Stats 204 Homework 5

Jordan Berninger

11/20/2019

1. In the malaria dataset, analyze the risk of malaria via logistic regression with age and the log-transformed antibody level as explanatory variables. Explain your model, analysis and conclusions.

We will use the *glm()* function in R to fit the following logistic regression model: $logit(p_i) = \beta_0 + \beta_1 * log(ab_i) + \beta_2 * age_i + \beta_3 * (age_i * log(ab_i))$ where $p_i = P(Y_i = 1|X)$. Since $log(ab)$ and age are continuous variables, the model only has one parameter associated with each of these variables - we do not have more parmaeters associated with the various factor levels. We also begin with the interaction effect in the model, which also only has one associated parameter. We fit the model:

```
data(malaria)
m1 <- glm(data = malaria, mal ~ age*log(ab), family = "binomial")
summary(m1)
```

```
##
## Call:
## glm(formula = mal ~ age * log(ab), family = "binomial", data = malaria)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8337  -0.7106  -0.4939   0.8566   2.4655
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.85855    2.43335   1.997   0.0459 *
## age         -0.32288    0.25507  -1.266   0.2056
## log(ab)     -1.24110    0.58514  -2.121   0.0339 *
## age:log(ab)  0.06138    0.05818   1.055   0.2915
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 116.652  on 99  degrees of freedom
## Residual deviance:  96.921  on 96  degrees of freedom
## AIC: 104.92
##
## Number of Fisher Scoring iterations: 4
```

The model summary tells us that the interaction effect is insignificant, so we will fit a model without it next. This next model is defined by: $logit(p_i) = \beta_0 + \beta_1 * log(ab_i) + \beta_2 * age_i$ where $p_i = P(Y_i = 1|X)$.

```
m2 <- glm(data = malaria, mal ~ age + log(ab), family = "binomial")
summary(m2)
```

```
##
## Call:
```

```
## glm(formula = mal ~ age + log(ab), family = "binomial", data = malaria)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8492  -0.7536  -0.4838   0.8809   2.5796
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.57234    0.95184   2.702 0.006883 **
## age         -0.06546    0.06772  -0.967 0.333703
## log(ab)     -0.68235    0.19552  -3.490 0.000483 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 116.652  on 99  degrees of freedom
## Residual deviance:  98.017  on 97  degrees of freedom
## AIC: 104.02
##
## Number of Fisher Scoring iterations: 5
```

This model indicates that the age variable is insignificant, so we will fit a model without it. This leaves us with the model: $logit(p_i) = \beta_0 + \beta_1 * log(ab_i)$ where $p_i = P(Y_i = 1|X)$

```
m3 <- glm(data = malaria, mal ~ log(ab), family = "binomial")
summary(m3)
```

```
##
## Call:
## glm(formula = mal ~ log(ab), family = "binomial", data = malaria)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9159  -0.7339  -0.4854   0.8813   2.4722
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.1552     0.8401   2.565 0.010305 *
## log(ab)      -0.7122     0.1932  -3.686 0.000228 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 116.652  on 99  degrees of freedom
## Residual deviance:  98.968  on 98  degrees of freedom
## AIC: 102.97
##
## Number of Fisher Scoring iterations: 4
```

At a confidence level of 0.05, we reject the null hypotheses that $\beta_0 = 0$ and $\beta_1 = 0$, which means that our model $logit(p_i) = \beta_0 + \beta_1 * log(ab_i)$ only has significant parameters. Additionally, this model has the lowest AIC and so it is our favorite model and it is the one we will use for estimation.

We now want the 95% confidence intervals for the parameter estimates, noting that the parameter associated with $log(ab)$ has a 95% confidence interval that is strictly less than 1, which means increases in $log(ab)$ translates to a decrease in the log-odds ratio of the risk of malaria.

```
exp(cbind(OR=coef(m3), confint(m3)))
```

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

```
##                    OR     2.5 %     97.5 %
## (Intercept) 8.6296891 1.7852011 49.6349805
## log(ab)     0.4905608 0.3246415  0.6977821
```

2. Fit a logistic regression model to the graft.vs.host data predicting gvhd response. Use different transformations of the index variable.

I have fit many, many model as part of this analysis and several interesting things jump out. - First, we note that we cannot include the *pnr* variable in the model since this is a unique identifier. - Second, we note that the model without any transformations produces a model with a perfect fit, as seen in the following model summary of model $m4$ below. - Third, We notice that doing the log-transformation on time removes this perfect fit problem without us removing any additional columns, as seen in the summary of model $m5$.

```
data(graft.vs.host)
m4 <- glm(data = graft.vs.host, gvhd ~ index + rcpage + donage + type + preg + time + dead,
          family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m4)
```

```
##
## Call:
## glm(formula = gvhd ~ index + rcpage + donage + type + preg +
##     time + dead, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -3.916e-05  -2.100e-08  -2.100e-08   2.100e-08   3.097e-05
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.902e+02  1.206e+05  -0.002    0.999
## index        5.747e+01  2.795e+04   0.002    0.998
## rcpage       3.124e+00  2.615e+03   0.001    0.999
## donage       4.421e+00  2.857e+03   0.002    0.999
## type         7.324e+01  4.092e+04   0.002    0.999
## preg         6.323e+01  5.046e+04   0.001    0.999
## time        -3.878e-01  1.911e+02  -0.002    0.998
## dead        -1.420e+02  8.505e+04  -0.002    0.999
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5.1049e+01  on 36  degrees of freedom
## Residual deviance: 6.9564e-09  on 29  degrees of freedom
## AIC: 16
##
## Number of Fisher Scoring iterations: 25
```

```
m5 <- glm(data = graft.vs.host, gvhd ~ index + rcpage + donage + type + preg  + log(time) + dead,
          family = binomial)
summary(m5)
```

```
##
## Call:
## glm(formula = gvhd ~ index + rcpage + donage + type + preg +
##     log(time) + dead, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.61488  -0.12283  -0.00525   0.04065   2.08429
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  15.5356    13.5678   1.145   0.2522
## index         1.8789     0.8866   2.119   0.0341 *
## rcpage        0.1407     0.1409   0.999   0.3179
## donage        0.1188     0.1412   0.841   0.4003
## type          1.7646     1.3905   1.269   0.2044
## preg          2.8669     2.1861   1.311   0.1897
## log(time)    -4.8121     2.6887  -1.790   0.0735 .
## dead         -3.5951     3.8373  -0.937   0.3488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 14.491  on 29  degrees of freedom
## AIC: 30.491
##
## Number of Fisher Scoring iterations: 8
```

Now we will keep all the variables, including log(time), as predictors and will investiage the impact of various transformations to index, paying attention to the significance of variables and the model AIC. The transformations we will consider are the log(index), square root(index), cube-root(index), index^2, and index^1.5. We will systematically fit this complete model with various transformations on index and we will see which model optimizes the AIC.

