4211 Homework 6

Matthew DeSantis

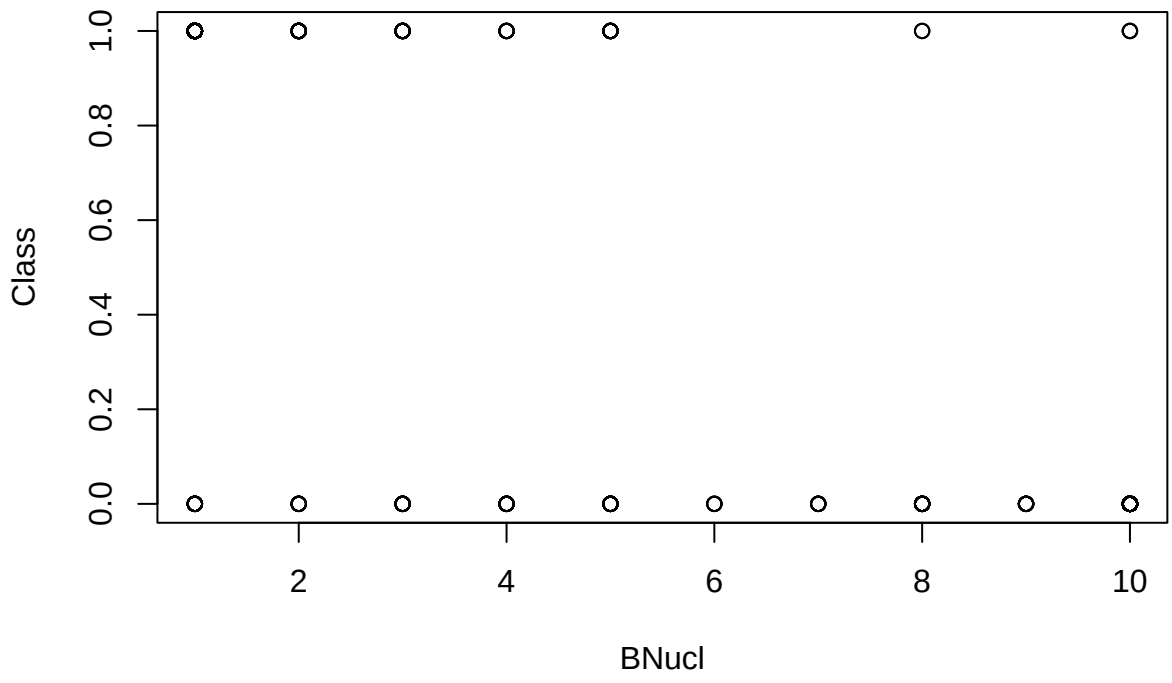
2023-03-23

1

(a)

i

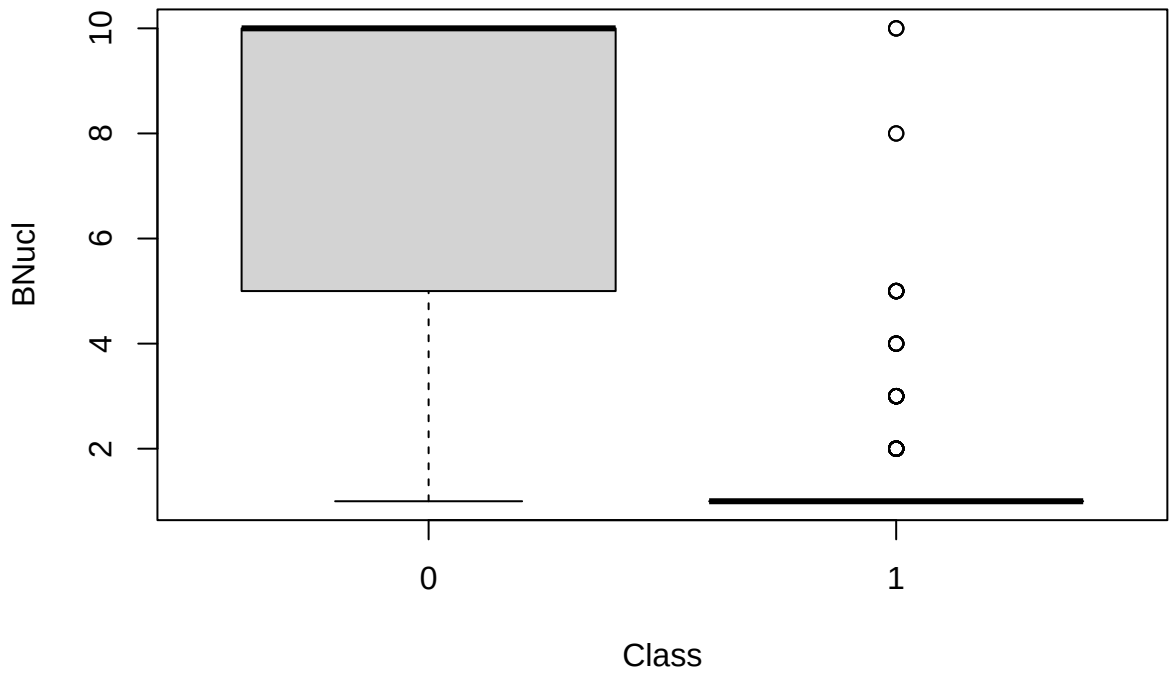
```
data("wbca")
plot(Class ~ BNucl, wbca)
```



Because "Class" is a binary variable, and BNucl only takes set values, plotting them in a traditional scatterplot does not produce informative results, as seen by the output. It is difficult to determine any trend from this.

