1 第一问：检查复共线性

源代码：

```
X=as.matrix(longley[,1:6])
y=as.matrix(longley[,7])
oldX=X
oldY=Y
xmean=colMeans(X)
xsd=apply(X,2,sd)*sqrt(15)
ymean=mean(y)
ysd=sd(y)*sqrt(15)
centered=X-matrix(rep(colMeans(X),16),byrow=T,nrow=16,ncol=6)
X=centered/matrix(rep(apply(X,2,sd)*sqrt(15),16),byrow=T,nrow=16,ncol=6)
y=(y-mean(y))/matrix(rep(sd(y)*sqrt(15),16),byrow=T,nrow=16,ncol=1)
fm1 <- lm(Employed ¬ ., data = as.data.frame(scale(longley)))
```

统计分析：

通过阅读题中所给的数据集说明，可以知道本问题中将”Employed” 作为因变量 y，将其他特征作为自变量 $X \triangleq (x_1, \cdots, x_6)$。再对 X,y 分别进行中心标准化，以消除量纲对数据分析的影响。记在中心标准化的过程中，用到的各分量的均值和标准差如下：y 的均值与标准差为 $\bar{y}$，s_y，各自变量的均值与标准差为 $\bar{x}_i$，s_i，$i = 1, \cdots, 6$

1.1 使用条件数检查复共线性

源代码：

```
kappa(t(X)%*%X, exact=T)
```

out: 12220.0098602771

统计分析：

上面用 kappa 函数求出的是矩阵 X’X 的条件数。由复共线性的条件数判别准则，当 X’X 的条件数小于 100 时，复共线性很小；在 100 到 1000 之间时，复共线性较强；大于 1000 时，回归自变量间存在严重的复共线性。这里条件数约为 12220 大于 1000，则可以判定 longley 数据集的自变量间存在严重的复共线性关系。

设 $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_p$ 为 X’X 的所有特征值，则 X’X 的条件数的定义为：

$$k = \frac{\lambda_1}{\lambda_p}$$

1.2 使用方差扩大因子法检查复共线性

源代码：

```
library(car)
vif(fm1)
```

out: GNP.deflator: 135.53243827999 GNP: 1788.513482718 Unemployed: 33.6188905960484 Armed.Forces: 3.58893019344552 Population: 399.151022312601 Year: 758.980597406813

统计分析：

使用 vif 函数求出了各自变量的方差扩大因子。当一个自变量的 vif 值大于 10 时，说明它与其余自变量之间由严重的复共线性。上面 6 个自变量中有 5 个自变量的 vif 值大于 10 甚至远大于 10，说明 longley 数据集的自变量之间存在复共线性。

其中，变量 x_i 的方差扩大因子 vif_i 的定义为：

$$vif_i = (1 - \frac{(\sum(y-\bar{y})(\hat{y}-\bar{y}))^2}{\sum(y-\bar{y})^2(\hat{y}-\bar{y})^2})$$

1.3 对相关系数阵绘图直观展示复共线性

源代码：

```
library(corrplot)
corrplot.mixed(cor(X))
```

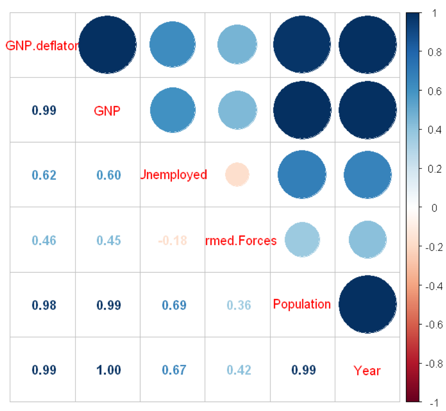


图 1: 相关系数图

统计分析：

此图中圆圈越大颜色越深说明相关系数越接近 1，变量间的相关性越强。可以看出图中有较多的圆圈是大且色深的，说明变量之间存在复共线性。

2　第二问：主成分回归

2.1　将回归方程化为典则形式

源代码：

```
phi=eigen(t(X)%*%X)$vectors
gam=diag(eigen(t(X)%*%X)$values)
Z=X%*%phi
```