Now we note the taking the log transformation to index produces a model with a perfect fit, which is strange.

```
m6 <- glm(data = graft.vs.host, gvhd ~ log(index) + rcpage + donage + type + preg  + log(time) + dead,
          family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m6)
```

```
##
## Call:
## glm(formula = gvhd ~ log(index) + rcpage + donage + type + preg +
##     log(time) + dead, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.052e-04  -2.100e-08  -2.100e-08   2.100e-08   9.481e-05
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2808.357 613425.551   0.005    0.996
## log(index)     285.932  60731.725   0.005    0.996
## rcpage          11.066   4148.006   0.003    0.998
## donage           8.749   4408.903   0.002    0.998
## type           155.548  58456.731   0.003    0.998
## preg           146.906 105023.363   0.001    0.999
## log(time)     -583.691 126212.630  -0.005    0.996
## dead          -628.261 145635.929  -0.004    0.997
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5.1049e+01  on 36  degrees of freedom
## Residual deviance: 4.4816e-08  on 29  degrees of freedom
## AIC: 16
##
## Number of Fisher Scoring iterations: 25
```

```
m7 <- glm(data = graft.vs.host, gvhd ~ sqrt(index) + rcpage + donage + type + preg  + log(time) + dead,
          family = binomial)
summary(m7)
```

```
##
## Call:
## glm(formula = gvhd ~ sqrt(index) + rcpage + donage + type + preg +
##     log(time) + dead, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.52690  -0.04907  -0.00010   0.00292   2.08684
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  26.3451    31.6767   0.832    0.406
## sqrt(index)   8.5999     5.3279   1.614    0.106
## rcpage        0.1773     0.1681   1.055    0.292
## donage        0.1910     0.2011   0.950    0.342
## type          2.9537     2.3767   1.243    0.214
## preg          4.0534     3.6397   1.114    0.265
```

```
## log(time)    -8.5711     7.1765  -1.194    0.232
## dead         -6.9930     8.8698  -0.788    0.430
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 11.402  on 29  degrees of freedom
## AIC: 27.402
##
## Number of Fisher Scoring iterations: 10
```

```
m8 <- glm(data = graft.vs.host, gvhd ~ I(index^(1/3)) + rcpage + donage + type + preg  + log(time) + dea
          family = binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m8)
```

```
##
## Call:
## glm(formula = gvhd ~ I(index^(1/3)) + rcpage + donage + type +
##     preg + log(time) + dead, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.48008  -0.00031   0.00000   0.00003   1.94992
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     63.3242    72.0059   0.879    0.379
## I(index^(1/3))  27.8807    23.8042   1.171    0.241
## rcpage           0.2939     0.2694   1.091    0.275
## donage           0.3315     0.3015   1.100    0.272
## type             6.5636     7.1293   0.921    0.357
## preg             8.6244    10.2715   0.840    0.401
## log(time)      -20.0079    19.5739  -1.022    0.307
## dead           -19.1871    20.7588  -0.924    0.355
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.0494  on 36  degrees of freedom
## Residual deviance:  9.4514  on 29  degrees of freedom
## AIC: 25.451
##
## Number of Fisher Scoring iterations: 12
```

```
m9 <- glm(data = graft.vs.host, gvhd ~ I(index^1.5) + rcpage + donage + type + preg  + log(time) + dead
          family = binomial)
summary(m9)
```

```
##
## Call:
```

```
## glm(formula = gvhd ~ I(index^1.5) + rcpage + donage + type +
##     preg + log(time) + dead, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.63417  -0.18607  -0.01955   0.09769   2.00902
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   12.8670    11.4040   1.128   0.2592
## I(index^1.5)   0.6277     0.3041   2.065   0.0390 *
## rcpage         0.1353     0.1326   1.020   0.3075
## donage         0.0824     0.1255   0.656   0.5116
## type           1.4163     1.2186   1.162   0.2451
## preg           2.5879     1.8706   1.383   0.1665
## log(time)     -3.8173     2.0964  -1.821   0.0686 .
## dead          -2.9214     3.2840  -0.890   0.3737
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 16.301  on 29  degrees of freedom
## AIC: 32.301
##
## Number of Fisher Scoring iterations: 8
```

```
m10 <- glm(data = graft.vs.host, gvhd ~ I(index^2) + rcpage + donage + type + preg  + log(time) + dead,
           family = binomial)
summary(m10)
```

```
##
## Call:
## glm(formula = gvhd ~ I(index^2) + rcpage + donage + type + preg +
##     log(time) + dead, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.74070  -0.23527  -0.03687   0.15130   1.95101
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.52377   10.61821   1.085   0.2778
## I(index^2)   0.24088    0.12003   2.007   0.0448 *
## rcpage       0.13579    0.12943   1.049   0.2941
## donage       0.05772    0.11640   0.496   0.6200
## type         1.25369    1.14933   1.091   0.2754
## preg         2.48426    1.74260   1.426   0.1540
## log(time)   -3.33890    1.85445  -1.800   0.0718 .
## dead        -2.57080    3.08351  -0.834   0.4044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 17.441  on 29  degrees of freedom
## AIC: 33.441
##
## Number of Fisher Scoring iterations: 8
```

For the models that include all the predictor variables, we see that the model which include index$^{1/3}$ has the lowest training set AIC (25.451) while not producing a perfect fit. Accordingly, we concluide that the squared transformation is the most apt. However, we notice that the other predictor varaibles have very high p-values across all the models. This indicates to me that we should fit models that have index (transformed) as the only predictor for modeling gvhd.

Now, we will remove the other variables since none of the model summary concluded that they were significant. We will now fit models that exclusively have index, iterating across the same transformation. We will include the log-transformation to index this time around. The following models are of the form: $logit(p_i) = \beta_0 + \beta_1 * \text{index}^*$, where index* is the transformation to the index variable

```
m11 <- glm(data = graft.vs.host, gvhd ~ sqrt(index), family = binomial)
summary(m11)
```

```
##
## Call:
## glm(formula = gvhd ~ sqrt(index), family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8695  -0.7391  -0.4275   0.8006   1.6919
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.6912     1.2978  -2.844  0.00445 **
## sqrt(index)   2.4153     0.8529   2.832  0.00463 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 38.381  on 35  degrees of freedom
## AIC: 42.381
##
## Number of Fisher Scoring iterations: 4
```

```
m12 <- glm(data = graft.vs.host, gvhd ~ I(index^(1/3)), family = binomial)
summary(m12)
```

```
##
## Call:
## glm(formula = gvhd ~ I(index^(1/3)), family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.8489  -0.7404  -0.3746   0.7960   1.6901
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -5.416      1.882  -2.877  0.00401 **
## I(index^(1/3))    4.128      1.441   2.866  0.00416 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 38.134  on 35  degrees of freedom
## AIC: 42.134
##
## Number of Fisher Scoring iterations: 4
```

```
m13 <- glm(data = graft.vs.host, gvhd ~ I(index^1.5), family = binomial)
summary(m13)
```

```
##
## Call:
## glm(formula = gvhd ~ I(index^1.5), family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9511  -0.7643  -0.6683   0.8631   1.6573
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.4392     0.5855  -2.458   0.0140 *
## I(index^1.5)   0.3102     0.1254   2.473   0.0134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 40.002  on 35  degrees of freedom
## AIC: 44.002
##
## Number of Fisher Scoring iterations: 5
```

```
m14 <- glm(data = graft.vs.host, gvhd ~ I(index^2), family = binomial)
summary(m14)
```

```
##
## Call:
## glm(formula = gvhd ~ I(index^2), family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.9706  -0.7867  -0.7377   0.9074   1.6271
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.17512    0.51285  -2.291   0.0219 *
## I(index^2)   0.13295    0.05683   2.340   0.0193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 40.642  on 35  degrees of freedom
## AIC: 44.642
##
## Number of Fisher Scoring iterations: 6
```

```
m15 <- glm(data = graft.vs.host, gvhd ~ log(index), family = binomial)
summary(m15)
```

```
##
## Call:
## glm(formula = gvhd ~ log(index), family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8021  -0.7491  -0.2654   0.7933   1.6780
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2930     0.5942  -2.176  0.02956 *
## log(index)    1.7380     0.6084   2.857  0.00428 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 37.740  on 35  degrees of freedom
## AIC: 41.74
##
## Number of Fisher Scoring iterations: 5
```

```
m16 <- glm(data = graft.vs.host, gvhd ~ index, family = binomial)
summary(m16)
```

```
##
## Call:
## glm(formula = gvhd ~ index, family = binomial, data = graft.vs.host)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9188  -0.7462  -0.5665   0.8256   1.6821
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9885     0.7479  -2.659  0.00784 **
## index         0.7747     0.2921   2.652  0.00799 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 39.211  on 35  degrees of freedom
## AIC: 43.211
##
## Number of Fisher Scoring iterations: 5
```

Overall, we see that these smaller models have higher AIC values, but that the model that has the log(index) has the lowest AIC out of this group of models as well (AIC = 41.74).

This analysis isn't completely black and white, but we conclude that the log transformation to index is the most appropraite. Our firs batch of models included all the predictor variables and showed that they are not significant covariates. We then applied logistic regression models on gvhd using only transformations to index as the predictor. In this case, we saw that the log(index) produced the smallest training set AIC, so that is our favorite.

If this were a linear regression situation, we could apply the Box Cox procedure directly, but since this is a logistic regression situation, I do not know how to adapt Box Cox accordingly, but I am sure it is possible.

