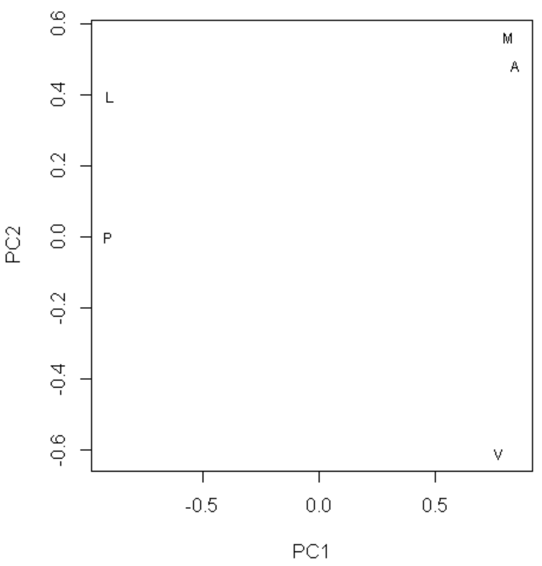
```
#  пакет psych

Синий – текст программы
Черный – вывод на консоль

# Обычно факторы, полученные методом главных компонент, не
# поддаются достаточно наглядной интерпретации.
# Поэтому следующим шагом факторного анализа служит
# преобразование (вращение) факторов таким образом, чтобы
# облегчить их интерпретацию.

#
# PCA  + principal
#
#  пакет psych
# используем  функцию principal – метод главных компонент
# с возможностью вращения
#
# principal(r, nfactors = 1, residuals = FALSE,
# rotate="varimax",n.obs=NA,
# covar=FALSE,scores=TRUE,missing=FALSE,impute="median",
# oblique.scores=TRUE,method="regression",...)

# Описание аргументов principal() в файле psych.doc

# загрузите пакет psych, если он не еще не установлен на ваш компьютер
install.packages("psych")

# загрузите библиотеку, если пакет уже установлен на ваш компьютер
library("psych")

# ввод данных
d<- read.table("pca2.csv",sep=";",dec=",")
data<-data.frame(d)
data
    V1   V2   V3    V4   V5
1 68.9 1060  7.8   5.5 25.3
2 68.1 1101  9.5  15.3 28.0
3 67.6 1147 10.1  30.2 30.0
4 69.2 1204 10.0  44.5 23.5
5 69.2 1602  9.8  58.6 18.0
6 64.6 1893  5.5  93.3 38.4
7 67.0 2777  6.2 122.0 29.6

# namesrow - имена объектов
namesrow<-c(1970,1975,1980,1985,1990,1995,1998); namesrow
[1] 1970 1975 1980 1985 1990 1995 1998

# Признаки: L – средняя продолжительность жизни; M – количество чиновников;
# A – количество автомобилей; P – доходы бедных; V – объемы продажи водки.
namescolumn=c("L","M","P","A","V")

numc <- ncol(data) # количество признаков (Переменных, столбцов)
[1] 5

numr <- nrow(data)  # количество объектов (строк)
[1] 7

# СТАНДАРТИЗАЦИЯ исходных данных вручную
# можно применить scale()
#
datas<-apply(data,2,function(x)(x-mean(x))/sd(x))
datas
             V1          V2         V3         V4          V5
[1,]  0.6706818 -0.76804053 -0.3191466 -1.1201407 -0.35403938
[2,]  0.1829132 -0.70251508  0.5640731 -0.8879205  0.07216089
[3,] -0.1219422 -0.62899871  0.8757977 -0.5348511  0.38786480
[4,]  0.8535951 -0.53790235  0.8238436 -0.1959992 -0.63817290
[5,]  0.8535951  0.09817403  0.7199354  0.1381134 -1.50635864
[6,] -1.9510744  0.56324495 -1.5140910  0.9603624  1.71382121
[7,] -0.4877686  1.97603770 -1.1504123  1.6404357  0.32472402
```

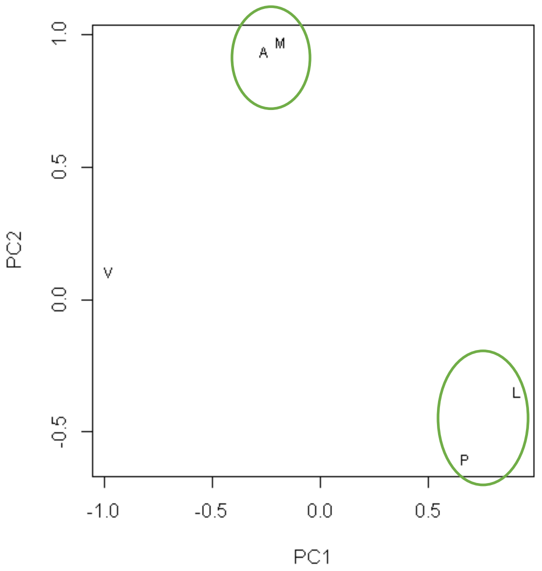
```
# Часть 1. PCA без вращения
# Количество факторов установим равным числу исходных
# переменных, так как сначала рассмотрим PCA без вращения
#
pc <- principal(datas,nfactors=numc); pc

PC1   PC2   PC3   PC4   PC5 h2      u2 com
V1 -0.90  0.40 -0.15  0.11  0.07  1 5.6e-16 1.5
V2  0.82  0.56  0.02  0.12 -0.07  1 1.2e-15 1.8
V3 -0.90  0.04  0.42  0.04 -0.02  1 2.0e-15 1.4
V4  0.85  0.49  0.19 -0.07  0.08  1 1.2e-15 1.7
V5  0.77 -0.61  0.10  0.12  0.05  1 2.0e-15 2.0

                       PC1  PC2  PC3  PC4  PC5
SS loadings           3.60 1.09 0.24 0.05 0.02
Proportion Var        0.72 0.22 0.05 0.01 0.00
Cumulative Var        0.72 0.94 0.99 1.00 1.00
Proportion Explained  0.72 0.22 0.05 0.01 0.00
Cumulative Proportion 0.72 0.94 0.99 1.00 1.00

Mean item complexity =  1.7
