STAT-S632

Assignment 2

John Koo

```
# packages, etc.
import::from(magrittr, `%>%`, `%<>%`)
dp <- loadNamespace('dplyr')
library(ggplot2)
import::from(GGally, ggpairs)
theme_set(theme_bw())
```

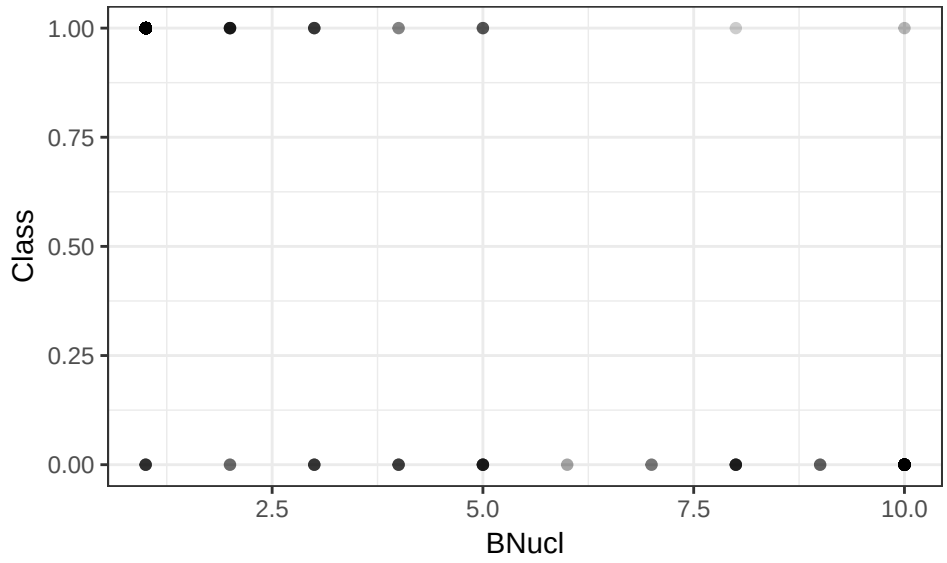
ELM 2.1

```
# get the data
wbca.df <- faraway::wbca
summary(wbca.df)
```

```
     Class            Adhes           BNucl           Chrom      
 Min.   :0.0000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.:0.0000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 2.000  
 Median :1.0000   Median : 1.000   Median : 1.000   Median : 3.000  
 Mean   :0.6505   Mean   : 2.816   Mean   : 3.542   Mean   : 3.433  
 3rd Qu.:1.0000   3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000  
 Max.   :1.0000   Max.   :10.000   Max.   :10.000   Max.   :10.000  
     Epith            Mitos           NNucl           Thick      
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 2.000  
 Median : 2.000   Median : 1.000   Median : 1.000   Median : 4.000  
 Mean   : 3.231   Mean   : 1.604   Mean   : 2.859   Mean   : 4.436  
 3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.: 4.000   3rd Qu.: 6.000  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000  
     UShap            USize       
 Min.   : 1.000   Min.   : 1.00  
 1st Qu.: 1.000   1st Qu.: 1.00  
 Median : 1.000   Median : 1.00  
 Mean   : 3.204   Mean   : 3.14  
 3rd Qu.: 5.000   3rd Qu.: 5.00  
 Max.   :10.000   Max.   :10.00  
```

Part a

```
ggplot(wbca.df) +
  geom_point(aes(x = BNucl, y = Class), alpha = .1)
```

