

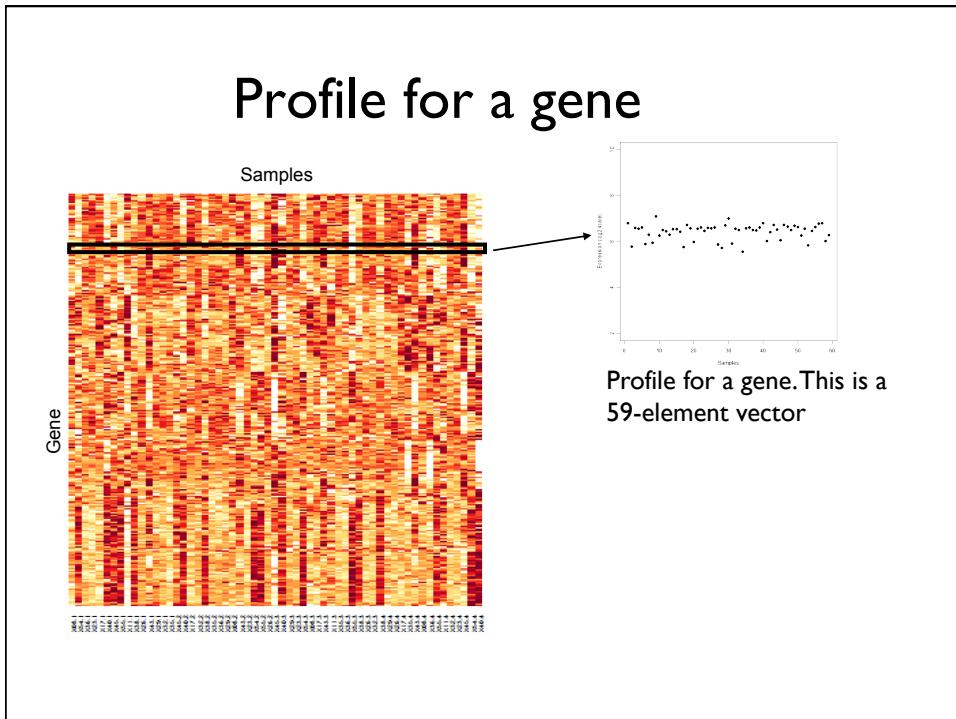
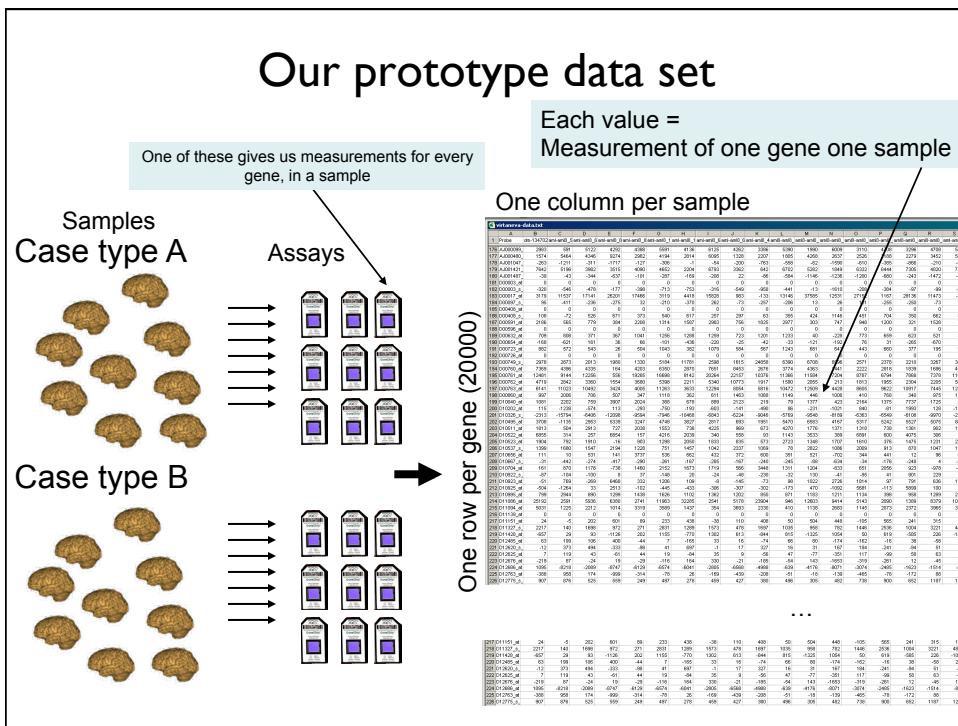
Statistical methods for  
high-dimensional biology