统计分析：

记线性回归模型为：

$$y = \alpha_0 1_n + X\beta + e$$
$$e \sim (0, \sigma^2 I_n)$$

则其典则形式为：

$$y = \alpha_0 1_n + Z\alpha + e$$
$$e \sim (0, \sigma^2 I_n)$$

其中 $Z \triangleq X\phi$, $\alpha \triangleq \phi'\beta$, $X'X = \phi\Gamma\phi'$，$\Gamma = diag(\lambda_1, \cdots, \lambda_p)$ 为 $X'X$ 的特征值构成的对角阵，$\phi = (\phi_1, \cdots, \phi_p)$ 为对应的标准正交的特征向量组成的列向量组。

这里使用 eigen 函数求出了特征值矩阵 gam 和特征向量矩阵 phi，并由此求出了 Z，相当于求出了回归方程的典则形式。

2.2　主成分个数选择

源代码：

```
(gam[1,1]/sum(diag(gam)))
((gam[1,1]+gam[2,2])/sum(diag(gam)))
((gam[1,1]+gam[2,2]+gam[3,3])/sum(diag(gam)))
```

out: 0.767229515961399　0.963119599170923　0.997023827904495

```
PCA=princomp(X)
summary(PCA, loadings=T)
screeplot(PCA, type="lines")
Z1=X%*%phi[,1:3]
```

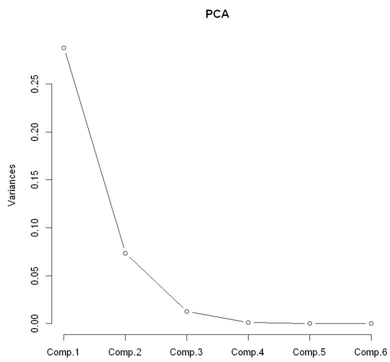


图 2: 碎石图

统计分析：

预先给定：当方差累计贡献率超过 97% 时，停止选入主成分。这里计算了前两个（方差前二大的）主成分的方差累积贡献率，在选入第三个主成分时就已经超过了 97%，于是选取三个主成分。从碎石图中也看出可以选取两个主成分。

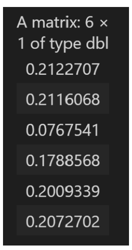
记：$\phi_0 = (\phi_1, \phi_2, \phi_3)$，$Z_1 = X\phi_0$，$\alpha_1 = \phi_0'\beta$

2.2.1 讨论：如果选取两个主成分

源代码：

```
alph1=solve(t(Z1)%*%Z1)%*%t(Z1)%*%y
(beta1=phi[,1:2]%*%alph1)
```

out:



A matrix: 6 × 1 of type dbl
0.2122707
0.2116068
0.0767541
0.1788568
0.2009339
0.2072702

图 3: 不合理的主成分回归结果

统计分析：

从上面的主成分回归结果可以看出，$\hat{\beta}$ 的分量的符号全为正，不符合变量的实际意义。如第三个分量"Unemployed" 和第四个分量"Armed.Forces" 越大，因变量"Employed" 应该越小，故它们的系数的符号应该为负值，所以这个估计不符合实际意义。

2.3 求主成分的最小二乘估计并代回原变量

源代码：

```
(alph0=mean(y))
alph1=solve(t(Z1)%*%Z1)%*%t(Z1)%*%y
beta1=phi[,1:3]%*%alph1
```

out: -4.92227786308419e-16

```
inter=ymean-ysd*t(beta2)%*%(as.vector(xmean/xsd))
```

out: -358.7128

```
betaz=ysd*(phi[,1:3]%*%alph2)/as.vector(xsd)
```

out: 0.094787894,0.012674214,-0.011614913,-0.005987296,0.153862145,0.202957527

```
oldX%*%betaz+rep(inter,16)-oldY
```

out:

A matrix: 16 × 1 of type dbl

1947	-0.17234061
1948	0.34533289
1949	-0.04291875
1950	0.14566152
1951	0.27508290
1952	0.55004065
1953	0.10213269
1954	-0.09371295
1955	-0.45603563
1956	-1.07693130
1957	-0.36063457
1958	0.04949157
1959	0.04162528
1960	-0.06684996
1961	0.03981460
1962	0.72024169

图 4: 主成分回归残差

统计分析：

将剩余的主成分对 y 做最小二乘回归：

$$\hat{\alpha_0} = \bar{y}$$

$$\hat{\alpha_1} = (Z_1'Z_1)Z_1'y$$

再返回到原来的自变量，得到 β 的最小二乘回归：

$$\hat{\beta} = \phi_0\hat{\alpha_1} \triangleq (\hat{\beta}_1, \cdots, \hat{\beta}_6)$$