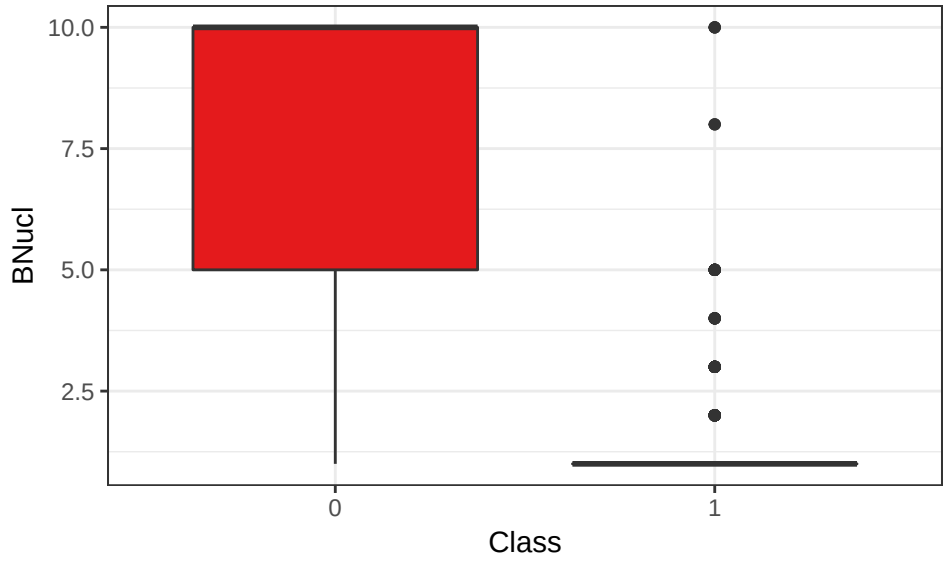


i

A scatterplot does not work well for `Class ~ BNucl` because the relationship is not regressive. `Class` is a quantitativve variable with only two levels, so a scatterplot will have a hard time showing any sort of trend between it and a predictor.

ii

```
wbca.df %<>% dp$mutate(Class = as.factor(Class))

ggplot(wbca.df) +
  geom_boxplot(aes(x = Class, group = Class, fill = Class, y = BNucl)) +
  scale_fill_brewer(palette = 'Set1') +
  guides(fill = FALSE)
```



```
table(dp$filter(wbca.df, Class == '0')$BNucl)
```

```
  1   2   3   4   5   6   7   8   9  10
 15   8  14  13  20   4   7  19   9 129
```

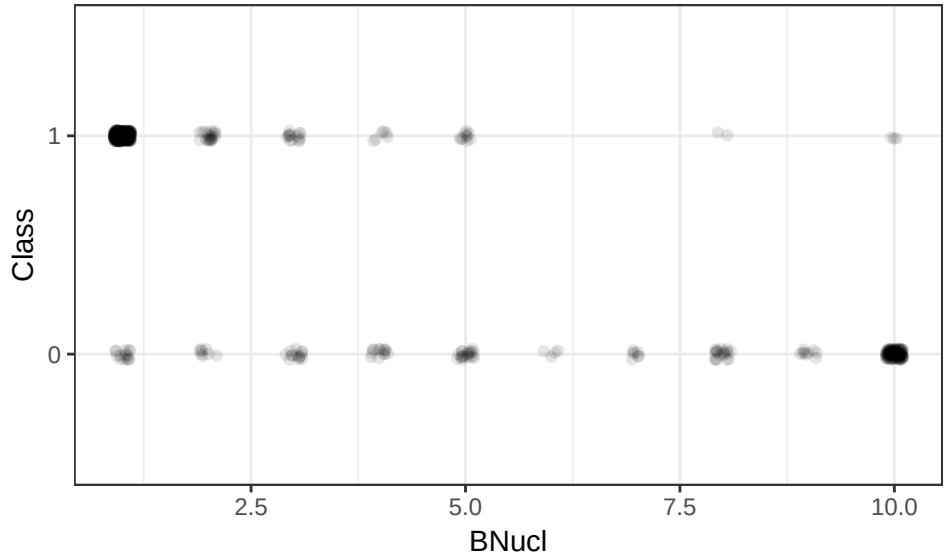
```
table(dp$filter(wbca.df, Class == '1')$BNucl)
```

```
  1   2   3   4   5   8  10
387  21  14   6  10   2   3
```

