## 1 11302.0 8459.32 15119.1         7.0
## 2  4414.2 3530.96  5480.4         7.5
## 3  1724.0 1431.97  2048.8         8.0
## 4   673.4  570.01   801.8         8.5
## 5   263.0  214.62   319.8         9.0
## 6   102.7   81.23   130.4         9.5
```

```r
#estimate hazards ratio
m <- flexsurvreg(Surv(cycles, cens) ~ stressquant, data = springs, dist = "weibullPH")
print(m)
```

```
## Call:
## flexsurvreg(formula = Surv(cycles, cens) ~ stressquant, data = springs,
##     dist = "weibullPH")
##
## Estimates:
##              data mean  est       L95%      U95%      se        exp(est)
## shape              NA   1.63e+00  1.33e+00  2.00e+00  1.71e-01        NA
## scale              NA   1.02e-16  3.93e-20  2.63e-13  4.08e-16        NA
## stressquant  8.25e+00   3.06e+00  2.39e+00  3.74e+00  3.43e-01  2.14e+01
##              L95%      U95%
## shape              NA        NA
```

```
## scale              NA        NA
## stressquant  1.09e+01  4.19e+01
##
## N = 60,  Events: 53,  Censored: 7
## Total time at risk: 169993
## Log-likelihood = -401.9, df = 3
## AIC = 809.8
```

2. Repeat previous problem but with stress as a factor. Interpret the multiplicative effect of stress in the AFT model by comparing the five higher stress levels with the lowest level. Repeat with the interpretation of the hazard function.

Multiplicative effect of stress based on the AFT model:

Compared to 700 newtons per mm^2 of stress, 750 newtons per mm^2 compresses the number of thousands of cycles until failure by a factor of 0.498.

Compared to 700 newtons per mm^2 of stress, 800 newtons per mm^2 compresses the number of thousands of cycles until failure by a factor of 0.069

Compared to 700 newtons per mm^2 of stress, 850 newtons per mm^2 compresses the number of thousands of cycles until failure by a factor of 0.024.

Compared to 700 newtons per mm^2 of stress, 900 newtons per mm^2 compresses the number of thousands of cycles until failure by a factor of 0.015.

Compared to 700 newtons per mm^2 of stress, 950 newtons per mm^2 compresses the number of thousands of cycles until failure by a factor of 0.012.

Multiplicative effect of stress on hazard function:

Compared to 700 newtons per mm^2 of stress, 750 newtons per mm^2 increases the hazards by a factor of 5.91.

Compared to 700 newtons per mm^2 of stress, 750 newtons per mm^2 increases the hazards by a factor of 923.

Compared to 700 newtons per mm^2 of stress, 750 newtons per mm^2 increases the hazards by a factor of 14300.

Compared to 700 newtons per mm^2 of stress, 750 newtons per mm^2 increases the hazards by a factor of 47800.

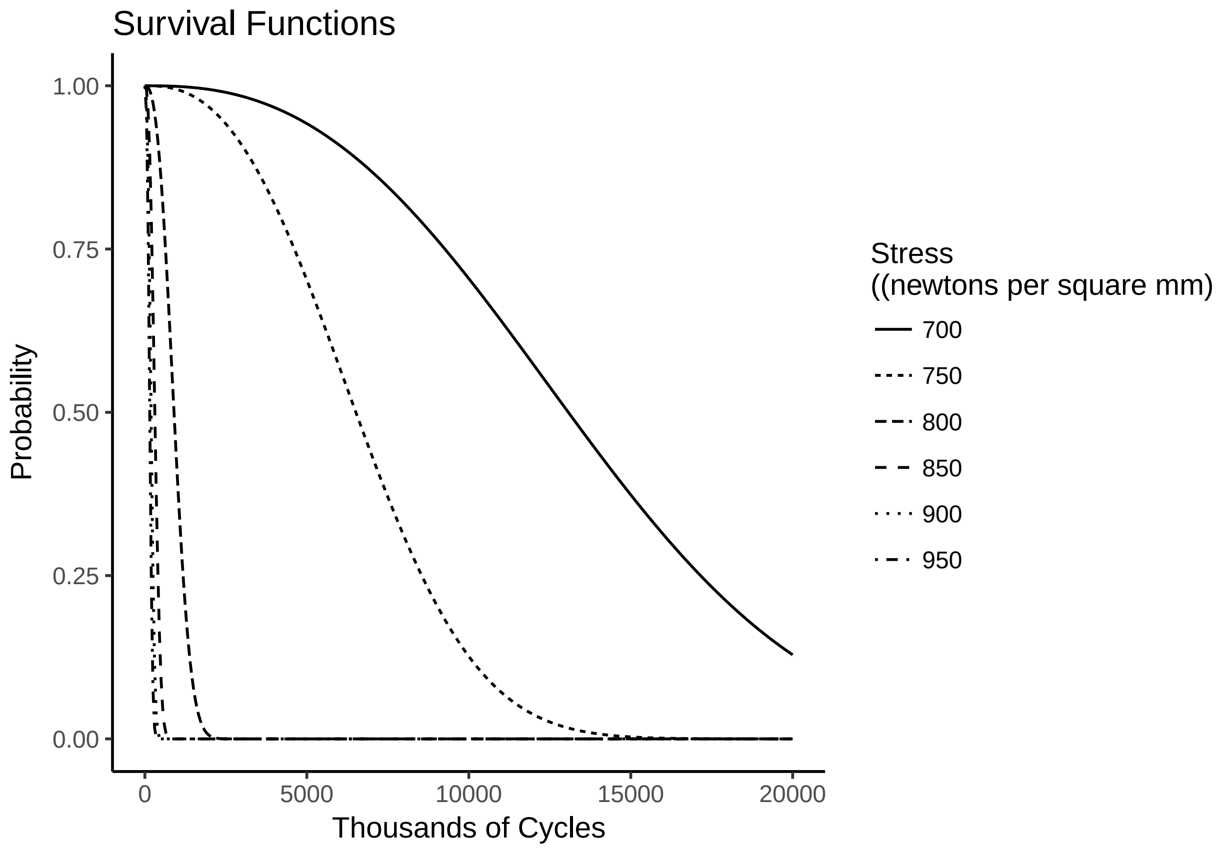
Compared to 700 newtons per mm^2 of stress, 750 newtons per mm^2 increases the hazards by a factor of 89600.

