Homework 3 Writeup

Joshua Burkhart

February 6, 2016

HW3

Input datasets are as below:

- a: raw intensity values from the original CEL files arranged in a matrix layout, where each column represents one hybridization, and rows stand for individual array features
- x: RMA normalized dataset in the assayData and annotation in the phenoData
- xq: single-cell gene expression levels measured by qPCR
- xql: single-cell gene expression measured by qPCR, with cells facing the blastocyst cavity labelled fluorescently

Let us assume our final goals are to build 1) a classifier that can be generalized for the microarray platform & probeset used in x and 2) a classifier that can be generalized for the qPCR assay used in xq that can predict the developmental stage better than each dataset's existing class distribution, 52% and 76%, respectively, and to compare their performance. We'd like to not only correctly label test data as "E3.25" or not in each experiment but build classifiers whose results may be interpreted by experts in the field to gain knowledge about any biological interactions that occur between selected features and developmental stage. We'll use the RMA normalized (x) dataset for 1 and the xq dataset for 2. The xql dataset would be of interest if it contained our target developmental stage, which it does not.

Even before considering the results from our EDA, we'll still reject metadata such as microarray date, number of cells, or index in the original dataset as feature candidates due to our noted constraint of creating a model that may increase biological insight into developmental stage. Additionally, we'll restrict our feature selection strategies to only those who avoid transforms, like PCA, and we'll restrict our models to avoid those whose results are notoriously difficult to interpret, like ANN.

Our null hypothesis shall be: All probe intensities in dataset x and gene expression values in dataset xq are evenly and randomly distributed among developmental stage E3.25 and the other developmental stages.

Analysis Plan

1. Extract feature candidates, probe intensities, gene expression levels, etc., from their original data sources and combine them into a single data table. This will allow for simpler handling for the remainder of the analysis.
2. Scan for and address missing data. Because we have so few samples and so many total features, we'll prefer to drop features than drop samples. We must scan probe intensities, gene expression levels, etc..
3. Convert feature datatypes as nessecary. For example, genotype is stored as one of two values: "FGF4-KO" and "WT". We'll change "FGF4-KO" to 1 and "WT" to 0.
4. Extract development stage and convert it to a numeric value. Currently, Embryonic.day is stored as one of three values: "E3.25", "E3.5", and "E4.5". We'll copy this column to a separate vector and store our positive class, "E3.25", as 1 and our negative class, "E3.5" and "E4.5", as 0.

5. Randomly split data into training and test sets. Seeing as we have so few samples, we'll use an 80%/20% training/test split.
6. Randomly split training data to prepare for cross validation. After heavy testing & tuning, we'll shoot for 3-fold Monte Carlo cross validation. Ideally we'd prefer more folds but appropreate computational resources are unavailable.
7. Z score transform all the training feature values for the xq dataset, storing the means and standard deviations for all columns so validation and test data can be transformed similarly later on. Putting features on the same scale allows some optimization algorithms, such as gradient descent, to converge more quickly and generally allows for more direct comparisons between feature distributions. This could be done for the x dataset too but it looks fairly evenly distributed as it is[A1], unlike the xq dataset[A2]. Also, Z scoring x takes a long time.
8. We'd like to explain a lot of variance without using too many features, as that can make things confusing. Normally we'd favor PCA as a logical next step to a learning system but, keeping in mind we'd like to create a model that's interpretable to biologists later on, we'd prefer to avoid it. Instead, we'll use another nonparametric multivariate filter that avoids feature transforms, a Markov Blanket[R1]. The difficulty with this technique is in constructing the nessecary bayesian net from which to extract the blanket. For the x dataset we'll need to filter the features apriori.
9. While the Markov Blanket feature selection approach will work well for the xq dataset containing only 33 features, the construction of a bayesian net for the x dataset (containing 45,102 features) requires unavailable computational resources. Thus, we'll use a nonparametric univariate filter, the Wilcoxon Rank Sum, to reduce our feature set before hand.
10. As there has been skepticism regarding the use of the Wilcoxon Rank Sum test to select features[R2], we'll remember that an accepted parametric univariate filter method, linear model fitting, assumes normally distributed feature values, while the Wilcoxon Rank Sum test makes no such assumption about the distribution of feature values as it simply relies on the ranking of feature values to discover features whose values for one class are greater or less than those of another. Also, we'll show that linear model fitting finds a similar set of features as the Wilcoxon Rank Sum approach anyhow, providing at least some sense of validation.
11. After extracting our Markov Blankets for each dataset, we'll train a support vector machine (SVM) with a linear kernel. This model selection was made because SVM's with linear kernels are thought to be rather straight forward to interpret[R3], which is a main concern.
12. The trained SVM will be run against the previously separated validation data and its error rate, a simple misclassification rate, will be recorded for each of the 3 cross-validation folds, along with the generated feature set and SVM.
13. Following the cross-validation, the feature sets and SVM's from the fold with the lowest error rate will be selected as our 'best model'.
14. To test our models, we'll apply Z score transformations to test data using stored means and standard deviations from the training data, select the feature sets, and run the SVM's from the best fold, and report our test error rate.

Part A (hw3A.Rmd)

Missing Data

302 na's reported[A3] in the xq gene expression data but nowhere else. The na's are from Spp1 (40), Prdm14 (40), Sdc4 (40), Morc1 (67), Tbpl1 (2), and Zp3 (113). We'll remove them[A4].

Class & Feature Distributions

Developmental stage (class) distributions are not uniform[A5] but with so few samples, we won't remove any.

Regarding feature vectors, genotypes are skewed[A6]. The probe intensities overall are slightly skewed[A7] while the probe intensity distributions all appear uniformly distributed with only two having noticable differences[A8]. It's odd to have noticably different distributions as they were all supposed to have been normalized in the x dataset. The gene expressions show wide variation among samples[A9].

Class distributions among features may not be visualized well but genotypes and cell types can be tested easily using Chi-squared tests[A10]. Genotypes are fairly well balanced among developmental stages but cell types are skewed.

Batch Effects

After plotting array index vs developmental stage, visual inspection of the plots shows obvious bias[A11]. We should consider the regularity of the positions in these datasets when separating our data into training and test sets, which can be addressed by randomly sampling when separating our training, test, and validation sets.

Plotting dates vs developmental stage for dataset x[A12]. Unfortunately, developmental stage appears highly correlated with both date variables over several years. This indicates unknown and unrecorded variables could be influencing the results of our data. Chi-squared tests agree in both cases, shown by extreemly low p-values[A13].

