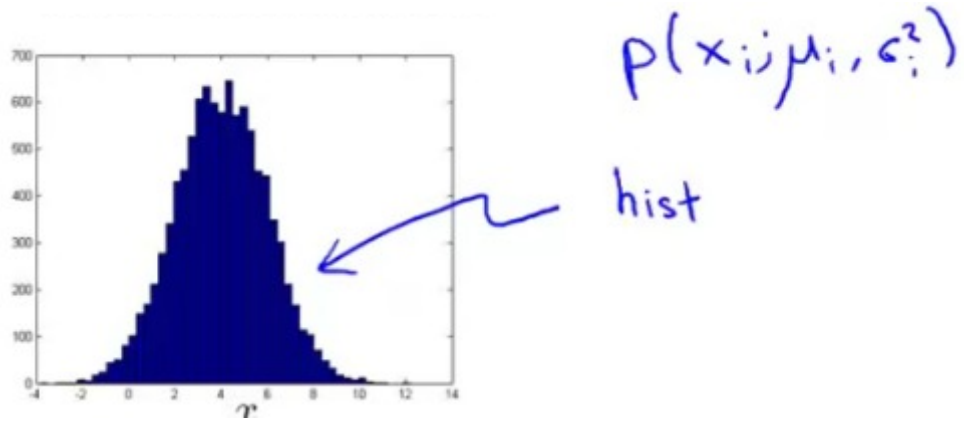
Choosing What Features to Use

The features will greatly affect how well our anomaly detection algorithm works.

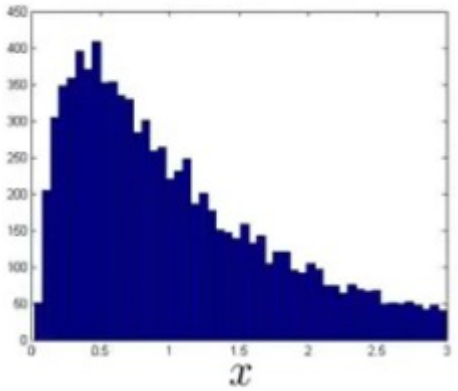
Non-Gaussian features:

We can check that our features are **gaussian** by plotting a histogram of our data and checking for the bell-shaped curve.



- Often still works if data is non-Gaussian

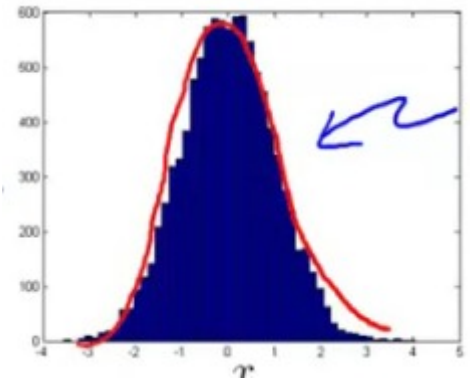
Non-Gaussian data might look like this:



- This doesn't look at all like a bell shaped curve, this is a very **asymmetric distribution**.

What we can do is play with **different transformations** of the data to make it look more Gaussian

- Might take a $\log$ transformation of the data
  - i.e. if we have some feature x_1, replace it with $\log(x_1)$



- This looks much more Gaussian

Some **transforms** we can try on an example feature x that does not have the bell-shaped curve are:

- $\log(x)$
- $\log(x+1)$
- $\log(x+c)$ for some constant
- $\sqrt{x}$
- $x^{1/3}$

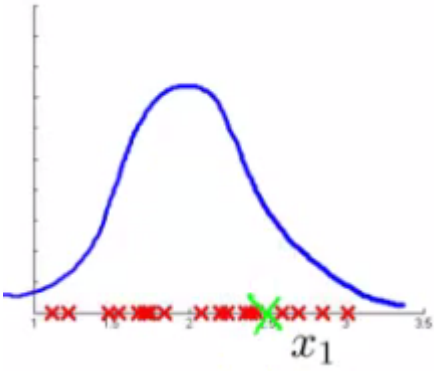
To summarize, if we plot a histogram with the data, and find that it looks pretty non-Gaussian, it's worth playing around a little bit with different transformations to see if we can make our data look a little bit more Gaussian, before we feed it to your learning algorithm.

Error analysis for anomaly detection

There is an **error analysis procedure** for anomaly detection that is very similar to the one in supervised learning.

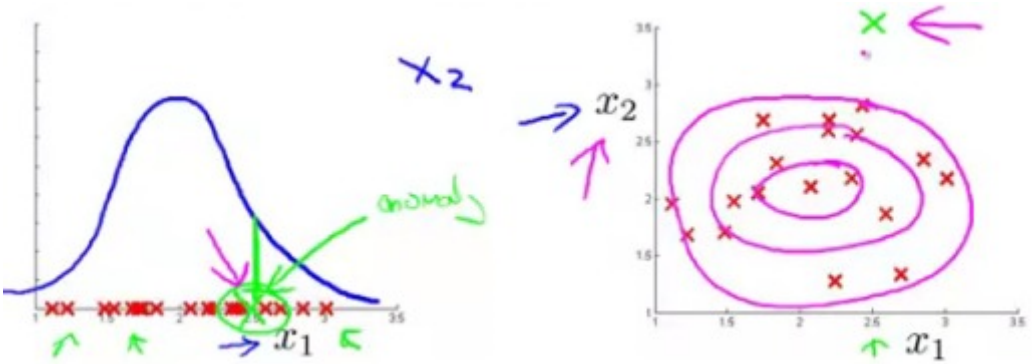
- Run algorithm on CV set
- See which one it got wrong
- Develop new features based on trying to understand why the algorithm got those examples wrong
  - Our goal is for $p(x)$ to be **large** for normal examples and **small** for anomalous examples.

One **common problem** is when $p(x)$ is similar or comparable for both types of examples (normal and anomalous examples).



In this case, we need to examine the anomalous examples that are giving high probability in detail and try to **figure out new features** that will better distinguish the data.

- Can looking at that example help develop a new feature (x_2) which can help distinguish further anomalous



- In general, choose features that might take on unusually large or small values in the event of an anomaly.

Monitoring computers in a data center (what features to use)

Choose features that might take on unusually large or small values in the event of an anomaly.

- x_1 = memory use of computer
- x_2 = number of disk accesses/sec
- x_3 = CPU load
- x_4 = network traffic

- We suspect CPU load and network traffic grow linearly with one another
  - If server is serving many users, CPU is high and network is high
  - Fail case is infinite loop, so CPU load grows but network traffic is low
    - New feature: x_5 = CPU load/network traffic
    - May need to do feature scaling

Video Question: Suppose your anomaly detection algorithm is performing poorly and outputs a large value of $p(x)$ for many normal examples and for many anomalous examples in your cross validation dataset. Which of the following changes to your algorithm is most likely to help?

- Try using fewer features.

> Try coming up with more features to distinguish between the normal and the anomalous examples.

- Get a larger training set (of normal examples) with which to fit $p(x)$.
- Try changing ϵ.