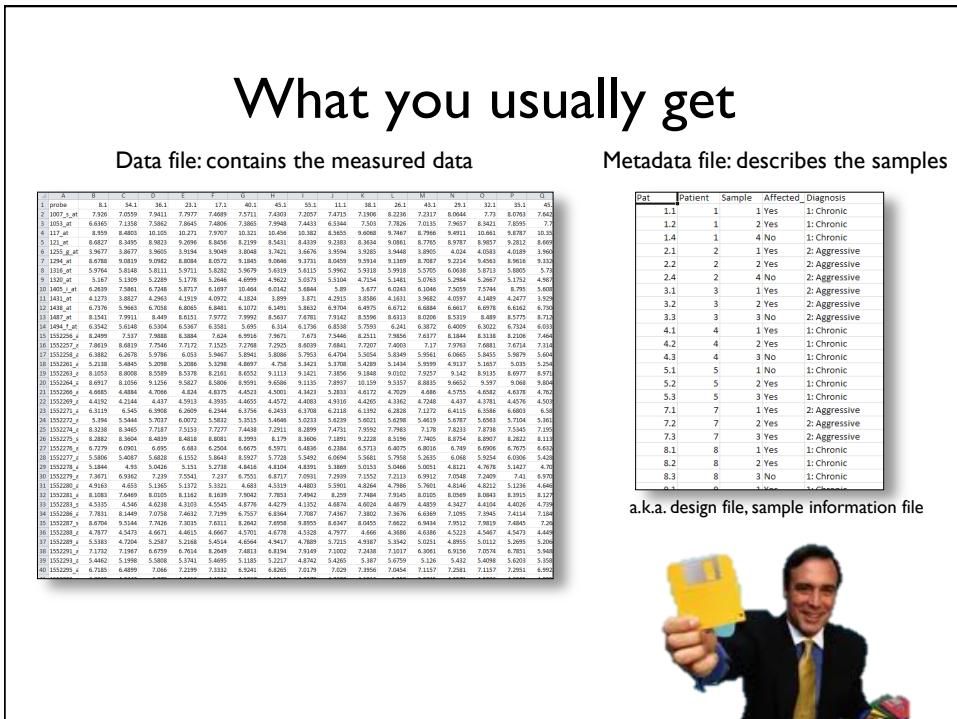
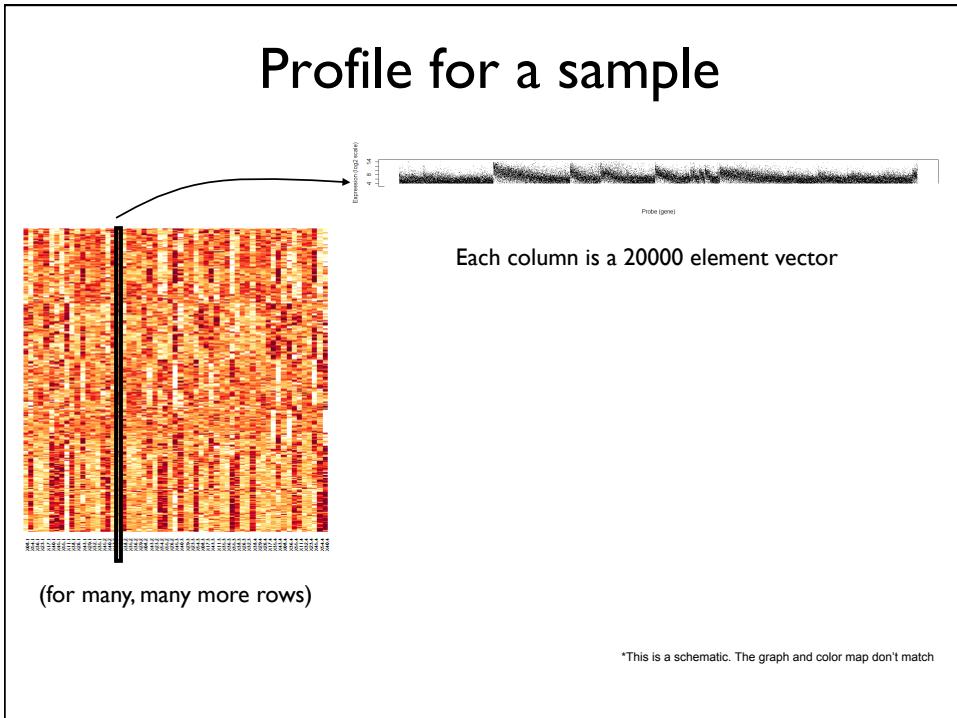
STAT 540 2016

**Data exploration**

Paul Pavlidis

Project pitches





	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	probe	8.1	54.1	36.1	23.1	17.1	40.1	45.1	55.1	11.1	38.1	26.1	43.1	29.1	32.1	35.1	45.	
2	1007_s_at	7.926	7.0559	7.9411	7.7977	7.4689	7.5711	7.4303	7.2057	7.4715	7.1906	8.2236	7.2317	8.0644	7.73	8.0763	7.642	
3	1053_at	6.6365	7.1358	7.5862	7.8645	7.4806	7.3865	7.9948	7.4433	6.5344	7.503	7.7826	7.0135	7.9657	8.3421	7.8595	7.7	
4	117_at	8.959	8.4803	10.105	10.271	7.9707	10.321	10.456	10.382	8.5655	9.6068	9.7467	8.7966	9.4911	10.661	9.8787	10.35	
5	121_at	8.6827	8.3495	8.9823	9.2696	8.8456	8.2199	8.5431	8.4339	9.2383	8.3634	9.0861	8.7765	8.9787	8.9857	9.2812	8.669	
6	1255_g_at	3.9677	3.8677	3.9605	3.9194	3.9049	3.8048	3.7421	3.6676	3.9594	3.9285	3.9448	3.8905	4.024	4.0583	4.0189	3.960	
7	1294_at	8.6788	9.0819	9.0982	8.8084	8.0572	9.1845	9.0646	9.3731	8.0459	9.5914	9.1369	8.7087	9.2214	9.4563	8.9616	9.332	
8	1318_at	5.9764	5.8148	5.8111	5.9711	5.8282	5.9679	5.6319	5.6115	5.9962	5.9318	5.9918	5.5705	6.0638	5.8713	5.8805	5.73	
9	1320_at	5.167	5.1309	5.2289	5.1778	5.2646	4.6999	4.9622	5.0373	5.5104	4.7154	5.1931	5.0763	5.2984	5.2667	5.1752	4.987	
10	1405_i_at	6.2639	7.5861	6.7248	5.871	6.997	10.464	6.0142	5.6044	5.29	5.677	5.73	6.1046	7.5059	7.5744	8.795	5.608	
11	1431_at	4.1273	3.8827	4.2963	4.19	3.97	3.83	3.899	3.97	3.95	3.97	3.97	4.331	3.962	4.0597	4.1489	4.2477	3.929
12	1438_at	6.7376	5.9663	6.7058	6.804	6.4831	6.107	6.1491	6.61	6.97	6.75	6.512	6.6884	6.6617	6.6978	6.6162	6.730	
13	1487_at	8.1541	7.9911	8.449	8.6151	7.9772	7.9992	8.5637	7.6781	7.9142	8.5596	8.6313	8.0206	8.5319	8.489	8.5775	8.712	
14	1494_f_at	6.3542	5.6148	6.5304	6.5367	6.3581	5.695	6.314	6.1736	6.8538	5.7593	6.241	6.3872	6.4009	6.3022	6.7324	6.033	
15	1552256_d	8.2499	7.537	7.9888	8.3884	7.624	6.9916	7.9671	7.673	7.5446	8.2511	7.9856	7.6377	8.1844	8.3138	8.2106	7.464	
16	1552257_d	7.8619	8.6819	7.7546	7.7172	7.1525	7.2768	7.2925	8.6039	7.6841	7.7207	7.4003	7.17	7.9763	7.6881	7.6714	7.314	
17	1552258_d	6.3882	6.2678	5.9786	6.053	5.9467	5.8941	5.8086	5.7953	6.4704	5.5054	5.8349	5.9561	6.0665	5.8455	5.9879	5.604	
18	1552261_d	5.2138	5.4845	5.2098	5.2086	5.3298	4.8697	4.758	5.3423	5.3708	5.4289	5.1434	5.9599	4.9137	5.1657	5.035	5.254	
19	1552263_d	8.1053	8.8008	8.5589	8.5378	8.2161	8.6552	9.1113	9.1421	7.3856	9.1848	9.0102	7.9257	9.142	8.9135	8.6977	8.971	
20	1552264_d	8.6917	8.1056	9.1256	9.5827	8.5806	8.9591	9.6586	9.1135	7.8937	10.159	9.5357	8.8835	9.6652	9.597	9.068	9.804	
21	1552266_d	4.6685	4.4884	4.7066	4.824	4.8375	4.4523	4.5001	4.3423	5.2833	4.6172	4.7029	4.686	4.5755	4.6582	4.6378	4.762	
22	1552269_d	4.4192	4.2144	4.437	4.5913	4.3935	4.4655	4.4572	4.4083	4.9316	4.4265	4.3352	4.7248	4.437	4.3781	4.4576	4.503	
23	1552271_d	6.3119	6.545	6.3908	6.2609	6.2344	6.3756	6.2433	6.3708	6.2118	6.1392	6.2828	7.1272	6.4115	6.3586	6.6803	6.58	
24	1552272_d																	
25	1552274_d																	
26	1552275_d																	
27	1552276_d																	
28	1552277_d																	
29	1552278_d																	
30	1552279_d																	
31	1552280_d																	
32	1552281_d																	
33	1552283_d																	
34	1552286_d																	
35	1552287_d																	
36	1552288_d																	
37	1552289_d																	
38	1552291_d																	
39	1552293_d																	
40	1552295_d																	