Plotting number of cells vs developmental stage[A14]. Like the dates above, the number of cells appears to correlate well with developmental stage. And again, a Chi-squared test agrees[A15].

Sadly, aside from randomly sampling when selecting our training, test, and validation sets, we're currently unable to adjust for the batch effects we discovered.

Part B (hw3B.Rmd)

Cross Validation

Three iterations of Monte Carlo cross validation was performed and the results from the iteration with the lowest error on the validation set were recorded. This included the feature means and standard deviations for both x and xq datasets, the percent similarity to the linear model feature filter for the x dataset (87.67%), the markov blankets for both x and xq datasets, the validation error for the selected fold for both x and xq datasets (0% validation error for both), and the trained SVM for both x and xq datasets.

Model Assessment & Results

Test data was transformed using the means and standard deviations recorded during cross validation and predictions were made made by the trained SVM's. The test data for the x dataset is classified correctly for all 20 test samples (0% test error) and the test data for xq is classified correctly for 26 of 27 test samples (3.7% test error).

References

[1] Xing, Eric P., Michael I. Jordan, and Richard M. Karp. "Feature selection for high-dimensional genomic microarray data." ICML. Vol. 1. 2001.

[2] McWeeney, S., Personal Communication, Feb. 4, 2016.

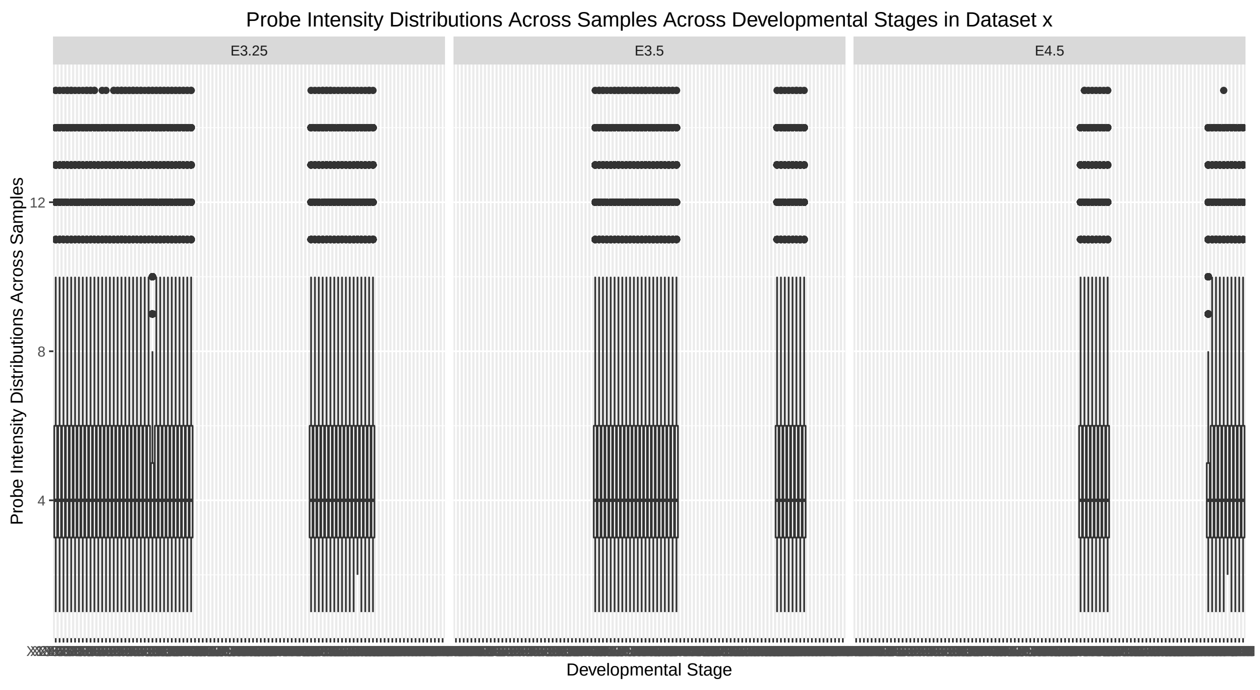
[3] Rosenbaum, Lars, et al. "Interpreting linear support vector machine models with heat map molecule coloring." J. Cheminformatics 3.11 (2011).

Appendix

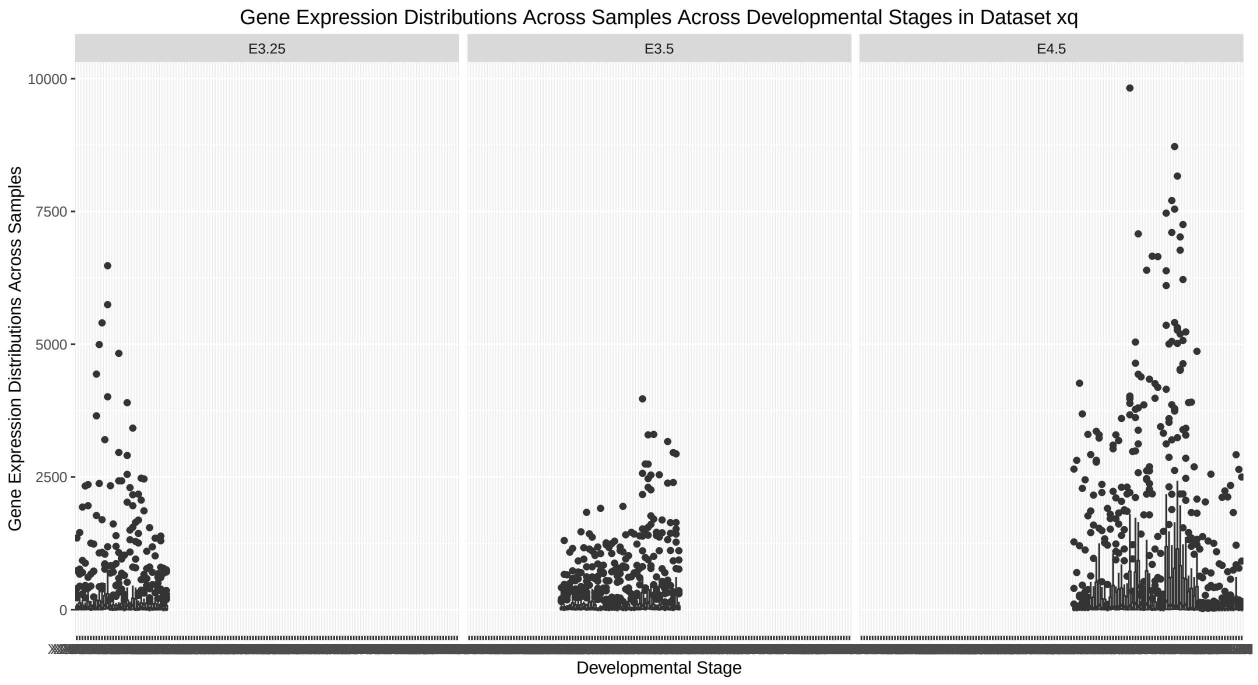
[0]

```
data("x")
data("xq")
data("xql")

#feature candidates
hw3A.x.genotypes <- x@phenoData@data$genotype
hw3A.x.probe_intensities <- data.frame(assayDataElement(x@assayData,'exprs'))
hw3A.xq.cell_type <- xq@phenoData@data$Cell.type
hw3A.xq.gene_expressions <- data.frame(assayDataElement(xq@assayData,'exprs'))
hw3A.xql.label <- xql@phenoData@data$Label
hw3A.xql.gene_expressions <- data.frame(assayDataElement(xql@assayData,'exprs'))

#classes
hw3A.x.classes <- x@phenoData@data$Embryonic.day
hw3A.xq.classes <- xq@phenoData@data$Embryonic.day
hw3A.xql.classes <- xql@phenoData@data$Embryonic.day

#additional data we'll consider when testing for batch effects
hw3A.x.pheno_dates <- as.Date(x@phenoData@data$ScanDate)
hw3A.x.proto_dates <- as.Date(x@protocolData@data$ScanDate)
hw3A.x.num_cells <- as.numeric(x@phenoData@data$Total.number.of.cells)
```

