

ClassAnaliz

A. Varykhin 24 01 2021

В данном документе представлена R-реализация некоторых алгоритмов Machine learning и Data-mining для решения задач классификации российских регионов по типу экономики.

Библиотеки используемых пакетов

```
library(openxlsx)
library(class)
library(gmodels)
library(lattice)
library(ade4)
library(cluster)
library(vegan)
library(pvclust)
library(tree)
library(randomForest)
library(e1071)
library(ggplot2)
library(ggthemes)
library(grDevices)
library(reshape2)
```

очистка области переменных

```
rm(list=ls(all=TRUE))
```

ЗАГРУЗКА ИЗ EXCEL ПЕРВИЧНЫХ ДАННЫХ

N=1 # номер листа с данными

```
x <- read.xlsx("D:/Machine learning/DATA/Region_Data4.xlsx", sheet = N)
```

ФОРМИРОВАНИЕ data.frame - ОСНОВНЫЕ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ РЕГИОНОВ РФ

```
xreg <- data.frame(P1=x$Pok_001, P2=x$Pok_002, P3=x$Pok_003, P4=x$Pok_004,
P5=x$Pok_005, P6=x$Pok_006, P7=x$Pok_007, P8=x$Pok_008, P9=x$Pok_009,
P10=x$Pok_010, P11=x$Pok_011, P12=x$Pok_012, P13=x$Pok_013, P14=x$Pok_014,
P15=x$Pok_015, P16=x$Pok_016, P17=x$Pok_017, P18=x$Pok_018)
```

```
#xreg
```

Формирование фактора тип экономики региона

```
(xreg_f <- as.factor(x$Pok_018))
```

```
## [1] A T P A T P P A P T A P T A P P P D D D P P P D P A T A T T D P T T A T
A T
## [39] T A P P A P D P P T P D A D A P A P D P T D D A D D D D P D T D D D T D
T D
## [77] D D D D P
## Levels: A D P T
```

ИСХОДНОЕ МНОЖЕСТВО ОСНОВНЫХ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ РЕГИОНОВ РФ 2018 г.

P1 - Площадь территории, тыс. км²
P2 - Численность населения на 1 января 2019 г., тыс. человек
P3 - Среднегодовая численность занятых, тыс. человек
P4 - Среднедушевые денежные доходы (в месяц), руб
P5 - Потребительские расходы в среднем на душу населения (в месяц), руб
P6 - Среднемесячная номинальная начисленная заработная плата работников организаций, руб.
P7 - Валовой региональный продукт в 2018 г. , млн руб
P8 - Инвестиции в основной капитал, млн руб
P9 - Основные фонды в экономике (по полной учетной стоимости; на конец года), млн руб.
P10 - Добыча полезных ископаемых , млн. руб
P11 - Обрабатывающие производства, млн руб
P12 - Обеспечение электрической энергией, газом и паром, млн руб
P13 - Водоснабжение; водоотведение, организация сбора и утилизации отходов, млн руб
P14 - Продукция сельского хозяйства , млн руб
P15 - Ввод в действие жилых домов, тыс. м²
P16 - Оборот розничной торговли, млн руб
P17 - Сальдированный финансовый результат деятельности организаций, млн руб.
P18 - Тип региональной экономики (A,D,P,T)
A - аграрный тип экономики
D - добывающий тип экономики
P - промышленный тип экономики
T - торговый тип экономики

```
#valreg <-xreg[,7] #валовой региональный продукт
#выбор исследуемого подмножества показателей регионов
w=c(10,11,14,16)
reg.ekon <-xreg[,w]
valreg <-rowSums(reg.ekon)
xreg_nor <-reg.ekon/valreg
```

```
### нормирующая функция-----
norm <-function(x){
  return ((x-min(x))/(max(x)-min(x)))
}
###-----
```

```
### нормирование исследуемых показателей
#xreg_nor <-as.data.frame(lapply(xreg_nor[1:4],norm))
#(summary(xreg_nor))
Y <-xreg_nor
```

Исходные данные для решения задач

print(Y)