Now what?



Ready for exploration																	
<ul style="list-style-type: none"> <li>Understand/get a feel for the data</li> <li>Formulate hypotheses / develop models</li> <li>Identify problems</li> </ul>																	

## The biggest mistake in data analysis

If you don't look at the data ... you are going to get in trouble.

- Not just at the beginning, but at every stage.
- That could mean making graphs, or staring at numbers. Probably both.
- “Sanity checks” should make up a lot of your effort.

**What are some “first questions” you might ask?**

### Some suggestions for first questions

- Is the file the expected size? Format?
- Do we have the expected number of samples?
  - Are there extra columns/rows in either file?
- Do the sample names in both files match?
  - Do not assume same ordering!
  - Watch for R mangling names
- Are the data row names in a format I can use
  - Gene names, probe identifiers, etc.
- Do we have all the sample information we want?
- Is the data numeric? Integers or decimal? Ratios (to what?)
- Are the data on a log scale? If so what is the base?
- Are there missing data points?
- Do factors have the expected number of levels?

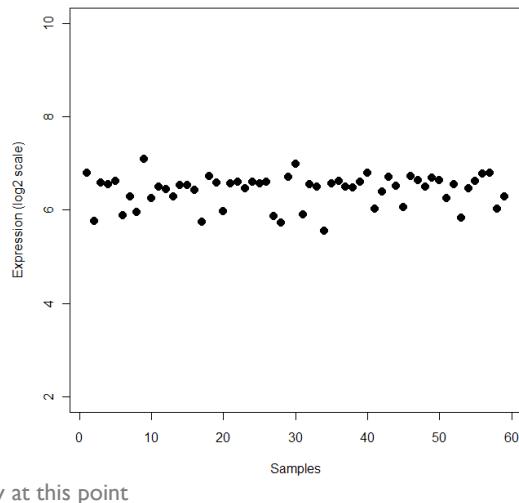
If you are the one generating the data, save yourself grief by paying attention to these issues up front.

**Document what you did and be consistent!**

If you are the analyst, hopefully you were involved in the design stage so there will be fewer surprises

## Data for one gene

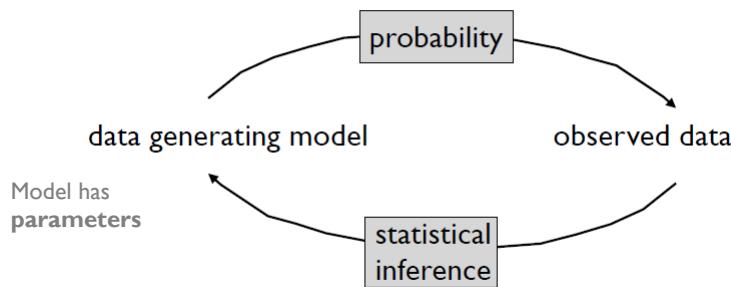
Why do the values vary?



Order of samples is arbitrary at this point

## Making sense of the data

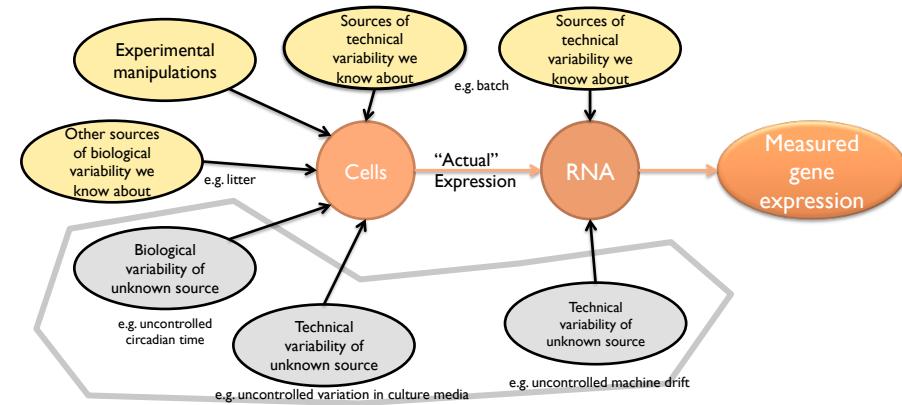
- Review: Data is what we observe, we want to infer something about “where it came from”



Jenny Bryan

## Model of where expression data comes from

- The measured expression level of gene<sub>i</sub> is the combination of many effects.
- Analysis goal is often to determine relative role of effects, separate interesting from “uninteresting”
- One person’s noise is another’s signal.



## Variability: friend and foe

- First line of defense: Know the enemy
  - You can only “correct” for things you know about.
  - Keep track of potential sources of variance: Batches of reagents, slides, personnel
- Design experiments to minimize impact of technical variability.
  - Avoid batches / minimize batch differences
  - Randomize design with respect to batches
- Replication
  - Biological (important)
  - Technical (*usually* less important but might need to convince yourself)
- Much more later in course

## Exploratory numbers

Use as a rough description of the data

- Range
- Mean, Mode, Variance
- Number of missing values

Examine these values per-sample and per-element (e.g. gene)!

## Exploratory graphics

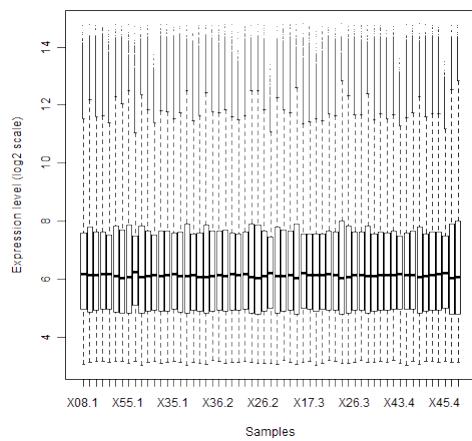
- “Exploratory analysis” is “compute a little and graph a lot”
- I’ll show a few simple approaches that are common in genomics
- Basis for examples: a data set of 59 Affymetrix microarrays, each of which have 54675 measurements. RNA was from blood cells of patients being treated for periodontal infection.