[1]



[2]



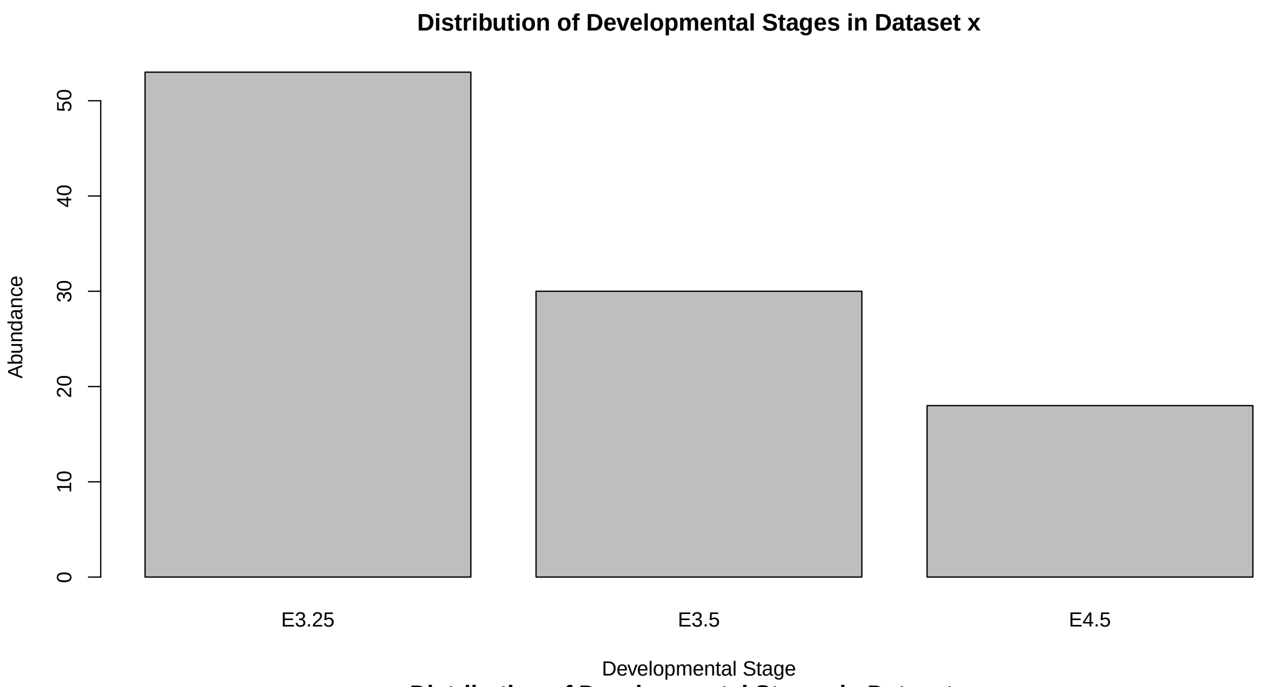
[3]

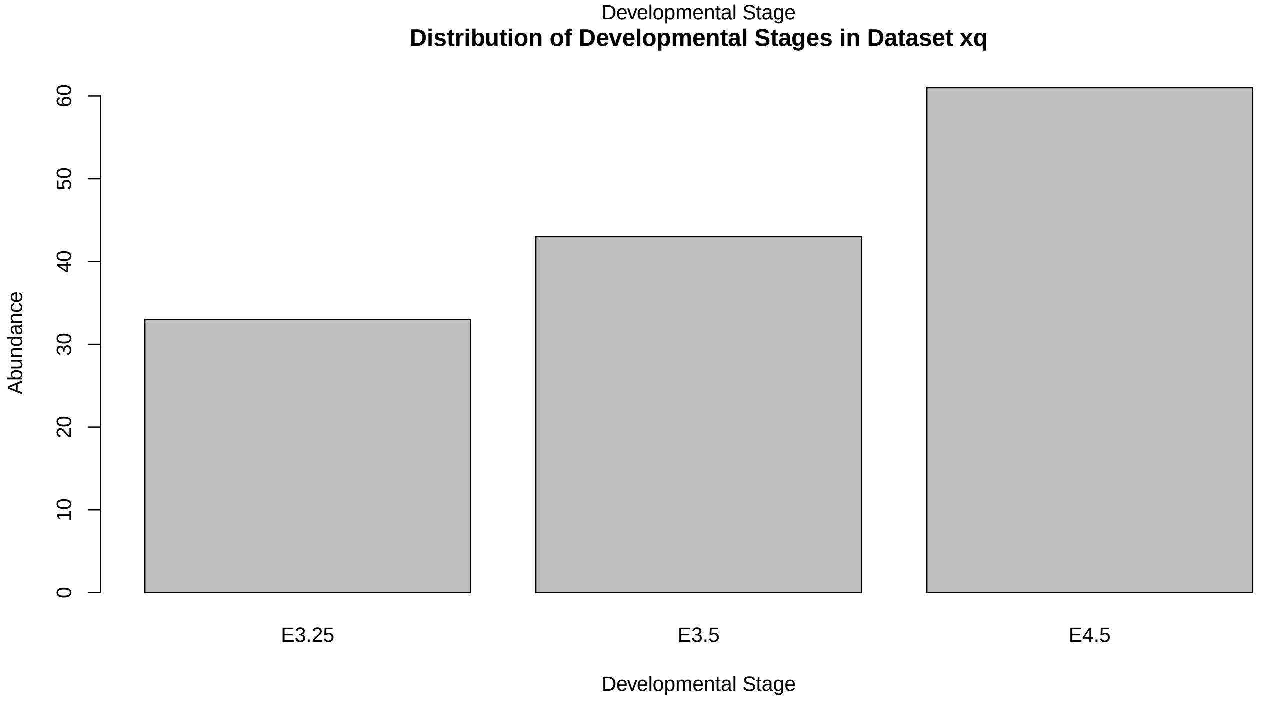
```r
sum(is.na(hw3A.x.genotypes))
sum(is.na(hw3A.x.probe_intensities))
sum(is.na(hw3A.xq.cell_type))
sum(is.na(hw3A.xq.gene_expressions))
sum(is.na(hw3A.xql.label))
sum(is.na(hw3A.xql.gene_expressions))
sum(is.na(hw3A.x.classes))
sum(is.na(hw3A.xq.classes))
sum(is.na(hw3A.xql.classes))
```