Test of the hypothesis that 5 components are sufficient.

The root mean square of the residuals (RMSR) is  0
 with the empirical chi square  0  with prob <  NA

Fit based upon off diagonal values = 1

# Результат работы principal – 29 выходных параметров.
# Полное описание вывода можно посмотреть в закладке «Environment».
# При необходимости значения параметров можно вывести на консоль,
# указав имя модели и имя параметра, например, pc0$n.obs,
# где n.obs – число объектов
pc$n.obs

[1] 7

# Основные параметры:
# PC1, PC2, ... - столбцы факторных нагрузок
# h2 - доля учтенной дисперсии, здесь 1,
# так как число факторов = числу переменных
# u2 - доля неучтенной дисперсии, здесь 0
#
# SS loadings    3.60, ...    - собственные значения
# Proportion Var  0.72    - первая компонента учитывает 72% дисперсии
# Cumulative Var        0.72 0.94 0.99 1.00 1.00
# Proportion Explained  - объясненная дисперсия
# Cumulative Proportion - накопленная объясненная дисперсия

# Матрица факторных нагрузок – первая часть массива pc$loading.
pc$loadings

Loadings:
   PC1    PC2    PC3    PC4    PC5   
V1 -0.896  0.398 -0.147  0.108       
V2  0.815  0.564         0.116       
V3 -0.905         0.421              
V4  0.847  0.486  0.188              
V5  0.772 -0.613         0.123       

                PC1   PC2   PC3   PC4   PC5
SS loadings    3.60 1.090 0.245 0.047 0.019
Proportion Var 0.72 0.218 0.049 0.009 0.004
Cumulative Var 0.72 0.938 0.987 0.996 1.000

# Матрица факторных нагрузок (обозначим ее Matf),
```

```
# имеет размерность  numc x numc, где numc - количество переменных
Matf<-pc$loadings[seq(numc),seq(numc)] ; Matf

          PC1          PC2         PC3         PC4         PC5
V1 -0.8961099  0.39790262 -0.14700564  0.10762138  0.07394265
V2  0.8150340  0.56360092  0.01613561  0.11612827 -0.06578346
V3 -0.9048438  0.04473558  0.42146450  0.03602025 -0.01807167
V4  0.8469578  0.48572104  0.18848317 -0.07130710  0.07827480
V5  0.7724239 -0.61325900  0.09947643  0.12270323  0.04819763

# для наглядности возьмем значения больше 0.1 по абсолютной величине
# и округлим результаты до трех знаков
Matf[-0.1<Matf & Matf<0.1 ]<- 0;
round(Matf,3)
      PC1    PC2    PC3   PC4 PC5
V1 -0.896  0.398 -0.147 0.108   0
V2  0.815  0.564  0.000 0.116   0
V3 -0.905  0.000  0.421 0.000   0
V4  0.847  0.486  0.188 0.000   0
V5  0.772 -0.613  0.000 0.123   0

# Матрица факторов (матрица главных компонентов) - она же матрица score
# Замечание: матрица уже нормирована (в отличии от функции princomp,
# которая вычисляет не нормированные значения, их приходится нормировать)
MatrS <- pc$scores; MatrS
            PC1        PC2         PC3        PC4         PC5
[1,] -0.6001053 -0.4653618 -2.00813484  0.1704057  0.06144998
[2,] -0.5397689 -0.7096035  0.16058903  0.6577804 -0.88730511
[3,] -0.3747882 -0.7903740  1.28475158  0.6689251 -0.35739385
[4,] -0.7243633  0.3389999  0.45986543 -0.1130010  2.01995880
[5,] -0.6619145  1.3009882  0.22746945 -1.4080954 -0.97696309
[6,]  1.5873946 -1.0194395  0.03784269 -1.2320003  0.20550992
[7,]  1.3135456  1.3447907 -0.16238334  1.2559855 -0.06525666