## Boxplots to compare samples

**Quick and dirty:** Reasonable tool for summarizing large amounts of data.

Not so great if your distribution is multimodal.

Don't use boxplots if you don't have to  
– show the actual data (esp. for small sample sizes!); use a strip plot. But during exploration it's more important to make it effective rather than "publication-ready"

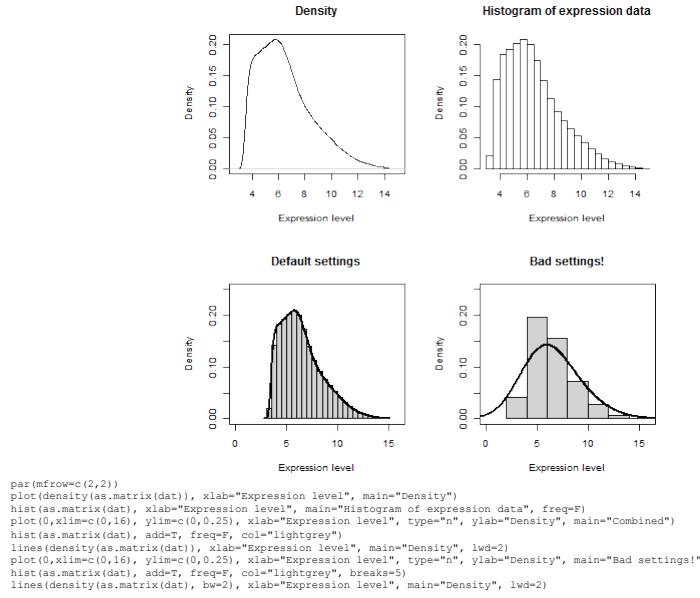


```
boxplot(dat, pch='.', xlab="Samples", ylab="Expression level (log2 scale)")
dev.print("basic.boxplot.png", device=png, width=500)
boxplot(dat[,1], pch='.', xlab=names(dat)[1], ylab="Expression level (log2 scale)")
```

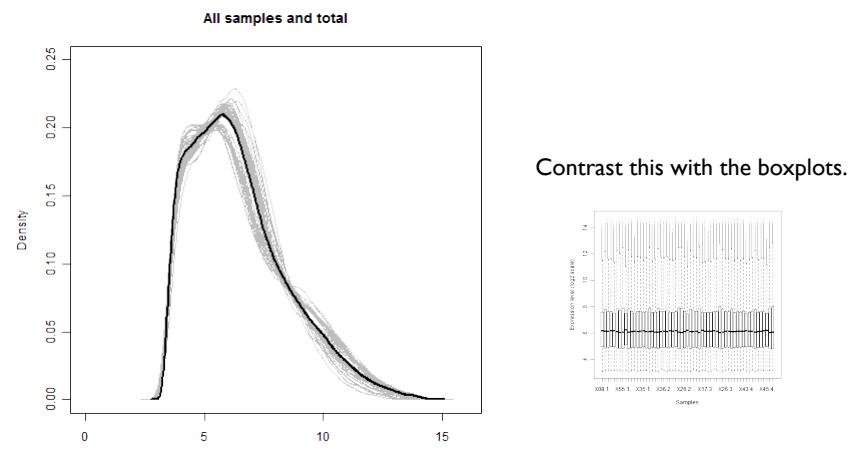
## Histograms

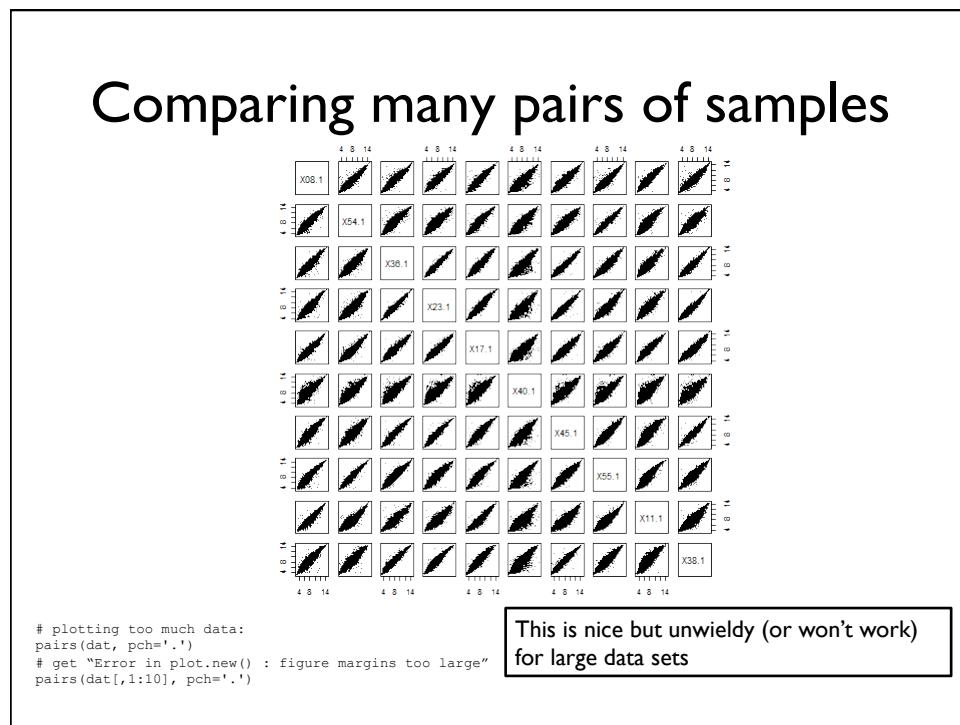
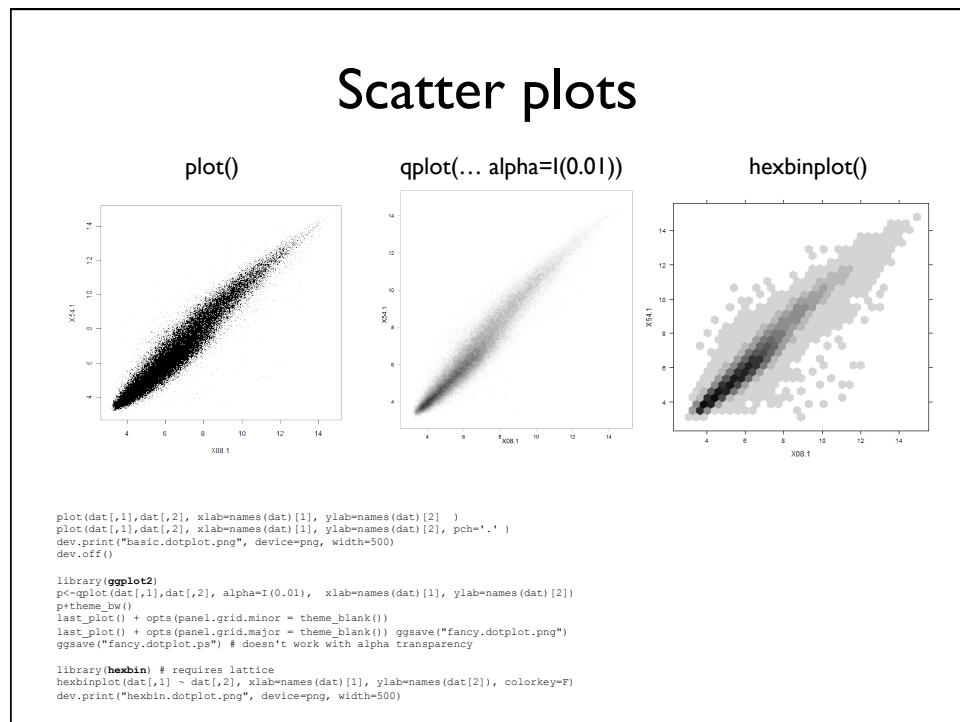
- Box plot's big brother but takes up more space.
- Consider using "density()" instead
- Choose sensible bin sizes and bandwidth