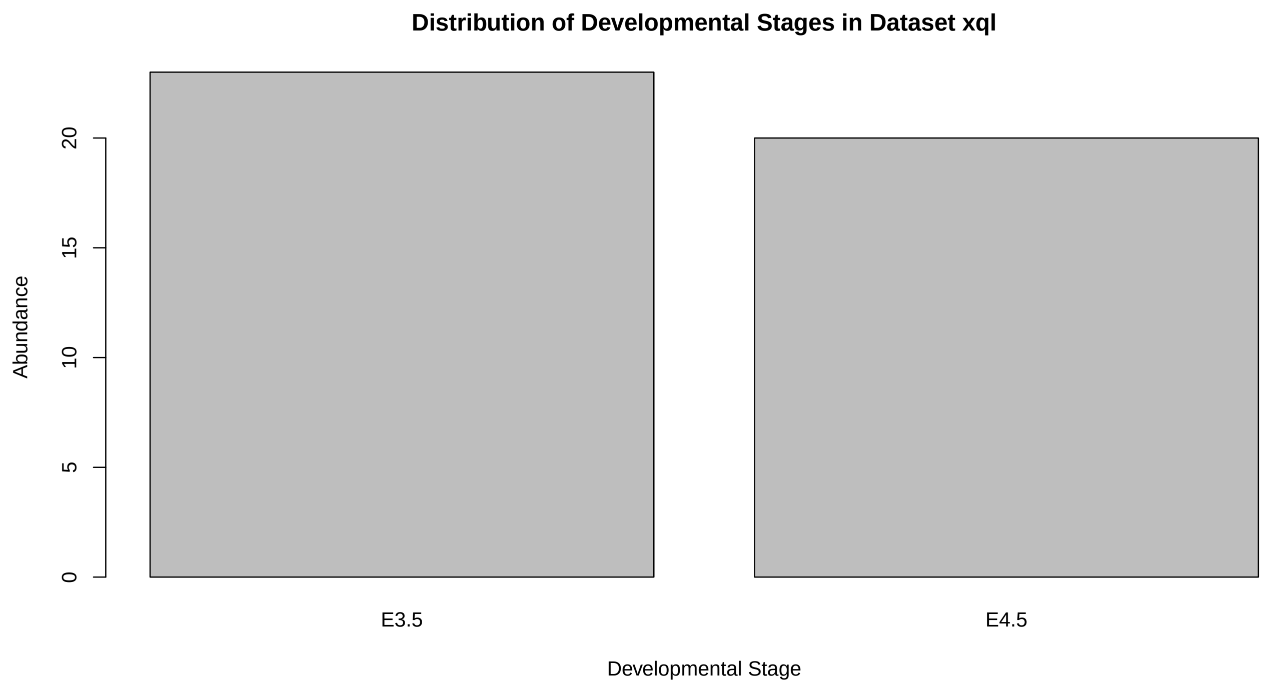
[4]

```r
t(hw3A.xq.gene_expressions) %>% summary()
hw3A.xq.gene_expressions <- na.omit(hw3A.xq.gene_expressions)
```

[5]

