cmpe320\proj3\proj3.py

```
#
#
#   File Name:    proj3.py
#   Date Created: Fri Apr 14 2023
#   Author:       Cole Cavanagh
#   Description:  CMPE 320 Project 3: The Central Limit Theorem
#
#

import matplotlib.pyplot as plt
import numpy as np
import scipy as sp

# PART 2.1: IID Random Variables U(0,1)

# Trial Variables
m = 100000
sum_size = [2, 6, 12]

# Subplot Setup
fig, axis = plt.subplots(3, 1)
plt.suptitle("Section 2.1: IID Random Variables")
plt.subplots_adjust(hspace = .5)

# Data and Plot Loop
for sum in sum_size:
    # Create sum data
    xd = np.random.rand(sum, m)
    xs = np.sum(xd, 0)
    xmin = np.amin(xs)
    xmax = np.amax(xs)

    # Create Gaussian PDF
    xr = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum /
(sum_size.index(sum) + 1)), step = .002)
    mean = sum / 2
    variance = sum / 12
    yr = np.exp(-(xr - mean) ** 2 / (2 * variance)) / np.sqrt(2 * np.pi * variance)

    # Print Values
    print("Section 2.1 Trial %s: SM: %s AM: %s SV: %s AV: %s" % ((sum_size.index(sum) + 1), mean,
np.mean(xs), variance, np.var(xs)))

    # Plot
    bin_edges = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum /
(sum_size.index(sum) + 1)), step = .1)
    axis[sum_size.index(sum)].hist(xs, bins = bin_edges, density = True, edgecolor = "black",
label = "Histogram")
    axis[sum_size.index(sum)].plot(xr, yr, 'r', label = "PDF")
    axis[sum_size.index(sum)].grid()
    axis[sum_size.index(sum)].legend()
    axis[sum_size.index(sum)].set_title("Trial %d: %d sums" % ((sum_size.index(sum) + 1), sum))
    axis[sum_size.index(sum)].set_xlabel("k")
    axis[sum_size.index(sum)].set_ylabel("Fk(k)")

plt.show()

# PART 2.2: IID Random Variables U(1,8)

# Trial Variables
m = 100000
sum_size = [2, 20, 40]

# Subplot Setup
fig, axis = plt.subplots(3, 1)
plt.suptitle("Section 2.2: IID Random Variables U(1,8)")
plt.subplots_adjust(hspace = .5)

# Data and Plot Loop
for sum in sum_size:
    # Create sum data
    xd = np.random.randint(low = 1, high = 9, size = (sum, m))
    xs = np.sum(xd, 0)
    xmin = np.amin(xs)
    xmax = np.amax(xs)

    # Create Gaussian PDF
    xr = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum / (sum_size.index(sum) + 1)), step = .1)
    mean = sum * 4.5
    variance = ((sum * 64) - 1) / 12
    yr = np.exp(-(xr - mean) ** 2 / (2 * variance)) / np.sqrt(2 * np.pi * variance)

    # Print Values
    print("Section 2.2 Trial %s: SM: %s AM: %s SV: %s AV: %s" % ((sum_size.index(sum) + 1), mean, np.mean(xs), variance, np.var(xs)))

    # Plot
    bin_edges = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum / (sum_size.index(sum) + 1)), step = 1)
    axis[sum_size.index(sum)].hist(xs, bins = bin_edges, density = True, edgecolor = "black", label = "Histogram")
    axis[sum_size.index(sum)].plot(xr, yr, 'r', label = "PDF")
    axis[sum_size.index(sum)].grid()
    axis[sum_size.index(sum)].legend()
    axis[sum_size.index(sum)].set_title("Trial %d: %d sums" % ((sum_size.index(sum) + 1), sum))
    axis[sum_size.index(sum)].set_xlabel("k")
    axis[sum_size.index(sum)].set_ylabel("Fk(k)")

plt.show()

# PART 2.3: IID Random Variables Fx(x)

# Trial Variables
m = 100000
sum_size = [5, 50, 150]

# Subplot Setup
fig, axis = plt.subplots(3, 1)
plt.suptitle("Section 2.3: IID Random Variables Fx(x)")
plt.subplots_adjust(hspace = .5)

# Data and Plot Loop
for sum in sum_size:
    # Create sum data
