G. Cowan

SDA Discussion Notes Week 6

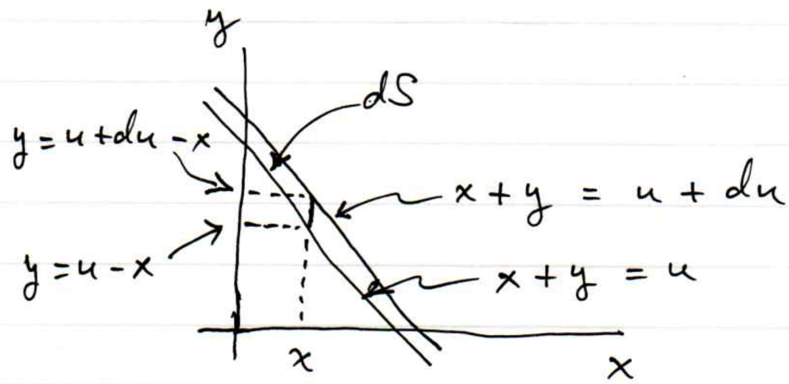
Problem Sheet 3 solutions

1) $x, y \sim f(x,y)$

Define $u = x + y$

From week 2 slides p. 9,

$$g(u)\,du = \int_{dS} f(x,y)\,dx\,dy$$



f approx. const in infinitesimal interval.

$$\Rightarrow g(u)\,du = \int_{-\infty}^{\infty} \int_{u-x}^{u+du-x} f(x,y)\,dy\,dx$$

$$g(u)\,\cancel{du} = \int_{-\infty}^{\infty} f(x, u-x)\,dx\,\cancel{du}$$

QED

Or carry out integration in opposite order,

$$\Rightarrow g(u) = \int_{-\infty}^{\infty} f(u-y, y)\,dy$$

2) $x \sim \frac{1}{\xi} e^{-x/\xi}$ $\quad x \geq 0$

$y \sim \frac{1}{\xi} e^{-y/\xi}$ $\quad y \geq 0$

x, y indep. $\Rightarrow$ $f(x,y) = f(x) \cdot f(y)$

$$f(x,y) = \frac{1}{\xi^2} e^{-(x+y)/\xi^2}, \quad x, y \geq 0$$

Let $u = x + y$

From Q1, $g(u) = \int_{-\infty}^{\infty} f(x, u-x)\, dx$

Integrand nonzero for $x \geq 0$ and $y = u - x \geq 0$

$\Rightarrow u \geq x$

$$\Rightarrow g(u) = \int_0^u \frac{1}{\xi^2} e^{-(x+u-x)/\xi} dx$$

$$= \frac{1}{\xi^2} e^{-u/\xi} \int_0^u dx$$

$$= \frac{u}{\xi^2} e^{-u/\xi}, \quad u \geq 0$$

(Special case of gamma dist.)

3) $x \sim f(x)$

$\rightarrow$ cumul. dist $F(x) = \int_{-\infty}^{x} f(x')\,dx'$

$\Rightarrow f(x) = \frac{dF}{dx}$

$r \sim \text{Uniform}[0,1]$

i.e. $g(r) = 1 \quad , \quad 0 \leq r \leq 1$

If $F(x) = r \Rightarrow x = F^{-1}(r)$

pdf of $x(r)$ is

$$p(x) = g(r)\left|\frac{dr}{dx}\right| = \frac{g(r)}{\left|\frac{dx}{dr}\right|}$$

inverse function theorem

$$\frac{dx}{dr} = \frac{d}{dr}F^{-1}(r) = \frac{1}{\frac{dF}{dx}(x(r))} = \frac{1}{f(x(r))}$$

$$\Rightarrow \frac{dr}{dx} = f(x)$$

$$\Rightarrow p(x) = \underbrace{1}_{g(r)} \times \underbrace{f(x)}_{\left|\frac{dr}{dx}\right|}$$

$$= f(x)$$

In [1]:

```
# simpleMC.py -- simple Monte Carlo program to make histogram of uniformly
# distributed random values and plot
# G. Cowan, RHUL Physics, October 2019

import matplotlib
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:

```
# generate data and store in numpy array, put into histogram

numVal = 10000
nBins = 100
xMin = 0.
xMax = 1.
xData = np.random.uniform(xMin, xMax, numVal)
xHist, bin_edges = np.histogram(xData, bins=nBins, range=(xMin, xMax))
```

In [3]:

```
# make plot and save in file

binLo, binHi = bin_edges[:-1], bin_edges[1:]
xPlot = np.array([binLo, binHi]).T.flatten()
yPlot = np.array([xHist, xHist]).T.flatten()
fig, ax = plt.subplots(1,1)
plt.gcf().subplots_adjust(bottom=0.15)
plt.gcf().subplots_adjust(left=0.15)
ax.set_xlim((xMin, xMax))
ax.set_ylim((0., 150))
plt.xlabel(r'$x$', labelpad=0)
plt.plot(xPlot, yPlot)
plt.show()
plt.savefig("uniformHist.png", format='png')
```