3. The dataset SpaceShuttle available in the R package vcd contains data for 24 space shuttle flights before the Challenger mission disaster in 1986. In particular, it contains the flight number, temperature and pressure at the time of the flight, and whether at least one primary O-ring suffered thermal distress. Use a logistic regression model to model the effect of temperature and pressure on the probability of thermal distress. Explain your model, analysis and conclusions.

We read in the data, drop one row that has missing data and notice that both temperature and pressure are continuous variables by default. Since we want to predict failure, we will use logistic regression models. We will first model failure with both temperature and pressure as continuous variables, and then we will switch them to factor and evalute the impact.

First, implement the following model: $logit(p_i) = \beta_0 + \beta_1 temperature_i + \beta_2 pressure_i$, where $P(Y_i = 1|X) = p_i$. This model's summary indicates the Pressure is an insignificant variable. However, looking at the data, we see there are 3 distinct pressure levels, so it makes sense to convert Pressure into a factor before dropping it from the model entirely.

```
library(vcd)
```

```
## Loading required package: grid
```

```
data(SpaceShuttle)

j1 <- glm(data = na.omit(SpaceShuttle), Fail ~ Temperature + Pressure, family = binomial)
summary(j1)
```

```
##
## Call:
## glm(formula = Fail ~ Temperature + Pressure, family = binomial,
##     data = na.omit(SpaceShuttle))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2130  -0.6089  -0.3870   0.3472   2.0928
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 14.359775   7.442899   1.929   0.0537 .
## Temperature -0.241540   0.109722  -2.201   0.0277 *
## Pressure     0.009534   0.008703   1.095   0.2733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 18.972  on 20  degrees of freedom
## AIC: 24.972
##
## Number of Fisher Scoring iterations: 5
```

Next, we have the model: $logit(p_i) = \beta_0 + \beta_1 temperature_i + \gamma_i$, where γ_i is the pressure effect and $P(Y_i = 1|X) = p_i$. This model summary indicates that factor(pressure) is still insignificnt.

```
j2 <- glm(data = na.omit(SpaceShuttle), Fail ~ Temperature + factor(Pressure), family = binomial)
summary(j2)
```

```
##
## Call:
## glm(formula = Fail ~ Temperature + factor(Pressure), family = binomial,
##     data = na.omit(SpaceShuttle))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2119  -0.6119  -0.3862   0.3485   2.0949
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          14.7970     7.9127   1.870   0.0615 .
## Temperature          -0.2410     0.1146  -2.104   0.0354 *
## factor(Pressure)100   0.5094     2.2407   0.227   0.8202
## factor(Pressure)200   1.4338     1.3306   1.078   0.2812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 18.971  on 19  degrees of freedom
```

```
## AIC: 26.971
##
## Number of Fisher Scoring iterations: 5
```

Now, we will also implement the model where both temperature and pressure are factor variables. Accordingly, we have the model: $logit(p_i) = \beta_0 + \alpha_i + \gamma_i$, where α_i is the temperature effect, γ_i is the pressure effect.

```
j3 <- glm(data = na.omit(SpaceShuttle), Fail ~ factor(Temperature) + factor(Pressure), family = binomial
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(j3)
```

```
##
## Call:
## glm(formula = Fail ~ factor(Temperature) + factor(Pressure),
##     family = binomial, data = na.omit(SpaceShuttle))
##
## Deviance Residuals:
##        1         2         3         5         6         7         8
## -0.00003   0.00006  -0.00003  -0.00003  -0.00006  -0.00003  -0.00003
##        9        10        11        12        13        14        15
## -0.00003   0.00003   0.00003   1.17741  -0.00003   0.00000   0.00003
##       16        17        18        19        20        21        22
##  0.00000  -1.17741  -1.17741  -0.00003  -0.00003  -0.00003   1.17741
##       23        24
##  0.00003  -0.00003
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            4.171e+01  3.256e+04   0.001    0.999
## factor(Temperature)57  2.157e+01  5.063e+04   0.000    1.000
## factor(Temperature)58 -3.714e-08  4.134e+04   0.000    1.000
## factor(Temperature)63 -3.091e-08  4.134e+04   0.000    1.000
## factor(Temperature)66 -6.327e+01  4.376e+04  -0.001    0.999
## factor(Temperature)67 -6.185e+01  3.558e+04  -0.002    0.999
## factor(Temperature)68 -6.327e+01  4.376e+04  -0.001    0.999
## factor(Temperature)69 -6.327e+01  4.376e+04  -0.001    0.999
## factor(Temperature)70 -2.157e+01  2.923e+04  -0.001    0.999
## factor(Temperature)72 -6.327e+01  4.376e+04  -0.001    0.999
## factor(Temperature)73 -6.327e+01  4.376e+04  -0.001    0.999
## factor(Temperature)75 -2.157e+01  2.923e+04  -0.001    0.999
## factor(Temperature)76 -4.313e+01  3.580e+04  -0.001    0.999
## factor(Temperature)78 -4.313e+01  4.134e+04  -0.001    0.999
## factor(Temperature)79 -4.313e+01  4.134e+04  -0.001    0.999
## factor(Temperature)81 -4.313e+01  4.134e+04  -0.001    0.999
## factor(Pressure)100   -4.171e+01  3.256e+04  -0.001    0.999
## factor(Pressure)200   -2.014e+01  1.434e+04  -0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 28.2672  on 22  degrees of freedom
## Residual deviance:  5.5452  on  5  degrees of freedom
## AIC: 41.545
##
## Number of Fisher Scoring iterations: 20
```

This model is not ideal because we have 17 different values for Temperature, 3 different values for Pressure and only 23 data points (after we remove one row with missing data). Accordingly, this model has very low Residual Deviance, but there is not much insight from the model because it is nearly a perfect fit. We also note that this same problem occurs when we have temperature as a factor and pressure as a continuous variables. It does not make sense to investigate an interaction effect between temperature and pressure, since that model would have more parameters than data points.

Now, we have seen that (1) converting temperature to a factor creates too many parameters for the number of data points to get a sensible model and (2) pressure is insignificant whether it is continuous or a factor. Accordingly, we conclude that pressure does not have a significant impact on o-ring failure. This leaves us with the model: $logit(p_i) = \beta_0 + \beta_1 \text{temperature}$.

```
j4 <- glm(data = na.omit(SpaceShuttle), Fail ~ Temperature, family = binomial)
summary(j4)
```

```
##
## Call:
## glm(formula = Fail ~ Temperature, family = binomial, data = na.omit(SpaceShuttle))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0611  -0.7613  -0.3783   0.4524   2.2175
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  15.0429     7.3786   2.039   0.0415 *
## Temperature  -0.2322     0.1082  -2.145   0.0320 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 20.315  on 21  degrees of freedom
## AIC: 24.315
##
## Number of Fisher Scoring iterations: 5
```

From the summary of model $j4$, using a confidence level of 0.05, we reject both null hypotheses that $\beta_0 = 0$ and $\beta_1 = 0$. We concluded that pressure does not have a significant impact on o-ring failure. We also conclude that as temperature rises, the log odds ratio of failure decreases. This means that as temperature goes up, the estimated probability of failure decreases.