A boxplot reveals that most of the values of `BNucl` for `Class == 0` are $geq5$, whereas most of the `BNucl` values for `Class = 1` are 1.

iii

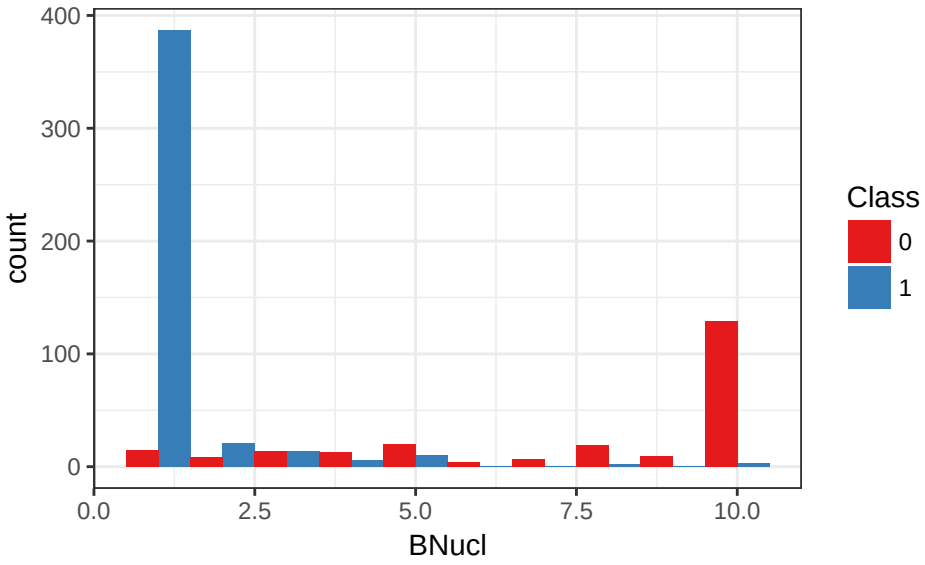
```
wbca.df %>%
  ggplot() +
  geom_jitter(aes(x = BNucl, y = Class),
              height = .025, width = .1, alpha = .1)
```



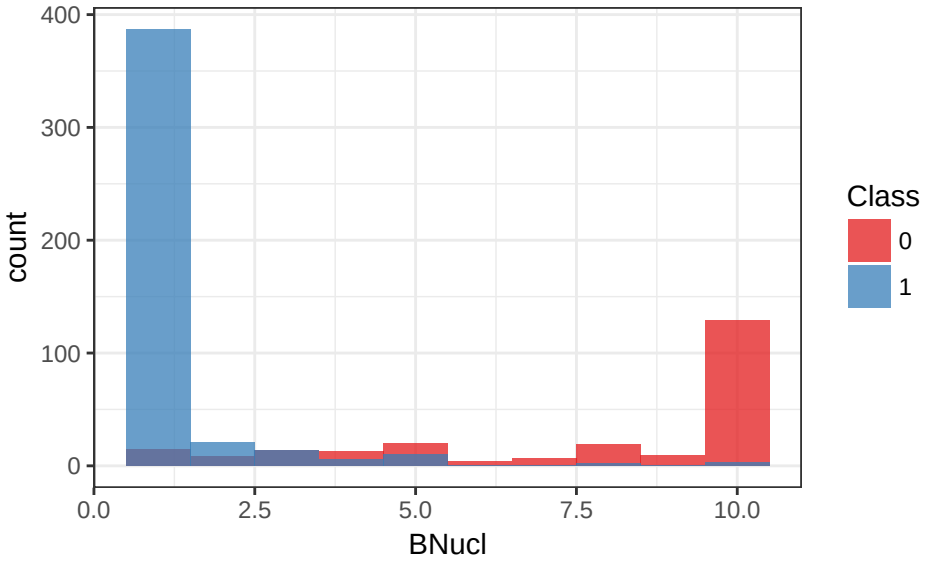
We can see that there is a concentration at `BNucl = 1` when `Class = 1` and at `BNucl = 10` when `Class = 0`.