##	P10	P11	P14	P16
## 1	0.1024607357	0.48925564	0.176916377	0.23136725
## 2	0.0004702848	0.39228216	0.152835387	0.45441217
## 3	0.0070610788	0.63264445	0.041831778	0.31846270
## 4	0.0062946660	0.36518432	0.178550653	0.44997036
## 5	0.0029042980	0.45653652	0.048061351	0.49249783
## 6	0.0036612406	0.77233396	0.040769188	0.18323561
## 7	0.0015789528	0.53479642	0.062721044	0.40090359
## 8	0.1412626871	0.30145091	0.227111091	0.33017531
## 9	0.0061860875	0.66402720	0.104654706	0.22513201
## 10	0.0025102570	0.51220351	0.021356685	0.46392954
## 11	0.0007240736	0.36139077	0.225483397	0.41240176
## 12	0.0029963642	0.54662759	0.102450806	0.34792524
## 13	0.0044219285	0.51692122	0.059618262	0.41903859
## 14	0.0004073696	0.33198793	0.259307072	0.40829763
## 15	0.0016961452	0.53906992	0.068067881	0.39116606
## 16	0.0057799303	0.66065459	0.061796314	0.27176917
## 17	0.0023675488	0.58979632	0.051511573	0.35632456
## 18	0.2517075032	0.36705018	0.013576118	0.36766620
## 19	0.5272218355	0.25586073	0.013373691	0.20354374
## 20	0.4244979655	0.27268070	0.012134587	0.29068675
## 21	0.0008776425	0.76185634	0.032174724	0.20509129
## 22	0.0230355529	0.72078980	0.043424792	0.21274986
## 23	0.0098563473	0.68108995	0.057009183	0.25204452
## 24	0.2124016359	0.40916960	0.004022608	0.37440615
## 25	0.0034276470	0.58540020	0.075600275	0.33557188
## 26	0.0076922322	0.40696974	0.145005841	0.44033219
## 27	0.0196143721	0.30366147	0.126594499	0.55012966
## 28	0.0000000000	0.01904875	0.543995445	0.43695580
## 29	0.0352299803	0.22928658	0.107467128	0.62801632
## 30	0.0185623909	0.36245810	0.135234114	0.48374539
## 31	0.5535642552	0.09910090	0.069922239	0.27741261
## 32	0.0433641213	0.60289176	0.087626538	0.26611758
## 33	0.0158102815	0.42851453	0.120562552	0.43511263
## 34	0.0073313229	0.06151103	0.167888779	0.76326887
## 35	0.0279392360	0.05450991	0.268340611	0.64921024
## 36	0.0011965844	0.14868402	0.237322573	0.61279682
## 37	0.0276980024	0.30584643	0.289773536	0.37668203
## 38	0.0044158525	0.12188256	0.152214814	0.72148677
## 39	0.0249903418	0.04920803	0.131673315	0.79412831
## 40	0.0103917312	0.29418510	0.192956361	0.50246680
## 41	0.1210905953	0.47968746	0.060630380	0.33859156
## 42	0.0023716270	0.54738816	0.150001942	0.30023827
## 43	0.0002507041	0.54868795	0.184725969	0.26633537
## 44	0.1832465456	0.51302879	0.060062687	0.24366198
## 45	0.2727184568	0.40645688	0.069463646	0.25136101
## 46	0.0025624089	0.50850581	0.096297229	0.39263455
## 47	0.1841707683	0.53912196	0.021027628	0.25567964
## 48	0.0029752176	0.49358627	0.087674284	0.41576423

```

## 49 0.0008186316 0.63257632 0.030708912 0.33589614
## 50 0.4151459868 0.26657341 0.082624839 0.23565577
## 51 0.0018704150 0.40397466 0.169664543 0.42449038
## 52 0.1670426813 0.49841540 0.040367941 0.29417397
## 53 0.0453231455 0.43000046 0.139309411 0.38536698
## 54 0.0162846347 0.52403595 0.077257147 0.38242226
## 55 0.0122672805 0.40167539 0.150432191 0.43562514
## 56 0.0234694047 0.60376330 0.026051904 0.34671539
## 57 0.6892816441 0.20286297 0.008220854 0.09963453
## 58 0.0366339551 0.67476950 0.054015521 0.23458102
## 59 0.0611765563 0.08104528 0.266281886 0.59149628
## 60 0.4861468985 0.00878109 0.104101684 0.40097033
## 61 0.2811066293 0.34792578 0.052480425 0.31848716
## 62 0.0061866052 0.39759068 0.162362975 0.43385973
## 63 0.2910036682 0.46942750 0.030307428 0.20926141
## 64 0.3948787320 0.33598888 0.041555341 0.22757704
## 65 0.5054732227 0.29954857 0.021596419 0.17338179
## 66 0.0751834806 0.44251036 0.066501734 0.41580443
## 67 0.0027496237 0.69173970 0.067695274 0.23781540
## 68 0.3658525723 0.31607420 0.051640443 0.26643279
## 69 0.0975016721 0.22863783 0.056512909 0.61734759
## 70 0.7338742772 0.03391619 0.023521003 0.20868853
## 71 0.3387092490 0.07432856 0.071566390 0.51539580
## 72 0.0977588340 0.60492083 0.037476188 0.25984414
## 73 0.0320091340 0.33473293 0.056708828 0.57654911
## 74 0.1297545702 0.38570873 0.024285486 0.46025121
## 75 0.1923883053 0.10900480 0.152317667 0.54628923
## 76 0.7596875562 0.02743551 0.016557379 0.19631956
## 77 0.8221584203 0.04875708 0.009013187 0.12007131
## 78 0.2206807563 0.14168004 0.123307776 0.51433143
## 79 0.8494578732 0.01327632 0.016787307 0.12047850
## 80 0.1388385453 0.49228899 0.000560935 0.36831153
## 81 0.0059311249 0.64552898 0.000000000 0.34853990

```

Анализ главных компонент (сокращение описания)

Многомерный анализ (Кендалл, Стьюарт, 1976) представляет собой часть статистики, которая выполняет обработку и интерпретацию результатов, полученных на основе наблюдений одновременно нескольких взаимосвязанных случайных переменных, каждая из которых представляется одинаково важной, по крайней мере, первоначально. В случае, если данные измерены в метрических шкалах и предполагаются линейные отношения между переменными, то в анализе применяются многомерные статистические методы, основанные на операциях матричной алгебры.

Во многих задачах обработки многомерных наблюдений исследователя интересует, в первую очередь, возможность снижения размерности, т.е. способ выделения небольшого набора исходных признаков или их линейных комбинаций, которые в наибольшей степени объясняют изменчивость наблюдаемых объектов. Это обусловлено следующими тремя причинами:

- возможностью наглядного представления (визуализация и ординация);
- стремлению к лаконизму исследуемых моделей

- необходимостью сжатия объемов информации.

Принцип ординации наблюдений (нем. Ordnung) заключается в использовании различных методов оптимального целенаправленного проецирования (projecting pursuit) облака точек из многомерного пространства в пространство малой размерности (с 2 или 3 осями координат). Для этого используются различные методы или их модификации, но в целом сущность ординации заключается в представлении исходной матрицы данных Y в виде совокупности p латентных переменных F , которые и являются осями ординационной диаграммы (двумерной при $p = 2$ или трехмерной в случае с тремя такими латентными переменными). Выбор осей ординации осуществляется с использованием принципа оптимальности: т.к. стремления достичь минимума потерь содержательной информации, имеющейся в исходных данных.

Как правило, первая ось $F1$ проводится через центр сгущения значений u_{ij} и совпадает с направлением наибольшей по длине оси эллипсоида рассеяния. Вторая ось $F2$, также проходит через центр распределения, но проводится перпендикулярно к первой и совпадает по направлению со второй из главных полуосей эллипсоида рассеяния. Эта операция обеспечивает формирование “картинки” данных с минимально возможными искажениями, а новые обобщающие переменные $F1$ и $F2$ становятся ортогональными (т.е. взаимно некоррелированными).