ii

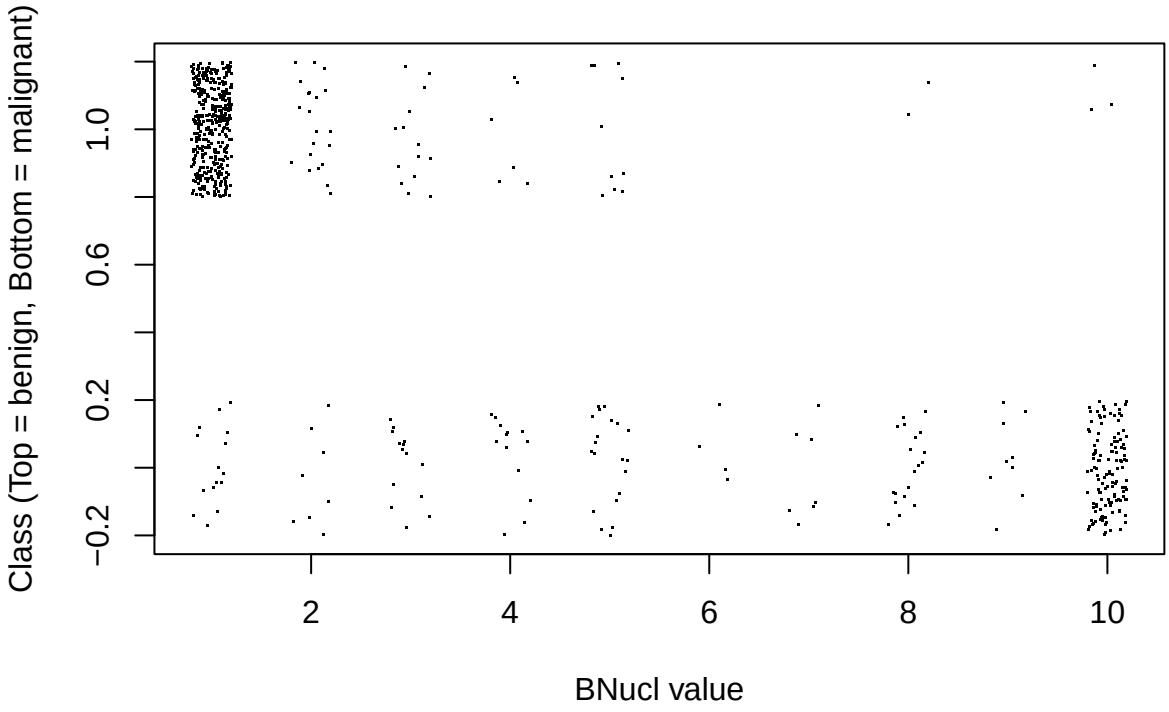
```
boxplot(BNucl~Class, wbca)
```



The boxplots show that the majority of those with malignant tumors have BNucl values from 10-5, although there is some spread, going all the way down to 0. Those with benign tumors have BNucl values much more tightly packed at 0, with a few outliers spanning the range of values.

iii

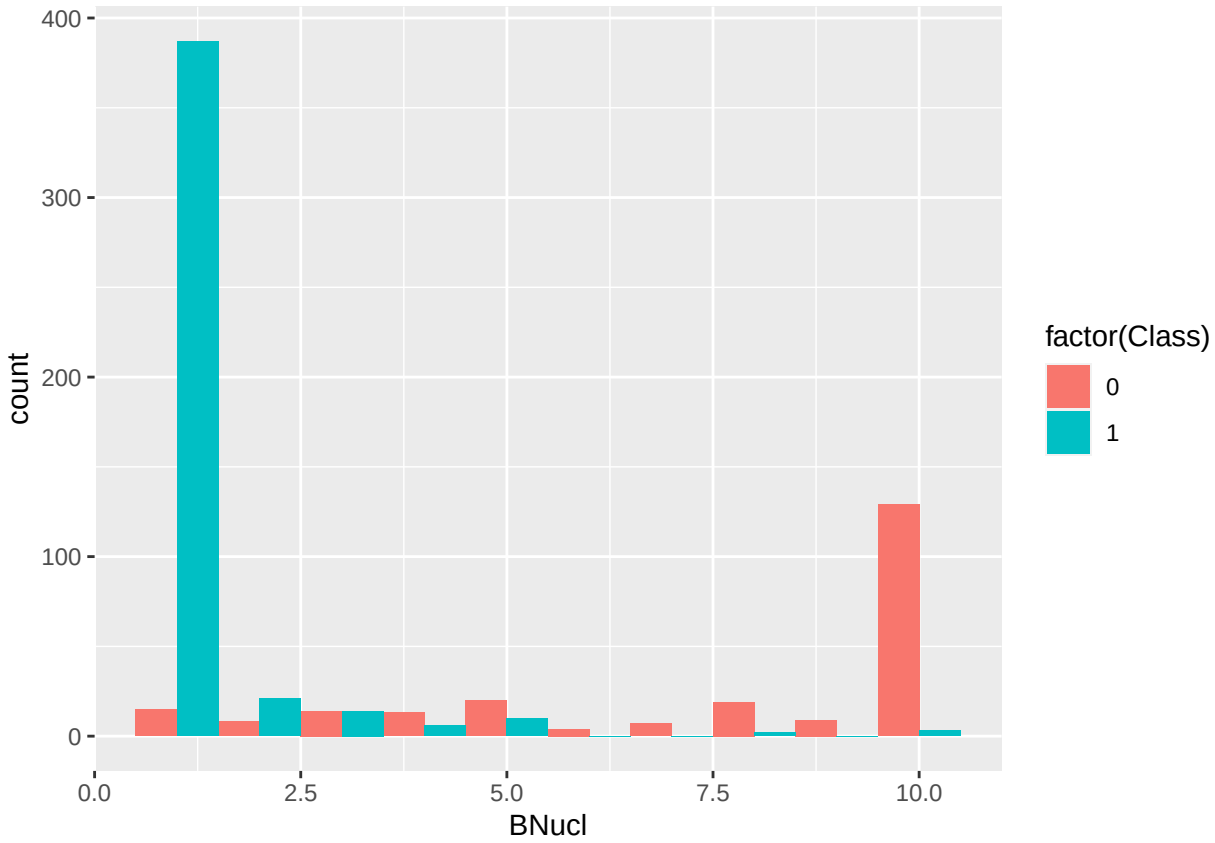
```
plot(jitter(Class) ~ jitter(BNucl), data = wbca, pch = ".", xlab = "BNucl value", ylab = "Class (Top = M
```



This plot has much the same interpretation as the boxplot, with malignant tumors having BNucl values focused at ten, with some spread in all values, and benign tumors being much more tightly packed at 0, with some outliers. This chart is a little more informative, perhaps, as it allows you to see the distribution for each value at a glance.