4. Problem 2.14 Albert & Rizzo

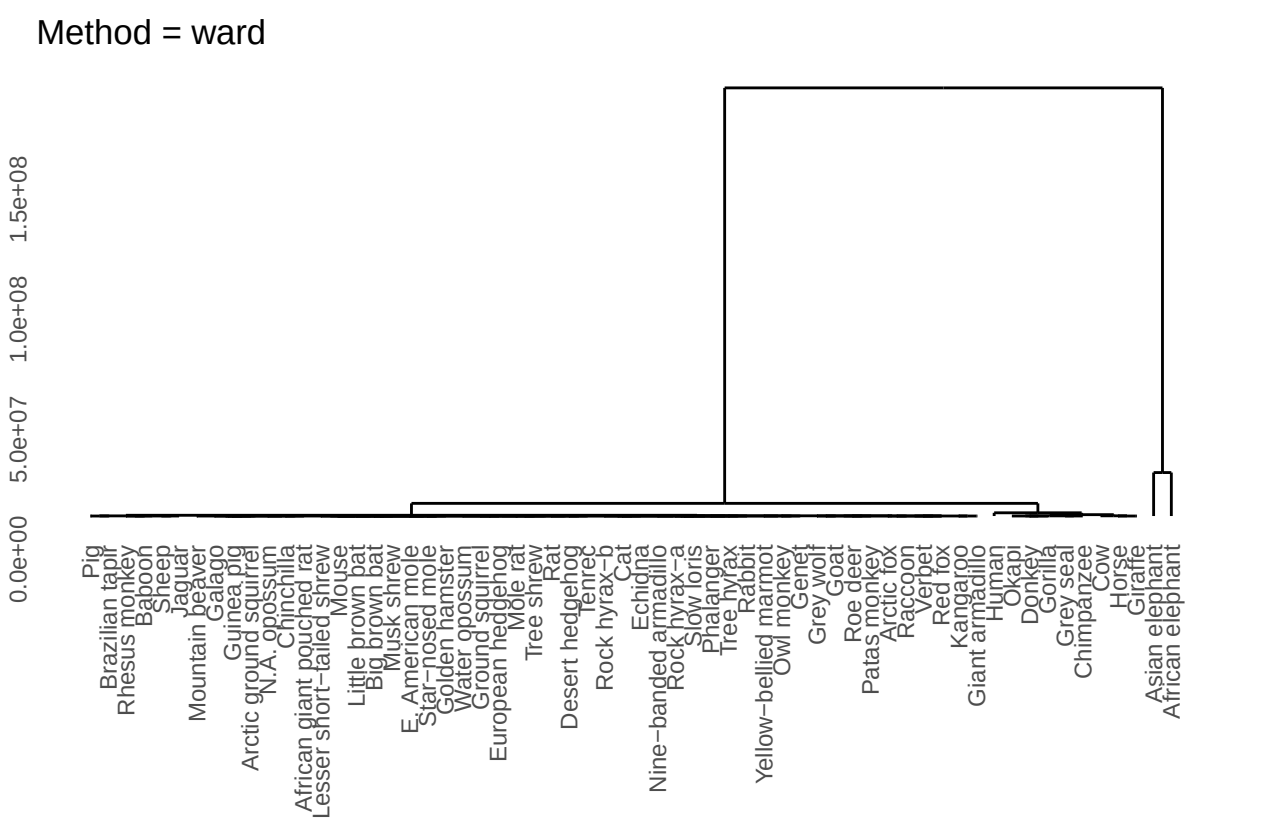
Refer to Example 2.10. Repeat the cluster analysis using Ward's minimum variance method instead of nearest neighbor (complete) linkage. Ward's method is implemented in hclust with method="ward" when

the first argument is the squared distance matrix. Display a dendrogram and compare the result with the dendrogram for the nearest neighbor method.

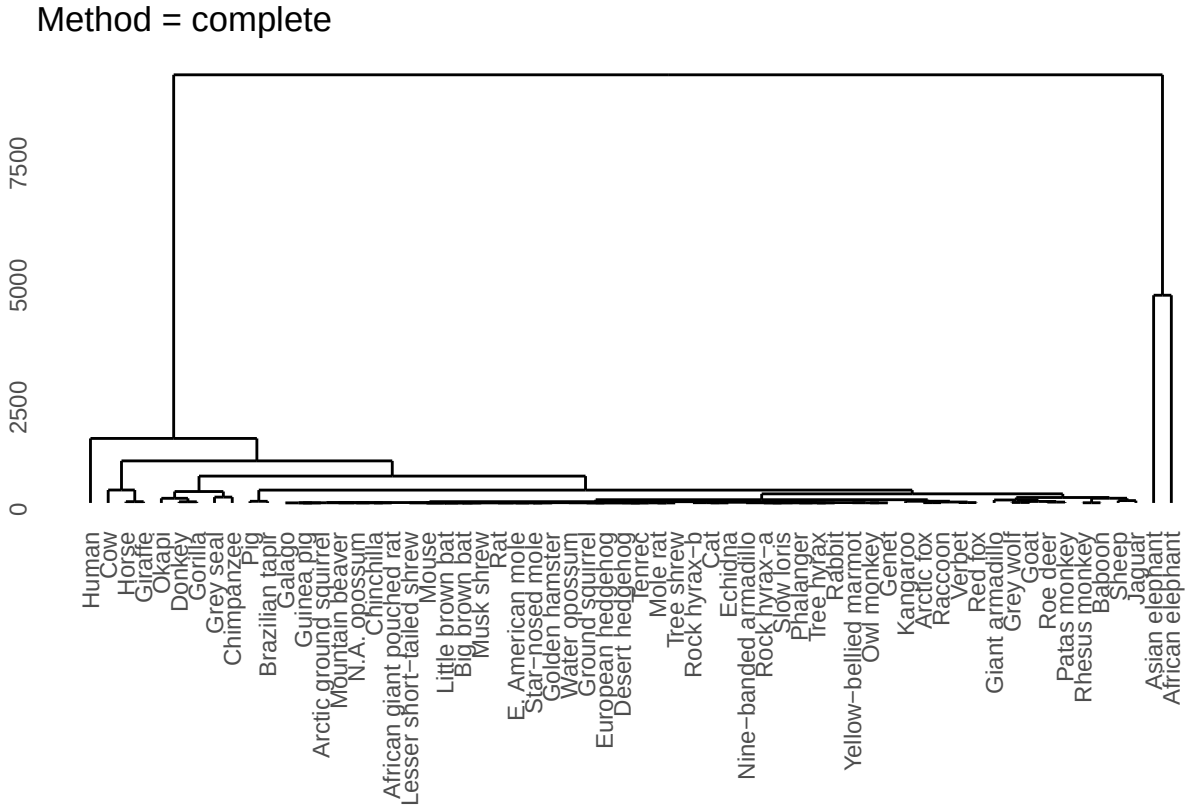
Since we are already familiar with this dataset, no EDA is necessary here. We will follow the instructions and Example 2.10 closely here.

The results for the 2 different methods are fairly similar. Both methods segment the elephants into a distinct branch with one of the first splits in the tree. We can also notice that the other large mammals (Cox, Giraffe, Horse, Okapi, Grey Seal, Human) are closely clustered in both dendrograms, albeit on different sides of the trees.

```
library(ggdendro)
data(mammals)

d = dist(mammals)

h = hclust(d^2, method="ward.D")
ggdendrogram(h) + ggtitle("Method = ward")
```



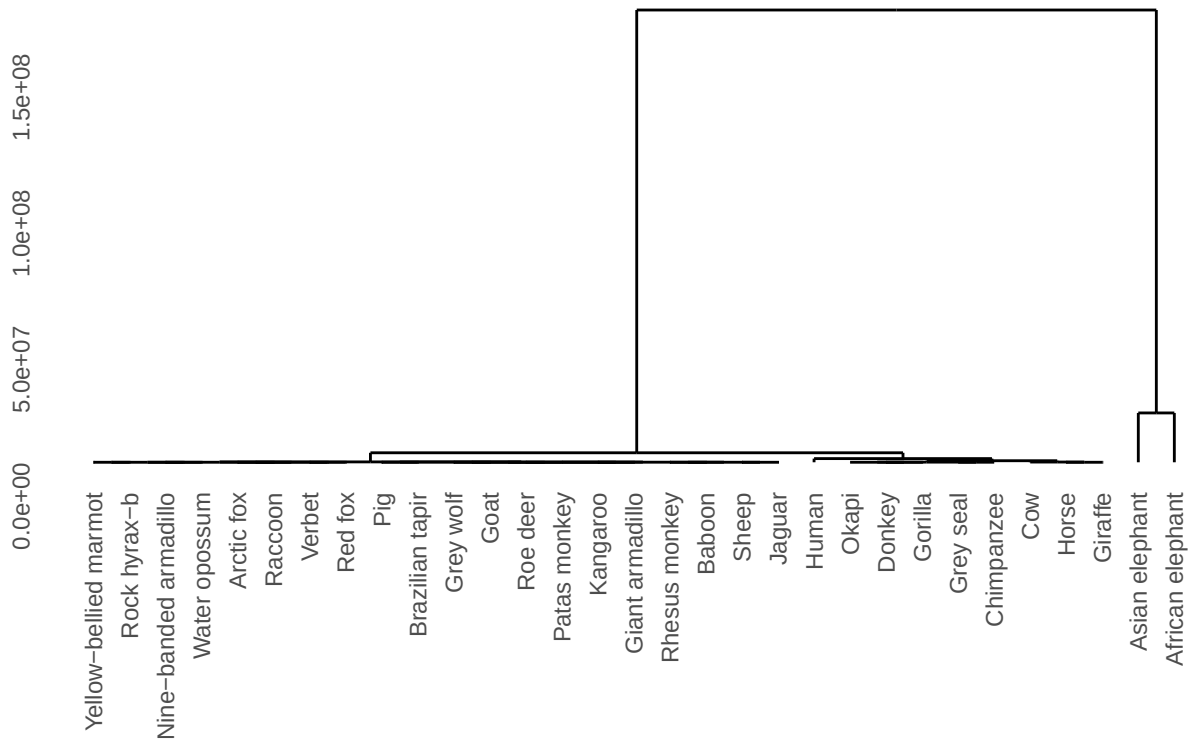

```
h2 = hclust(d, method="complete")
ggdendrogram(h2) + ggtitle("Method = complete")
```