```r
springs$stress <- factor(springs$stress,
                         levels = c('700', '750', '800', '850', '900', '950'))
m <- flexsurvreg(Surv(cycles, cens) ~ stress, dist = "weibull", data = springs)
print(m)
```

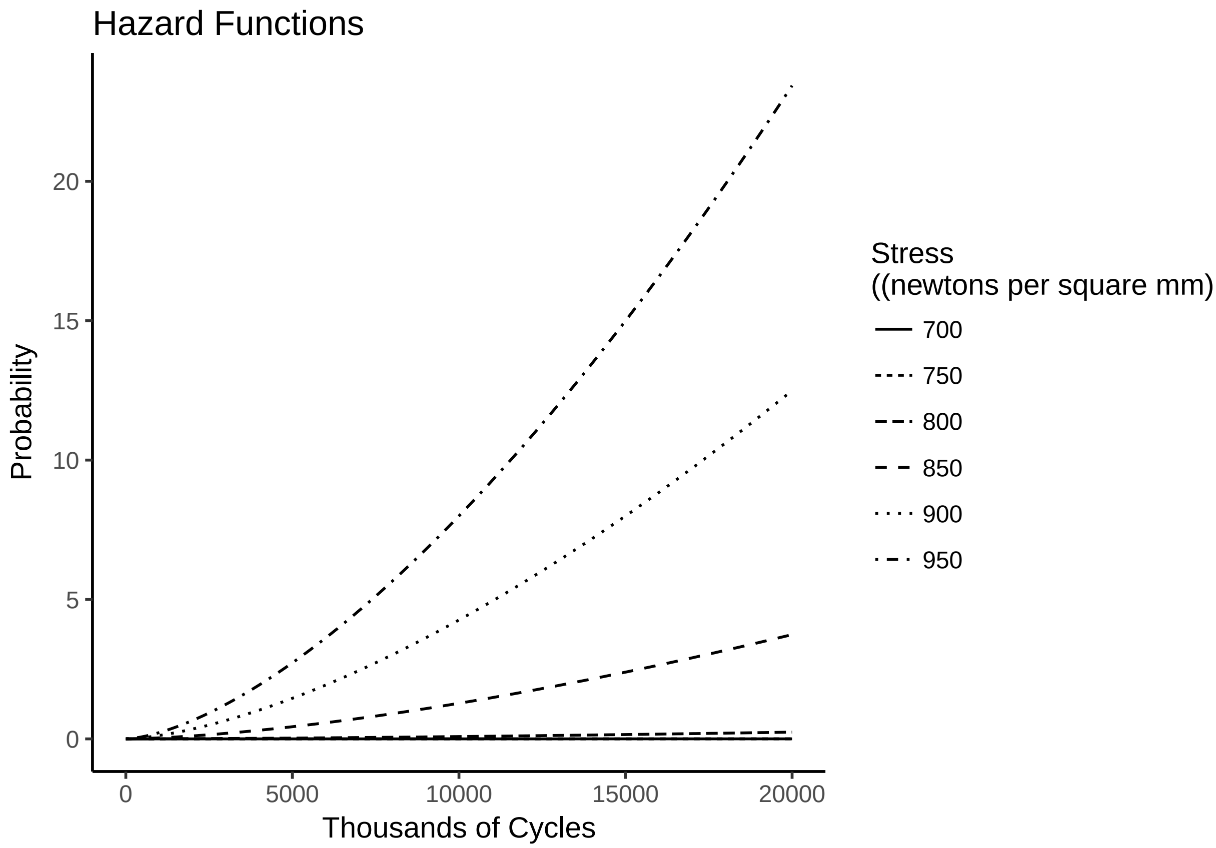
```
## Call:
## flexsurvreg(formula = Surv(cycles, cens) ~ stress, data = springs,
##     dist = "weibull")
##
## Estimates:
##            data mean    est          L95%         U95%         se
## shape               NA      2.55098      2.04234      3.18630      0.28944
```

