DS Visualisation and Analysis

Abbey Waldron

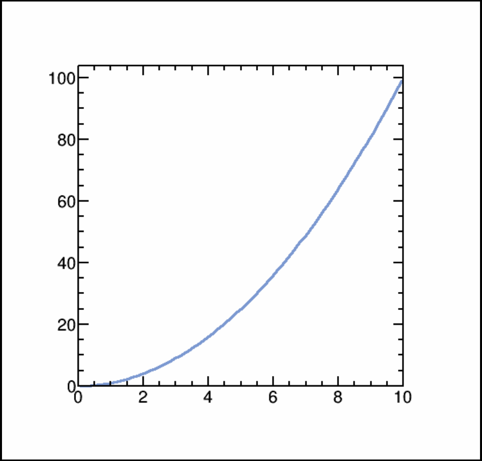
This Week - Fitting, Interpolation and Prediction

1. Show last week's work to the class
2. Fitting
3. Interpolation and Prediction
4. Start work on problems

Random Student Generator

Correlations

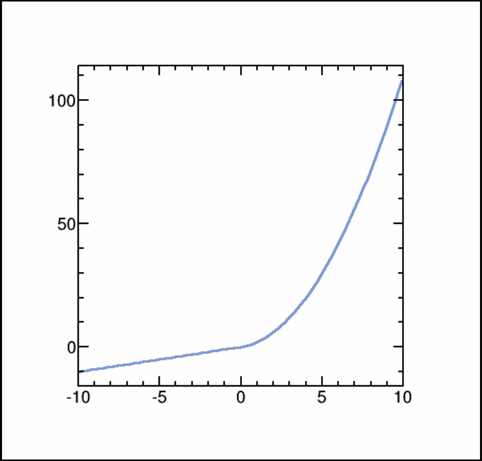
Functions Q1



Q1 Solution

$$f(x) = x^2$$

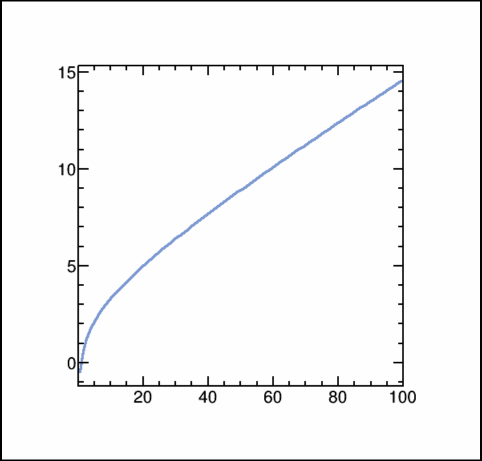
Functions Q2



Q2 Solution

$$f(x) = x + 0.5x^2(1 + \mathrm{erf}(x))$$

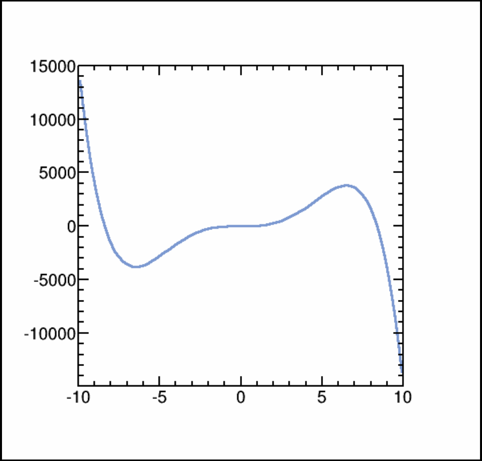
Functions Q3



Q3 Solution

$$f(x) = \log x + 0.1x$$

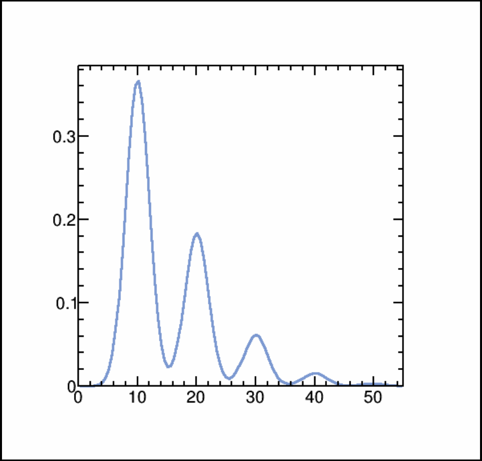
Functions Q4



Q4 Solution

$$f(x) = 35x^3 - 0.5x^5$$

Functions Q5



Q5 Solution

$\mu = 10$, $\sigma = 2$, $\lambda = 1$

$$f(x) = \sum_{i=1}^{n} \texttt{dnorm}(x, i\mu, \sigma) * \texttt{dpois}(i, \lambda)$$

This Week...