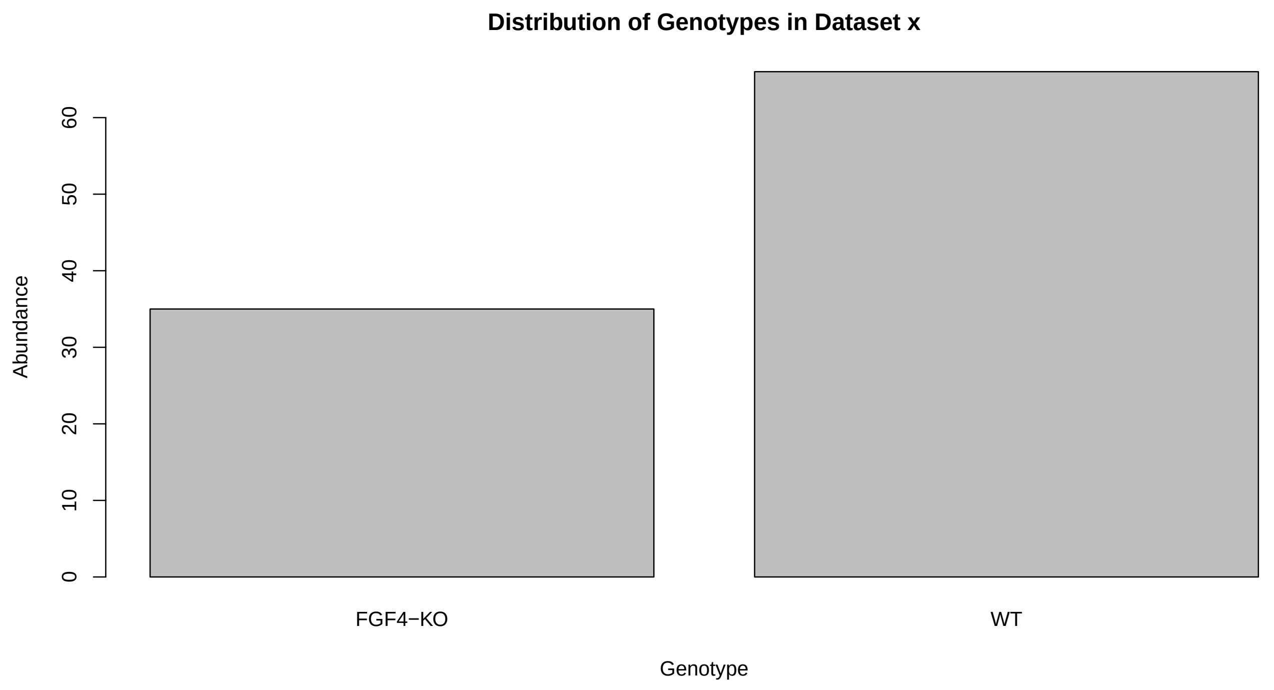

Distribution of Developmental Stages in Dataset x
Abundance
0
10
20
30
40
50
E3.25
E3.5
E4.5
Developmental Stage


Distribution of Developmental Stages in Dataset xq
Abundance
0
10
20
30
40
50
60
E3.25
E3.5
E4.5
Developmental Stage



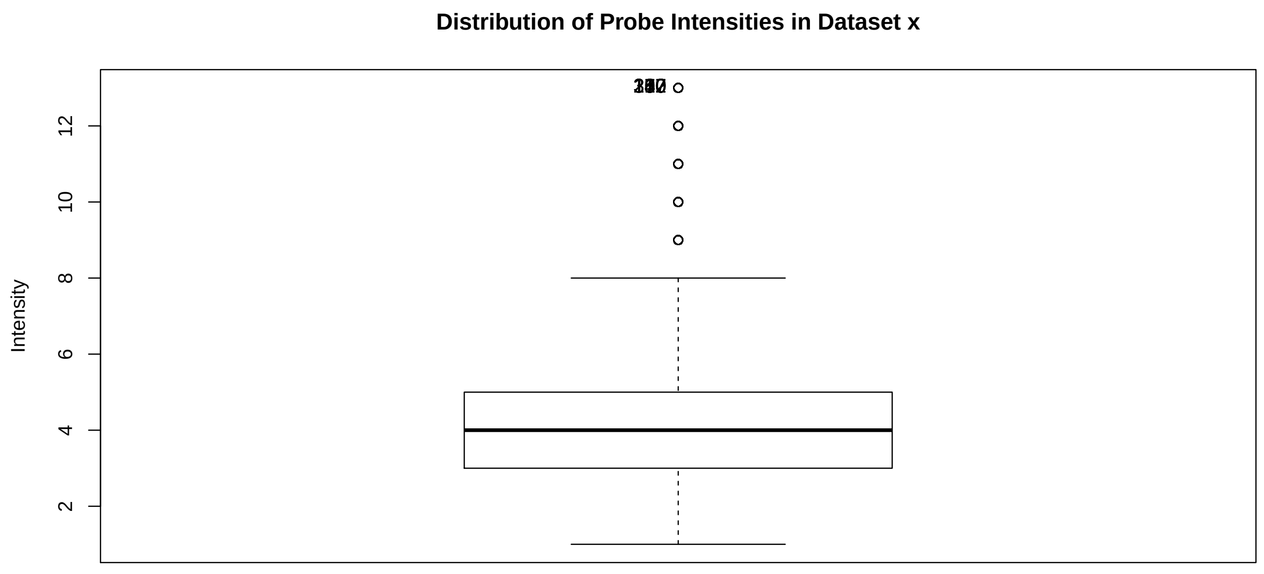
[6]

```r
plot(x=hw3A.x.genotypes,
     xlab="Genotype",
     ylab="Abundance",
     main="Distribution of Genotypes in Dataset x")
```



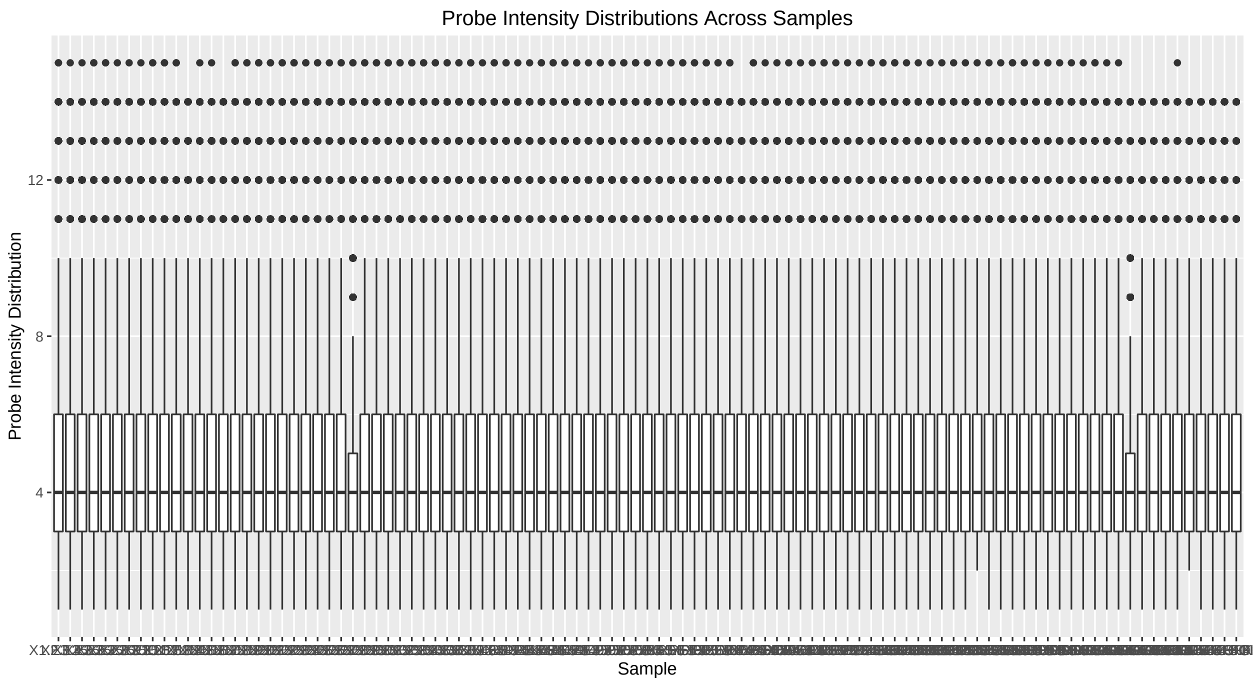
[7]

```
range(hw3A.x.probe_intensities) %>% invisible()
rdiff <- round(range(hw3A.x.probe_intensities)[2] - range(hw3A.x.probe_intensities)[1])
bins <- seq(1:rdiff)
Boxplot(.bincode(hw3A.x.probe_intensities[,1],bins),
     ylab="Intensity",
     main="Distribution of Probe Intensities in Dataset x") %>% invisible()
```



