题目1

1. 实现PCA算法

：代码在附录

1) function [pcs, cprs_data, cprs_c] = pca_compress(data, rerr)

：压缩数据函数

2) function recon_data = pca_reconstruct(pcs, cprs_data, cprs_c)

：恢复（重构）原始数据，保存在新文件"recon_data.xls"

2. CPA（Component Principal Analysis）

```
It is normal regression (non ill-conditioned).
Linear-Regression equation:
y_ =

0.338*x4 - 1.55*x2 - 1.866*x3 - 1.522*x1 + 0.439*x5 - 1.371*x6 - 2.127*x7 + 0.923*x8 - 1.458*x9 - 0.032*x10 + 44.093

F=292.210755
Fa=1.833746
Linear relationship exists
CI :(y0-10.72, y0+10.72)>>
```

1) 判断是否回归分析：It is normal regression (non ill-conditioned).

2）实现线性回归

"Linear-Regression equation:

y_ =0.338*x4 - 1.55*x2 - 1.866*x3 - 1.522*x1 + 0.439*x5 - 1.371*x6 - 2.127*x7 + 0.923*x8 - 1.458*x9 - 0.032*x10 + 44.093

3）检验

F=292.210755

Fa=1.833746

"Linear relationship exists. "

4）置信区间

CI :(y0-10.72，y0+10.72)

附录：MATLAB 代码

【hw5】

```
% Read data X,Y & set significance level rerr = 0.05
X=xlsread('counties.xlsx','','C2:P3115');
Y=xlsread('counties.xlsx','','Q2:Q3115');
rerr = 0.05;

% PCA compress
[pcs, cprs_data, cprs_c] = pca_compress(X, rerr);

% PCA reconstruct
pca_reconstruct(pcs, cprs_data, cprs_c);

% excute function
linear_regression(Y,cprs_data',rerr);
```

【linear_regression】

```
function linear_regression(y,x,alpha)
% relevant data of x,y
[N,n]=size(x);
ex=mean(x);
ey=mean(y);
var_x=var(x);
var_y=var(y);

%regression
for i=1:n
    x(:,i)=(x(:,i)-ex(:,i))/sqrt(var_x(:,i));
end

y=(y-ey)/sqrt(var_y);

x=x';
% new relevants
Ye=mean(y);
Xe=mean(x);

[Q,lamda]=eig(x*x');
lamda=abs(sum(lamda));
lamda_max=max(lamda);

t=1;
maxr=find(lamda==lamda_max);
while (lamda(maxr(1))>=lamda_max/10)&&(t<=n)
    Qm(:,t)=Q(:,maxr(1));
    t=t+1;
    Q(:,maxr(1))=[];
    lamda(maxr(1))=[];
    maxr=find(lamda==max(lamda));
    if t==n+1
        maxr=1;
        lamda=0;
    end
end

if t==n+1
```

```
    fprintf("It is normal regression (non ill-conditioned).\n");
    c_=inv(x*x')*x*y;
    c0_=Ye-mean(x')*c_;
    for i=1:n
      c(i)=sqrt(var_y/var_x(i))*c_(i);
    end
    c0=ey-ex*c';
else
   fprintf("It is ill-conditioned regression.\n");
    z=Qm'*x;
    lumda=inv(z*z');
    d=lumda*Qm'*x*y;
    c_=Qm*d;
    c0_=Ye-mean(x')*c_;

    for i=1:n
        c(i)=sqrt(var_y/var_x(i))*c_(i);
    end
    c0=ey-c*ex';
end

for i=1:n
    x_(i)=sym(['x',num2str(i)]);
end

y_=0;

for i=1:n
    y_=y_+round(c(i),3)*x_(i);
end

fprintf('Linear-Regression equation:')
y_=vpa(y_+ round(c0,3))

% TSS = ESS + RSS
TSS=0;
for i=1:N
    TSS=TSS+(y(i)-Ye)^2;
end
ESS=sum((x'*c_+c0_-mean(y)).^2);
RSS=TSS-ESS;

% F-test, df=(n, N-n-1)
F=(N-n-1)*ESS/(n*RSS);
Fa=finv(1-alpha,n,N-n-1);
fprintf('F=%f\nFa=%f\nLinear relationship exists\n',F,Fa);
% if F > Fa
% null hypothesis is not true, there be linear relationship
if F>Fa
    S=sqrt(var_y)*sqrt(RSS/(N-n-1));
    Z=norminv(1-alpha/2,0,1);
    fprintf('CI :(y0-%4.2f£¬y0+%4.2f)',S*Z, S*Z);
else
    fprintf("can't do linear regression");
end
end
```

【pca_compress】

```
function[pcs,cprs_data,cprs_c]=pca_compress(data,rerr)
x=data;
[N,n]=size(x); % the number of row & column
ex=mean(x); % avearage of x
varx=var(x); % variance of x

%normalization
for i=1:n
    x(:,i)=(x(:,i)-ex(:,i))/sqrt(varx(:,i));
end

%eigenvalue
x=x';
[Q,l]=eig(x*x');
r=abs(sum(l));

%getting maximum eigen value, optimal one
t=1;
k=0;
ls=sum(r);
maxr=find(r==max(r));
while (k/ls<1-rerr)&&(t<=n)
    k=k+r(maxr);
    Qm(:,t)=Q(:,maxr);
    t=t+1;
    Q(:,maxr)=[];
    r(maxr)=[];
    maxr=find(r==max(r));
end

if t==(n+1)
    sprinrf("There isn't linear correlation in the matrix, it couldn't be
compressed.")
else
    pcs=Qm;
    cprs_data=Qm'*x;
    cprs_c=[ex' varx'];
end
end
```

【pca_reconstruct】

```
function recon_data=pca_reconstruct(pcs,cprs_data,cprs_c)
[n,~]=size(pcs);
% reconstruct standard variables
recon_data=pcs*cprs_data;
% nomarlizing recon_data
for i=1:n
    recon_data(i,:)=recon_data(i,:).*sqrt(cprs_c(i,2))+cprs_c(i,1);
end
% rewrite datas in new document
xlswrite('recon_data.xls',recon_data');
end
```