yohandi - assignment 10

1a.
$$M_{X_2}(k) : E(e^{tX_2})$$

= $E(e^{t(Y-X_1)})$

= $E(e^{tY}) E(e^{-tX_1})$

= $(1-2k)^{-r/2} (1-2k)^{r/2}$

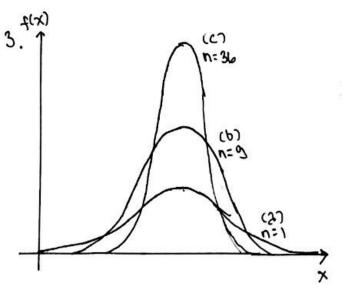
= $(1-2k)^{r/2}$

b.
$$\chi_2 \sim \chi^2 (r-r_1)$$

2. $M_Y(k) : E(e^{k\chi^2})$
: $\int_{0}^{\infty} e^{\chi^2} \frac{1}{\sqrt{2\pi}} e^{-\frac{\chi^2}{2}} d\chi$

let
$$W: \chi\sqrt{1-2k}$$

 dw
 $d\chi = \sqrt{1-2k}$
 \Rightarrow |My(t) = $\int_{-\infty}^{\infty} e^{-\frac{w^2}{2}} \frac{1}{\sqrt{2\pi}} \frac{dw}{\sqrt{1-2k}}$
 $=\frac{1}{\sqrt{1-2k}}$
this implies that $Y \sim \chi^2(1)$



2.
$$P(X < 6.0171) = \phi\left(\frac{6.0171 - 6.05}{\sqrt{0.0004}}\right) \approx 0.050$$

 $Y \sim B(9, 0.050)$
 $b. P(Y \le 2) = \sum_{i=0}^{2} {9 \choose i} (0.050)^{i} (1-0.050)^{9-i} \approx 0.992$

$$= \int_{-\infty}^{\infty} e^{\chi^{2}(t-\frac{1}{2})} \frac{1}{\sqrt{2\pi}} d\chi \qquad 62W = \frac{\frac{7}{2}}{\sqrt{\frac{7}{2}^{2}+\frac{7}{2}}} = \frac{\frac{7}{2}}{\sqrt{\frac{1}{2}}} \qquad \text{where} \qquad u \wedge \chi^{2}(2)$$

this implies that Witca)

$$f(v) = \frac{d}{dv}F(v)$$

$$= \frac{1}{\pi(2-v^2)^{3/2}} \left(\int_{0}^{\infty} \frac{1}{2^2 \cdot 2^2} \int_{0}^{\infty} \frac{1}{2^2 \cdot 2^2} \int_{0}^{\infty} \frac{1}{2^2} \int_{0$$

$$\frac{1}{2} \int_{0}^{\infty} \frac{1}{2} \int_$$

e. In succinct way, both (a)2(b) are mainly different on their dependencies.

Since 21,22, and 23 have independent

distribution,

Z1 is independent with (222+232)

however,

Zi might not be independent with 12122

yonardi -assignment 10 (computer-based)
$$\frac{S \left| -1 \right| 1}{5(5) \left| 0.5 \right| 0.5}$$

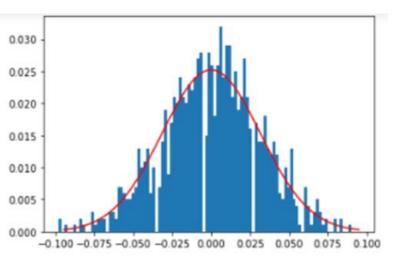
$$E(X) = E\left(\frac{1}{N}\sum_{i=1}^{N}S_{i}\right) = \frac{1}{N}\sum_{i=1}^{N}E(S_{i}) = 0$$

from the obtained Var(X),

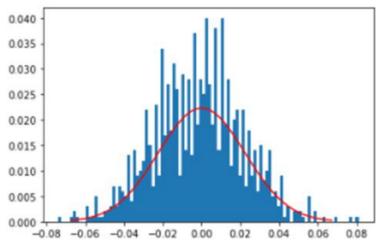
the bigger N, the smaller var(x)

=> the graph tends to be more compact

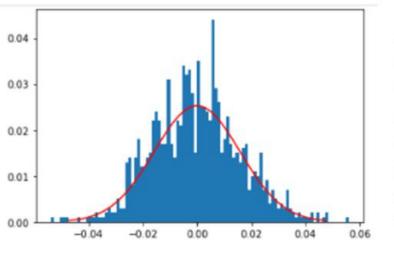
N=1000



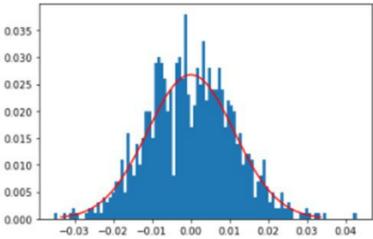
N=2000



N=4000



N=8000



```
import matplotlib.pvplot as plt
import numpy as no
import random
import scipy.stats
total experiment = 4
N list = [(1000 * 2 ** i) for i in range(total experiment)]
def random walk(N):
    return sum([(random.randint(0, 1) * 2 - 1) for i in range(N)]) / N
def experiment(ith, N, M = 1000, K = 100);
    ith list = [0.002, 0.00125, 0.001, 0.00075]
    sample X = np.arrav([random walk(N) for i in range(M)])
    ax = plt.figure().add subplot(111)
    n. bins. patches = ax.hist(sample X, bins = K, cumulative = 0, weights = np.zeros_like(sample X) + 1. / len(sample X))
    ax.legend
    mean = 0
    variance = 1 / N
    sigma = math.sart(variance)
    X = np.linspace(mean - 3 * sigma, mean + 3 * sigma, 100)
    Y = [i * ith list[ith] for i in scipv.stats.norm.pdf(X, mean, sigma)]
    plt.plot(X, Y, color = "r")
    plt.show()
for i in range(total experiment):
    experiment(i, N_list[i])
```

import math