## Histogram and density examples

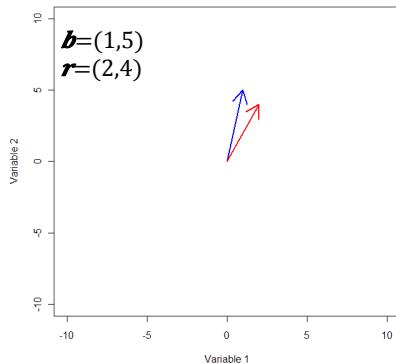


## Overlay densities of all samples

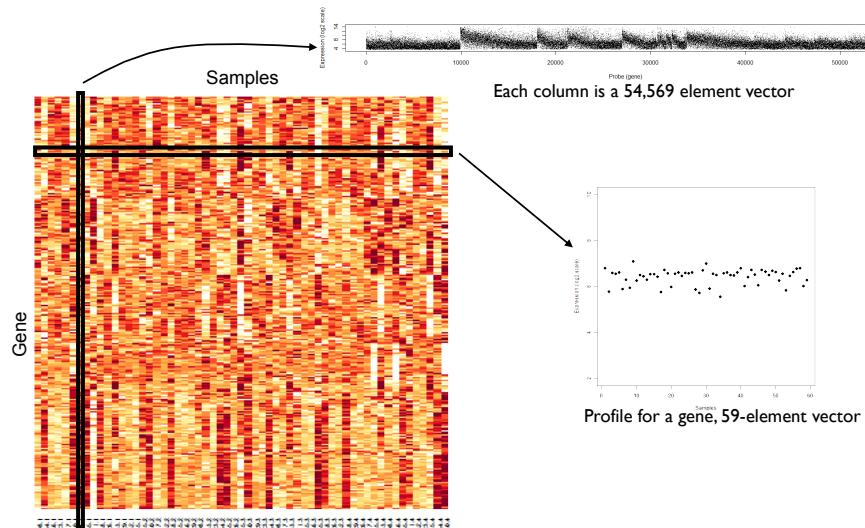




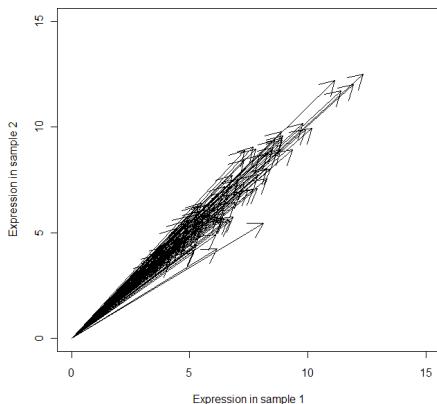
## Digression into vectors



Rows and columns of your data matrix are vectors



## First two dimensions of a bunch of gene vectors



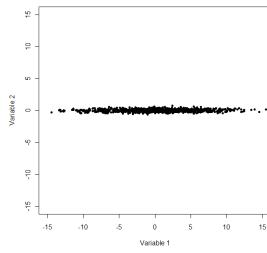
Each vector is a gene (just 100 shown).  
Variables/Coordinates are “assayed samples”



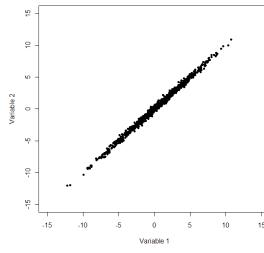
Without the arrows you can plot all the genes  
Do the genes form “interesting patterns” in “sample space”?

## Data as points in space

- Distributed with a certain shape (and orientation.)
- Want to find “interesting patterns” in these “clouds”
- Do we need all 59 (or 54569) dimensions?
  - Often we are interested in finding interesting lower-dimensional structure in the (or representations of) data (regression, clustering, PCA)
- Motivating examples:



One of these dimensions is useless

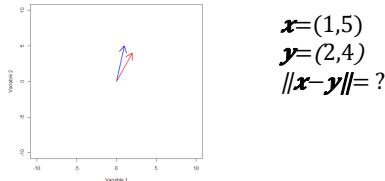


Dimensions are redundant

## Comparing vectors: Euclidean distance

- The length of the difference between two vectors.
- In high dimensions, distances get big.

$$d(x, y) = \|x - y\| = \sqrt{\sum (x_i - y_i)^2}$$

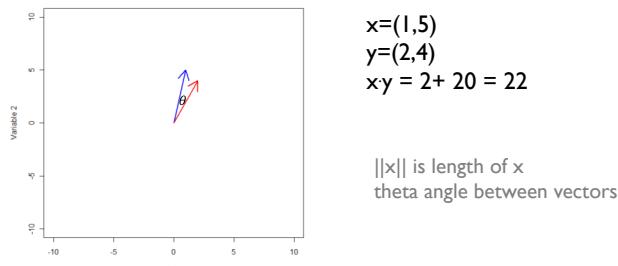


For matrices in R: `dist()`

## Comparing vectors: Dot product

- Also known as **inner product** or **scalar product**
- If two vectors are “nearer” each other, it’s bigger
- Can be negative, unlike Euclidean distance

$$x \cdot y = \sum x_i y_i = \cos \theta \|x\| \|y\|$$



In R: `%*%`

## Covariance

Recall: Sample variance       $s^2(\mathbf{x}) = \frac{\sum_i^N (x_i - \bar{x})^2}{N - 1}$

Sample covariance       $\text{Cov}(\mathbf{x}, \mathbf{y}) = \frac{\sum_i^N (x_i - \bar{x})(y_i - \bar{y})}{N - 1}$

“Are you far from your mean when  
I am far from mine?”

