```
% Problem 1
n = 457
n_red = 248
p_hat = n_red / n
P = 0.5
delta = 0

% calculate S
sigma = sqrt(((P*(1-P))/n))

% calculate Z stat
Z = (p_hat - P) / sigma

% one-sided p value
p_one_sided = 1-normcdf(Z)

% two-sided p value
p_two_sided = 2*(1-normcdf(Z))

% create plot 1
x = linspace(0.4, 0.6, 100)
mu = 0.5;
sigma = 0.0234;
y = normpdf(x,mu,sigma)
plot(x, y)
hold on
% vertical line
line([p_hat, p_hat], [0, 18])

% new p-hat
```

```
Z_2 = (0.57 - P) / sigma

% create plot 2
x = linspace(0.4, 0.6, 100)
mu = 0.5;
sigma = 0.0234;
y = normpdf(x,mu,sigma)
plot(x, y)
hold on
% vertical line
line([p_hat, p_hat], [0, 18])
hold on
line([P-(p_hat-P), P-(p_hat-P)], [0, 18])


% one-sided p value
p_one_sided_2 = 1-normcdf(Z_2)

% two-sided p value
p_two_sided_2 = 2*(1-normcdf(Z_2))

% new sample size
n = 272
n_red = 150
p_hat = n_red / n
P = 0.5
delta = 0

% calculate S
sigma = sqrt(((P*(1-P))/n))
```

```
% calculate Z stat
Z = (p_hat - P) / sigma

% one-sided p value
p_one_sided = 1-normcdf(Z)

% two-sided p value
p_two_sided = 2*(1-normcdf(Z))


% problem 2
std_error_1 = 1 / sqrt(25)
z_1 = (3.25-3.5) / std_error_1
p_1 = 1- normcdf(z_1)

std_error_2 = 1 / sqrt(100)
z_2 = (3.25-3.5) / std_error_2
p_2 = 1- normcdf(z_2)


% problem 3
xbar = 5.25
mu = 6
n = 6
sigma = 1.25
std_err = sigma / sqrt(n)
T = (xbar - mu) / std_err
df = n-1
tinv(0.05, n-1)
```

```
p = tcdf(T, df)


%problem 4
x1 = [24 25 28 28 28 29 29 31 31 35 35 35]'
x2 =  [21 22 24 27 27 28 29 32 32]'

xbar1 = mean(x1)
xbar2 = mean(x2)

[h,p,ci,stats] = ttest2(x1, x2, 'vartype',↙
'equal', 'Tail', 'right')

%problem 5
n1 = 55
n2 = 149
phat = (24+54)/(55+149)
phat1 = 24/55
phat2 = 54/149
Z_num = (phat1-phat2)
Z_denom = sqrt(phat*(1-phat)*((1/n1) + (1/n2)))
Z = Z_num / Z_denom

Z_crit = norminv(0.95)

p = 1-normcdf(Z)

%problem 6
```

```
ps = [0.1 0.2 0.5]
ns=2:1:50
i = 1
for p = ps
    l=[]
    for n = ns
        x=0:1:n
        binos = binopdf(x, n, p)
        norms = normpdf(x, n*p,sqrt(n*p*(1-p)))
        maxval = max(abs(binos-norms))
        l = [l maxval]
    end
    subplot(3, 1, i)
    plot(l)
    title(['p = ' num2str(p)])
    ylabel('maximum difference')
    if i == 3
        xlabel('n')
    end
    i = i+1
end
```