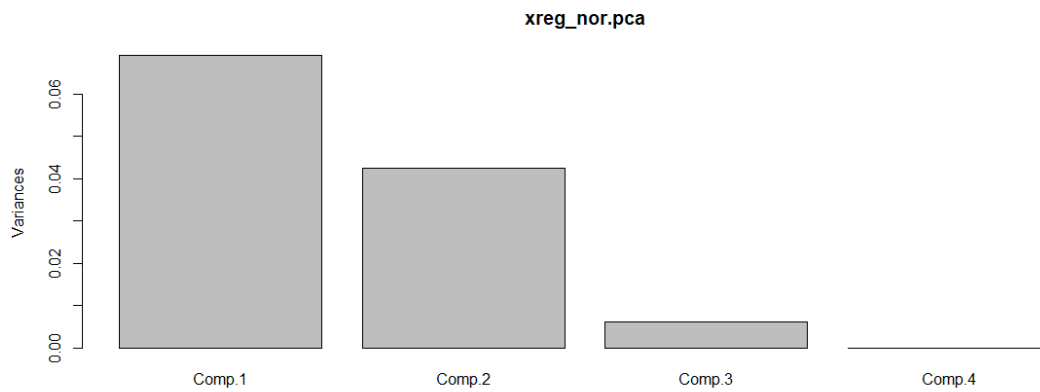
Большинство методов снижения размерности основано на анализе собственных значений и собственных векторов, который является важнейшим разделом линейной (матричной) алгебры. Классическим методом снижения размерности данных является анализ главных компонент (PCA, Principal Components Analysis), который широко используется в различных областях науки и техники. Снижение размерности исходного пространства методом PCA можно представить как последовательный, итеративный процесс, который можно оборвать на любом шаге u . Вследствие ортогональности системы координат главных компонент (т.е. фактически, их взаимной независимости) нет необходимости перестраивать матрицы счетов (scores) F и нагрузок (loadings) U при изменении числа компонент: последующие столбцы или строки просто прибавляются или отбрасываются.

Расчет главных компонент.

```
#xreg_nor.pca <- princomp(scale(xreg_nor))
#эту команду центрирования и нормирования целесообразно
#использовать для первичных не нормированных данных

xreg_nor.pca <- princomp(xreg_nor)
#эту команду целесообразно использовать для
#предварительно нормированных данных

#график главных компонент
plot(xreg_nor.pca)
```



#основные характеристики компонент

#Standard deviation - стандартное отклонение

#Proportion of Variance - доля вариации

#Cumulative Proportion - суммарная доля

```
harF <-summary(xreg_nor.pca)
```

```
print(harF)
```

Importance of components:

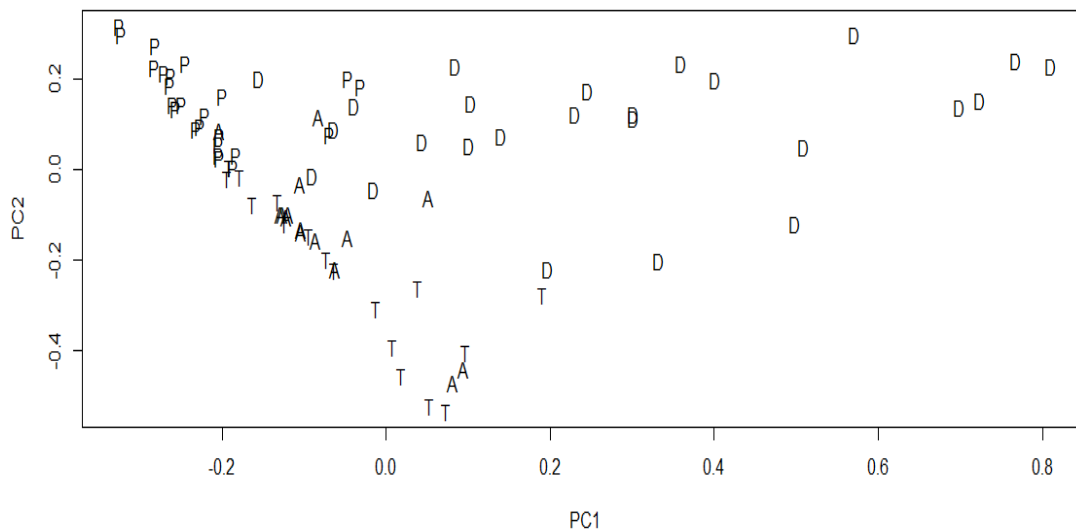
	Comp.1	Comp.2	Comp.3	Comp.4
## Standard deviation	0.2628116	0.2058287	0.07756685	0
## Proportion of Variance	0.5880696	0.3607043	0.05122616	0
## Cumulative Proportion	0.5880696	0.9487738	1.00000000	1

#Графическое представления регионов в пространстве главных компонент-----

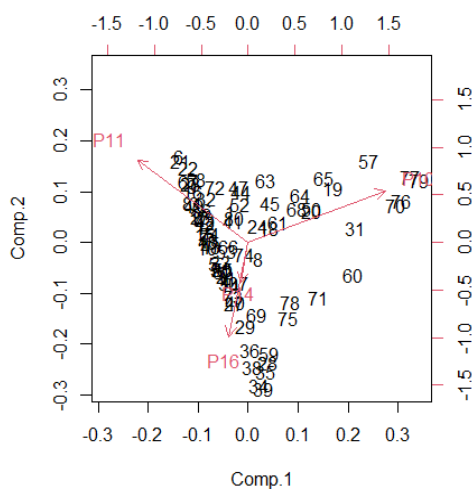
```
xreg_nor.p <- predict(xreg_nor.pca)
```

```
plot(xreg_nor.p[,1:2], type="n", xlab="PC1", ylab="PC2")
```

```
text(xreg_nor.p[,1:2], labels=abbreviate(xreg_f[,1], method="both.sides"))
```