What happens if the data are mean centered?

## Comparing vectors: Pearson correlation

- Two vectors are pos. correlated if they point “in the same direction”
  - Equivalently, if the angle between them is small
- Always between  $1$  and  $-1$ , no matter how many dimensions.
- What happens if the data is **standardized** (mean 0, var 1)?

Covariance:

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Variances: “normalizes” the covariance

In R: `cov()` and `cor()`; `scale()` for standardizing

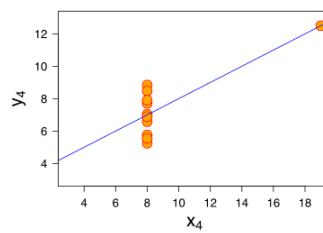
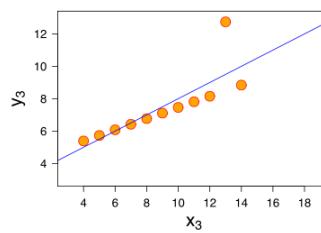
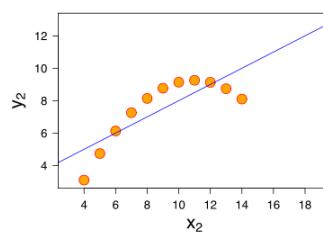
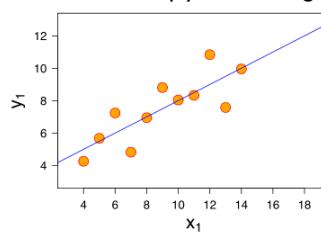
**Self-study exercise:** think about the metrics on the last few slides for cases of:

- orthogonal vectors
- vectors with opposite direction
- vectors with same direction
- High-dimensional data vs. lower-dimensional data
- What happens if variance of one vector is zero
- etc...

Supplement with sketches and/or plots in R

## Look at the data!

Correlation does not imply “interesting”, and lack of correlation does not mean “boring”

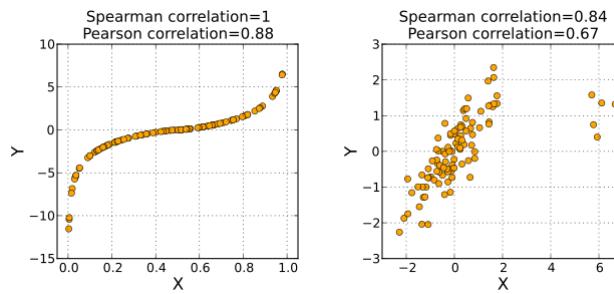


[http://en.wikipedia.org/wiki/File:Anscombe%27s\\_quartet\\_3.svg](http://en.wikipedia.org/wiki/File:Anscombe%27s_quartet_3.svg)

## Remedy for some problems: ranks

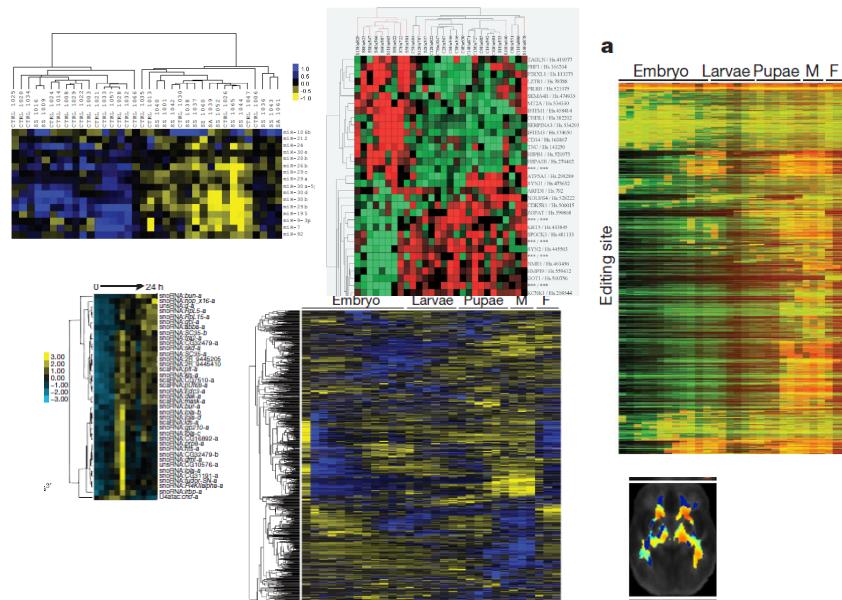
The correlation of the ranks is robust to “non-linear” relationships and outliers.

- Annoyance: dealing with ties



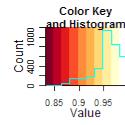
Wikipedia

## Heat maps

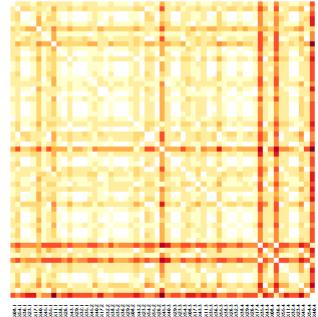


## Using heat maps to compare assays

- Pairwise Pearson correlations of entire sample expression profiles

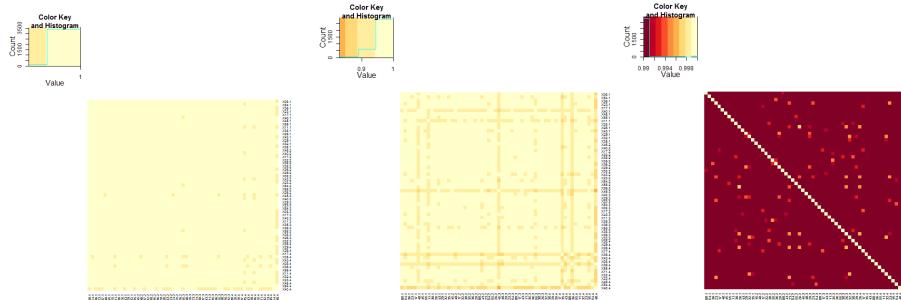


- This method loses some information, but is suitable for quite large data sets.
- Not for comparing many features (genes)!



```
library(gplots)
library(RColorBrewer)
cols<-c(rev(brewer.pal(9,"YlOrRd")), "#FFFFFF")
heatmap.2(cor(dat), Rowv=NA, Colv=NA, symm=T,
trace="none", dendrogram="none", col=cols, cexCol=0.5,
cexRow=0.5)
dev.print("heatmap.png", device=png, width=500)
```