    xd = np.random.rand(sum, m)
    xd = (- np.log(1 - xd) / .5) + 2
    xs = np.sum(xd, 0)
    xmin = np.amin(xs)
    xmax = np.amax(xs)

    # Create Gaussian PDF
    xr = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum /
(sum_size.index(sum) + 1)), step = .002)
    mean = (sum * 2) / .5
    variance = sum * 4
    yr = np.exp(-(xr - mean) ** 2 / (2 * variance)) / np.sqrt(2 * np.pi * variance)

    # Print Values
    print("Section 2.3 Trial %s: SM: %s AM: %s SV: %s AV: %s" % ((sum_size.index(sum) + 1), mean,
np.mean(xs), variance, np.var(xs)))

    # Plot
    bin_edges = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum /
(sum_size.index(sum) + 1)), step = 1)
    axis[sum_size.index(sum)].hist(xs, bins = bin_edges, density = True, edgecolor = "black",
label = "Histogram")
    axis[sum_size.index(sum)].plot(xr, yr, 'r', label = "PDF")
    axis[sum_size.index(sum)].grid()
    axis[sum_size.index(sum)].legend()
    axis[sum_size.index(sum)].set_title("Trial %d: %d sums" % ((sum_size.index(sum) + 1), sum))
    axis[sum_size.index(sum)].set_xlabel("k")
    axis[sum_size.index(sum)].set_ylabel("Fk(k)")

plt.show()

# PART 2.4: Bernoulli Random Variables

# Trial Variables
m = 100000
sum_size = [4, 8, 150]

# Subplot Setup
fig, axis = plt.subplots(3, 2)
plt.suptitle("Section 2.1: IID Random Variables")
plt.subplots_adjust(hspace = .5)

# Data and Plot Loop
for sum in sum_size:
    # Create sum data
    xd = np.less(np.random.rand(sum, m), .6)
    xd = np.where(xd == True, 1, 0)
    xs = np.sum(xd, 0)
    xmin = np.amin(xs)
    xmax = np.amax(xs)

    # Create Gaussian PDF
    xr = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum /
(sum_size.index(sum) + 1)), step = .002)
    mean = sum * .6
    variance = sum * .6 * (1 - .6)
    yr = np.exp(-(xr - mean) ** 2 / (2 * variance)) / np.sqrt(2 * np.pi * variance)

    # Create PMF
    xm = np.arange(start = xmin - (sum / (sum_size.index(sum) + 1)), stop = xmax + (sum /
(sum_size.index(sum) + 1)), step = 1)
    ym = np.exp(-(xm - mean) ** 2 / (2 * variance)) / np.sqrt(2 * np.pi * variance)

    # Print values
    print("Section 2.4 Trial %s: SM: %s AM: %s SV: %s AV: %s" % ((sum_size.index(sum) + 1), mean,
np.mean(xs), variance, np.var(xs)))

    # Plot
    bin_edges = np.arange(start = xmin - ((sum / (sum_size.index(sum) + 1)) - .5), stop = xmax +
(sum / (sum_size.index(sum) + 1)), step = 1)
    # Hist and PDF
    axis[sum_size.index(sum), 0].hist(xs, bins = bin_edges, density = True, edgecolor = "black",
label = "Histogram")
    axis[sum_size.index(sum), 0].plot(xr, yr, 'r', label = "PDF")
    axis[sum_size.index(sum), 0].grid()
    axis[sum_size.index(sum), 0].legend()
    axis[sum_size.index(sum), 0].set_title("Trial %d: %d sums" % ((sum_size.index(sum) + 1),
sum))
    axis[sum_size.index(sum), 0].set_xlabel("k")
    axis[sum_size.index(sum), 0].set_ylabel("Fk(k)")

    # Hist and PMF
    axis[sum_size.index(sum), 1].hist(xs, bins = bin_edges, density = True, edgecolor = "black",
label = "Histogram")
    axis[sum_size.index(sum), 1].stem(xm, ym, 'r', label = "PMF")
    axis[sum_size.index(sum), 1].grid()
    axis[sum_size.index(sum), 1].legend()
    axis[sum_size.index(sum), 1].set_title("Trial %d: %d sums" % ((sum_size.index(sum) + 1),
sum))
    axis[sum_size.index(sum), 1].set_xlabel("k")
    axis[sum_size.index(sum), 1].set_ylabel("Fk(k)")

plt.show()
```