iv

```
ggplot(wbca.df) +
  geom_histogram(aes(x = BNucl, fill = Class),
                 colour = NA, position = 'dodge', binwidth = 1) +
  scale_fill_brewer(palette = 'Set1')
```



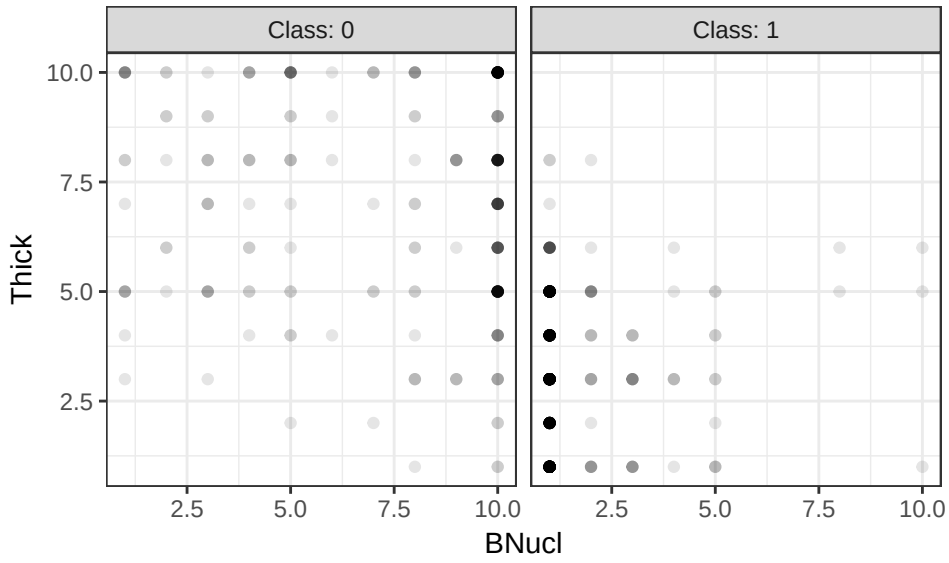

```
# i personally think this looks better ...
ggplot(wbca.df) +
  geom_histogram(aes(x = BNucl, fill = Class),
                 colour = NA, position = 'identity', binwidth = 1, alpha = .75) +
  scale_fill_brewer(palette = 'Set1')
```



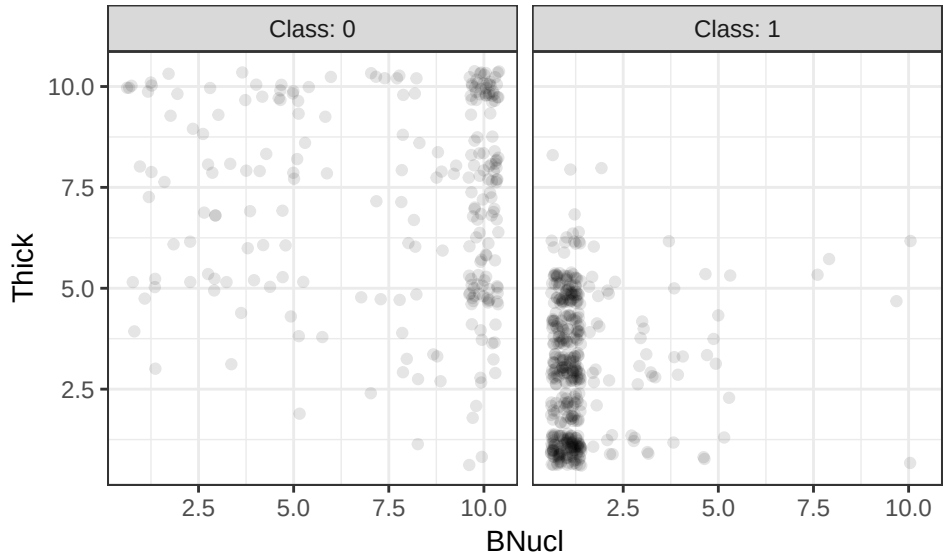
This confirms what we saw in the previous plots. For each `Class`, the distribution of `BNucl` is heavily skewed to the left and to the right.