iv

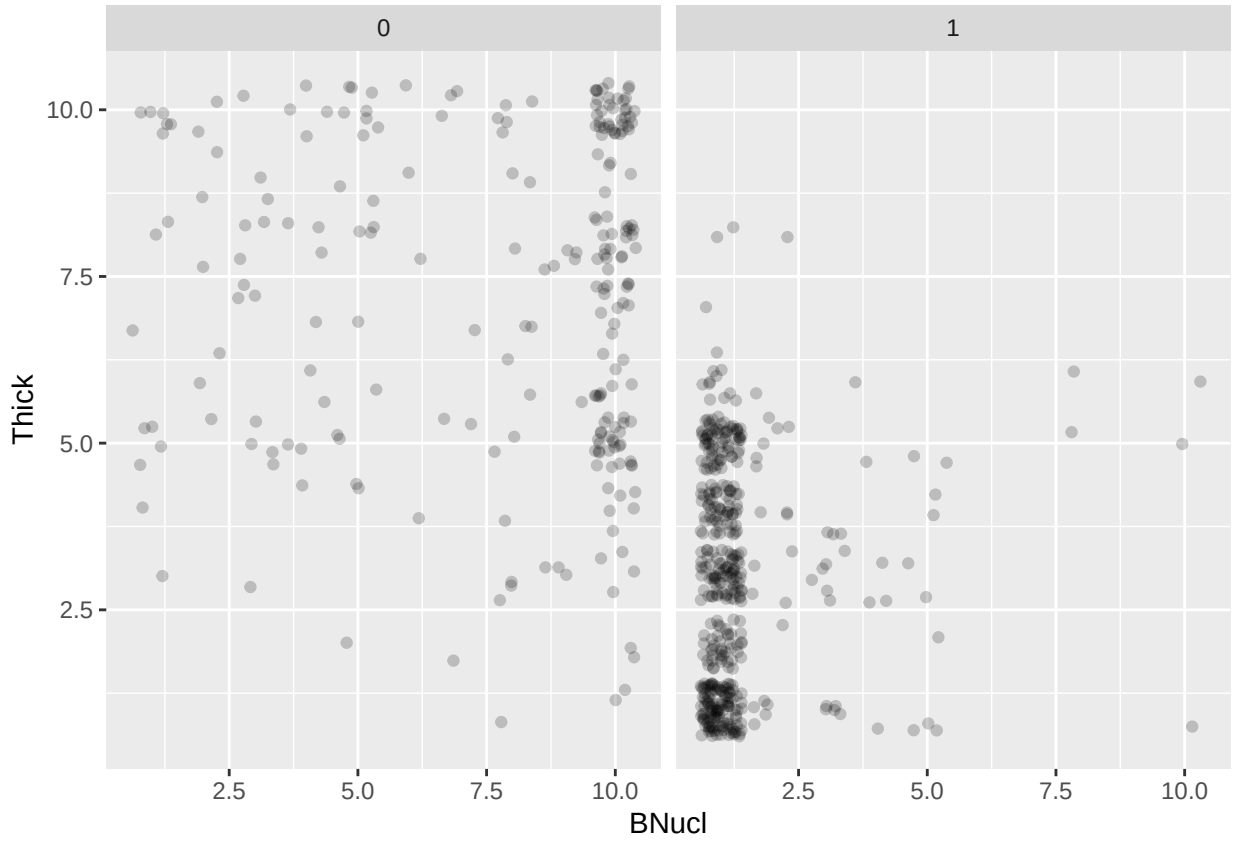
```
ggplot(wbca, aes(x = BNucl, fill = factor(Class))) + geom_histogram(bins = 10, position = "dodge")
```



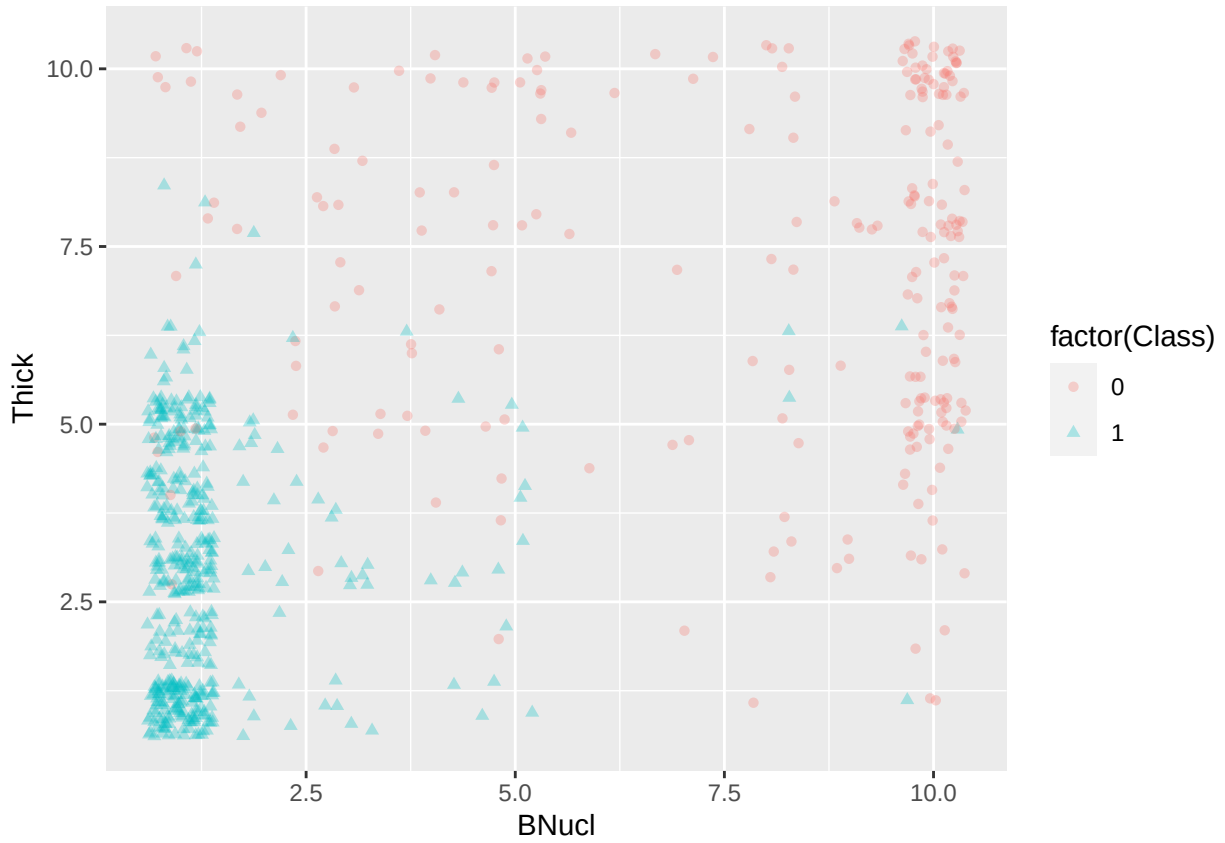
Once again, the interpretation of the chart is mostly the same as the other two. One advantage of this chart is that it makes it visually obvious that there were more benign tumors than malignant ones.