When we subset to the big mammals, very different dendograms in terms of the order of the names. However, there are certain, meaningful similarities, such as the 2 elephants being in the same terminal branch in both dendograms. Looking at the ordering of the distances, we see that the 2 methods have the same first 6, and last 2, mammals in order.

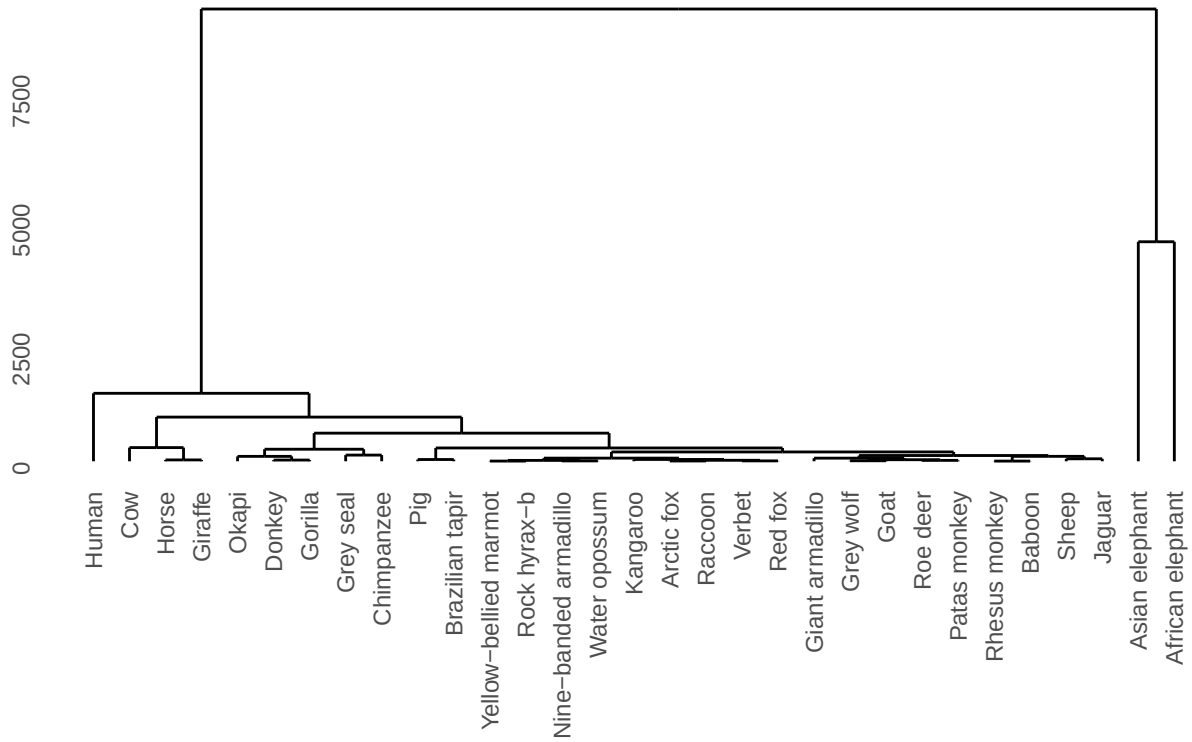
```
big = subset(mammals, subset=(body > median(body)))

d = dist(big)

h = hclust(d^2, method="ward.D")
ggdendrogram(h) + ggtitle("Method = ward")
```

Method = ward




```
h2 = hclust(d, method="complete")
ggdendrogram(h2) + ggtitle("Method = complete")
```

Method = complete



The ordering of the mammal's names is very similar for the 2 methods we tested. While the dendograms look a little different, the underlying orders are quite similar. It is interesting and noteworthy that we see similar outputs for the 2 methods considering that the Ward method uses the squared distance matrix and the Complete method uses the standard distance matrix.

5. Problem 2.15 Albert & Rizzo

After cluster analysis, one is often interested in identifying groups or clusters in the data. In a hierarchical cluster analysis such as in Example 2.10, this corresponds to cutting the dendrogram (e.g. Figure 2.20) at a given level. The cutree function is an easy way to find the corresponding groups. For example, in Example 2.10, we saved the result of our complete-linkage clustering in an object h. To cut the tree to form five groups we use cutree with k=5: $g = cutree(h, 5)$. Display g to see the labels of each observation. Summarize the group sizes using $table(g)$. There are three clusters that have only one mammal. Use $mammals[g > 2]$ to identify which three mammals are singleton clusters.

We will use our clusterings from the full dataset, not the subset of large animals. We split these dendograms into 5 clusters, as instructed, and we go through them serially, noting that the clusters are very, very similar.

```
k <- 5
# ward = h, complete = h2
g <- cutree(h, k)
table(g)
```

```
## g
##  1  2  3  4  5
## 20  8  1  1  1
```

```
g2 <- cutree(h2, k)
table(g2)
```

```
## g2
##  1  2  3  4  5
## 25  3  1  1  1
```

For both the ward and complete dendogram, There are 3 singleton clusters, which are identical for the 2 methods.

```
g[g == 5]
```

```
## African elephant
##                5
```

```
g2[g2 == 5]
```

```
## African elephant
##                5
```

```
g[g == 4]
```

```
## Human
##     4
```

```
g2[g2 == 4]
```

```
## Human
##     4
```

```
g[g == 3]
```

```
## Asian elephant
##              3
```

```
g2[g2 == 3]
```

```
## Asian elephant
##              3
```

The 2nd cluster for the Complete method only has 3 members, all 3 of which are in the 2nd cluster from the Ward method. We also note that these are medium sized mammals.

```
print("Ward")
```

```
## [1] "Ward"
```

```
g[g == 2]
```

```
##        Cow     Donkey      Horse    Giraffe    Gorilla  Grey seal
##          2          2          2          2          2          2
##      Okapi Chimpanzee
##          2          2
```

```
print("Complete")
```

```
## [1] "Complete"
```

```
g2[g2 == 2]
```

```
##     Cow   Horse Giraffe
##       2       2       2
```

Finally, we see significant overlap, albeit different size and orders, of the 1st clusters from each method. These animal are all medium-small sized and, from what I can tell, there is very little variance in the sizes across these species.

```
print("Ward")
```

```
## [1] "Ward"
```

```
g[g == 1]
```

```
##            Arctic fox             Grey wolf                  Goat
##                     1                     1                     1
##              Roe deer                Verbet Nine-banded armadillo
##                     1                     1                     1
##          Patas monkey         Water opossum         Rhesus monkey
##                     1                     1                     1
##             Kangaroo Yellow-bellied marmot                 Sheep
##                     1                     1                     1
##               Jaguar                Baboon       Giant armadillo
##                     1                     1                     1
##          Rock hyrax-b               Raccoon                   Pig
##                     1                     1                     1
##       Brazilian tapir              Red fox
##                     1                     1
```

```
print("Complete")
```

```
## [1] "Complete"
```

```
g2[g2 == 1]
```

```
##            Arctic fox             Grey wolf                  Goat
##                     1                     1                     1
##              Roe deer                Verbet Nine-banded armadillo
##                     1                     1                     1
##                Donkey          Patas monkey               Gorilla
##                     1                     1                     1
##             Grey seal         Water opossum         Rhesus monkey
##                     1                     1                     1
##             Kangaroo Yellow-bellied marmot                 Okapi
##                     1                     1                     1
##                 Sheep                Jaguar            Chimpanzee
##                     1                     1                     1
##                Baboon       Giant armadillo          Rock hyrax-b
##                     1                     1                     1
##               Raccoon                   Pig       Brazilian tapir
##                     1                     1                     1
##               Red fox
##                     1
```

Inspecting the 5 clusters from each method using the cuttree fcuntion reveals that these 2 methods produce nearly identical clusterings of the mammals, despite using different distance metrics in their algorithms (recall ward uses the squared distance matrix). This is a striking result. We could investigate what happens when we cluster into more or less groups, but that is not within the scope of this problem.