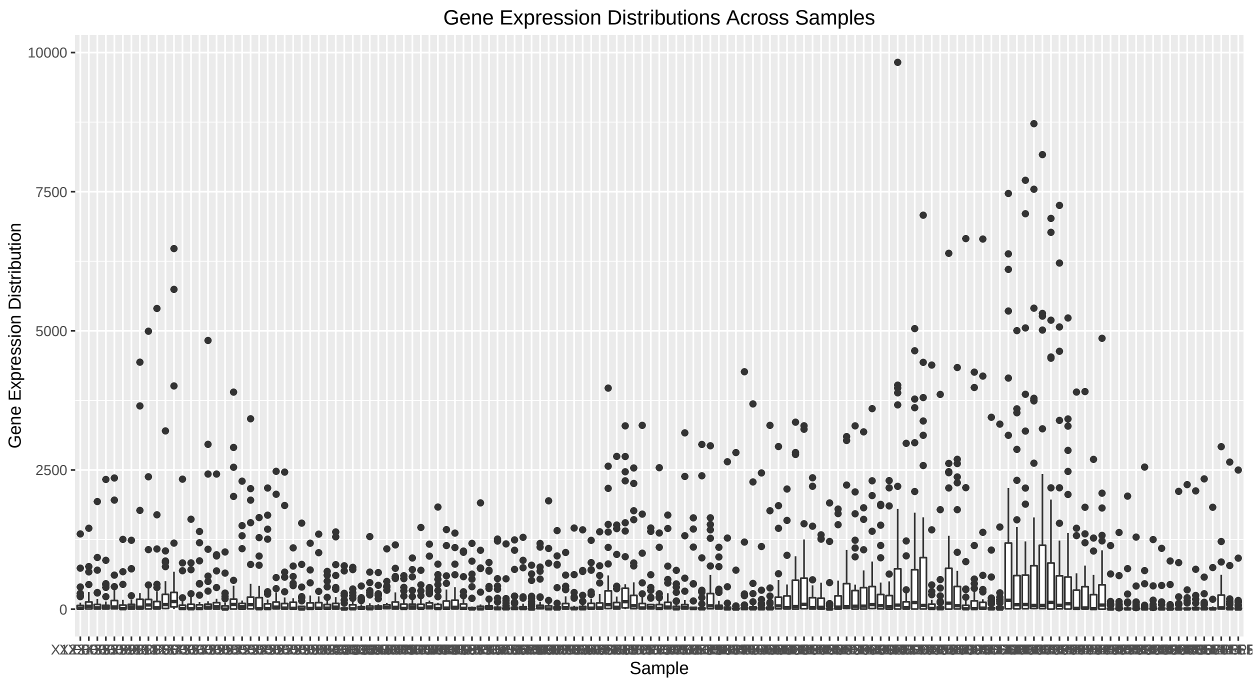
[8]

```
mm <- melt(hw3A.x.probe_intensities)
ggplot(mm) +
  geom_boxplot(
    aes(y=round(value),x=variable,colour=floor(value))) +
  labs(x="Sample",
       y="Probe Intensity Distribution",
       title="Probe Intensity Distributions Across Samples")
```



[9]

```r
mm2 <- melt(hw3A.xq.gene_expressions)
ggplot(mm2) +
  geom_boxplot(
    aes(y=round(value),x=variable,colour=floor(value))) +
  labs(x="Sample",
       y="Gene Expression Distribution",
       title="Gene Expression Distributions Across Samples")
```



[10]

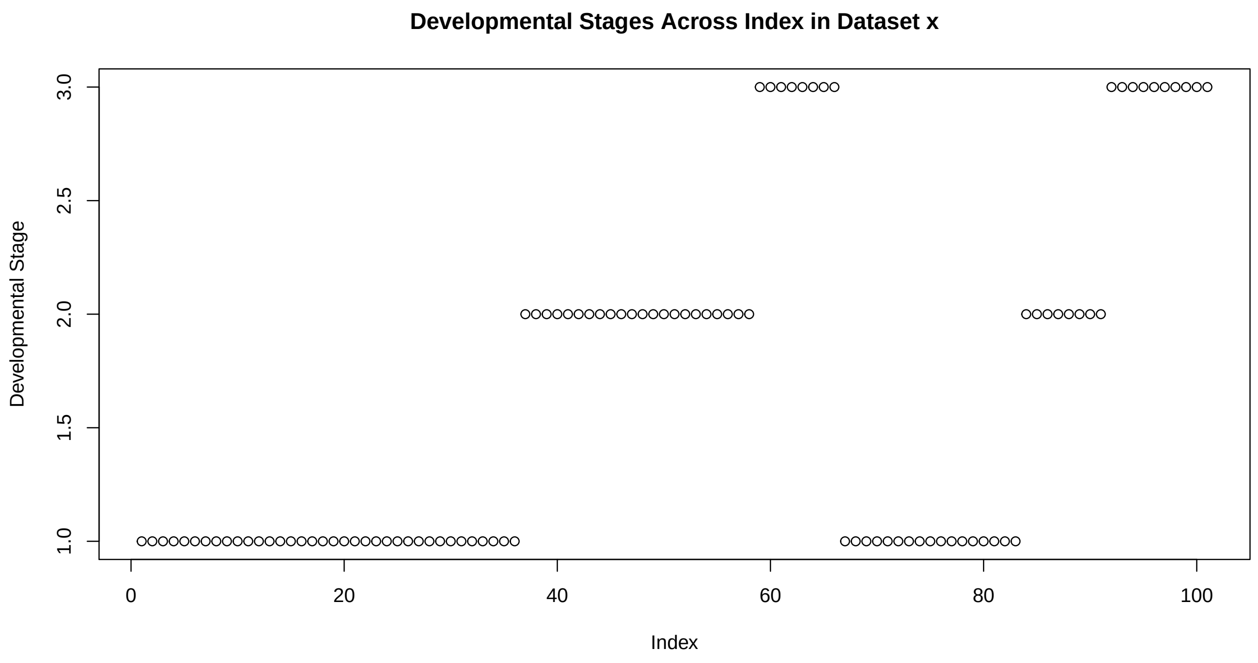
```
table(x=hw3A.x.genotypes,y=hw3A.x.classes) %>% chisq.test()
```