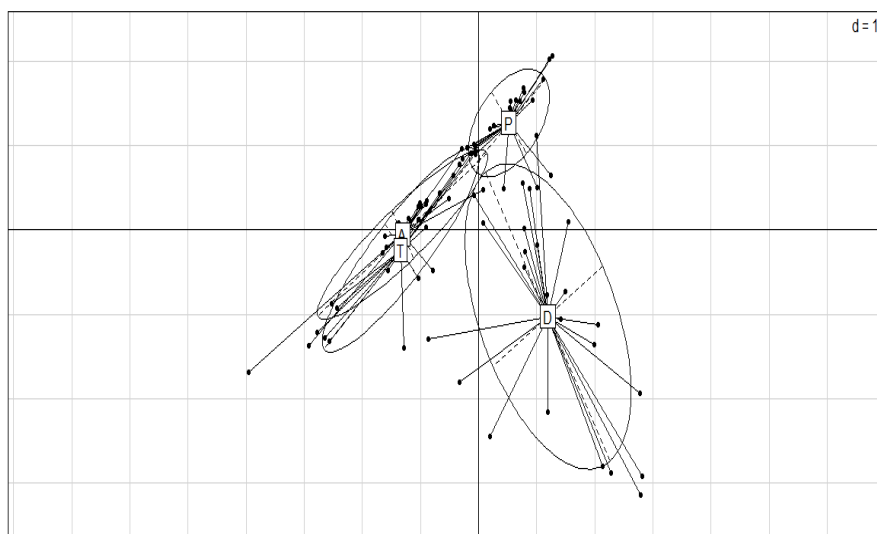
```
biplot(xreg_nor.pca)
```



```
loadings(xreg_nor.pca)
```

```
##
## Loadings:
##      Comp.1 Comp.2 Comp.3 Comp.4
## P10  0.773  0.364  0.141  0.500
## P11 -0.624  0.586  0.133  0.500
## P14      -0.285 -0.817  0.500
## P16 -0.108 -0.666  0.543  0.500
##
##              Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.00  1.00  1.00  1.00
## Proportion Var    0.25  0.25  0.25  0.25
## Cumulative Var    0.25  0.50  0.75  1.00
```

```
xreg_nor.dudi <- dudi.pca(xreg_nor[,1:4], scannf=FALSE)
s.class(xreg_nor.dudi$li, xreg_f)
```



Главные компоненты с использованием пакета 'vegan'

#Формирование компонент и их параметры

```
mod.pca <- rda(Y ~ 1)
```

```
#summary(mod.pca)
```

#ПРЕДСТАВЛЕНИЕ РЕГИОНОВ В ПРОСТРАНСТВЕ ДВУХ ПЕРВЫХ КОМПОНЕНТ

```
pca.scores <- as.data.frame(summary(mod.pca)$sites[,1:2])
```

```
pca.scores <- cbind(pca.scores, xreg_f)
```