(b)

```
ggplot(wbca, aes(x=BNucl, y = Thick)) + geom_point(alpha = 0.2, position = position_jitter()) + facet_g
```




```
ggplot(wbca, aes(x=BNucl, y = Thick, shape = factor(Class), color = factor(Class))) + geom_point(alpha =
```



The first plot is much more readable than the second. As for interpretation, it appears that BNucl is much more useful for predicting Class, though Thick does also appear to have some correlation.

(c)

```
mod1 = glm(Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick + UShap + USize, family = binor
summary(mod1)
```

```
##
## Call:
## glm(formula = Class ~ Adhes + BNucl + Chrom + Epith + Mitos +
##     NNucl + Thick + UShap + USize, family = binomial, data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.48282  -0.01179   0.04739   0.09678   3.06425
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.16678    1.41491   7.892 2.97e-15 ***
## Adhes       -0.39681    0.13384  -2.965  0.00303 **
## BNucl       -0.41478    0.10230  -4.055 5.02e-05 ***
## Chrom       -0.56456    0.18728  -3.014  0.00257 **
## Epith       -0.06440    0.16595  -0.388  0.69795
```

```
## Mitos       -0.65713    0.36764  -1.787  0.07387 .
## NNucl       -0.28659    0.12620  -2.271  0.02315 *
## Thick       -0.62675    0.15890  -3.944 8.01e-05 ***
## UShap       -0.28011    0.25235  -1.110  0.26699
## USize        0.05718    0.23271   0.246  0.80589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.464  on 671  degrees of freedom
## AIC: 109.46
##
## Number of Fisher Scoring iterations: 8
```

```
diff = mod1$null.deviance - mod1$deviance
df = mod1$df.null - mod1$df.residual
1 - pchisq(diff, df)
```

```
## [1] 0
```

We can use the difference between the residual deviance of the full model and the null model to test whether the model with predictors included is better than without. However, this is not a very good test, as it will always favor more predictors, regardless of how correlated they are.

(d)

```
mod2 = step(mod1)
```

```
## Start:  AIC=109.46
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
##     UShap + USize
##
##         Df Deviance    AIC
## - USize  1   89.523 107.52
## - Epith  1   89.613 107.61
## - UShap  1   90.627 108.63
## <none>       89.464 109.46
## - Mitos  1   93.551 111.55
## - NNucl  1   95.204 113.20
## - Adhes  1   98.844 116.84
## - Chrom  1   99.841 117.84
## - BNucl  1  109.000 127.00
## - Thick  1  110.239 128.24
##
## Step:  AIC=107.52
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
##     UShap
##
##         Df Deviance    AIC
```

```
## - Epith  1   89.662 105.66
## - UShap  1   91.355 107.36
## <none>       89.523 107.52
## - Mitos  1   93.552 109.55
## - NNucl  1   95.231 111.23
## - Adhes  1   99.042 115.04
## - Chrom  1  100.153 116.15
## - BNucl  1  109.064 125.06
## - Thick  1  110.465 126.47
##
## Step:  AIC=105.66
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap
##
##         Df Deviance    AIC
## <none>       89.662 105.66
## - UShap  1   91.884 105.88
## - Mitos  1   93.714 107.71
## - NNucl  1   95.853 109.85
## - Adhes  1  100.126 114.13
## - Chrom  1  100.844 114.84
## - BNucl  1  109.762 123.76
## - Thick  1  110.632 124.63
```

```
summary(mod2)
```

```
##
## Call:
## glm(formula = Class ~ Adhes + BNucl + Chrom + Mitos + NNucl +
##     Thick + UShap, family = binomial, data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.44161  -0.01119   0.04962   0.09741   3.08205
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.0333     1.3632   8.094 5.79e-16 ***
## Adhes        -0.3984     0.1294  -3.080  0.00207 **
## BNucl        -0.4192     0.1020  -4.111 3.93e-05 ***
## Chrom        -0.5679     0.1840  -3.085  0.00203 **
## Mitos        -0.6456     0.3634  -1.777  0.07561 .
## NNucl        -0.2915     0.1236  -2.358  0.01837 *
## Thick        -0.6216     0.1579  -3.937 8.27e-05 ***
## UShap        -0.2541     0.1785  -1.423  0.15461
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.662  on 673  degrees of freedom
## AIC: 105.66
##
## Number of Fisher Scoring iterations: 8
```

(e)

```
dafunc = function(data, mod, cutoff = 0.5, trues = FALSE){
  mod$diag = ifelse(mod$fitted.values>=cutoff, 1, 0)
  mod$correct = data == mod$diag

  if(trues){
    truepositives = sum((mod$diag == 0) & (mod$correct == TRUE))
    truenegatives = sum((mod$diag == 1) & (mod$correct == TRUE))
    return(c(truepositives, truenegatives))
  }

  falsepositives = sum((mod$diag == 0) & (mod$correct == FALSE))
  falsenegatives = sum((mod$diag == 1) & (mod$correct == FALSE))

  return(c(falsepositives, falsenegatives))
}
```

```
dafunc(wbca$Class, mod2)
```

```
## [1]  9 11
```