由于做过中心标准化，下面将方程还原至中心标准化之前的变量，仍用 $x_1, \cdots, x_6$ 表示。

$$\frac{\hat{y}-\bar{y}}{s_y} = \sum_{i=1}^{6}\hat{\beta}_i\frac{x_i-\bar{x}_i}{s_i}$$

即为：

$$\hat{y} = \bar{y} - s_y\sum_{i=1}^{6}\hat{\beta}_i\frac{\bar{x}_i}{s_i} + s_y\sum_{i=1}^{6}\frac{\hat{\beta}_i}{s_i}x_i$$

代入数据计算得到如下的主成分回归方程：

$$\hat{y} = -358.7128 + 0.0948x_1 + 0.0127x_2 - 0.0116x_3 - 0.006x_4 + 0.1539x_5 + 0.2030x_6$$

计算残差向量如图 4 所示，可见残差向量各分量均较小，拟合效果较好。

3 第三问：岭回归

源代码：

```
betahat <- function(k){
      return (solve(t(X)%*%X+k*diag(6))%*%t(X)%*%y)
}
```

统计分析：

这里是定义了岭回归估计量 $\hat{\beta}(k) = (X'X + kI_p)^{-1}X'y \triangleq (\hat{\beta}_1(k), \cdots, \hat{\beta}_6(k))$。其中，k 被称为岭参数。下面用三种方法来估计岭参数：

3.1 岭迹法

源代码：

```
dd<- ...
    as.data.frame(t(rbind(seq(0,0.03,0.00001),sapply(seq(0,0.03,0.00001),betahat))))
write.csv(dd,"dd.csv")
```

```
df=pd.read_csv("dd.csv")
df.iloc[:,1:].plot(x="V1",legend=False)
import matplotlib.pyplot as plt
plt.savefig("lingji.png")
```

out:

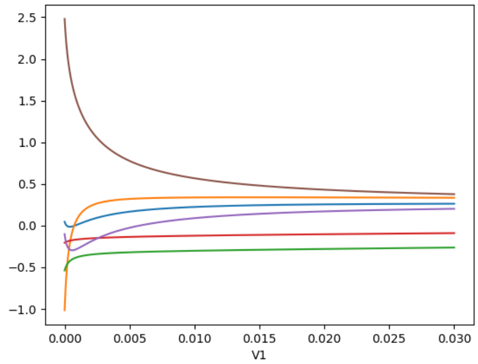


图 5: 岭迹图

```
(inter1=ymean-ysd*t(betahat(0.02))%*%(as.vector(xmean/xsd)))
```

out: -575.2279

```
(betal1=ysd*(betahat(0.02))/as.vector(xsd))
```

out: 0.0831301699985482, 0.0119777673450668,-0.0105029944945249, -0.00518425678976126,0.0865205092723138,0.318236949235663

```
oldX%*%betal1+rep(inter1,16)-oldY
```

out:

A matrix: 16 × 1 of type dbl

1947	-0.226034712
1948	0.242128635
1949	0.060508476
1950	0.218804834
1951	0.254467931
1952	0.550710788
1953	0.061286102
1954	0.068981728
1955	-0.393896107
1956	-1.069366807
1957	-0.405802089
1958	0.170139122
1959	0.031297170
1960	-0.119157072
1961	0.009129163
1962	0.546802838

图 6: 岭估计 (k=0.020) 的残差

统计分析：

岭迹法中岭参数的选取标准为：

- 使各个回归系数的岭估计大体上稳定
- 各个回归系数的岭估计值的符号比较合理
- 残差平方和不要上升太多

于是岭参数 k 不宜太大或太小，故我们选取了 k=0.020。

由于做过中心标准化，下面将方程还原至中心标准化之前的变量，仍用 $x_1, \cdots, x_6$ 表示。

$$\frac{\hat{y}-\bar{y}}{s_y} = \sum_{i=1}^{6} \hat{\beta}_i(0.02)\frac{x_i-\bar{x}_i}{s_i}$$

即为：

$$\hat{y} = \bar{y} - s_y\sum_{i=1}^{6} \hat{\beta}_i(0.02)\frac{\bar{x}_i}{s_i} + s_y\sum_{i=1}^{6} \frac{\hat{\beta}_i(0.02)}{s_i}x_i$$