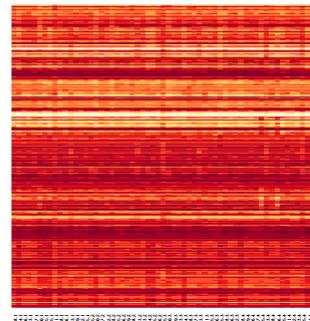
## Choose an appropriate range



R sets “reasonable” ranges for you; but you can control it (`breaks`) and will often want to.  
 Depending on the distribution of values, might want to do  
`diag(m)<-NA` so the diagonal doesn’t dominate the color space (say, if the correlations are all <<1)

## Setting up data heat maps

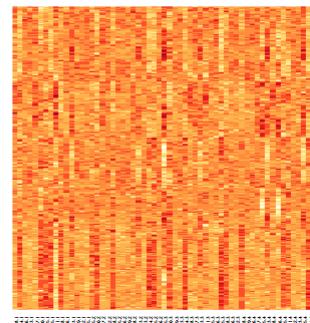
- I've taken 500 rows at random from my big data set
- The heat map here was created using this data "as is".
- I did cheat a little: The row ordering is not random.
- Note that this lecture doesn't get into dealing with missing data – how to evaluate it and visualize it



```
heatmap.2(bitOfData, Rowv=NA, Colv=NA, scale=NULL, trace="none", dendrogram="none", col=cols, cexCol=0.5, labRow=NA)
```

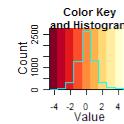
## Revision |

- Same input data, but each row is scaled to have mean 0 and variance 1 (z-scores)
  - Subtract the mean; divide by the standard deviation. use `scale()` on the data rows.
- It is now easier to compare the rows and seem some structure.
- But looks kind of bland compared to ones you often see in the literature



Note: R heatmap scales rows by default. You must disable this to give yourself control over how it turns out.

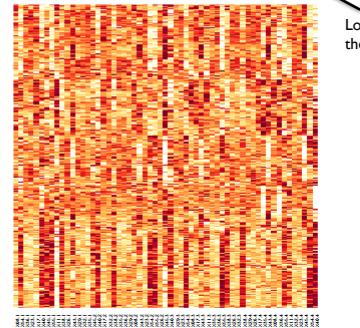
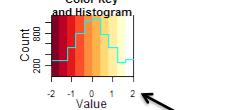
```
heatmap.2(scaledBit, Rowv=NA, Colv=NA, scale=NULL, trace="none", dendrogram="none", col=cols, cexCol=0.5, cexRow=0.5, labRow=NA)
```



## Revision 2: Adjusting contrast

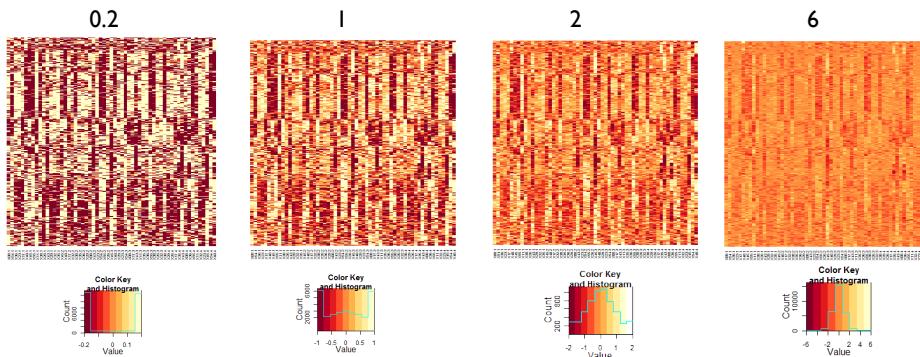
- Same input data again, but after scaling I clipped the range of values to  $\{-2, 2\}$
- This means “anything more than two standard deviations from the row mean is set to 2”
- Now extremes have no effect; “Higher contrast”
- Limit values of the range limit of 2 or 3 are usually good.

Heat maps you see in the literature are almost always set up this way. Be aware of what's really going on.



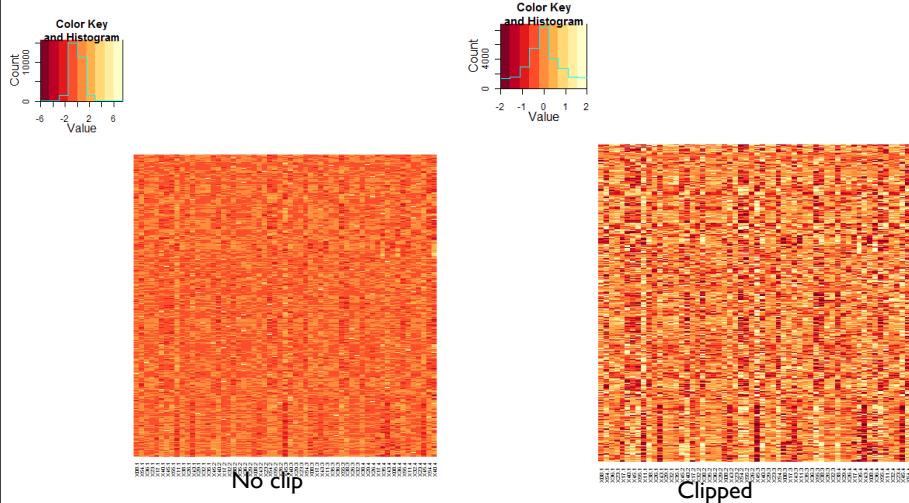
```
limit<-2
trim.scaledBit<-scaledBit
trim.scaledBit[which(trim.scaledBit < -limit)]<- -limit
trim.scaledBit[which(trim.scaledBit > limit)]<- limit
heatmap.2(trim.scaledBit, Rowv=NA, Colv=NA, scale=NULL, trace="none",
col=cols, cexCol=0.5, cexRow=0.5, labRow=NA)
```

## Varying clip limit



## Another reason why clipping is and isn't good

I added sparse random “spiky” values to the data (I just multiplied them by 3)  
 Clipping hides these “outliers” but allows us to see variation in the bulk of the data



## Plotting too much data



The entire data set.

If the cells are less than 1 pixel,  
 everything starts to turn to mush  
 and can even be misleading.

(This won't work in R unless  
 you print directly to a file.)