```
## scale               NA  15094.22234  10229.60127  22272.18264   2995.99974
## stress750       0.16667     -0.69660     -1.17285     -0.22035      0.24299
## stress800       0.16667     -2.67667     -3.15287     -2.20047      0.24296
## stress850       0.16667     -3.75081     -4.21040     -3.29123      0.23449
## stress900       0.16667     -4.22366     -4.68376     -3.76356      0.23475
## stress950       0.16667     -4.47013     -4.93010     -4.01015      0.23469
##              exp(est)     L95%         U95%
## shape               NA           NA           NA
## scale               NA           NA           NA
## stress750      0.49828      0.30949      0.80224
## stress800      0.06879      0.04273      0.11075
## stress850      0.02350      0.01484      0.03721
## stress900      0.01464      0.00924      0.02320
## stress950      0.01145      0.00723      0.01813
##
## N = 60,  Events: 53,  Censored: 7
## Total time at risk: 169993
## Log-likelihood = -378.9, df = 7
## AIC = 771.8
```

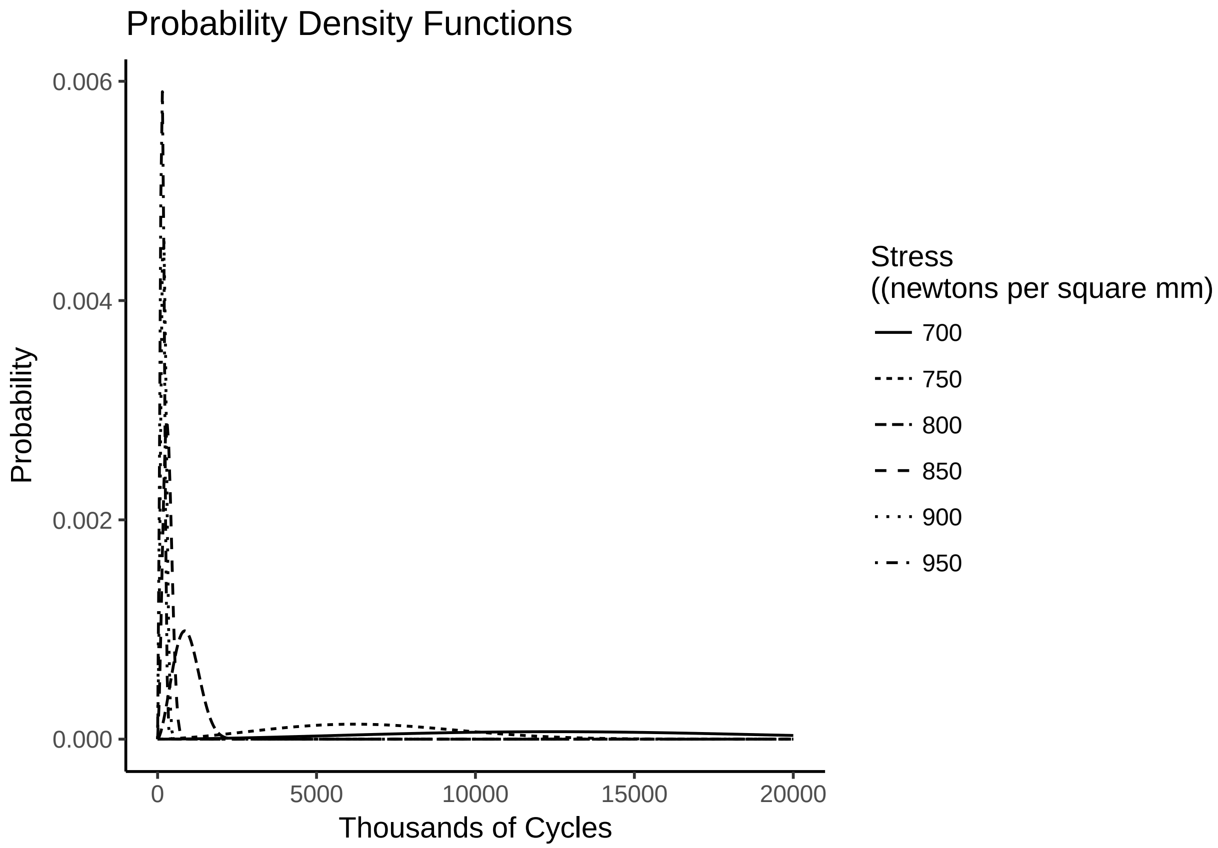
```r
#survival function plots
d <- data.frame(stress = levels(springs$stress))
d <- summary(m, newdata = d, t = seq(0, 20000, by = 50), type = "survival", tidy = TRUE)

p <- ggplot(d, aes(x = time, y = est))
p <- p + geom_line(aes(linetype = stress))
p <- p + labs(x = "Thousands of Cycles",
              y = "Probability",
              title = 'Survival Functions',
              linetype = 'Stress \n((newtons per square mm)')
p <- p + theme(legend.position = 'none')
p <- p + theme_classic()
plot(p)
```