代入数据计算得到如下的岭回归方程：

$$\hat{y} = -575.2279 + 0.0831x_1 + 0.0120x_2 - 0.0105x_3 - -0.0052x_4 + 0.0865x_5 + 0.3182x_6$$

计算残差向量如图 6 所示，可见残差向量各分量均较小，拟合效果较好。

3.2 方差扩大因子法

源代码：

```
chat <- function(k){
    return ...
        (max(diag(solve(t(Z)%*%Z+diag(k,6))%*%t(Z)%*%(Z)%*%solve(t(Z)%*%Z+diag(k,6)))))
}
plot(seq(0.01,0.2,0.001),lapply(seq(0.01,0.2,0.001),chat))
abline(h=10,v=0.024)
```

out:

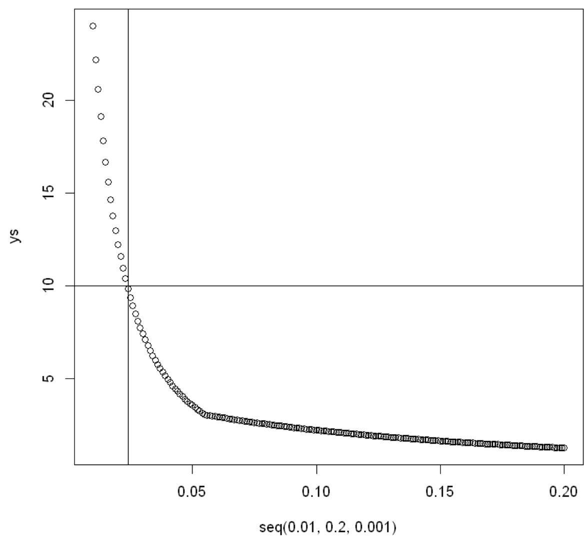


图 7: 方差扩大因子图

```
(inter2=ymean-ysd*t(betahat(0.024))%*%(as.vector(xmean/xsd)))
```

out: -538.6605

```
(betal2=ysd*(betahat(0.024))/as.vector(xsd))
```

out: 0.0845302428494129,0.0119224614765523,-0.0102338477540399,
-0.00490294566821215,0.0942791278588035,0.298918108195488

```
oldX%*%betal2+rep(inter2,16)-oldY
```

out:

A matrix: 16 × 1 of type dbl	
1947	-0.22610922
1948	0.23238637
1949	0.08098000
1950	0.22132618
1951	0.26055286
1952	0.55784188
1953	0.06050241
1954	0.09956186
1955	-0.40042113
1956	-1.08878812
1957	-0.42584304
1958	0.18815495
1959	0.01563704
1960	-0.13535664
1961	0.01930399
1962	0.54027060

图 8: 岭估计 (k=0.024) 的残差

统计分析：

根据前面给出的方差扩大因子的定义，可以知道矩阵 $c(k)=(X'X+kI)^{-1}X'X(X'X+kI)^{-1}$ 的对角元 $c_{ii}(k)$ 为岭估计的方差扩大因子。原则为：选择使得所有 $c_{ii}(k)$ 均不超过 10 的 k 值。上面图像的 y 轴代表的是当前 k 下 $c_{ii}(k)$ 中的最大值，当 $c_{ii}(k)$ 中的最大值都小于 10 时，则所有 $c_{ii}(k)$ 均不超过 10。故选择 k=0.024。

由于做过中心标准化，下面将方程还原至中心标准化之前的变量，仍用 $x_1,\cdots,x_6$ 表示。

$$\frac{\hat{y}-\bar{y}}{s_y}=\sum_{i=1}^{6}\hat{\beta}_i(0.024)\frac{x_i-\bar{x}_i}{s_i}$$

即为：

$$\hat{y}=\bar{y}-s_y\sum_{i=1}^{6}\hat{\beta}_i(0.024)\frac{\bar{x}_i}{s_i}+s_y\sum_{i=1}^{6}\frac{\hat{\beta}_i(0.024)}{s_i}x_i$$

代入数据计算得到如下的岭回归方程：

$$\hat{y}=-538.6605+0.0845x_1+0.0119x_2-0.0102x_3-0.0049x_4+0.0943x_5+0.2989x_6$$

计算残差向量如图 8 所示，可见残差向量各分量均较小，拟合效果较好。