(f)

```
dafunc(wbca$Class, mod2, 0.9)
```

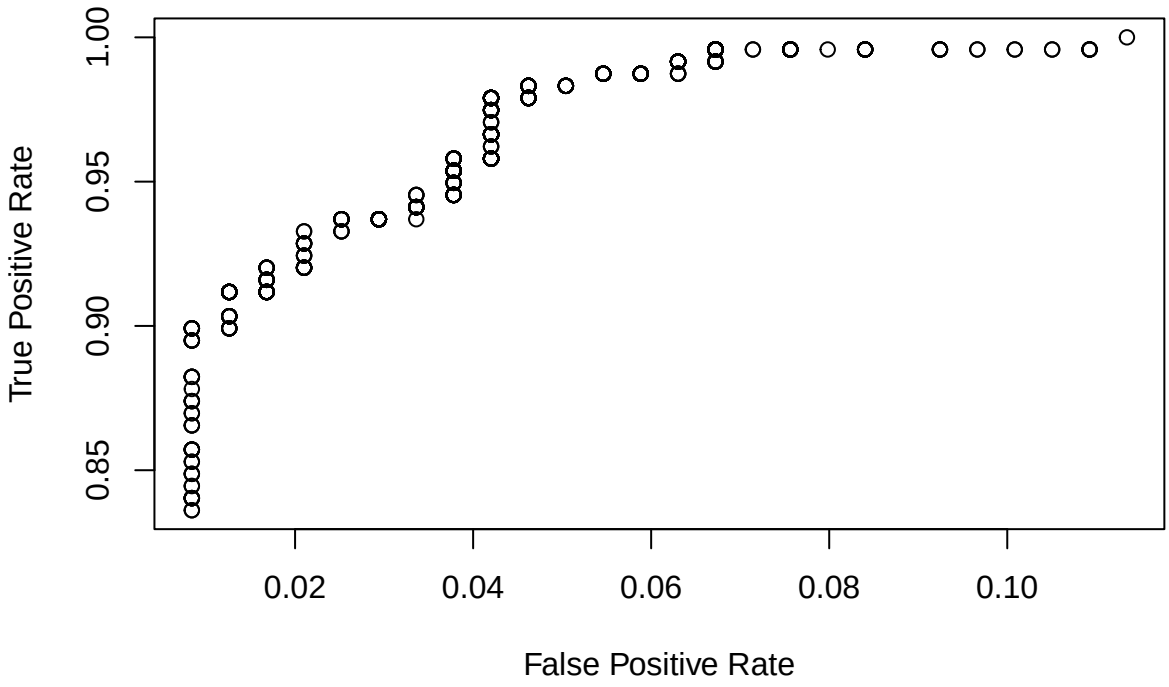
```
## [1] 16  1
```

(g)

```
dafunk = function(data, mod, cutoff = 0.5){
  truepos = dafunc(data, mod, cutoff, trues = TRUE)[1]
  falsepos = dafunc(data, mod, cutoff)[1]
  size = length(data)
  ben = sum(data)
  pos = size-ben
  tpr = truepos/pos
  fpr = falsepos/pos
  return(c(fpr, tpr))

}
```

```
x = seq(0.05,0.95,by = 0.001)
ROC = lapply(x, dafunk, data = wbca$Class, mod = mod2)
ROC = as.data.frame(do.call(rbind, ROC))
plot(ROC, xlab = "False Positive Rate", ylab = "True Positive Rate")
```



(h)

```
wbca$test = ((1:nrow(wbca))%%3)==0
wbcatrain = wbca[wbca$test==FALSE,]
wbcatest = wbca[wbca$test==TRUE,]
mod3 = step(glm(Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick + UShap + USize, family =
```

```
## Start:  AIC=77.65
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
##     UShap + USize
##
##         Df Deviance    AIC
## - Epith  1   58.340 76.340
## - USize  1   58.880 76.880
## <none>       57.651 77.651
## - Mitos  1   60.712 78.712
## - UShap  1   61.450 79.450
## - Chrom  1   65.983 83.983
## - BNucl  1   67.373 85.373
## - NNucl  1   67.538 85.538
## - Adhes  1   68.073 86.073
## - Thick  1   71.162 89.162
##
```

```
## Step:  AIC=76.34
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap +
##     USize
##
##         Df Deviance    AIC
## - USize  1   59.536 75.536
## <none>       58.340 76.340
## - Mitos  1   61.264 77.264
## - UShap  1   61.702 77.702
## - Chrom  1   66.515 82.515
## - BNucl  1   67.402 83.402
## - NNucl  1   67.556 83.556
## - Adhes  1   68.310 84.310
## - Thick  1   72.311 88.311
##
## Step:  AIC=75.54
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap
##
##         Df Deviance    AIC
## <none>       59.536 75.536
## - UShap  1   61.894 75.894
## - Mitos  1   62.329 76.329
## - Chrom  1   66.762 80.762
## - NNucl  1   67.576 81.576
## - BNucl  1   68.332 82.332
## - Adhes  1   68.359 82.359
## - Thick  1   72.363 86.363
```

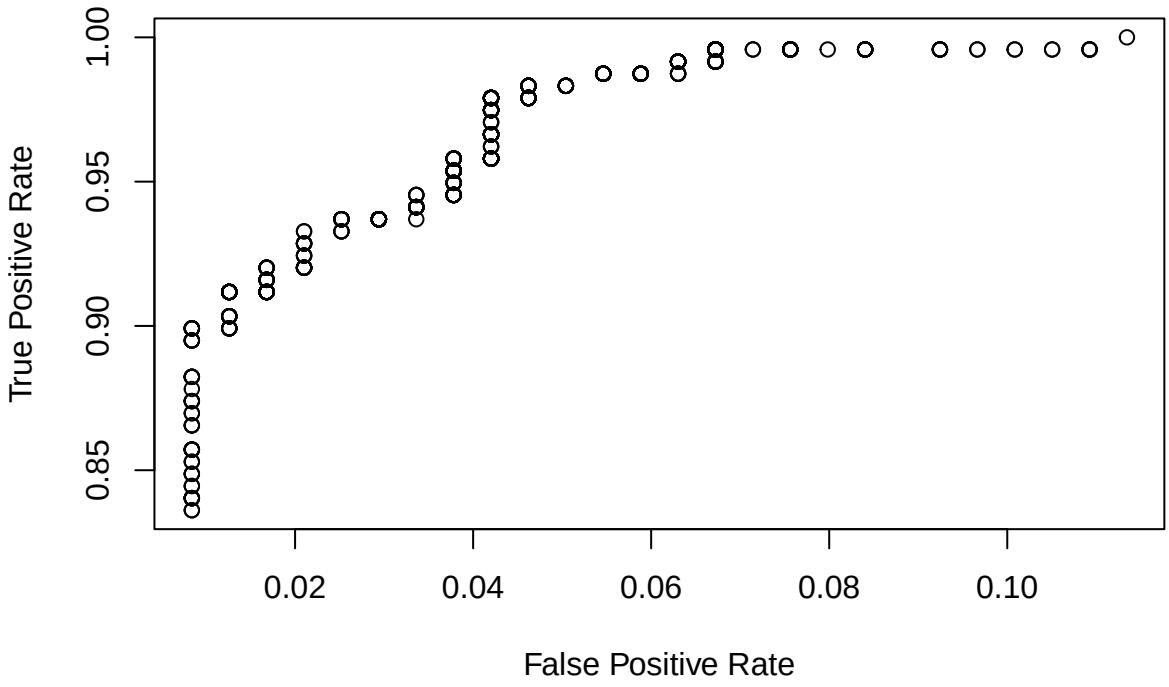
```
dafunc2(wbcatest, mod2)
```

```
## [1] 2 4
```

```
dafunc2(wbcatest, mod2,0.9)
```

```
## [1] 2 1
```

```
x = seq(0.05,0.95,by = 0.001)
ROC2 = lapply(x, dafunk, data = wbca$Class, mod = mod2)
ROC2 = as.data.frame(do.call(rbind, ROC2))
plot(ROC2, xlab = "False Positive Rate", ylab = "True Positive Rate")
```

