**Bad Honnef School**

Peter Young

**Problem 2, Solution**

Determination of $\langle x^4 \rangle / \langle x^2 \rangle^2$ for a set of 1000 points.

Source code is given in perl and python at the bottom. There are two versions in python, the first uses basic python only, the second uses numpy extensions and is more concise and elegant. (Thanks to Matt Wittmann for doing the python scripts.)

Usage of perl script: `sol_HW2.pl data.HW2`

Usage of python scripts:

```
python sol_HW2.py data.HW2
python sol_HW2_numpy.py data.HW2
```

OUTPUT

```
 Overall average is     1.8215
 Jackknife average is   1.8215 +/- 0.0368
```

The first line is just the standard result where each average is replaced by the average over the data. The second line is the jackknife estimate which includes an error bar. The two estimates agree to the number of digits shown. Using the full precision of the numbers, and extrapolating the bias as described in the lecture notes, I get a result of 1.8212. However, the difference between 1.8215 and 1.8212 is much less than the error bar, and so, as also discussed in the lecture notes, elimination of the bias is not necessary when you have a large number of (statistically independent) data points.
(b) A value of 1.8 corresponds to a rectangular distribution centered at the origin, and this is indeed the distribution used to generate the data. The jackknife estimate agrees with this to within the calculated error bar.

```
================================================
Perl Code:
================================================

#!/usr/bin/perl

$n = 0;
$x2_tot = 0; $x4_tot = 0;
#
# read in the data
#
while(<>)
{
    @line = split;
    $x2[$n] = $line[0]**2;
    $x4[$n] = $x2[$n]**2;
    $x2_tot += $x2[$n];
    $x4_tot+= $x4[$n];
    $n++;
}
#
# Do the jackknife averages
#
for ($i = 0; $i < $n; $i++)
{
```

```
    $x2_jack[$i] = ($x2_tot - $x2[$i]) / ($n - 1);
    $x4_jack[$i] = ($x4_tot - $x4[$i]) / ($n - 1);
}
$x2_av = $x2_tot / $n;
$x4_av = $x4_tot / $n;
$g_av = $x4_av / $x2_av**2;

$g_jack_av = 0; $g_jack_err = 0;
for ($i = 0; $i < $n; $i++)
{
    $dg = $x4_jack[$i] / $x2_jack[$i]**2;
    $g_jack_av += $dg;
    $g_jack_err += $dg**2;
}
$g_jack_av /= $n;
$g_jack_err /= $n;
$g_jack_err = sqrt(($n - 1) * abs($g_jack_err - $g_jack_av**2));

printf " Overall average is   %8.4f\n", $g_av;
printf " Jackknife average is %8.4f +/- %6.4f \n", $g_jack_av, $g_jack_err;
```

```
============================================
Python Code (using basic python constructs only):
============================================

import fileinput
from math import *

x2 = []; x2_tot = 0.
x4 = []; x4_tot = 0.

for line in fileinput.input():  # similar to perl's while(<>)
    line = line.split()
    x2_i = float(line[0])**2
    x4_i = x2_i**2
    x2.append(x2_i)
    x4.append(x4_i)
    x2_tot += x2_i
    x4_tot += x4_i

n = len(x2)

#
# Do the jackknife averages
#
x2_jack = []
x4_jack = []

for i in xrange(n):
    x2_jack.append((x2_tot - x2[i]) / (n - 1))
    x4_jack.append((x4_tot - x4[i]) / (n - 1))

x2_av = x2_tot / n
x4_av = x4_tot / n
```

```
g_av = x4_av / x2_av**2

g_jack_av = 0.; g_jack_err = 0.

for i in xrange(n):
    dg = x4_jack[i] / x2_jack[i]**2
    g_jack_av += dg
    g_jack_err += dg**2

g_jack_av /= n
g_jack_err /= n
g_jack_err = sqrt((n - 1) * abs(g_jack_err - g_jack_av**2))

print " Overall average is   %8.4f" % g_av
print " Jackknife average is %8.4f +/- %6.4f" % (g_jack_av, g_jack_err)
```

```
==============================================
Python Code (using numpy extensions. This is much more
compact and elegant):
==============================================

import sys
import numpy as np

fname = sys.argv[1] if len(sys.argv) > 1 else 'data.txt'
x = np.loadtxt(fname)

x2 = x**2
x4 = x**4

x2_tot = np.sum(x2)
x4_tot = np.sum(x4)

n = len(x)

x2_jack = (x2_tot - x2) / (n - 1)
x4_jack = (x4_tot - x4) / (n - 1)

x2_av = x2_tot / n
x4_av = x4_tot / n
g_av = x4_av / x2_av**2

g_jack = x4_jack / x2_jack**2
g_jack_av = np.average(g_jack)
g_jack_err = np.sqrt(n - 1) * np.std(g_jack)

print " Overall average is   %8.4f" % g_av
print " Jackknife average is %8.4f +/- %6.4f" % (g_jack_av, g_jack_err)
```