# Найдены:
# матрица нагрузок  Matrf;
# матрица факторов  MatrS
# Далее требуется:
# Выбрать количество главных компонент (факторов).
# Построить график - исходные признаки
# в пространстве главных компонет (факторов).
# Дать интерпретацию полученных результатов (см. pca2.r и pca2.doc).

------------------------------------------------
# Часть 2. Факторный анализ. PCA с вращением.
# Факторизация + вращение + интерпретация
# Факторизация: пусть первичная факторное решение найдено методом PCA.
# Число факторов возьмем равное 2,
# так 2 собственных числа >1 (3.60 1.09), объясняют 94% дисперсии.
# то есть только первые 2 фактора будут учитываться.
# Выполним вращение методом varimax
nf<-2; nf                 #  число факторов
[1] 2
pcV <- principal(datas,nfactors=2,rotate="varimax",covar=FALSE,scores=TRUE)
pcV$loadings
Loadings:
   RC1    RC2
V1  0.917 -0.347
V2 -0.183  0.974
V3  0.675 -0.605
V4 -0.261  0.941
V5 -0.980  0.107
                 RC1   RC2
SS loadings    2.359 2.331
Proportion Var 0.472 0.466
Cumulative Var 0.472 0.938

# Матрица факторных нагрузок
```

```
# размерность  numc x nf = 2x2,  где nf - количество факторов
Matrfv <- pcv$loadings[seq(numc),seq(nf)]; Matrfv

          RC1        RC2
V1  0.9168985 -0.3473279
V2 -0.1830640  0.9738655
V3  0.6747362 -0.6045450
V4 -0.2605299  0.9409498
V5 -0.9804206  0.1072414

#Теперь основные факторный нагрузки на 1-й фактор - V1="L", V3="P" и V5="V"
# на 2 фактор -- V2="M" и V4="A"
# признак V3="V" относится и к первому и ко 2-му факторам,
# при этом, не коррелирует с другими признаками
# такие "непонятные" переменные можно оставить в анализе, как отдельный
# 3-ий фактор.

# Матрица факторов (матрица главных компонентов) - матрица score
MatrSv <- pc$scores; MatrSv

             RC1        RC2
[1,]  0.09935460 -0.7528723
[2,] -0.11530756 -0.8840768
[3,] -0.28939965 -0.8254727
[4,]  0.75337526 -0.2684192
[5,]  1.38551549  0.4593996
[6,] -1.84545689  0.3916216
[7,]  0.01191875  1.8798197

# Найдены:
# матрица нагрузок  Matrfv
# матрица факторов  MatrfvS
# Далее требуется:
# Построить график - исходные признаки
# в пространстве главных компонет (факторов).
# Дать интерпретацию полученных результатов (см. pca2.r И pca2.doc).