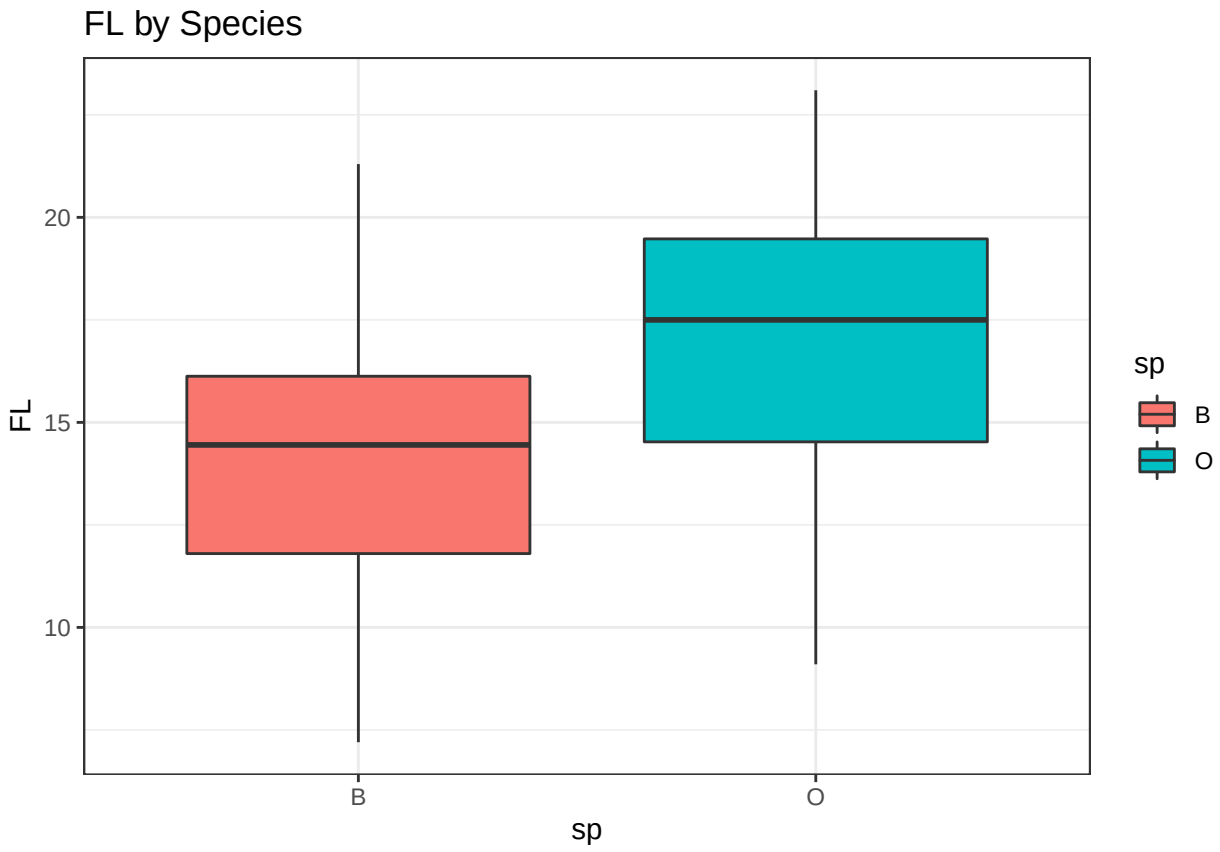
There was a last minute addendum to this question, I concluded that I did not need to change the above response, which I had completed before the change.

6. The dataset crabs from the library MASS contains 200 rows and 8 columns describing 5 morphological measurements on 50 crabs each of two color forms and both sexes, of the species Leptograpsus variegatus. Is there evidence from the morphological data alone of a division of two forms of crabs?

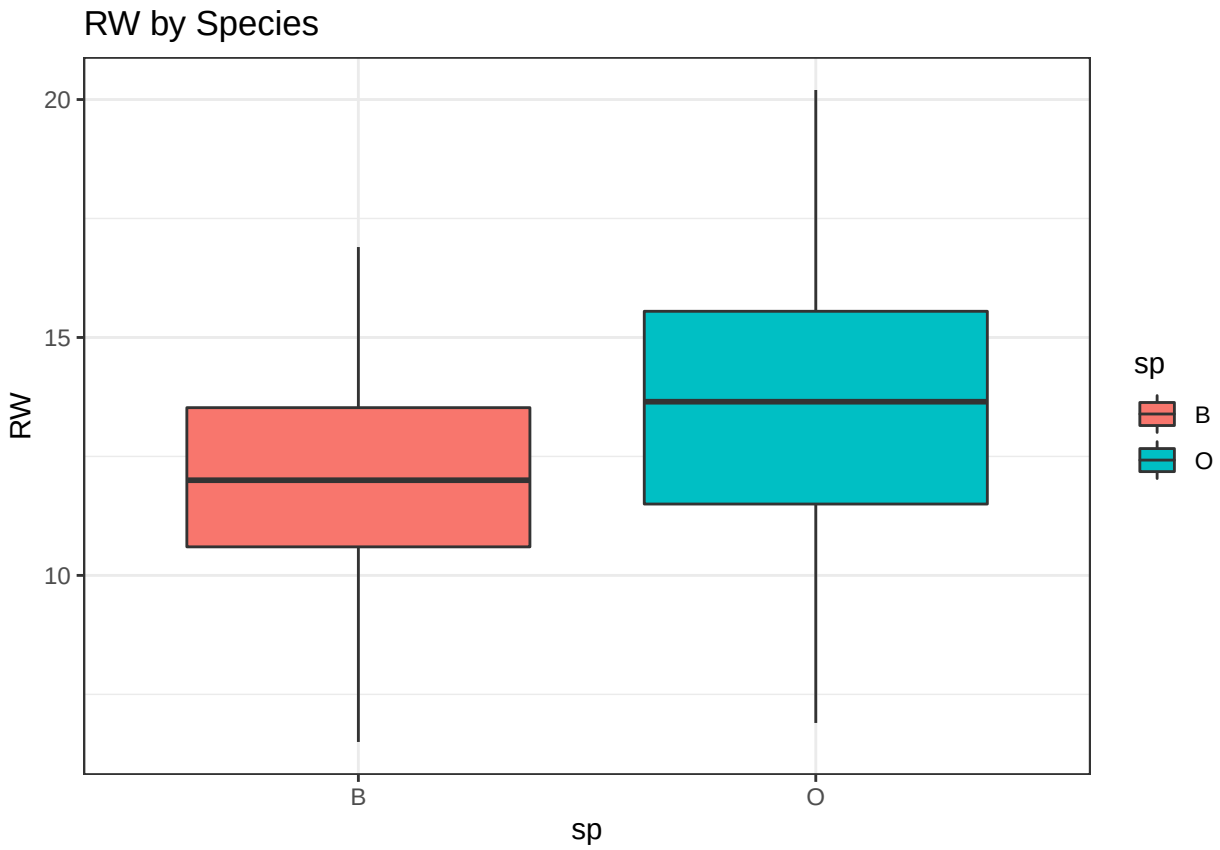
We are interested in seeing if there are natural distinctions between the colors in the morphological measurements. We will perform principal component analysis on these variables and then we will visualize the principal components with the crab form types to see if there is an underlying distinction.