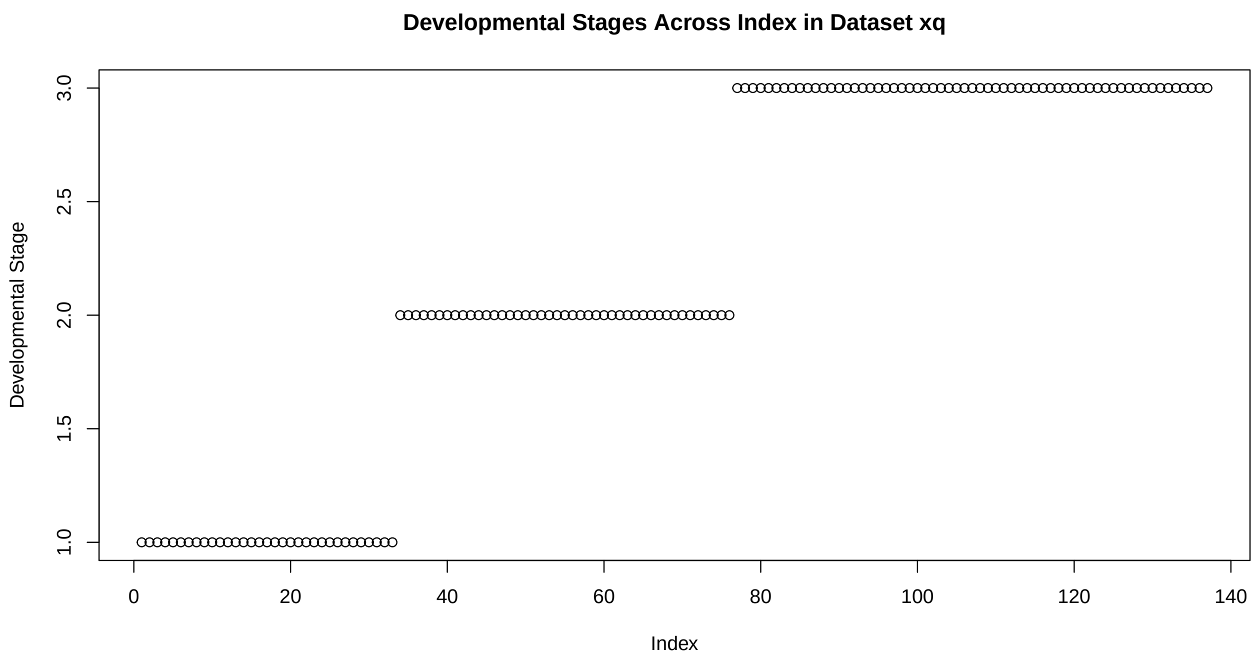
```

	Pearson's Chi-squared test

data:  .
X-squared = 4.4735, df = 2, p-value = 0.1068
```

```
table(hw3A.xq.cell_type,hw3A.xq.classes) %>% chisq.test()
```

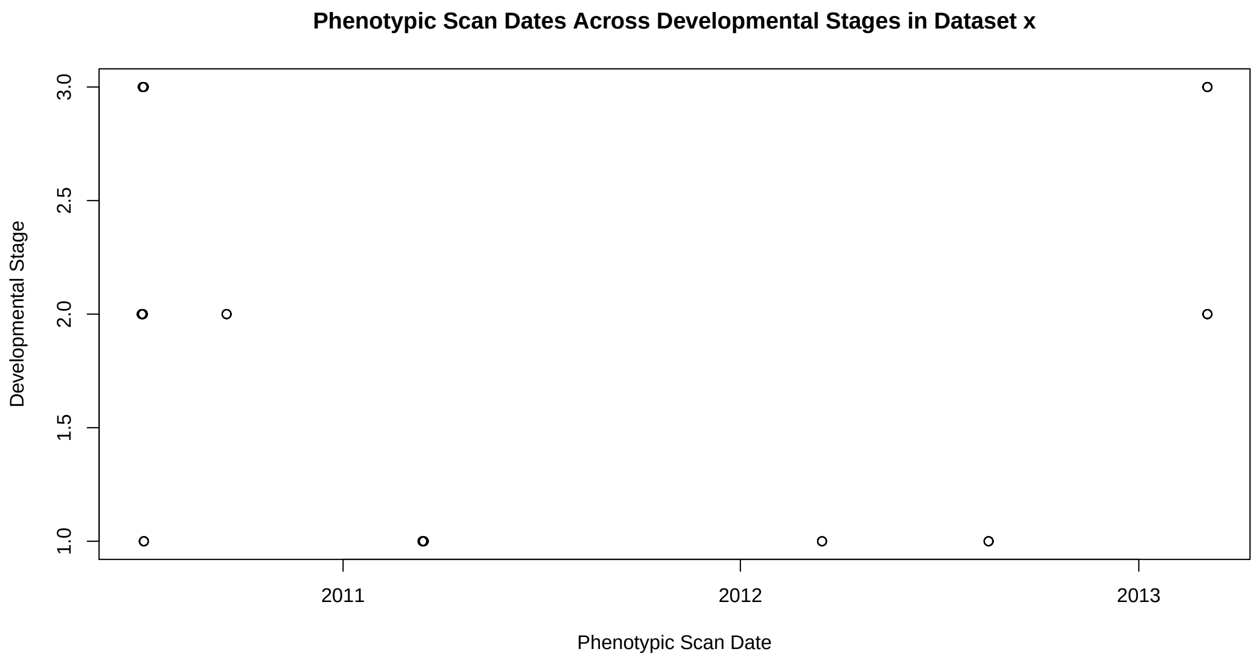
```

	Pearson's Chi-squared test

data:  .
X-squared = 137, df = 4, p-value < 2.2e-16
```