```r
#plot hazard function
d <- data.frame(stress = levels(springs$stress))
d <- summary(m, newdata = d, t = seq(0, 20000, by = 50), type = "hazard", tidy = TRUE)

p <- ggplot(d, aes(x = time, y = est))
p <- p + geom_line(aes(linetype = stress))
p <- p + labs(x = "Thousands of Cycles",
              y = "Probability",
              title = 'Hazard Functions',
              linetype = 'Stress \n((newtons per square mm)')
p <- p + theme_classic()
plot(p)
```




```r
#plot probability density
d <- data.frame(stress = levels(springs$stress))
d <- summary(m, newdata = d,
             t = seq(0, 20000, by = 50),
             fn = function(t, ...) dweibull(t, ...), tidy = TRUE)

p <- ggplot(d, aes(x = time, y = est))
p <- p + geom_line(aes(linetype = stress))
p <- p + labs(x = "Thousands of Cycles",
              y = "Probability",
              title = 'Probability Density Functions',
              linetype = 'Stress \n((newtons per square mm)')
p <- p + theme_classic()
plot(p)
```




```r
#estimate the expected number of thousands of cycles for each of these six stress levels
d <- data.frame(stress = levels(springs$stress))
summary(m, newdata = d, type = "mean", tidy = TRUE)
```

```
##       est    lcl     ucl stress
## 1 13399.6 9369.2 19899.9    700
## 2  6676.7 5052.0  8606.1    750
## 3   921.8  721.0  1170.3    800
## 4   314.9  246.8   397.1    850
## 5   196.2  156.2   250.6    900
## 6   153.4  120.9   199.7    950
```

```r
#estimate hazards ratio
m <- flexsurvreg(Surv(cycles, cens) ~ stress, data = springs, dist = "weibullPH")
print(m)
```

```
## Call:
## flexsurvreg(formula = Surv(cycles, cens) ~ stress, data = springs,
##     dist = "weibullPH")
##
## Estimates:
##            data mean  est       L95%      U95%      se        exp(est)
## shape            NA   2.55e+00  2.04e+00  3.18e+00  2.89e-01        NA
## scale            NA   2.19e-11  1.01e-13  4.75e-09  6.00e-11        NA
## stress750  1.67e-01   1.78e+00  5.95e-01  2.96e+00  6.03e-01  5.91e+00
## stress800  1.67e-01   6.83e+00  5.19e+00  8.46e+00  8.35e-01  9.23e+02
## stress850  1.67e-01   9.57e+00  7.30e+00  1.18e+01  1.16e+00  1.43e+04
```

```
## stress900  1.67e-01   1.08e+01  8.28e+00  1.33e+01  1.27e+00  4.78e+04
## stress950  1.67e-01   1.14e+01  8.78e+00  1.40e+01  1.34e+00  8.96e+04
##            L95%      U95%
## shape            NA        NA
## scale            NA        NA
## stress750  1.81e+00  1.93e+01
## stress800  1.80e+02  4.74e+03
## stress850  1.48e+03  1.38e+05
## stress900  3.94e+03  5.79e+05
## stress950  6.51e+03  1.23e+06
##
## N = 60,  Events: 53,  Censored: 7
## Total time at risk: 169993
## Log-likelihood = -378.9, df = 7
## AIC = 771.8
```

Marginal Effects for the Barnacles Model

1. For each location report the estimated discrete marginal effect of RC length between 10 and 15mm, and 15 and 20mm. Also estimate the discrete marginal effect of location at RC lengths of 10, 15, and 20mm. Interpret the results.

The discrete marginal effect of increasing RC length from 10 to 15 mm is a 0.48 g increase in dry weight at Barca and 0.52 g increase in dry weight at Lens.

The discrete marginal effect of increasing RC length from 15 to 20 mm is a 0.89 g increase in dry weight at Barca and 0.99 g increase in dry weight at Lens.

```r
m <- glm(DW ~ log2(RC) * F, data = barnacle, family = tweedie(link.power = 0, var.power = 1.7))

#for each location estimate discrete marginal effect of RC length
#between 10 and 15, and 15 and 20 mm
trtools::margeff(m,
        a = list(RC = 15, F = c('barca', 'lens')),
        b = list(RC = 10, F = c('barca', 'lens')),
        cnames = c('barca', 'lens'))
```

```
##       estimate       se  lower  upper tvalue   df pvalue
## barca   0.4787 0.004160 0.4706 0.4869  115.1 1996      0
## lens    0.5154 0.004349 0.5069 0.5239  118.5 1996      0
```

```r
trtools::margeff(m,
        a = list(RC = 20, F = c('barca', 'lens')),
        b = list(RC = 15, F = c('barca', 'lens')),
        cnames = c('barca', 'lens'))
```

```
##       estimate      se  lower  upper tvalue   df pvalue
## barca   0.8908 0.01169 0.8679 0.9137  76.20 1996      0
## lens    0.9870 0.01285 0.9618 1.0122  76.82 1996      0
```

2. For each location, report the instantaneous marginal effect of RC length at lengths of 10, 15, and 20 mm. Interpret the results.

At a RC length length of 10 mm, the expected change in dry weight is 0.00006 g per mm increase in RC length.

At a RC length length of 15 mm, the expected change in dry weight is 0.0001 g per mm increase in RC length.

At a RC length length of 20, the expected change in dry weight is 0.0002 g per mm increase in RC length.