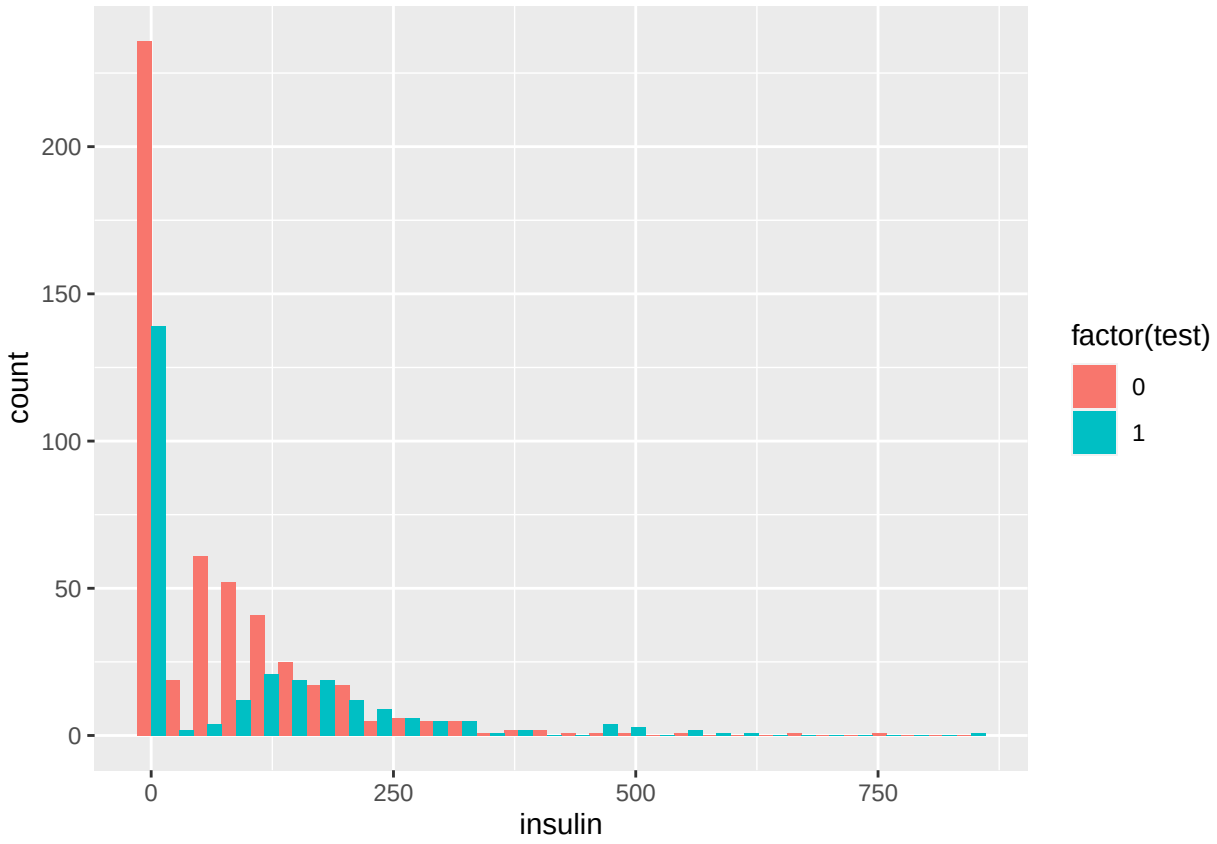


The outcome seems more or less the same. There are fewer errors, but this is because we are testing it with less data.

2

(a)

```
data(pima)
ggplot(pima, aes(x = insulin, fill = factor(test))) + geom_histogram(bins = 30, position = "dodge")
```

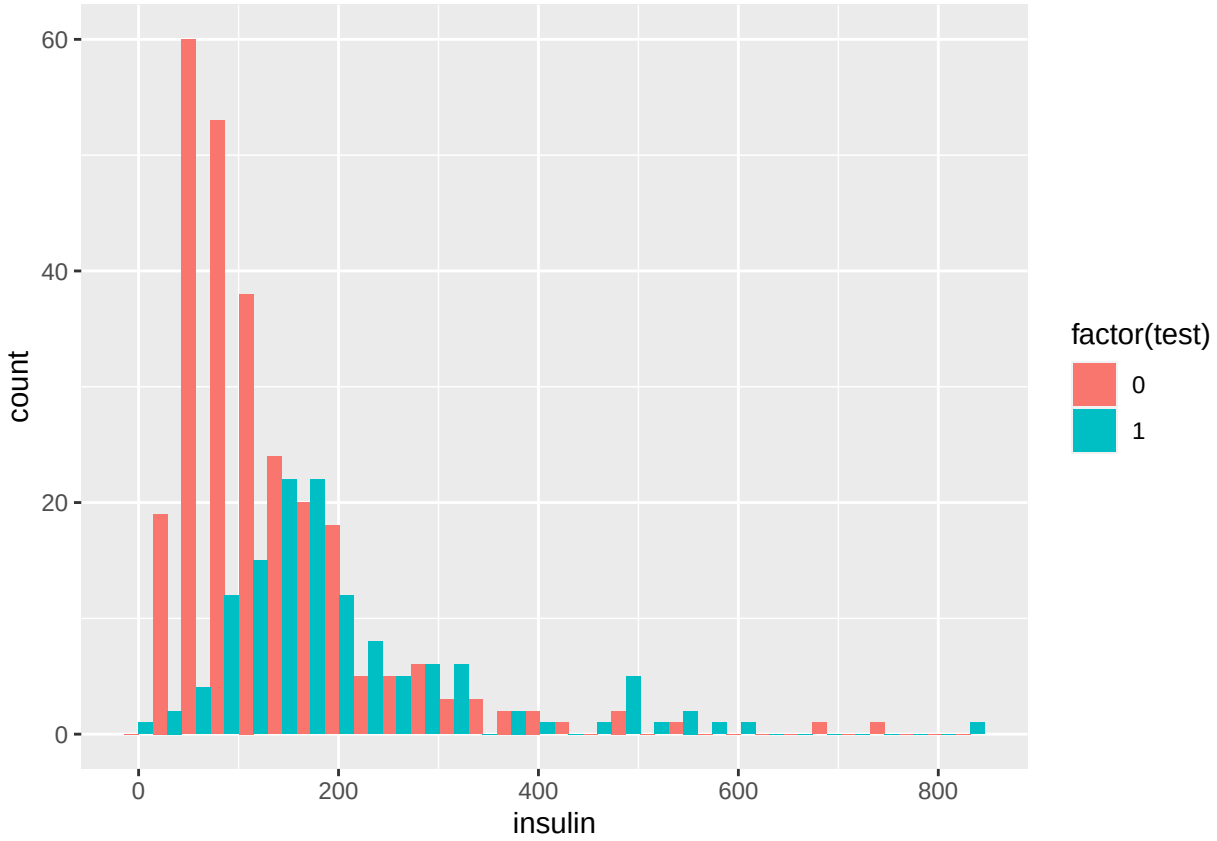


There appears to be many measurements of 0, which shouldn't be possible for this kind of test.