Part b

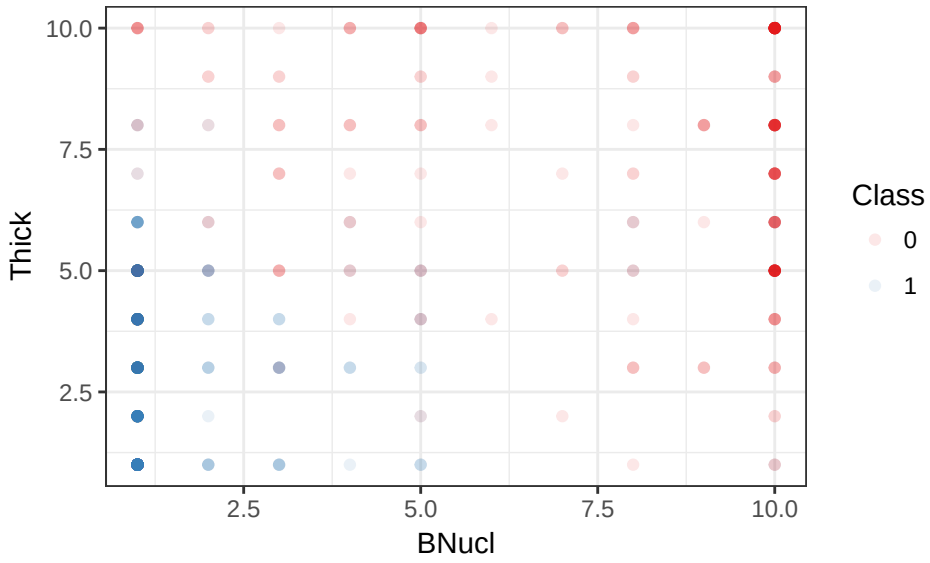
```
ggplot(wbca.df) +
  geom_point(aes(x = BNucl, y = Thick), alpha = .1) +
  facet_wrap(~ Class, labeller = 'label_both')
```




```
ggplot(wbca.df) +
  geom_jitter(aes(x = BNucl, y = Thick), alpha = .1) +
  facet_wrap(~ Class, labeller = 'label_both')
```




```
ggplot(wbca.df) +
  geom_point(aes(x = BNucl, y = Thick, colour = Class),
             alpha = .1) +
  scale_colour_brewer(palette = 'Set1')
```



Both appear to provide information in classifying `Class`. `BNucl` appears to provide more information.

Part c

```
full.mod <- glm(Class ~ ., data = wbca.df, family = binomial)
summary(full.mod)
```

```
Call:
glm(formula = Class ~ ., family = binomial, data = wbca.df)

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
```

Residual deviance: 89.464
Degrees of freedom: 671 ($n - p + 1$)

No, for that, you need the null deviance, which compares an intercept model with the model in question.

Part d

From previous exploration, we saw that starting from the full model tends to result in a model with lower AIC compared to starting from an intercept-only model and stepping forward. This is not guaranteed, but for the sake of keeping this short, we'll start from the full model. (I also tried forward selection and surprise, surprise, it didn't lower the AIC as much.)

```
back.mod <- step(full.mod, direction = 'both')
```

```
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
+ USize  1   89.464 109.46
- 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
```

```
+ Epith  1   89.523 107.52
+ USize  1   89.613 107.61
- 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(back.mod)
```

```
Call:
glm(formula = Class ~ Adhes + BNucl + Chrom + Mitos + NNucl +
    Thick + UShap, family = binomial, data = wbca.df)

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
```