```
trtools::margeff(m,
        a = list(RC = c(10, 15, 20) + 0.001, F = 'barca'),
        b = list(RC = c(10, 15, 20), F = 'barca'),
        cnames = c('@10', '@15', '@20'))
```

```
##        estimate           se      lower      upper tvalue   df pvalue
## @10 0.00006255 0.0000004217 0.00006172 0.00006337 148.31 1996      0
## @15 0.00013301 0.0000014325 0.00013020 0.00013582  92.85 1996      0
## @20 0.00022719 0.0000034136 0.00022050 0.00023389  66.56 1996      0
```

3. For each location, report the estimated percent change in expected RC length between 10 and 15 mm, and 15 and 20 mm. Also estimate the percent change between the locations at RC lengths of 10, 15, and 20 mm. Interpret the results.

The estimated percent change in dry weight for an increase in RC length from 10 to 15 mm is 219% at Barca, and 231% at Lens.

The estimated percent change in dry weight for an increase in RC length from 15 to 20 mm is 128% at Barca, and 134% at Lens.

The estimated percent increase in dry weight from Barca to Lens is 2.3% at an RC length of 10mm, 6.0% at an RC length of 15mm, and 8.7% at an RC length of 20mm.

```
trtools::margeff(m,
        a = list(RC = 15, F = c('barca', 'lens')),
        b = list(RC = 10, F = c('barca', 'lens')),
        type = 'percent',
        cnames = c('barca', 'lens'))
```

```
##       estimate    se lower upper tvalue   df pvalue
## barca    219.0 2.239 214.6 223.4  97.83 1996      0
## lens     230.6 2.407 225.9 235.3  95.81 1996      0
```

```
trtools::margeff(m,
        a = list(RC = 20, F = c('barca', 'lens')),
        b = list(RC = 15, F = c('barca', 'lens')),
        type = 'percent',
        cnames = c('barca', 'lens'))
```

```
##       estimate    se lower upper tvalue   df pvalue
## barca    127.7 1.134 125.5 130.0  112.7 1996      0
## lens     133.6 1.206 131.2 135.9  110.7 1996      0
```

```
trtools::margeff(m,
        a = list(RC = c(10, 15, 20), F = c('lens')),
        b = list(RC = c(10, 15, 20), F = c('barca')),
        type = 'percent',
        cnames = c('10', '15', '20'))
```

```
##    estimate    se   lower  upper tvalue   df         pvalue
## 10    2.254 1.169 -0.0385  4.546  1.928 1996 0.05397078120
## 15    5.964 1.040  3.9254  8.003  5.737 1996 0.00000001113
## 20    8.679 1.521  5.6956 11.662  5.706 1996 0.00000001332
```

White Sturgeon Sexual Maturity

1. Estimate the sequential regression model using family = cratio. Do not specify paralle = TRUE. Report the parameter estimates returned by vglm. Plot the estimated probability for each category of sexual maturity as a function of size and se. The format of the plot can be similar to that we used for the impairment data in lecture but with size on the x-axis and one panel for each sex. Interpret the odds ratios for the effects of size and sex (apply the exponential function to the parameter estimates).

The odds of a sturgeon being at least mature increases by a factor of 1.04 for each 10 cm increase in midpoint size.

The odds of being ripe increases by a factor of 1.0 for each 10 cm increase in midpoint size.

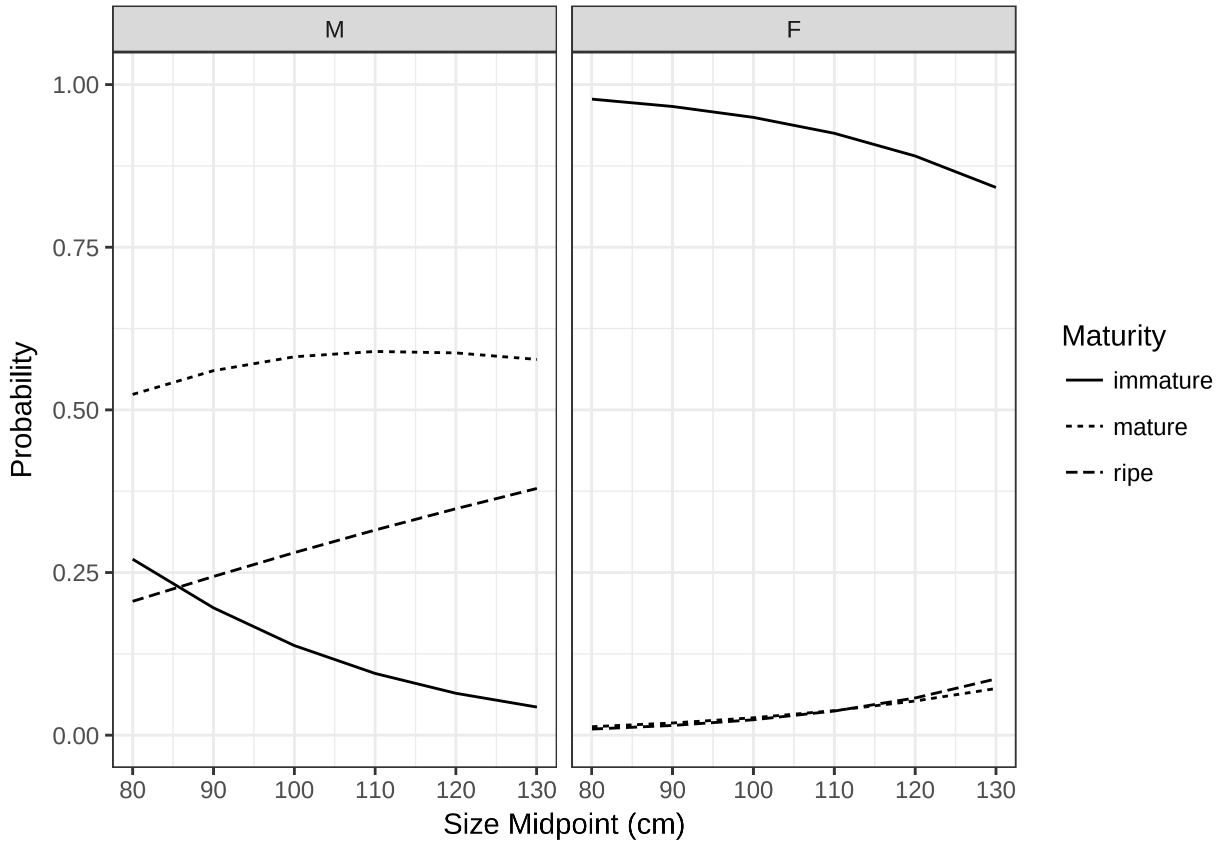
The odds of a sturgeon being at least mature increases by a factor of 117.90 if the sturgeon is a male.

The odds of a male sturgeon being at ripe is decreases by a factor of 0.54.