(b)

```
pima["insulin"][pima["insulin"] == 0] = NA
ggplot(pima, aes(x = insulin, fill = factor(test))) + geom_histogram(bins = 30, position = "dodge")
```

```
## Warning: Removed 374 rows containing non-finite values (‘stat_bin()‘).
```



After removing the values, we have a much more reasonable looking graph. It appears that showing signs of diabetes is associated with higher insulin levels.

(c)

```
pima["diastolic"][pima["diastolic"] == 0] = NA
pima["triceps"][pima["triceps"] == 0] = NA
pmod1 = glm(test ~ pregnant+glucose+diastolic+triceps+insulin+bmi+diabetes+age, family = binomial, data
summary(pmod1)
```

```
##
## Call:
## glm(formula = test ~ pregnant + glucose + diastolic + triceps +
##     insulin + bmi + diabetes + age, family = binomial, data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7742  -0.6593  -0.3615   0.6385   2.5617
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.007e+01  1.215e+00  -8.286  < 2e-16 ***
## pregnant     8.253e-02  5.543e-02   1.489  0.13650
## glucose      3.829e-02  5.769e-03   6.637 3.21e-11 ***
```

```
## diastolic   -1.563e-03  1.182e-02  -0.132  0.89476
## triceps      1.083e-02  1.702e-02   0.636  0.52482
## insulin     -8.299e-04  1.307e-03  -0.635  0.52538
## bmi          7.187e-02  2.687e-02   2.675  0.00748 **
## diabetes     1.129e+00  4.250e-01   2.658  0.00787 **
## age          3.415e-02  1.837e-02   1.859  0.06301 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.70  on 393  degrees of freedom
## Residual deviance: 344.14  on 385  degrees of freedom
##   (374 observations deleted due to missingness)
## AIC: 362.14
##
## Number of Fisher Scoring iterations: 5
```

374 observations were removed, leaving 394 with which to fit the model. The removed observations were removed because of the presence of NA values in them.

(d)

```
pima2 = drop_na(pima)
pmod2 = glm(test ~ pregnant+glucose+diastolic+triceps+insulin+bmi+diabetes+age, family = binomial, data
pmod3 = glm(test ~ pregnant+glucose+diastolic+bmi+diabetes+age, family = binomial, data = pima2)

diff2 = pmod3$deviance - pmod2$deviance
df2 = pmod3$df.residual - pmod2$df.residual
1 - pchisq(diff2, df2)
```

```
## [1] 0.658784
```

In order to fit both models with the same dataset, I dropped all rows containing NA values, then fit my models with that dataset. I then compared them with a Chisquare test. Because the result is insignificant, we are justified in dropping insulin and triceps.

(e)

```
pmod4 = step(pmod2)
```

```
## Start:  AIC=362.14
## test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age
##
##             Df Deviance    AIC
## - diastolic  1   344.16 360.16
## - insulin    1   344.55 360.55
```

```
## - triceps    1   344.55 360.55
## <none>           344.14 362.14
## - pregnant   1   346.38 362.38
## - age        1   347.72 363.72
## - diabetes   1   351.62 367.62
## - bmi        1   351.77 367.77
## - glucose    1   397.20 413.20
##
## Step:  AIC=360.16
## test ~ pregnant + glucose + triceps + insulin + bmi + diabetes +
##     age
##
##            Df Deviance    AIC
## - insulin   1   344.55 358.55
## - triceps   1   344.56 358.56
## <none>          344.16 360.16
## - pregnant  1   346.39 360.39
## - age       1   347.77 361.77
## - diabetes  1   351.71 365.71
## - bmi       1   352.15 366.15
## - glucose   1   397.54 411.54
##
## Step:  AIC=358.55
## test ~ pregnant + glucose + triceps + bmi + diabetes + age
##
##            Df Deviance    AIC
## - triceps   1   344.99 356.99
## <none>          344.55 358.55
## - pregnant  1   346.89 358.89
## - age       1   348.04 360.04
## - diabetes  1   351.94 363.94
## - bmi       1   352.18 364.18
## - glucose   1   411.40 423.40
##
## Step:  AIC=356.99
## test ~ pregnant + glucose + bmi + diabetes + age
##
##            Df Deviance    AIC
## <none>          344.99 356.99
## - pregnant  1   347.35 357.35
## - age       1   348.84 358.84
## - diabetes  1   352.75 362.75
## - bmi       1   361.58 371.58
## - glucose   1   412.11 422.11
```

```
summary(pmod4)
```

```
##
## Call:
## glm(formula = test ~ pregnant + glucose + bmi + diabetes + age,
##     family = binomial, data = pima2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.8772  -0.6514  -0.3641   0.6483   2.5819
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.016534   1.083354  -9.246  < 2e-16 ***
## pregnant      0.084233   0.055037   1.530 0.125896
## glucose       0.036462   0.004978   7.325 2.38e-13 ***
## bmi           0.078848   0.020399   3.865 0.000111 ***
## diabetes      1.141246   0.422040   2.704 0.006849 **
## age           0.034447   0.017809   1.934 0.053086 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.70  on 393  degrees of freedom
## Residual deviance: 344.99  on 388  degrees of freedom
## AIC: 356.99
##
## Number of Fisher Scoring iterations: 5
```

pregnant, glucose, bmi, diabetes, and age are selected. 394 observations.