## Choice of colours

- R defaults: ketchup and mustard
- RColorBrewer: good maps based on work of visualization expert (matrix2png uses them too)
- Red-black-green you often see in papers is **bad choice** for colour blind individuals.
- Blue-black-yellow is better
- Greyscale: loss of dynamic range, but cheaper to publish!
- My favourite: matrix2png "black body"
- Humans can't really tell the difference between a 8 and 16 colour scale.

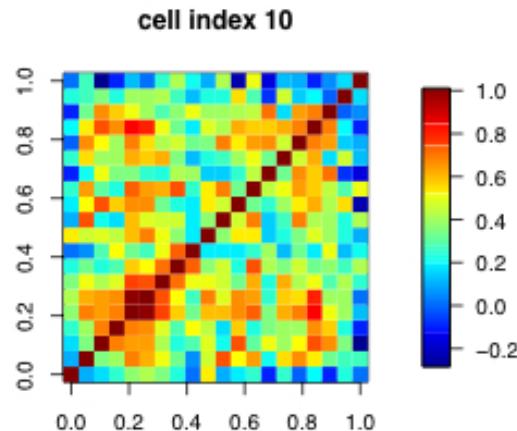
## Divergent vs. sequential maps

Colours pass through black at (in this case) zero. Yellow="Below the mean", Blue="Above the mean" for the row. This can be the right thing to do, especially if your original data are naturally "symmetric" around zero (or some other value). Otherwise it might just be confusing.

Colours go from light to dark. Darker colours mean "higher" by default in RColorBrewer; I usually reverse to make dark mean less.

YlOrRd
YlOrBr
YlOrBu
YlGn
Reds
RdYl
Purples
PuRd
PuBu
OrRd
Oranges
Greys
Greens
GnBu
BuGn
Blues

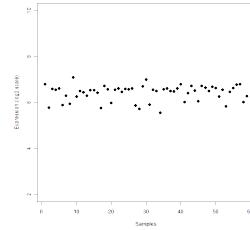
## A confusing heat map (at least for me; one can get used to anything)



<http://mikelove.wordpress.com/category/visualization/>

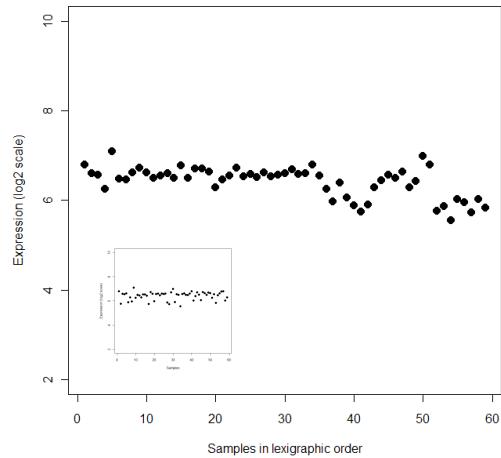
## Slicing and dicing

- I mean: Viewing data arranged or grouped by factors of interest
- In my example, I have multiple “visits” for each “patient”
  - Any interesting trends we can spot?
  - Are intra-subject samples more similar than inter-subject samples?



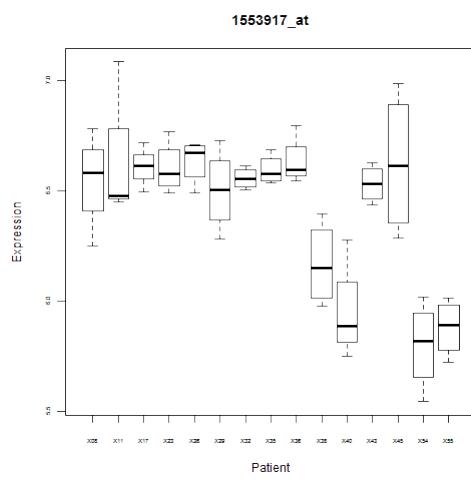
## Playing around with one gene

- I sorted the samples by their names
- Is this structure meaningful? Is it interesting?
- Maybe we need more information.
- (And some statistics)



## Explicitly group by patient

- We're developing the concept that we can look for "factors" that "explain" some of the variation.
- Is this interesting? We don't know, but it suggests that accounting for Patient differences might be reasonable.
- Need to get more comprehensive view ...

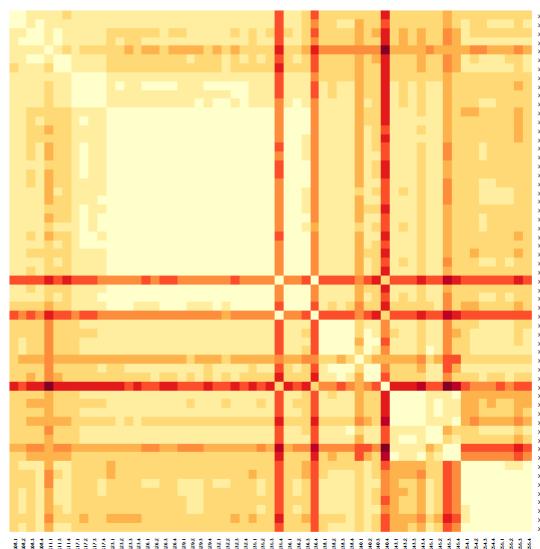
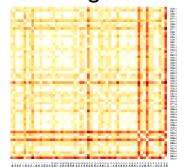


(Boxplots not ideal for this, but easy)

## Arrange data by Patient

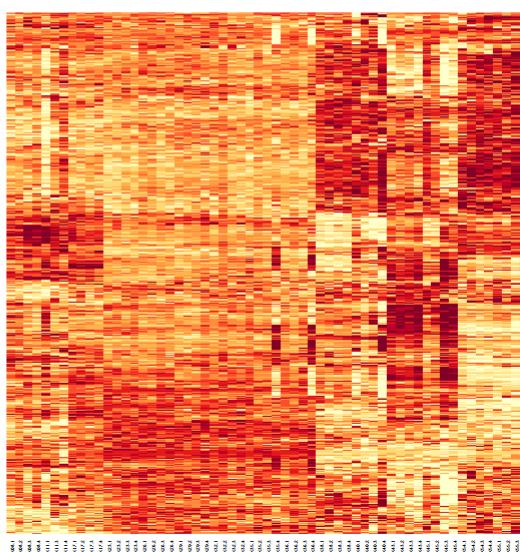
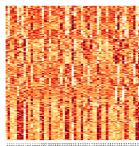
The sample correlations again;  
dark = low values

Original



## View of (part of) the data organized by Patient

(Again, row order  
is not random  
either)



## Summary

- Sanity checks
- Use graphs to look at the data, slicing and dicing – build up a library of techniques that work for your data
- Additional exploratory techniques will be discussed later in the course
  - Clustering
  - PCA
- Also: missing values