First, some bar plots show indications that there are significant morphological differences across the color types:

```
ggplot(data = crabs, aes(x=sp,y=FL,factor=sp,fill=sp))+geom_boxplot()+theme_bw() +
  ggtitle("FL by Species")
```

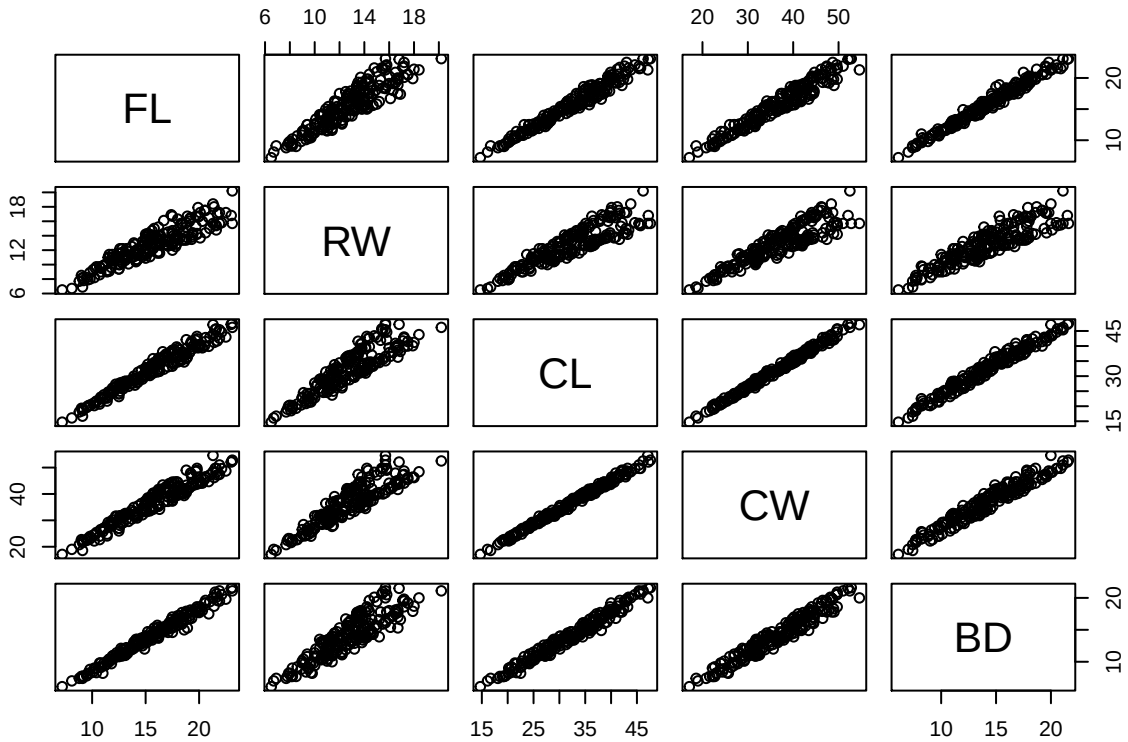



```
ggplot(data = crabs, aes(x=sp,y=RW,factor=sp,fill=sp))+geom_boxplot()+theme_bw() +
  ggtitle("RW by Species")
```



We also see very high multicollinearity in the morphological variables:

```
crabs %>% dplyr::select(-c(sex,index,sp)) %>% pairs()
```



```
pcdata = subset(crabs, select = -c(sp,sex,index))
crabs.pc = princomp(pcdata, cor=TRUE)
summary(crabs.pc)
```

```
## Importance of components:
##                           Comp.1     Comp.2      Comp.3      Comp.4
## Standard deviation     2.188341 0.38946785 0.215946693 0.105524202
## Proportion of Variance 0.957767 0.03033704 0.009326595 0.002227071
## Cumulative Proportion  0.957767 0.98810400 0.997430593 0.999657664
##                             Comp.5
## Standard deviation     0.0413724263
## Proportion of Variance 0.0003423355
## Cumulative Proportion  1.0000000000
```

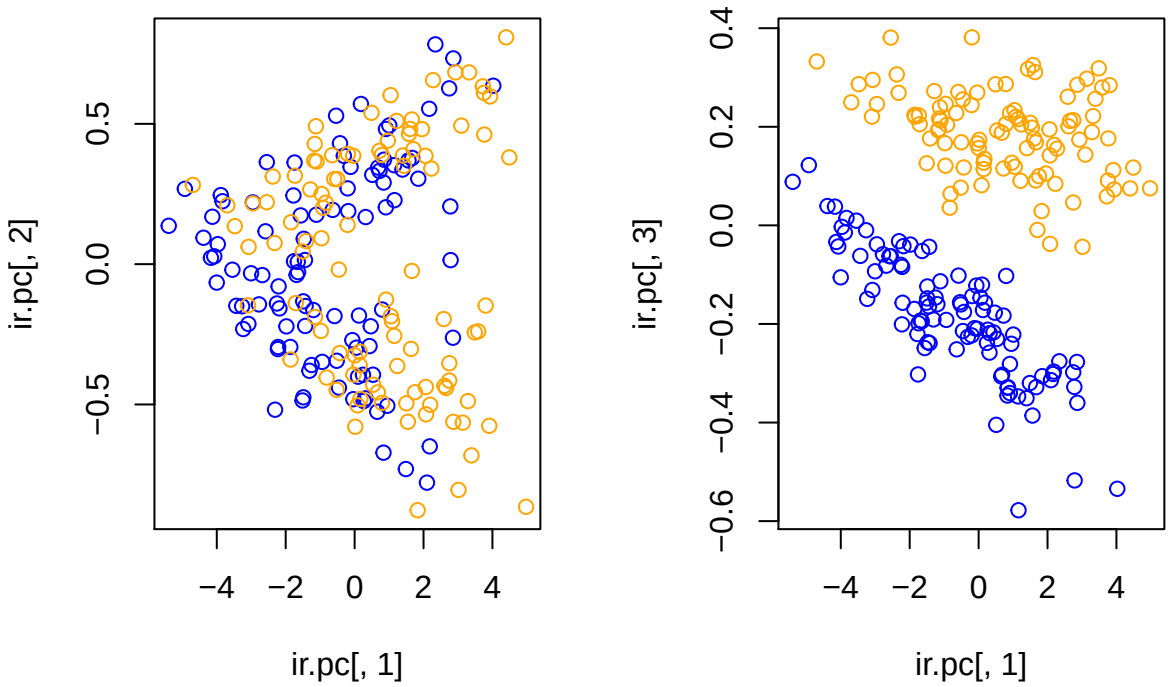
```
crabs.pc$loadings
```

```
##
## Loadings:
##    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## FL  0.452  0.138  0.531  0.697
## RW  0.428 -0.898
## CL  0.453  0.268 -0.310        -0.792
## CW  0.451  0.181 -0.653         0.575
## BD  0.451  0.264  0.443 -0.707  0.176
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## SS loadings       1.0    1.0    1.0    1.0    1.0
## Proportion Var    0.2    0.2    0.2    0.2    0.2
## Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