(f)

```
data(pima)
pima["insulin"][pima["insulin"] == 0] = NA
pima["diastolic"][pima["diastolic"] == 0] = NA
pima["triceps"][pima["triceps"] == 0] = NA

missing = c()
for(x in 1:nrow(pima)){
  miss = all(!is.na(pima[x,]))
  missing = append(missing, miss)
}
pima$not_missing = missing

pmod5 = glm(test ~ not_missing, family = binomial, data = pima)
summary(pmod5)
```

```
##
## Call:
## glm(formula = test ~ not_missing, family = binomial, data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9596  -0.9596  -0.8949   1.4121   1.4892
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.5366     0.1072  -5.007 5.52e-07 ***
```

```
## not_missingTRUE  -0.1718     0.1515  -1.134    0.257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.48  on 767  degrees of freedom
## Residual deviance: 992.20  on 766  degrees of freedom
## AIC: 996.2
##
## Number of Fisher Scoring iterations: 4
```

Missingness is not associated with the test result.

```
pmod5 = glm(test ~ pregnant + glucose + bmi + diabetes + age , family = binomial, data = pima)
summary(pmod5)
```

```
##
## Call:
## glm(formula = test ~ pregnant + glucose + bmi + diabetes + age,
##     family = binomial, data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7147  -0.7336  -0.4255   0.7397   2.8533
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.673124   0.689980 -12.570  < 2e-16 ***
## pregnant     0.119458   0.031653   3.774 0.000161 ***
## glucose      0.032893   0.003403   9.667  < 2e-16 ***
## bmi          0.079550   0.013810   5.760  8.4e-09 ***
## diabetes     0.891487   0.292239   3.051 0.002284 **
## age          0.012230   0.009085   1.346 0.178247
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.48  on 767  degrees of freedom
## Residual deviance: 732.51  on 762  degrees of freedom
## AIC: 744.51
##
## Number of Fisher Scoring iterations: 5
```

Because we showed that missingness has no association with the test value, we can add in the data that had missing values.

(g)

```r
quantile(pima$bmi)
```

```
##   0%  25%  50%  75% 100%
##  0.0 27.3 32.0 36.6 67.1
```

```r
beta = 0.079550
inte = c(confint(pmod5)[4],confint(pmod5)[10])
```

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

```r
lower = inte[1]
upper = inte[2]

exp(beta*27.3) - exp(beta*36.6)
```

```
## [1] -9.611589
```

```r
c(exp(lower*27.3) - exp(lower*36.6), exp(upper*27.3) - exp(upper*36.6))
```

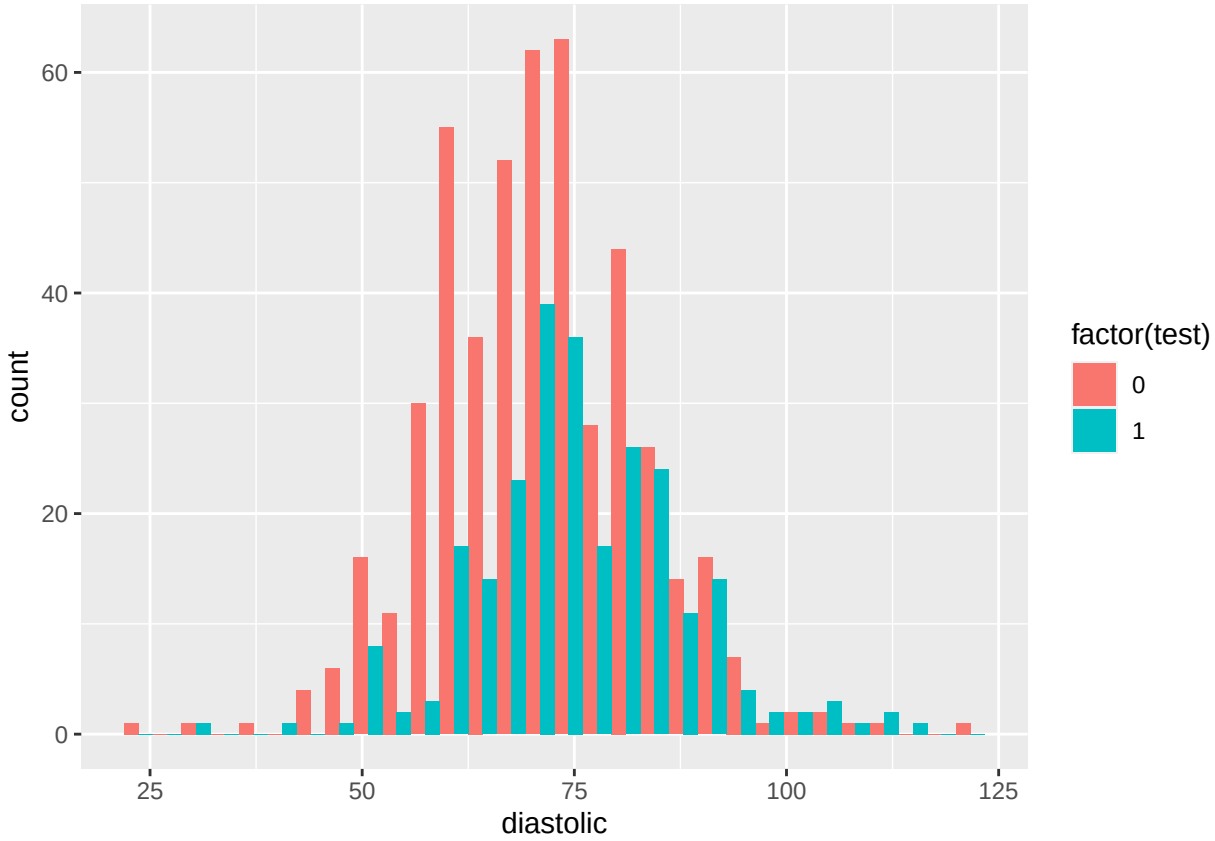
```
## [1]  -2.725706 -32.066276
```

(h)

```r
ggplot(pima, aes(x = diastolic, fill = factor(test))) + geom_histogram(position = "dodge")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 35 rows containing non-finite values (`stat_bin()`).
```




```
t.test(pima[pima$test==0 & !is.na(pima$diastolic),"diastolic"], pima[pima$test==1 & !is.na(pima$diastoli
```

```
##
##  Welch Two Sample t-test
##
## data:  pima[pima$test == 0 & !is.na(pima$diastolic), "diastolic"] and pima[pima$test == 1 & !is.na(pi
## t = -4.6643, df = 504.72, p-value = 3.972e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.316023 -2.572156
## sample estimates:
## mean of x mean of y
##  70.87734  75.32143
```

As seen by the graph and test, women who test positive do have higher diastolic blood pressure. However, the diastolic blood pressure is not significant in the model. Significance in the regression is calculated in the presence of all the other predictors, however. The other predictors predict much better than diastolic does, so diastolic is not significant.

3

(a)

It is a part of the exponential family.

(b)

Because it contains the term exp(yˆk), Weibull does not belong to the exponential family.

(c)

It is a part of the exponential family. Choose phi = sigmaˆ-2, a() = 1/phi, theta = -1/2muˆ2, b() = -sqrt(-2theta), c() = 1/2ln(phi/2piyˆ3) - phi/2y

(d)

Because both the term nln(theta) and ln(n+y-1 y) in the density cannot be separated into terms of n and theta, it doesn't belong to exponential family.