

```

# -*- coding: utf-8 -*- import sympy from Incertitudes import * print "Hi... I am defining two
variables: x and sigma" x, sigma, cste = sympy.symbols("x", "sigma", "cste") print "Now, I'll build
two functions..." func = x-x + sigma/sigma func2 = x**sigma + func*cste print "func = " + str(func)
print "func2 = " + str(func2) print "Now here is the incertitude on func: " print
symbolic_abs_error(func, ["x", "sigma", "cste"]) print spec_latex(symbolic_abs_error(func, ["x",
"sigma"])) print "Now here is the incertitude on func2: " print symbolic_abs_error(func2, ["x",
"sigma"]) print spec_latex(symbolic_abs_error(func2, ["x", "sigma"])) print "Now calculating func
with x = 1.0 ± 0.1 and sigma = 3.0 ± 0.001" print pprint_incert(eval_with_incert(func, {"x":[1.0,
0.1], "sigma":[3.0, 0.001], "cste":[1E2, 0.0001]})) print "Now calculating func2 with x = 1.0 ± 0.1
and sigma = 3.0 ± 0.001" print pprint_incert(eval_with_incert(func2, {"x":[1.0, 0.1], "sigma":[3.0,
0.001], "cste":[1E2, 0.0001]})) print "Now building a table..." functions = [[func, "unit"],[func2,
"pow"]] constants = {"cste":[3.0, 0.001]} tab = IncertTableau(["x", "sigma"], functions, constants)
print "Now adding measures..." tab.add_line({"x":[1.0, 0.1], "sigma":[3.0, 0.001]})
tab.add_line({"x":[6, 0.1], "sigma":[0.2, 0.1]}) print "Importing from csv..."
tab.add_csv("sample_in.csv") print "... and displaying result" print tab tab.sort_lines("sigma")
print tab print tab.get_columns(["x", "pow"]) tab.export_to_csv("sample_out.csv", ["x", "Deltax",
"sigma", "Deltasigma", "unit", "pow", "Deltapow"]) tab.export_to_gnuplot("sample_out.dat", ["x",
"Deltax", "sigma", "Deltasigma", "unit", "pow", "Deltapow"]) instructions = "" set encoding utf8
set terminal pdfcairo color enhanced font "Helvetica, 10" lw 2 solid size 15cm, 10cm set output
"gnuplot.pdf" set fit errorvariables f(x) = a + b*x + c*x**2 fit f(x) 'gnutemp.dat' using %x:%{pow}
via a, b, c print sprintf("REGLIN_COEFFS = {'a':%f, %f}, 'b':%f, %f}, 'c':%f, %f})", a, a_err, b,
b_err, c, c_err) set key top left set xlabel "x" set ylabel "pow" plot 'gnutemp.dat' using
%x:%{pow}:%{Deltax}:%{Deltapow} with xyerrorbars title "Mesures", f(x) title "Régression" ""
reglin_results = tab.gnuplotize(instructions, verbose = True, temp_filename = "gnutemp.dat") a =
reglin_results["a"][0] Deltaa = reglin_results["a"][1] for key in reglin_results: print "Coeff %s: %f ±
%f" %(key, reglin_results[key][0], reglin_results[key][1]) print "Now calculating averages and
standart deviation..." print pprint_incert(average_with_incert([1.0, 1.0, 0.0, 2.0])) print
pprint_incert(tab.get_average_for_column("sigma")) raise print "Now printing two datasets into
the same plot..." tabs = [] indexes = ["0", "1"] for index in indexes: tab = IncertTableau(["x",
"sigma"], functions, constants) tab.add_csv("multi_in_" + index + ".csv") tabs.append(tab)
multi_instructions = "" set encoding utf8 set terminal pdfcairo color enhanced font "Helvetica,
10" lw 2 solid size 15cm, 10cm set output "gnuplot_multi.pdf" "" for index in indexes:
multi_instructions += "f" + index + "(x) = a" + index + " + b" + index + "*x + c" + index + "*x**2\n"
multi_instructions += "set fit errorvariables\n" multi_instructions += "fit f" + index + "(x)
'gnutemp.dat' index " + index + " using %x:%{pow} via a" + index + ", b" + index + ", c" + index
+ "\n" multi_instructions += "print sprintf(\"REGLIN_COEFFS = {'a" + index + "':%f, %f}, 'b" +
index + "':%f, %f}, 'c" + index + "':%f, %f})\", a" + index + ", a" + index + "_err, b" + index + ", b"
+ index + "_err, c" + index + ", c" + index + "_err)\n" multi_instructions += "" set key top left set
xlabel "x" set ylabel "pow" plot 'gnutemp.dat' index 0 using %x:%{pow}:%{Deltax}:%{Deltapow}
with xyerrorbars title "Mesures 1", f0(x) title "Régression", 'gnutemp.dat' index 1 using
%x:%{pow}:%{Deltax}:%{Deltapow} with xyerrorbars title "Mesures 2", f1(x) title "Régression"
"" print gnuplotize(tabs, multi_instructions, temp_filename = "gnutemp.dat", verbose = True)

```