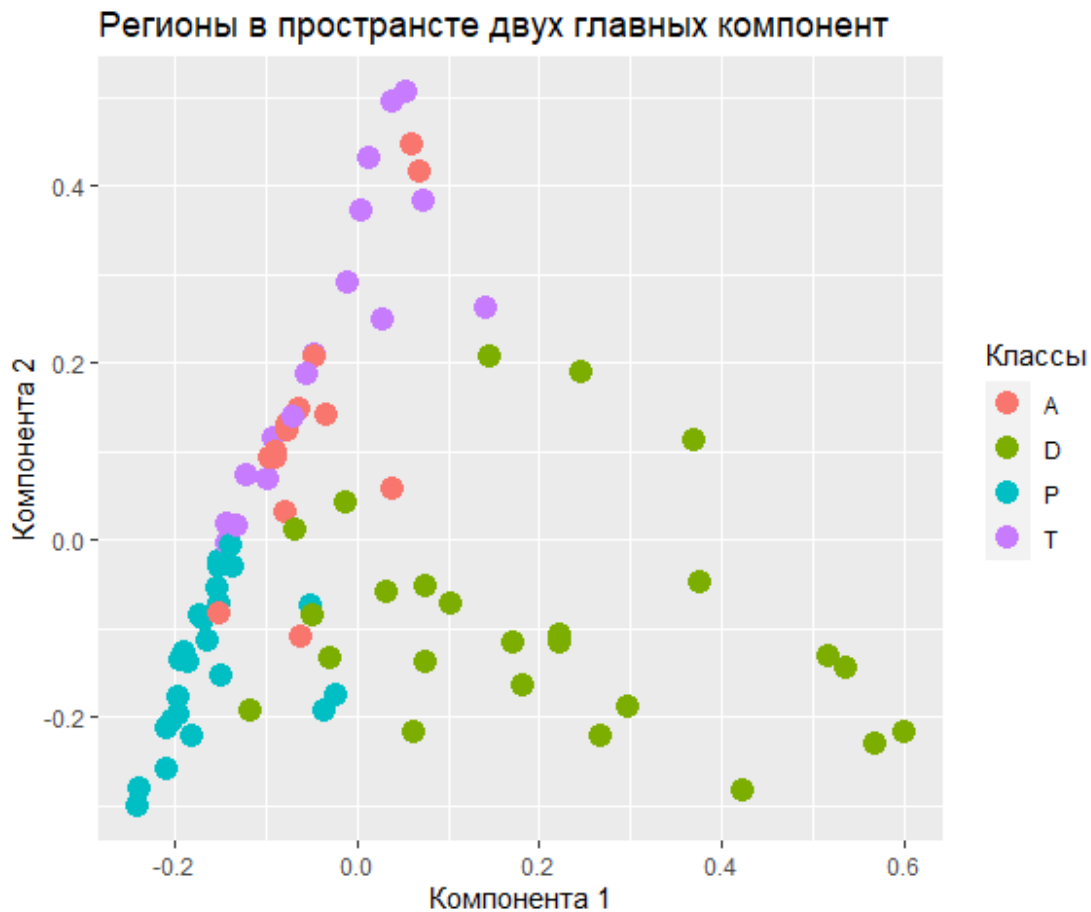
```
print(pca.scores)
```

##		PC1	PC2	xreg_f
## row1	-0.061551616	-0.109200404	A	
## row2	-0.092390347	0.114138740	T	
## row3	-0.185646084	-0.137370989	P	
## row4	-0.076915843	0.131312982	A	
## row5	-0.120654156	0.073367700	T	
## row6	-0.241385479	-0.299390587	P	
## row7	-0.150735887	-0.023477004	P	
## row8	0.038238179	0.057588107	A	
## row9	-0.195072055	-0.196434239	P	
## row10	-0.143558640	0.017354280	T	
## row11	-0.076750682	0.124305370	A	
## row12	-0.152339636	-0.053244217	P	
## row13	-0.142190311	-0.003926219	T	
## row14	-0.064005193	0.147283107	A	
## row15	-0.152026015	-0.030592208	P	
## row16	-0.196200994	-0.176559731	P	
## row17	-0.171832429	-0.085449404	P	
## row18	0.074675054	-0.050979386	D	
## row19	0.297512797	-0.188042789	D	
## row20	0.223791695	-0.107236997	D	
## row21	-0.239629030	-0.281138414	P	
## row22	-0.208846954	-0.258118758	P	
## row23	-0.201594910	-0.203055669	P	
## row24	0.032353815	-0.059115296	D	
## row25	-0.168245647	-0.089967947	P	
## row26	-0.093678997	0.092488745	A	
## row27	-0.047264991	0.210056746	T	
## row28	0.069801868	0.416137718	A	
## row29	-0.009543214	0.289950853	T	
## row30	-0.070021515	0.138209632	T	
## row31	0.377587181	-0.048212321	D	
## row32	-0.148201448	-0.154061435	P	
## row33	-0.097840545	0.067827919	T	
## row34	0.039446810	0.494419483	T	
## row35	0.060660198	0.446326673	A	
## row36	0.005559422	0.371917437	T	
## row37	-0.034610593	0.140644602	A	
## row38	0.013646191	0.431303662	T	
## row39	0.053891195	0.504848675	T	
## row40	-0.046338371	0.206341280	A	
## row41	-0.051542615	-0.074049267	P	

```
## row42 -0.150649801 -0.070707790 P
## row43 -0.150790066 -0.082717974 A
## row44 -0.023657045 -0.174096597 P
## row45 0.076151622 -0.138429847 D
## row46 -0.138343120 -0.005371802 P
## row47 -0.035000225 -0.191859207 P
## row48 -0.132796418 0.015039021 T
## row49 -0.190264005 -0.127177612 P
## row50 0.223551764 -0.116313935 D
## row51 -0.095103232 0.092822616 A
## row52 -0.029657743 -0.133824127 D
## row53 -0.078154128 0.030475108 A
## row54 -0.136263907 -0.030320846 P
## row55 -0.088386195 0.092342598 A
## row56 -0.164648430 -0.113432200 P
## row57 0.423587332 -0.281563516 D
## row58 -0.181803406 -0.220625394 P
## row59 0.072151478 0.383132090 T
## row60 0.369754270 0.112432823 D
## row61 0.103187942 -0.071048792 D
## row62 -0.090202466 0.098818169 A
## row63 0.062049707 -0.216852512 D
## row64 0.181658763 -0.164044003 D
## row65 0.266973983 -0.221603840 D
## row66 -0.067060544 0.012766000 D
## row67 -0.209784258 -0.212609165 P
## row68 0.170792776 -0.115714748 D
## row69 0.028894346 0.248310501 T
## row70 0.518195625 -0.130201395 D
## row71 0.246551331 0.190395645 D
## row72 -0.115902952 -0.191476841 D
## row73 -0.054557648 0.186327268 T
## row74 -0.011732545 0.042108186 D
## row75 0.141587366 0.262983404 T
## row76 0.537217889 -0.145205724 D
## row77 0.569559055 -0.228781406 D
## row78 0.146138489 0.207065898 D
## row79 0.601393266 -0.216154208 D
## row80 -0.047770161 -0.084638901 D
## row81 -0.193418916 -0.136447372 P
```

#График регионов по классам в пространстве двух главных компонент 1

```
g3 <-ggplot(data= pca.scores, aes(x=PC1,y=PC2,colour = xreg_f),colors =
"Accent")
g3 <-g3 +geom_point(size=4)
g3 <-g3 +labs(title ="Регионы в пространсте двух главных компонент",
x="Компонента 1", y="Компонента 2",colour = "Классы")
#g3 <-g3 + theme_fivethirtyeight() + scale_color_fivethirtyeight()
#g3 <-g3 + theme_stata() + scale_colour_stata()
g3
```



#графическое представление классов регионов в пространстве двух главных компонент 2

Составляем таблицу для "каркаса" точек на графике

```
l <- lapply(unique(pca.scores$xreg_f), function(c)
{ f <- subset(pca.scores, xreg_f == c); f[chull(f), ]})
hull <- do.call(rbind, l)
```