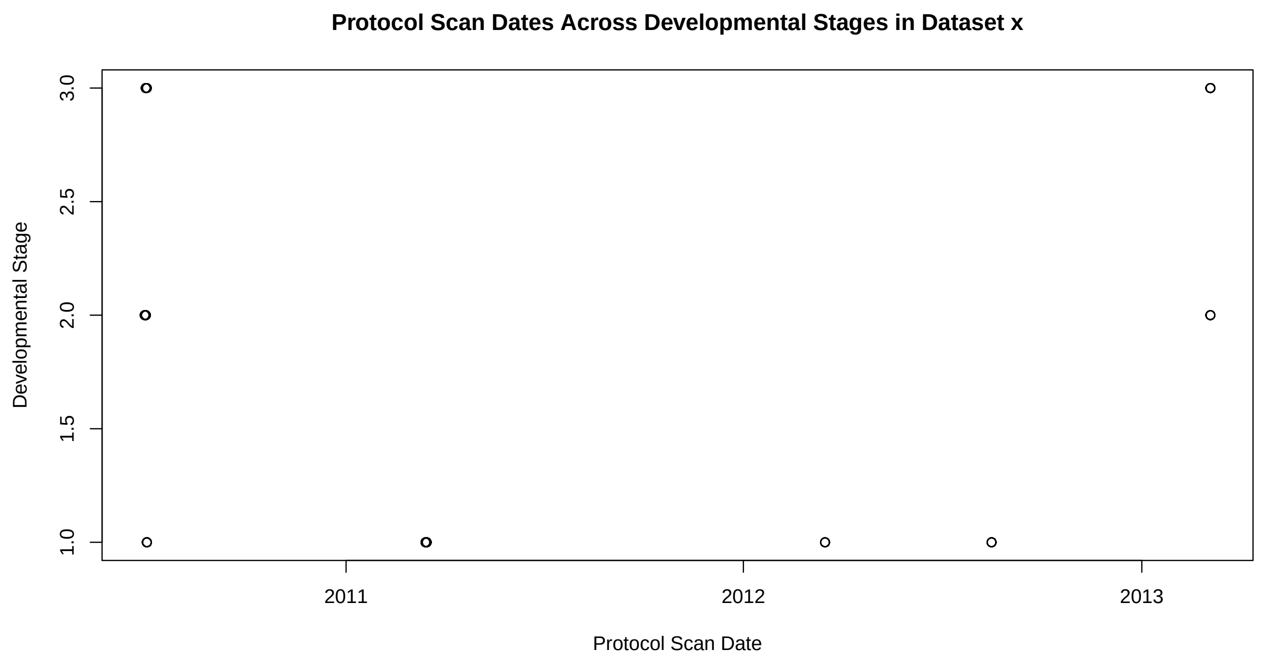
[11]




Developmental Stages Across Index in Dataset xq
Developmental Stage
3.0
2.5
2.0
1.5
1.0
0
20
40
60
80
100
120
140
Index

[12]


Phenotypic Scan Dates Across Developmental Stages in Dataset x
Developmental Stage
3.0
2.5
2.0
1.5
1.0
2011
2012
2013
Phenotypic Scan Date



[13]

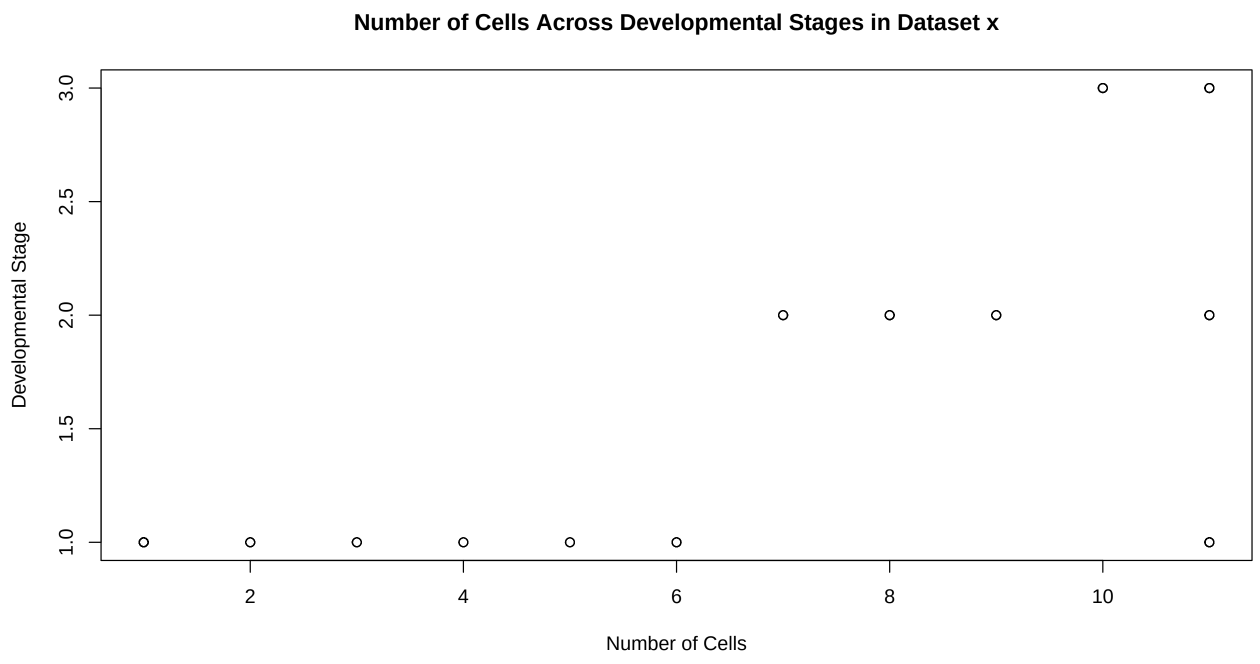
```r
table(hw3A.x.pheno_dates,hw3A.x.classes) %>% chisq.test()
```

```

    Pearson's Chi-squared test

data:  .
X-squared = 114.83, df = 16, p-value < 2.2e-16
```

```r
table(hw3A.x.proto_dates,hw3A.x.classes) %>% chisq.test()
```

```

    Pearson's Chi-squared test

data:  .
X-squared = 96.497, df = 14, p-value = 2.217e-14
```

[14]



[15]

```r
table(hw3A.x.num_cells,hw3A.x.classes) %>% chisq.test()
```

```
    Pearson's Chi-squared test

data:  .
X-squared = 136.28, df = 20, p-value < 2.2e-16
```