```
#convert proportions into frequencies
mysturgeon <- sturgeon
mysturgeon$immature <- with(sturgeon, round(immature * count))
mysturgeon$mature <- with(sturgeon, round(mature * count))
mysturgeon$ripe <- with(sturgeon, round(ripe * count))

m <- vglm(cbind(immature, mature, ripe) ~ midpoint + sex,
          family = cratio, data = mysturgeon)
cbind(coef(m), confint(m))
```

```
##                            2.5 %   97.5 %
## (Intercept):1 -7.14678 -9.0267143 -5.26686
## (Intercept):2 -1.14776 -2.7920186  0.49650
## midpoint:1     0.04211  0.0300134  0.05421
## midpoint:2     0.01027  0.0002563  0.02028
## sexM:1         4.76983  4.0671162  5.47254
## sexM:2        -0.60812 -1.1512293 -0.06502
```

```
#plot
d <- expand.grid(sex = c("M","F"), midpoint = seq(80, 130, by = 10))
d <- cbind(d, predict(m, newdata = d, type = "response"))
d <- gather(d, key = "maturity", value = "probability", immature, mature, ripe)

p <- ggplot(d, aes(x = midpoint, y = probability, linetype = maturity))
p <- p + geom_line() + ylim(0,1) + theme_bw() + facet_wrap(~sex)
p <- p + labs(x = 'Size Midpoint (cm)', y = 'Probability', linetype = 'Maturity')
plot(p)
```