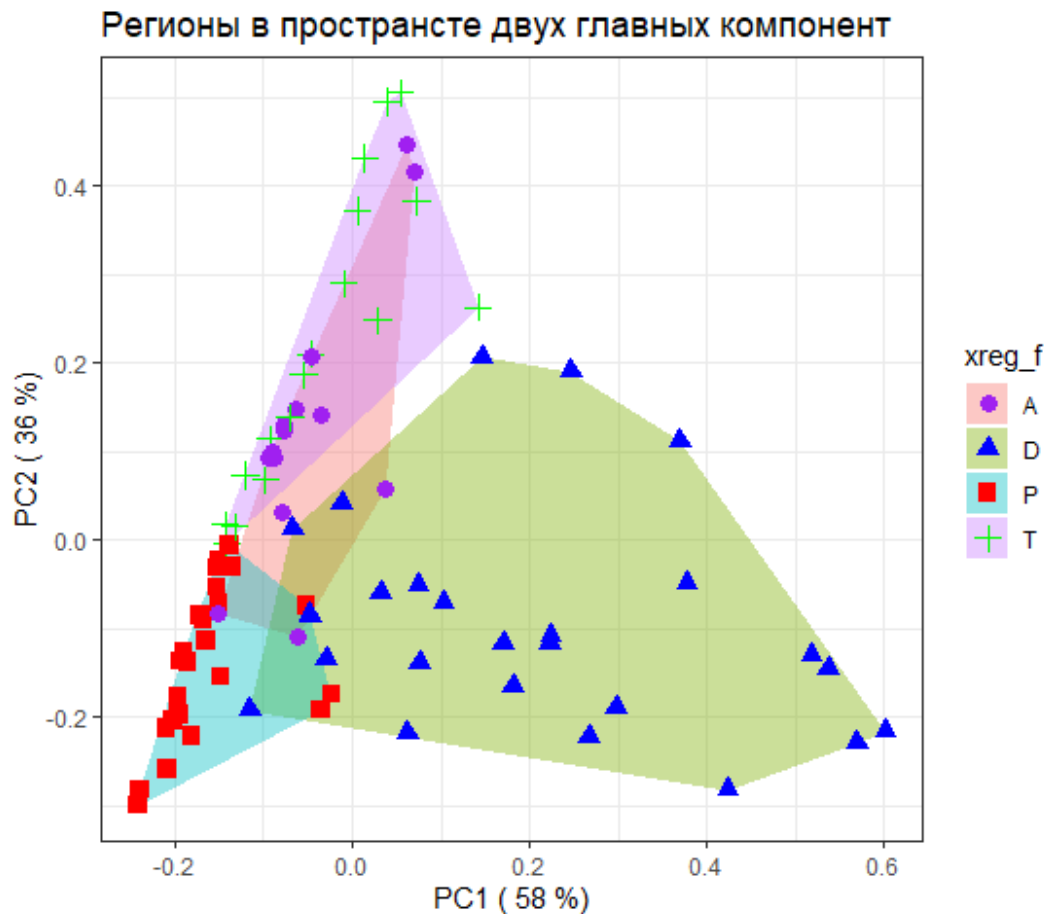
Включаем в названия осей доли объясненной дисперсии

```
axX <- paste("PC1 (",
as.integer(100*mod.pca$CA$eig[1]/sum(mod.pca$CA$eig)), "%)")
```

```
axY <- paste("PC2 (",
```

```
as.integer(100*mod.pca$CA$eig[2]/sum(mod.pca$CA$eig)), "%)")
```

```
ggplot() +
geom_polygon(data=hull, aes(x=PC1, y=PC2, fill=xreg_f),
alpha=0.4, linetype=0) +
labs(title = "Регионы в пространстве двух главных компонент")+
geom_point(data=pca.scores, aes(x=PC1, y=PC2, shape= xreg_f,
colour=xreg_f), size=3) +
scale_colour_manual( values = c('purple', 'blue', 'red', 'green'))+
xlab(axX) + ylab(axY) + coord_equal() + theme_bw()
```



Модели для предсказания класса объектов

Классификация – наиболее часто встречающаяся задача машинного обучения, и заключается в построении моделей, выполняющих отнесение интересующего нас объекта к одному из нескольких известных классов. Существуют сотни методов классификации, которые можно использовать для предсказания значения отклика с двумя и более классами.

Выделим основания для классификации методов классификации объектов.

1. Поставленная цель исследования

- объяснение внутренних механизмов изучаемых процессов;
- прогнозирование отклика.

В первом случае создаваемая модель должна в явном виде отображать их в виде наглядной схемы, либо осуществлять сравнительную оценку силы влияния отдельных переменных. Примерами хорошо интерпретируемых моделей классификации являются деревья решений, логистическая регрессия и модели дискриминации.

Во втором случае, когда основной задачей является достижение высокой общей точности предсказаний (overall accuracy). Изучаемый процесс представляется в виде “черного ящика”, а решающие процедуры могут иметь большое (до десятков тысяч) или неопределенное число трудно интерпретируемых параметров.

2.Парадигма машинного обучения

- геометрическая;
- вероятностная;
- логическая.

При геометрическом подходе каждый объект рассматривается как точка в многомерной системе координат. Геометрическая модель разделения на классы строится в пространстве признаков с применением таких геометрических понятий, как прямые, плоскости (в общем виде “гиперплоскости”) и криволинейные поверхности. Примеры моделей, реализующих геометрическую парадигму: логистическая регрессия, метод опорных векторов и дискриминантный анализ. Другим важным геометрическим понятием является функция расстояния между объектами, которая приводит к классификатору по ближайшим соседям и k-средних.

Вероятностный подход заключается в предположении о существовании некоего случайного процесса, который порождает значения целевых переменных, подчиняющиеся вполне определенному, но неизвестному нам распределению вероятностей. Примером модели вероятностного характера является байесовский классификатор, формирующий решающее правило по принципу апостериорного максимума.

Модели логического типа по своей природе наиболее алгоритмичны, поскольку легко выражаются на языке правил, понятных человеку, таких как: $if < \text{условие} > = 1 then Y = < \text{класс} i >$ Примером таких моделей являются деревья классификации.

3 Природа наблюдаемых признаков(шкалы измерения)

- бинарные (0/1);
- категориальные (номинальные);
- счетные (порядковые);
- метрические.

Имеются определенные нюансы при использовании перечисленных типов признаков в качестве предикторов в тех или иных методах классификации. Однако принципиально важное значение имеет, к какому типу признаков относится отклик: задача классификации предполагает, что он измерен в бинарных, категориальных или, отчасти, в порядковых шкалах. Если отклик измеряется в метрической шкале имеем задачу регрессии.

При решении задачи классификации российских регионов по типу экономики отклик измеряется в категориальной (номинальной) шкале, а исходные признаки в метрической.

4 Полнота и характер априорной информации о классах (отклике)

- обучение с учителем;
- обучение без учителя.