# Интерпретация. Сравнение.
# Признаки: L - средняя продолжительность жизни; M - количество чиновников;
# A - количество автомобилей; P - доходы бедных; V - объемы продажи водки.

# Графики. Признаки в пространстве главных компонент (рис a),
# PCA без вращения
plot(Matrf[,seq(2)],type="n",xlab="PC1",ylab="PC2")
text(Matrf[,seq(2)],as.character(namescolumn),cex=0.75)

# Графики. Признаки в пространстве главных факторов (рис b),
# PCA с вращением
plot(Matrfv[,seq(2)],type="n",xlab="PC1",ylab="PC2")
text(Matrfv[,seq(2)],as.character(namescolumn),cex=0.75)
```

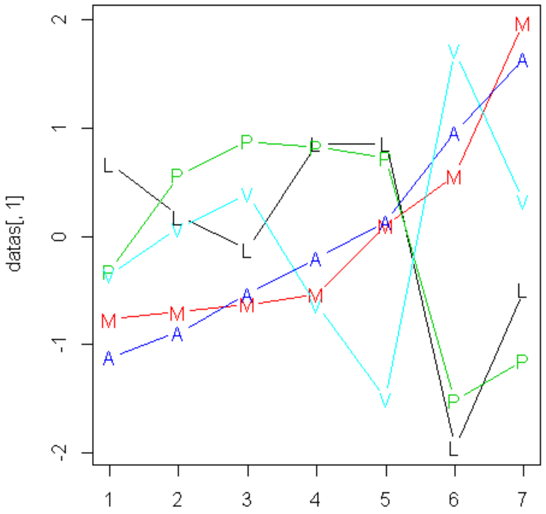


(a)

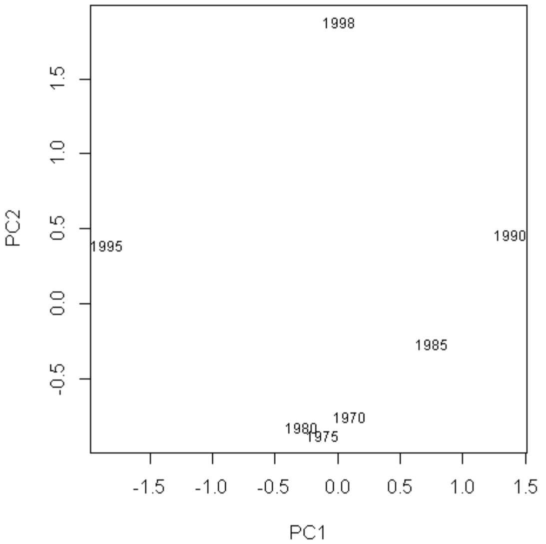


(b)

```
# Метод главных факторов (a) имеет высоконагруженный исходными показателями
# первый фактор, включающий в себя максимум разброса значений переменных,
# вычисляемых в проекции на первый фактор. Это затрудняет интерпретацию.
#
# Вращение варимакс является ортогональным вращением факторных осей с целью
# максимизировать дисперсию квадратов нагрузок фактора по всем переменным
# в факторной матрице. Это облегчает интерпретацию факторов (b).
#

# Наблюдается корреляция переменных A и M, а также   P и L. Эти две группы
# ортогональны, разные по знаку. Чем больше число автомобилей и чиновников,
# тем меньше продолжительность жизни и доходы бедных (и наоборот).
# Таким образом, 2-ой фактор можно считать уровнем благополучия.
# Заметим, что знаки «плюс-минус» в факторной таблице не несут смысловой нагрузки.
# Важно, одного они знака или нет.
# 1-ый фактор противопоставил два признака V и L.
# Однако признак V не коррелирует ни с обной переменной.
# Поэтому трактовать 1-ый фактор трудно.
# Такие факторы, можно рассматривать в данном случае, как третий фактор.
# Или убрать из рассмотрения. Возможно следует добавить еще новые факторы
# и повторить исследование.


# Посмотрим исходные нормирования значения признаков.
# По оси x - номер года, по оси y - значение признака.
plot(datas[,1],ylim=range(datas),type='n', col='black')
for (i in 1:numc) {lines(datas[,i],col=i,pch=namescolumn[i], type='b')}

# Более благополучные года 1970,1975,1980,
# 1995,1998 - не благополучные.
```




```
namesrow<-c(1970,1975,1980,1985,1990,1995,1998); namesrow

# Посмотрим объекты в пространстве двух факторов
plot(MatrSv[,seq(2)],xlab="PC1",ylab="PC2")
text(MatrSv[,seq(2)],as.character(namesrow),cex=0.75)
```




```
# 2-ой фактор, делит объекты на две части,
# чем меньше, тем более благополучный год 1970,1975,1980,
# чем больше, тем менее благополучный год.
# Самый неблагополучный 1998 – самое большое число чиновников
# и низкий доход у бедного населения.
# «Золотая» середина – 1990, все вмеру, объемы продажи алкоголя резко снизились.
```