A Appendix A: Testing whether a fleece density score would improve classification using follicle density as a proxy

We first add follicle density (”Fn”) to the full set of 10 on-sheep traits

```
> form.11 <- formula(CrimpType ~ StapMaxD + StapMinD + StapArea +
        CompEx + Softness + Lustre + Whiteness + PeelScore +
        CrimpFreq + Zigzag + Fn)
```

then run the recursive partitioning algorithm on 11 traits and get

```
> rpart.11 <- rpart(form.11,jan20sf2.df)
> rpart.11
n= 306

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 306 148 stretched (0.516339869 0.091503268 0.392156863)
   2) CompEx< 3.5 204  64 stretched (0.686274510 0.132352941 0.181372549)
     4) Zigzag>=1.5 178  47 stretched (0.735955056 0.061797753 0.202247191)
       8) Zigzag< 3.5 158  35 stretched (0.778481013 0.069620253 0.151898734) *
       9) Zigzag>=3.5 20   8 unfolded (0.400000000 0.000000000 0.600000000)
        18) StapMinD>=1.85 11   4 stretched (0.636363636 0.000000000 0.363636364) *
        19) StapMinD< 1.85 9   1 unfolded (0.111111111 0.000000000 0.888888889) *
     5) Zigzag< 1.5 26  10 unaligned (0.346153846 0.615384615 0.038461538) *
   3) CompEx>=3.5 102  19 unfolded (0.176470588 0.009803922 0.813725490) *
> rpart(form.all,jan20sf2.df)
n= 306
```

So it does not use Fn. End of story as far as classification trees go, adding Fn is useless. We might just look at how important it thinks Fn is

```
> rpart.11$variable.importance
    CompEx     Zigzag  PeelScore   StapMaxD   StapArea         Fn   StapMinD
49.2260711 24.6984711  9.9092431  9.3724380  8.6731148  6.2480532  4.4285790
    Lustre  CrimpFreq
 3.8188483  0.9104377
```

So it rates it higher than Lustre and StapMinD, but does not use it? I dont understand that, I guess these ratings are not the full story.

If we try again with discriminant functions, we get with 11 traits

```
> lda.11 <- lda(form.11,data=jan20sf2.df)
> lda.11
Call:
lda(form.11, data = jan20sf2.df)
```

```
Prior probabilities of groups:
 stretched  unaligned   unfolded 
0.51546392 0.09278351 0.39175258 

Group means:
          StapMaxD StapMinD  StapArea   CompEx Softness   Lustre Whiteness
stretched 4.592667 2.248000 10.754667 2.880000 3.500000 3.373333  3.320000
unaligned 6.618519 2.966667 20.937037 1.814815 2.185185 2.222222  3.185185
unfolded  3.711404 1.838596  7.138596 3.789474 3.982456 3.745614  3.543860
          PeelScore CrimpFreq   Zigzag       Fn
stretched  3.633333  3.720000 2.600000 70.00333
unaligned  2.481481  4.596296 1.444444 62.82593
unfolded   4.350877  4.006140 3.324561 78.37632

Coefficients of linear discriminants:
                  LD1          LD2
StapMaxD  -0.183753929  0.774899465
StapMinD  -0.094024204  2.134902277
StapArea  -0.005777724 -0.425642225
CompEx     0.761674049 -0.574463023
Softness   0.166620075  0.485939660
Lustre    -0.041094681  0.172516176
Whiteness -0.139053473 -0.548224971
PeelScore  0.266050275 -0.090423885
CrimpFreq  0.086778227 -0.389109861
Zigzag     0.551376396 -0.295712978
Fn         0.007500229  0.004482359

Proportion of trace:
  LD1   LD2 
0.875 0.125 
>
```

which is almost exactly the same as the 10 trait discriminant function analysis with Fn added on the end but with insignificant coefficients. So again, the procedure declines the opportunity to use Fn.

Just to check further, we look at the confusion table

```
> table(predicted=plda.11$class,actual=ct306)
           actual
predicted   stretched unaligned unfolded
  stretched       123        11       19
  unaligned         7        16        1
  unfolded         20         0       94
>
```

Exactly the same except one extra observation is missing due to a missing Fn value.

The conclusion is clear. Fn contributes nothing extra to classification. It may substitute for other variables, but it adds nothing extra. It is likely that straight density is important, but in complicated ways - for example high density with organised between-follicle spaces would be different from high density with random spacing, from the point of view of classifying crimp types.