Обучение с учителем (Supervised learning) — один из разделов машинного обучения, посвященный решению следующей задачи. Имеется множество объектов (ситуаций) и множество возможных ответов (откликов, реакций). Существует некоторая зависимость между ответами и объектами, но она неизвестна. Известна только конечная совокупность прецедентов — пар «объект, ответ», называемая обучающей выборкой. На основе этих данных требуется восстановить зависимость, то

есть построить алгоритм, способный для любого объекта выдать достаточно точный ответ. Для измерения точности ответов определённым образом вводится функционал качества. Под учителем понимается либо сама обучающая выборка, либо тот, кто указал на заданных объектах правильные ответы. При решении задачи классификации российских регионов по типу экономики предварительно сформирована информация об отклике для всех исследуемых регионов для определенного момента времени. То есть имеется исходная обучающая выборка.

Обучение без учителя (Unsupervised learning) — один из разделов машинного обучения. Изучает широкий класс задач обработки данных, в которых известны только описания множества объектов (обучающей выборки), и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами. Основной тип задач обучения без учителя это кластеризация. *Кластерный анализ (Data clustering)* — задача разбиения заданной выборки объектов (ситуаций) на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались.

Задача классификации российских регионов без учителя.

```
# формирование полного исходного описания регионов
xreg_nor <- cbind(xreg_nor, xreg_f)
```

```
#РАСЧЕТ МАТРИЦЫ НЕСХОДСТВА РЕГИОНОВ
```

```
#МНОГОМЕРНОЕ МАСШТАБИРОВАНИЕ И ПЕРЕХОД В ДВУХМЕРНОЕ ПРОСТРАНСТВО k=2
```

```
#возможные метрики несходства
```

```
#1. euclidean - Евклидово расстояние
```

```
#2. manhattan - манхэттенское расстояние
```

```
#3. gower - расстояние Гауэра
```

```
xreg_nor.dist <- daisy(xreg_nor[,1:4], metric="manhattan")#вектор несходства
```

```
mat.dist <- as.matrix(xreg_nor.dist)#матрица несходства
```

```
#вызов dij as.matrix(xreg_nor.dist)[i,j]
```

```
#МАТРИЦА НЕСХОДСТВА РОССИЙСКИХ РЕГИОНОВ ПО ТИПУ ЭКОНОМИКИ (5 первых регионов)
```

```
head(mat.dist, n=5)
```

```
##           1           2           3           4           5           6           7
## 1 0.0000000 0.44608984 0.4609685 0.44047478 0.5222612 0.5661566 0.4301542
## 2 0.4460898 0.00000000 0.4939062 0.06307929 0.2095481 0.7664855 0.2872459
## 3 0.4609685 0.49390616 0.0000000 0.53645308 0.3605294 0.2793790 0.2066603
## 4 0.4404748 0.06307929 0.5364531 0.00000000 0.2677593 0.8142993 0.3392242
## 5 0.5222612 0.20954807 0.3605294 0.26775934 0.0000000 0.6331088 0.1858392
##           8           9          10          11          12          13          14
## 1 0.3756095 0.3495431 0.5110203 0.45920306 0.3478599 0.4306738 0.5186422
## 2 0.4301362 0.5549217 0.2629574 0.14580360 0.3137430 0.2571814 0.2129434
## 3 0.6623871 0.1884114 0.2909337 0.55518136 0.1801631 0.2367248 0.6146205
## 4 0.3670569 0.5976858 0.3219568 0.09386549 0.3628865 0.3034738 0.1615128
## 5 0.6348163 0.5347316 0.1113340 0.35484409 0.2891452 0.1469185 0.4224914
##          15          16          17          18          19          20          21
## 1 0.4192262 0.42360174 0.45099598 0.5710914 0.8495222 0.7627135 0.5452014
## 2 0.2960272 0.54736415 0.39882285 0.5024744 1.0535031 0.8480554 0.7399631
## 3 0.1978789 0.09594935 0.09508332 0.5876999 1.0403215 0.8348738 0.2584238
## 4 0.3477712 0.59094053 0.44922399 0.4945574 1.0418543 0.8364066 0.7933440
## 5 0.2050798 0.44145732 0.27342003 0.4976064 1.0486351 0.8431873 0.6106396
```