Part e

```
# put predictions in data frame
wbca.df %<>%
  dp$mutate(class.phat = predict(back.mod, newdata = wbca.df,
                                 type = 'response'),
            class.pred = dp$if_else(class.phat > .5, '1', '0'))

# confusion matrix
confusion.matrix <- table(wbca.df$class.pred, wbca.df$Class)
print(confusion.matrix)
```

```
    0   1
0 227   9
1  11 434
```

```
# error rate
1 - sum(diag(confusion.matrix)) / sum(confusion.matrix)
```

```
[1] 0.02936858
```

Our model made 9 Type I errors and 11 Type II errors, with an overall error rate of 2.9%.

Part f

```
# put predictions in data frame
wbca.df %<>%
  dp$mutate(class.pred = dp$if_else(class.phat > .9, '1', '0'))

# confusion matrix
confusion.matrix <- table(wbca.df$class.pred, wbca.df$Class)
print(confusion.matrix)
```

```
    0   1
0 237  16
1   1 427
```

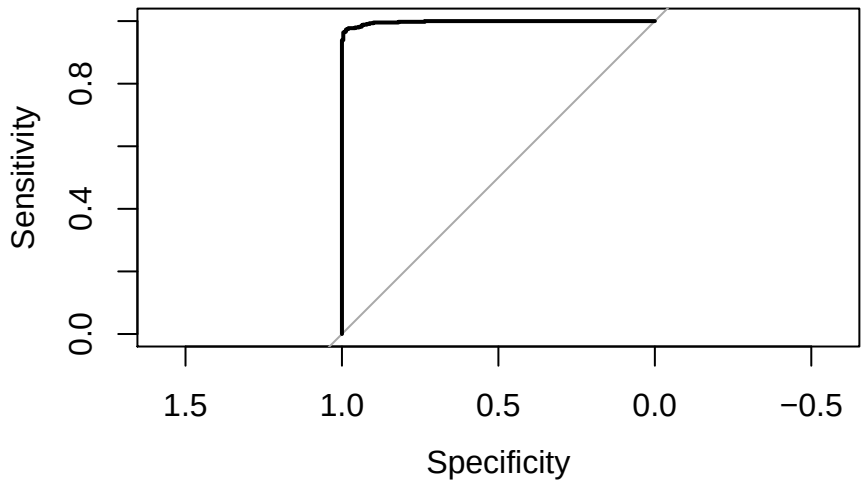
```
# error rate
1 - sum(diag(confusion.matrix)) / sum(confusion.matrix)
```

```
[1] 0.02496329
```

Setting the cutoff at 0.9 results in 16 Type I errors and 1 Type II error (which is expected by shifting the cutoff up).

Part g

```
plot(pROC::roc(wbca.df$Class, wbca.df$class.phat))
```



```
pROC::roc(wbca.df$Class, wbca.df$class.phat)
```

```
Call:
roc.default(response = wbca.df$Class, predictor = wbca.df$class.phat)

Data: wbca.df$class.phat in 238 controls (wbca.df$Class 0) < 443 cases (wbca.df$Class 1).
Area under the curve: 0.9974
```

Computing the AUC adjusts for class imbalance. If there is severe class imbalance, high Type I or Type II error rates could have only a negligible effect on the overall error rate.

In this case, the model performs well in separating the two classes, to the extent that changing the cutoff for $\hat{p}$ does not change the error rates too much.