I will show you lots of things. Don't panic.

- R makes your life very easy
- You will see this all again in statistics/machine learning class

Interpolation/Prediction

Interpolation - finding values between two known points

Prediction - estimating values outside the range of the data

Fitting vs Interpolation

Interpolation - when you have no or very small errors, for example in your model calculations

Fitting - to deal with errors in your data

Interpolation in R

`smooth.spline()`

Regression Fitting

We want to get a function that describes our data well, but we know there are uncertainties in the data causing some scatter in the data points

Fitting in R

Linear:

```
lm(y ~ x)
```

Non-linear:

```
nls( y ~ f(x), data = , start = list(p0 = , p1 = , …))
```

Calculating sensible starting parameters will make your life easier

Under and Over Fitting

If you define a suitably complex function, you can get it to pass through all of your data points (like with the splines). However, this does not mean that the features in you function really exist, rather they are probably caused by statistical noise - **over fitting**.

If you try to fit a straight line to a non-linear functional relationship then you will not be able to well describe the behaviour of the data - **under fitting**.

Checking for under/over fitting

1. Look at the data and fit and use your brain
2. Ask if the function you have used is the simplest one that could describe the data?
3. Ask if the fit describes the data well?
4. Test the fit on a subset of the data (training set)

Goodness of Fit: χ^2 Test

Calculate the value of the test statistic:

$$\chi^2 = \sum_{i=1}^{n} \frac{(o_i - e_i)^2}{e_i}$$

Where o_i are the observed (data) values, e_i are the expected values, in this case what our fit predicts, and n is the number of fitted data points.

Then compare to the appropriate χ^2 distribution to calculate the probability.

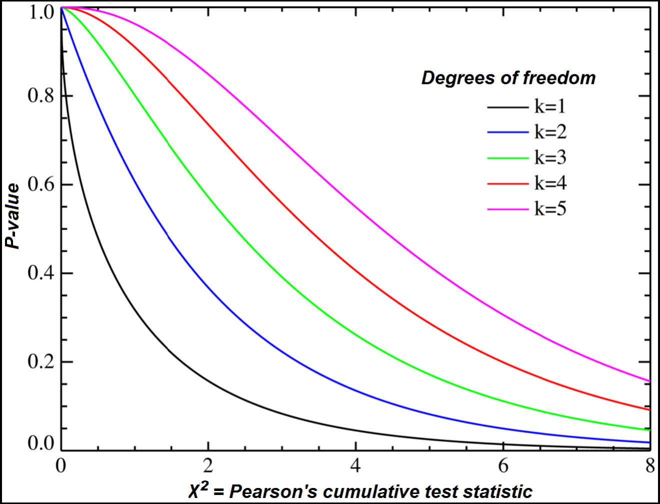
χ^2 Distribution

You need to know the number of degrees of freedom, *k*, in this case:

$$k = n - 1 - p$$

Where *n* is the number of data points, and *p* is the number of free parameters in the fit.

χ^2 Distributions



Wikipedia

Goodness of Fit in R

```
e <- predict(fit_function, x)
chisq.test(y, p = e/sum(e))
```

Prediction

``Prediction is hard, especially about the future.''

Prediction Ranges

Be careful...

How to stay out of trouble:

- Don't make predictions
- If you must make predictions, don't make them too far in the future
- What is too far in the future?

Now all the warnings are over...

Making predictions in R:

- Find a variable that can be used as a predictor for another (`plot()`, `pairs()`, ...)
- Fit a suitable function (`lm()` or `nls()`)
- extrapolate that fit funtion and its errors (`predict()`)

Week 4 Problems

1. Fitting Data
2. Making predictions

Backup Slides