```r
#odds ratios
round(exp(cbind(coef(m), confint(m))), 2)
```

```
##                        2.5 % 97.5 %
## (Intercept):1   0.00  0.00   0.01
## (Intercept):2   0.32  0.06   1.64
## midpoint:1      1.04  1.03   1.06
## midpoint:2      1.01  1.00   1.02
## sexM:1        117.90 58.39 238.06
## sexM:2          0.54  0.32   0.94
```

2. Repeat the previous problem except use a proportional odds model with family = propodds.

The odds of a sturgeon being at least mature increases by a factor of 1.03 for each 10 cm increase in midpoint size.

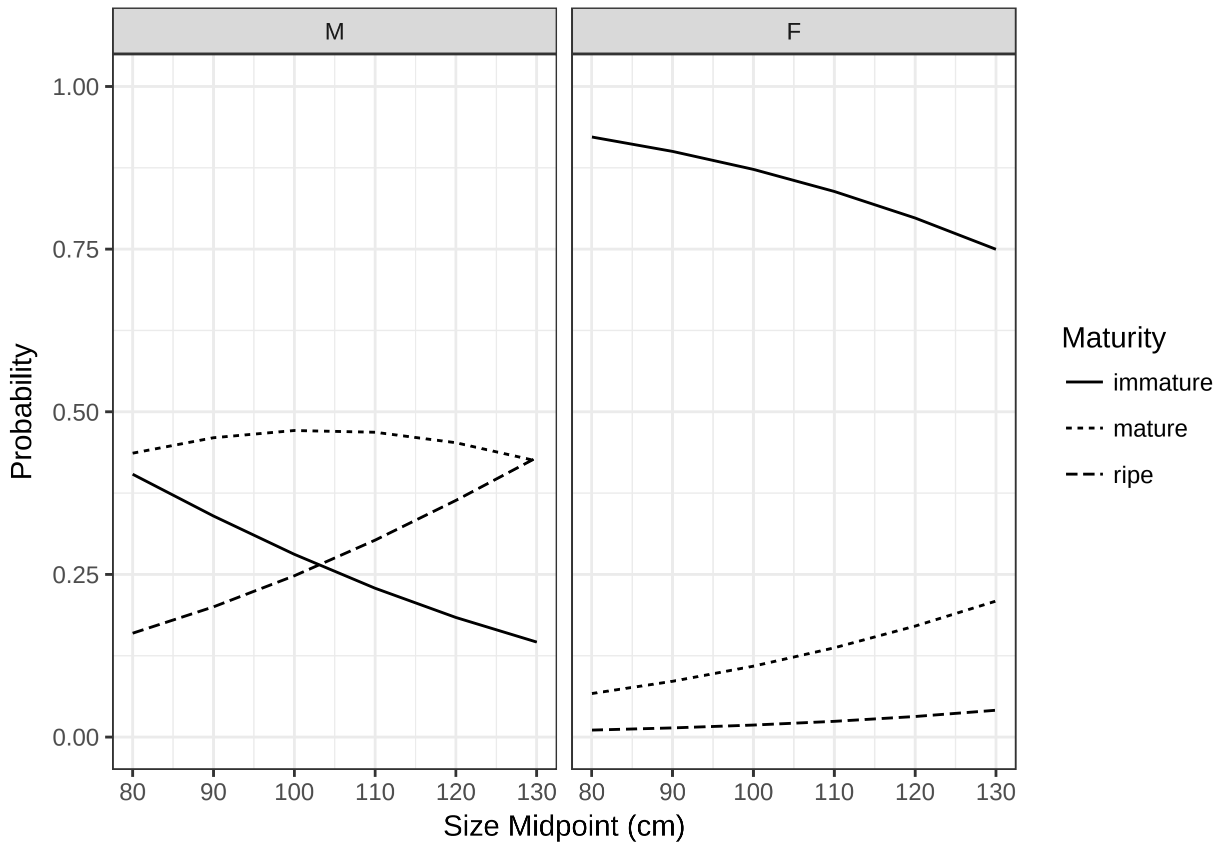
The odds of a sturgeon being at least mature increases by a factor of 17.53 if the sturgeon is male.

```r
#convert proportions into frequencies
mysturgeon <- sturgeon
mysturgeon$immature <- with(sturgeon, round(immature * count))
mysturgeon$mature <- with(sturgeon, round(mature * count))
mysturgeon$ripe <- with(sturgeon, round(ripe * count))

m <- vglm(cbind(immature, mature, ripe) ~ midpoint + sex,
          family = propodds, data = mysturgeon)
```

```r
cbind(coef(m), confint(m))
```

```
##                              2.5 %   97.5 %
## (Intercept):1 -4.67890 -5.86485 -3.49295
## (Intercept):2 -6.72868 -7.97521 -5.48214
## midpoint       0.02755  0.01983  0.03527
## sexM           2.86369  2.46721  3.26017
```

```r
#plot
d <- expand.grid(sex = c("M","F"), midpoint = seq(80, 130, by = 10))
d <- cbind(d, predict(m, newdata = d, type = "response"))
d <- gather(d, key = "maturity", value = "probability", immature, mature, ripe)

p <- ggplot(d, aes(x = midpoint, y = probability, linetype = maturity))
p <- p + geom_line() + ylim(0,1) + theme_bw() + facet_wrap(~sex)
p <- p + labs(x = 'Size Midpoint (cm)', y = 'Probability', linetype = 'Maturity')
plot(p)
```




