Not surprisingly, some of you had some transects in which you detected no magpies. Unless you deal with this correctly, it is likely to create an error message that says something like:

'Error in unmarkedFrameDS(y = as.matrix(yDat), siteCovs = covs, survey = "line", : tlength should be a vector with length(tlength)==nrow(y)'

How to deal with transects with no detections is shown in the Chandler reading assignment, but it is a little hard to follow without some explanation. Onn page 2, he starts the script by loading the unmarked package and reading in a file with transect ID and perpendicular distance for each detection:

```
> library(unmarked)
> dists <- read.csv(system.file("csv", "distdata.csv", package="unmarked"))
> head(dists, 10)
distance transect
1 1 a
2 18 a
3 7 a
4 2 a
5 13 b
6 3 b
7 5 b
8 10 b
9 6 c
10 9 c
```

Then he does this:

```
> levels(dists$transect) <- c(levels(dists$transect), "g")
```

Notice that he has just added transect g to the set of values (levels) in the variable 'transects' in the dataframe 'dists'. This is because there were no detected animals in transect g, so the program does not even know it exists, just based on the distances that were recorded and read into 'dists' from distdata.csv.

> levels(dists\$transect)
[1] "a" "b" "c" "d" "e" "f" "g"

Notice that transect g has been added, as needed. Now, he runs formatDistData to create ydat:

```
> yDat <- formatDistData(dists, distCol="distance",
transectNameCol="transect", dist.breaks=c(0, 5, 10, 15, 20))
```

When you examine ydat...

... you see that transect g is included, correctly showing that no animals were detected at any distance on that transect.

Now you can read in the covariate data... (he's just entering it by hand in this example)

> (covs <- data.frame(canopyHt = c(5, 8, 3, 2, 4, 7, 5), habitat = c('A','A','A','B','B','B'), row.names=letters[1:7]))

canopyHt habitat

a 5 A b 8 A c 3 A d 2 A e 4 B f 7 B g 5 B

and the rows align correctly with the information in ydat, so when he runs unmarkedFrameDS(), everything will work, as long as tlengths is a vector of transect lengths equal to the number of transects (7 in this example).

```
umf <- unmarkedFrameDS(y=as.matrix(yDat), siteCovs=covs, survey="line",
dist.breaks=c(0, 5, 10, 15, 20), tlength=rep(100, 7),
unitsIn="m")
```

(He's got 7 transects that were all 100m, in this example.)

If you have a file that you are reading into the dataframe covs, like this:

covs <- read.csv("Lpnp.Data.Cov.csv",dec=".", header=TRUE)
covs\$trans <- as.factor(covs\$trans) #convert transects into a categorical variable</pre>

then you can include a variable 'length' for each transect, and then when you run unmarkedFrameDS(), you can use tlength = covs\$length.