```
ir.pc <- predict(crabs.pc)
```

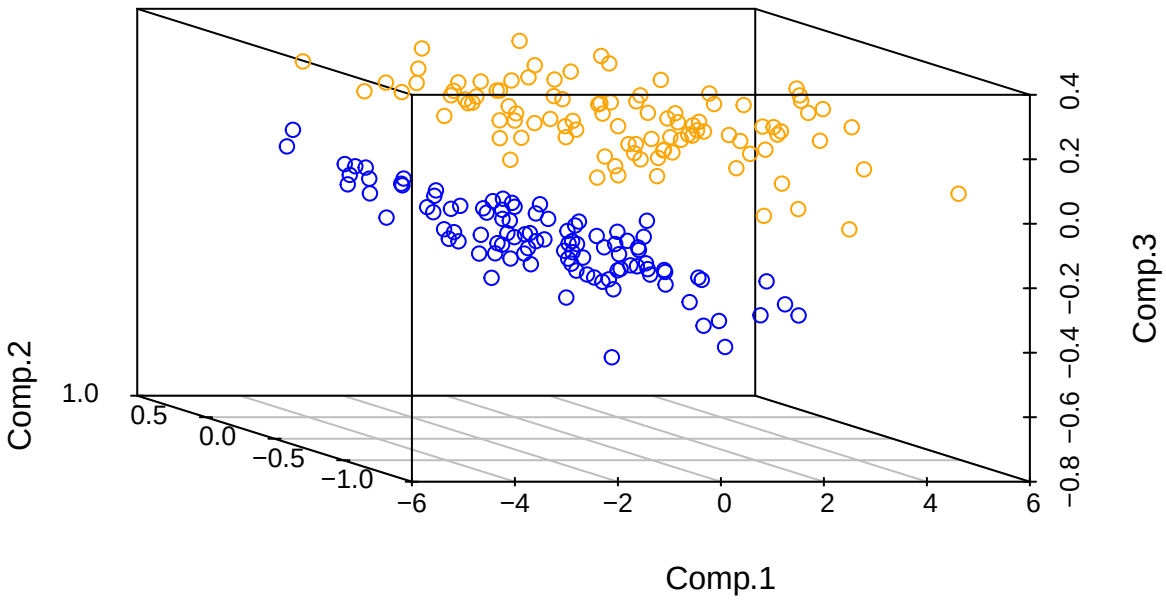
We see that almost 99% of the variance in the 2 groups, which means we lose very little information while reducing the dimentsions from 5 to 3. When we project the data on the plans defined by the principal components, we see that there is a clear distinct between the orange and blue crabs. This distinction is not captured by only the first 2 principal components.

```
par(mfrow = c(1,2))
plot(ir.pc[,1], ir.pc[,2], col = ifelse(crabs$sp == "B", "blue", "orange"))
plot(ir.pc[,1], ir.pc[,3], col = ifelse(crabs$sp == "B", "blue", "orange"))
```



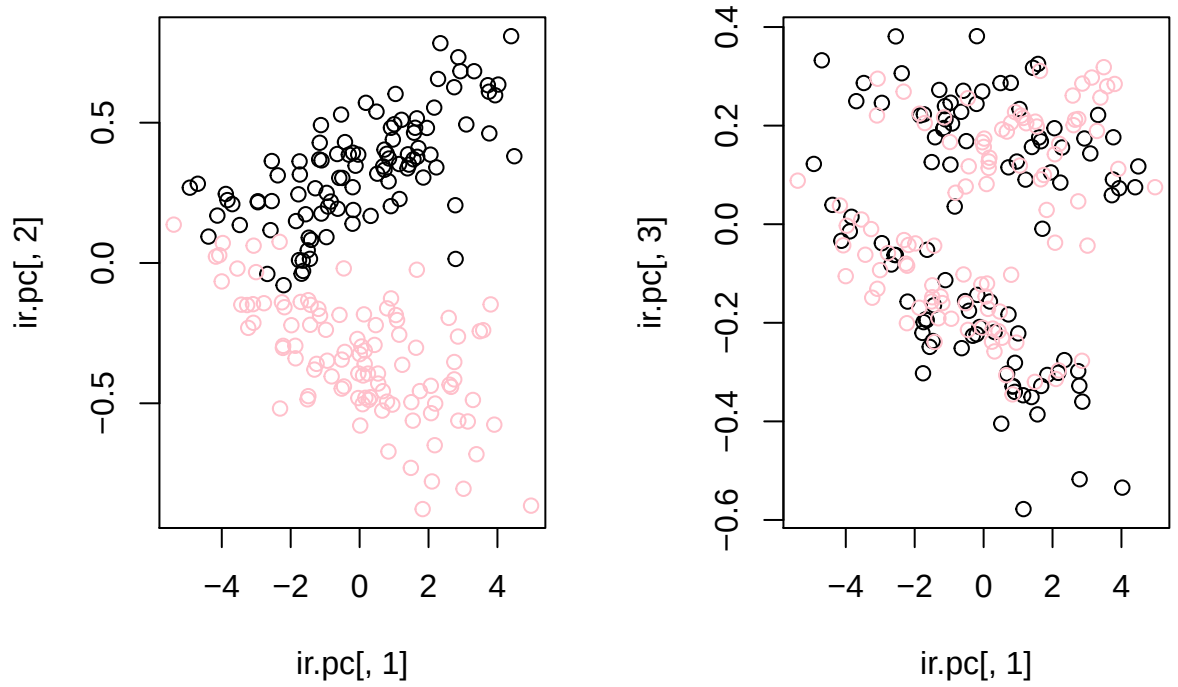
The 3-D scatter plot shows gives an insightful view into the separation of our data across the first 3 principal components.

```
scatterplot3d(ir.pc[,1:3],
              color = ifelse(crabs$sp == "B", "blue", "orange"),
              angle = 150)
```

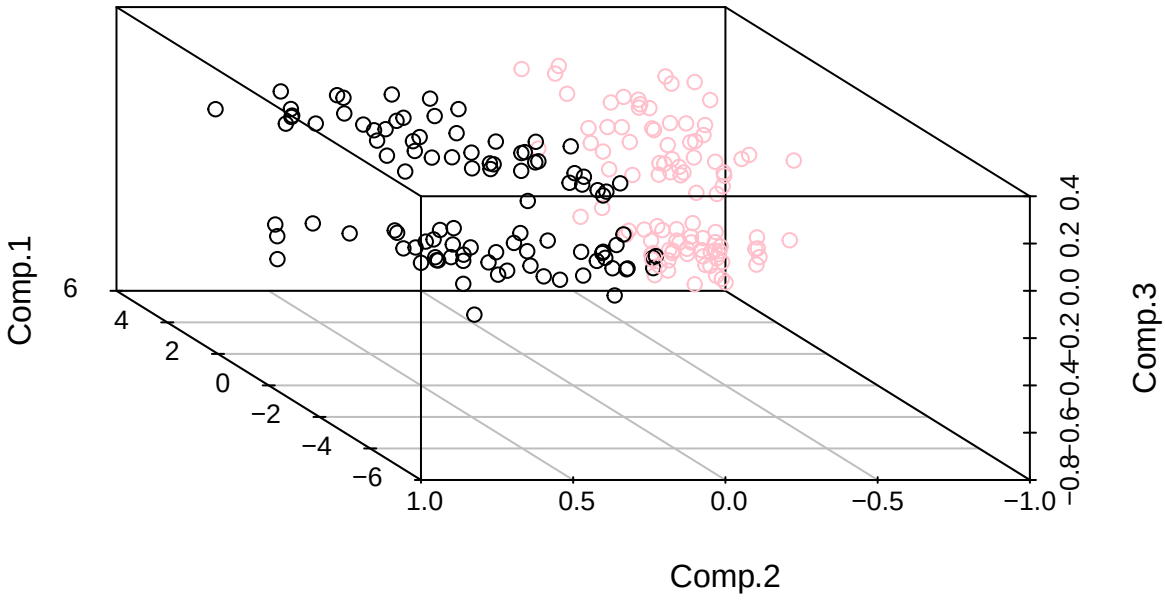


It is also interesting to see that the same principal component analysis captures the distiction between the sexes. In the following plots we see that the first 2 principal components models the distinction between the gender of the crabs.

```
par(mfrow = c(1,2))
plot(ir.pc[,1], ir.pc[,2], col = ifelse(crabs$sex == "M", "black", "pink"))
plot(ir.pc[,1], ir.pc[,3], col = ifelse(crabs$sex == "M", "black", "pink"))
```




```
scatterplot3d(ir.pc[,1:3],
              color = ifelse(crabs$sex == "M", "black", "pink"),
              angle = 300)
```



This is a cool analysis. We started with 5 continuous variables with high multicollinearity. Through PCA, we were able to reduce the data to 3 dimensions that account for 98% of of the variance and are useful in distinguishing between both sex and color of the crabs. Given the morphological dimensions of any crabs, we should be confident in our ability to classify it.