```r
#odds ratios
round(exp(cbind(coef(m), confint(m))), 2)
```

```
##                       2.5 % 97.5 %
## (Intercept):1  0.01  0.00   0.03
## (Intercept):2  0.00  0.00   0.00
## midpoint       1.03  1.02   1.04
## sexM          17.53 11.79  26.05
```

3. Repeat the previous problem except use a multinomial logit model. Be sure to set the baseline category to the first (immature) category using the option refLevel = 'immature'.

The odds of a sturgeon being mature vs immature increases by a factor of 1.04 for every 10 cm increase in midpoint size.

The odds of a sturgeon being ripe vs mature increases by 1.05 for every 10 cm increase in midpoint size.

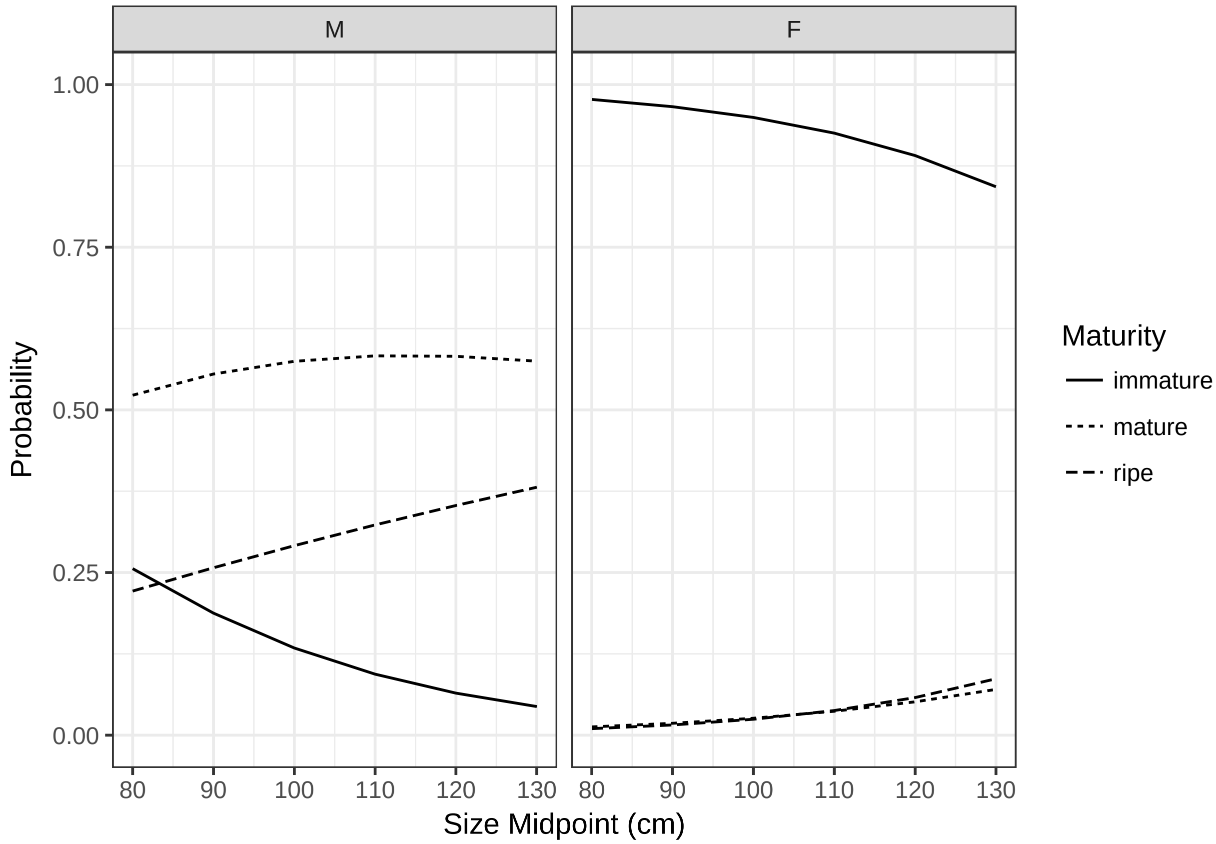
The odds of a sturgeon being mature vs immature increases by a factor of 156.62 if the sturgeon is a male.

The odds of a sturgeon being ripe vs immature increases by a factor of 84.22 if the sturgeon is a male.

```
#convert proportions into frequencies
mysturgeon <- sturgeon
mysturgeon$immature <- with(sturgeon, round(immature * count))
mysturgeon$mature <- with(sturgeon, round(mature * count))
mysturgeon$ripe <- with(sturgeon, round(ripe * count))

m <- vglm(cbind(immature, mature, ripe) ~ midpoint + sex,
          family = multinomial(refLevel = "immature"), data = mysturgeon)
cbind(coef(m), confint(m))
```

```
##                               2.5 %   97.5 %
## (Intercept):1 -7.31012  -9.37534 -5.24490
## (Intercept):2 -8.26348 -10.30410 -6.22286
## midpoint:1     0.03713   0.02388  0.05037
## midpoint:2     0.04607   0.03309  0.05904
## sexM:1         5.05381   4.29514  5.81248
## sexM:2         4.43343   3.68800  5.17887
```

```
#plot
d <- expand.grid(sex = c("M","F"), midpoint = seq(80, 130, by = 10))
d <- cbind(d, predict(m, newdata = d, type = "response"))
d <- gather(d, key = "maturity", value = "probability", immature, mature, ripe)

p <- ggplot(d, aes(x = midpoint, y = probability, linetype = maturity))
p <- p + geom_line() + ylim(0,1) + theme_bw() + facet_wrap(~sex)
p <- p + labs(x = 'Size Midpoint (cm)', y = 'Probability', linetype = 'Maturity')
plot(p)
```




```r
#odds ratios
round(exp(cbind(coef(m), confint(m))), 2)
```

```
##                          2.5 % 97.5 %
## (Intercept):1   0.00  0.00   0.01
## (Intercept):2   0.00  0.00   0.00
## midpoint:1      1.04  1.02   1.05
## midpoint:2      1.05  1.03   1.06
## sexM:1        156.62 73.34 334.45
## sexM:2         84.22 39.96 177.48
```