Part h

For this, we will use stepwise model selection starting from the full model and using the AIC to pick the best model. Then the AUC and error rates will be measured on the test set.

```
# reset the data
wbca.df <- faraway::wbca %>%
  dp$mutate(Class = as.factor(Class))

# rows to use for test data
test.ind <- seq(3, nrow(wbca.df), 3)

# split the data
test.df <- wbca.df[test.ind, ]
train.df <- wbca.df[-test.ind, ]

# build model on test data
final.mod <- glm(Class ~ ., data = train.df, family = binomial) %>%
  step(direction = 'both')
```

```
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
+ Epith  1   57.651 77.651
- 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
+ USize  1   58.340 76.340
+ Epith  1   58.880 76.880
- 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
```

```
summary(final.mod)  # not surprising that it's the same as before
```

```
Call:
glm(formula = Class ~ Adhes + BNucl + Chrom + Mitos + NNucl +
    Thick + UShap, family = binomial, data = train.df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.03312  -0.01224   0.04042   0.08373   2.85056

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  11.5571     1.8285   6.321  2.6e-10 ***
Adhes        -0.4249     0.1441  -2.949  0.00318 **
BNucl        -0.3341     0.1187  -2.815  0.00487 **
Chrom        -0.5963     0.2422  -2.462  0.01382 *
Mitos        -0.5822     0.4872  -1.195  0.23207
NNucl        -0.4192     0.1604  -2.614  0.00895 **
Thick        -0.6037     0.1924  -3.138  0.00170 **
UShap        -0.2943     0.2034  -1.447  0.14795
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 592.796  on 453  degrees of freedom
Residual deviance:  59.536  on 446  degrees of freedom
```

```
AIC: 75.536

Number of Fisher Scoring iterations: 9
```

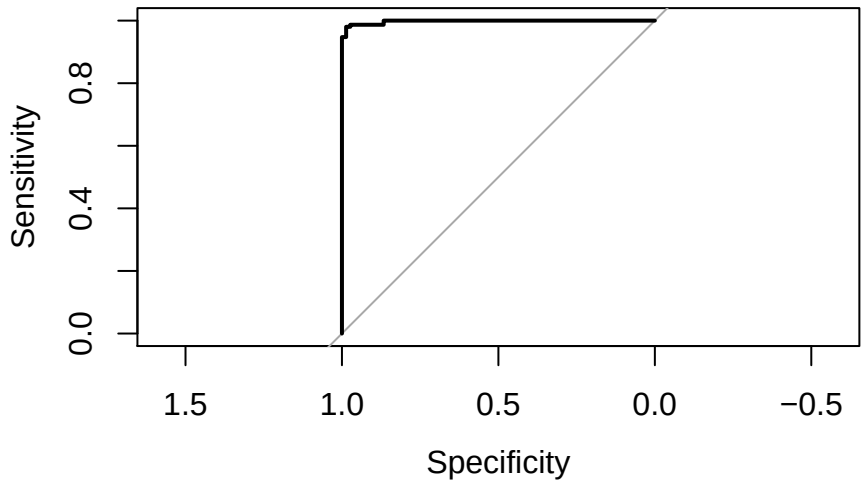
```
# predict on test set and compute metrics
test.df %<>%
  dp$mutate(class.phat = predict(final.mod, newdata = test.df,
                                 type = 'response'),
            class.pred = dp$if_else(class.phat > .5, '1', '0'))

# confusion matrix using cutoff of .5
confusion.matrix <- table(test.df$class.pred, test.df$Class)
print(confusion.matrix)
```

```
     0   1
  0 70   2
  1  5 150
```

```
# error rate using cutoff of .5
1 - sum(diag(confusion.matrix)) / sum(confusion.matrix)
```

```
[1] 0.030837
```

```
# AUC
plot(pROC::roc(test.df$Class, test.df$class.phat))
```




```
pROC::roc(test.df$Class, test.df$class.phat)
```

```
Call:
roc.default(response = test.df$Class, predictor = test.df$class.phat)

Data: test.df$class.phat in 75 controls (test.df$Class 0) < 152 cases (test.df$Class 1).
Area under the curve: 0.9976
```

We see no performance degradation. Error (using 0.5 as the cutoff for $\hat{p}$), increases from 2.9% to 3.1%. The AUC increases by a negliglble amount. In this case, the AIC does a good job at regularizing when using it as a model selection criterion. In fact, the models from parts (d) and (h) use the same regressors.