1. Chinese monophthongs

Pinyin	Example Character	Meaning	IPA
nà	那	that	a
ni√	你	you	i
nu√	努	NA	u
er√	耳	ear	a
bó	脖	neck	o
è	饿	hungry	ɤ
yè (represented as ie)	夜	night	e
yü√	雨	to rain	y

4.a. How compact are the ``clouds'' of exemplars for any given type? How much do they overlap? How do you explain this observation?

For a, ü, i, ie, o, the clouds are pretty compact. I think this means two things:

1) one compact cloud represent one single vowel, meaning that different vowels will not share the same space. Or in other words different vowels will not have overlaps in their clouds.
2) Ideally for each type of the vowel, each data type should locate the same spot on the chart, where we would end up having one circle overlapped 10 times for each type instead of a cloud (imaging we play a record instead of recording ourselves saying the same vowel 10 times). However I think this experiment represents the data collection in the real world, where we have lots of ever changing parameters that can impact our data collection process. Thus we for each type we have a cloud which has some overlaps but also are unique individually.

4.b. Are there any outliers? To what do you attribute them?

For er and e, there are two “outliers” for each that are still close to the main clouds. I think it could be because of the tounes. In this case, I think these data points are still useful and can be used in the training data.
For u, there is one super outlier, which is really close to the ie cloud. I think it might be caused by the tone (third) and nasal n. In this case, I think this outlier data point should be thrown away because it might just be a mistake which was created when generating the data.

This also made me wonder how do different tones influence the formants of vowels?

4.c. Human first-language learners do not know, in advance, how many vowel types their language will have. Given data like that you have extracted, how would you construct a program that learned the optimal number and identity of vowels in a vowel system?

I would use a cluster algorithm. For example we can train a system using kNN algorithm (for example, set k to 10) on the training data. Then for a given testing data point, the system will assign a cluster/cloud to it. Then we would be able to identify the vowel.