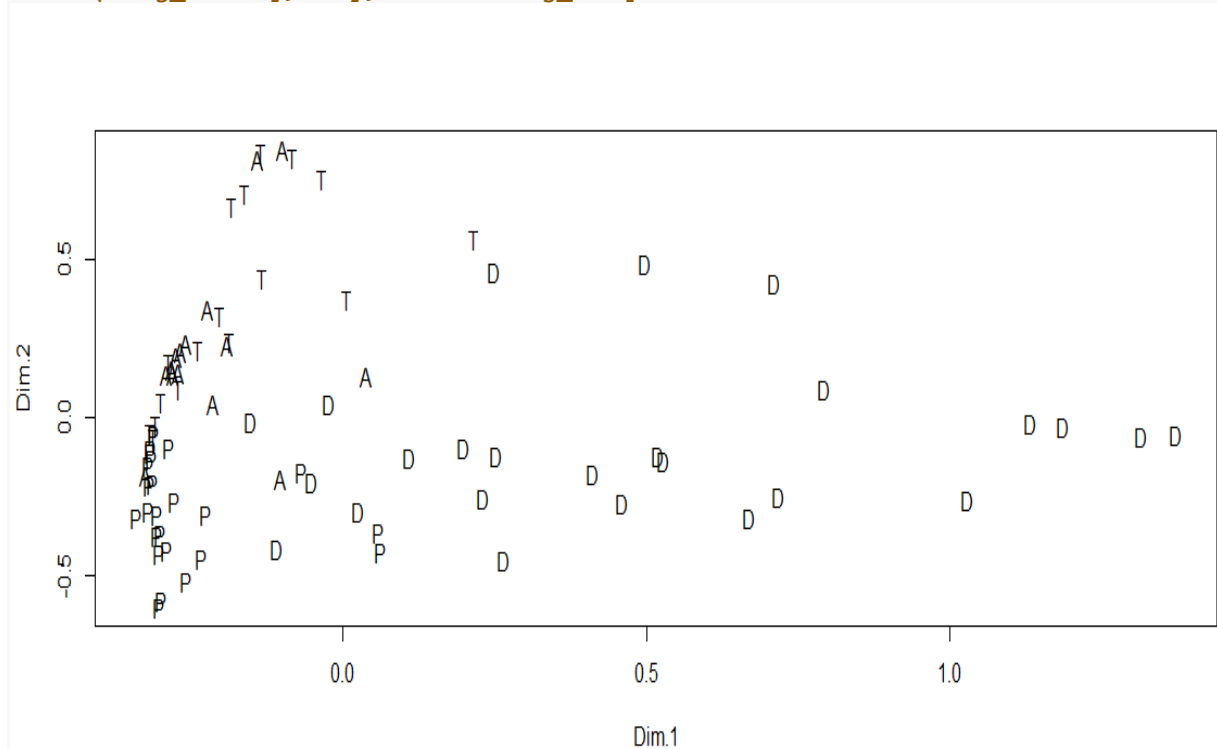
##	22	23	24	25	26	27	28
## 1	0.4630683	0.4250232	0.5059596	0.4006984	0.41792989	0.6375248	1.1453352
## 2	0.7021458	0.5963877	0.4576376	0.3921508	0.04381905	0.2297232	0.7823201
## 3	0.2114257	0.1328363	0.5225680	0.1017554	0.45134942	0.6579660	1.2413136
## 4	0.7446927	0.6389346	0.5001845	0.4404318	0.08636596	0.2269580	0.7308896
## 5	0.5687691	0.4809066	0.4189947	0.3138519	0.20346485	0.3057501	0.9918682
##	29	30	31	32	33	34	35
## 1	0.7932981	0.50475629	0.9942978	0.2967729	0.4074908	1.0638032	1.0185345
## 2	0.4167277	0.09485066	1.1061879	0.5070069	0.1031447	0.6615423	0.6755445
## 3	0.8067157	0.54037269	1.1491873	0.1641956	0.4082598	1.1422668	1.1562691
## 4	0.4139625	0.09208551	1.0945392	0.5495538	0.1456917	0.6286703	0.6213488
## 5	0.4544999	0.20566171	1.1450417	0.4527605	0.1708144	0.7900510	0.8040532
##	36	37	38	39	40	41	42
## 1	0.8836715	0.5163439	0.9802390	1.1255221	0.5742791	0.2517084	0.2540071
## 2	0.4871963	0.3283317	0.5420403	0.7284724	0.1961941	0.4160512	0.3140147
## 3	0.9796498	0.6535960	1.0268142	1.1668728	0.6769187	0.3059140	0.2163403
## 4	0.4431968	0.2652524	0.5430328	0.7257073	0.1419984	0.4585981	0.3644077
## 5	0.6191204	0.5330118	0.6693079	0.8146570	0.3247028	0.3078125	0.3855845
##	43	44	45	46	47	48	49
## 1	0.2044201	0.2337074	0.3805030	0.3610349	0.3117775	0.3774552	0.49569914
## 2	0.3765928	0.6070458	0.5728458	0.2366315	0.6610806	0.2076181	0.48128501
## 3	0.2857884	0.3888328	0.5865785	0.2572746	0.3542194	0.2862881	0.03486688
## 4	0.3793579	0.6495927	0.6153927	0.2866430	0.7036275	0.2568039	0.53478400
## 5	0.4576321	0.4976717	0.5824329	0.2004103	0.5277038	0.1534672	0.35207959
##	50	51	52	53	54	55	56
## 1	0.6339475	0.38624626	0.2730969	0.3079995	0.3716707	0.40851579	0.45971161
## 2	0.8293514	0.05984358	0.5454113	0.1651423	0.2951363	0.04238044	0.46896052
## 3	0.8977559	0.46772090	0.3199632	0.4052880	0.2172170	0.46193812	0.08932205
## 4	0.8177026	0.07758068	0.5879582	0.2076892	0.3376832	0.08492736	0.51150743
## 5	0.8936104	0.24320638	0.4120345	0.2673338	0.2201511	0.22346765	0.33558376
##	57	58	59	60	61	62	63
## 1	1.173642	0.3774553	0.8989891	1.1065785	0.5315316	0.40498497	0.3770859
## 2	1.377623	0.6373020	0.6224738	0.9713532	0.5612727	0.04110487	0.7353574
## 3	1.364441	0.1677633	1.1031983	1.2477267	0.5694373	0.47185647	0.5678852
## 4	1.365974	0.6798489	0.5682781	0.9597045	0.5496239	0.06481273	0.7779044
## 5	1.372755	0.5158336	0.7509825	1.0785659	0.5652428	0.23516786	0.6019807
##	64	65	66	67	68	69	70
## 1	0.5848360	0.8060250	0.3688744	0.4178644	0.5969148	0.7719607	1.262827
## 2	0.7888169	1.0100059	0.2498828	0.6034738	0.7307646	0.5199336	1.466808
## 3	0.7756353	0.9968243	0.3802682	0.1699175	0.7372003	0.8080132	1.453626
## 4	0.7771681	0.9983571	0.2924297	0.6531108	0.7191158	0.5171685	1.455159
## 5	0.7839489	1.0051378	0.1814391	0.5096742	0.7330547	0.4557974	1.461940
##	71	72	73	74	75	76	77
## 1	1.0405541	0.2882842	0.6903637	0.5123556	0.8096991	1.314454	1.439395
## 2	0.7984452	0.6198544	0.3073516	0.2702467	0.5675902	1.518435	1.643376
## 3	1.1166318	0.1813955	0.5958230	0.5289640	1.0472793	1.505253	1.630195
## 4	0.7956801	0.6624014	0.3045864	0.3085303	0.5648250	1.506786	1.631728
## 5	0.5574943						

```
## 5 0.7644159 0.4864777 0.2436072 0.2537005 0.6950634 1.513567 1.638508
0.6297130
##          79          80          81
## 1 1.493994 0.3527109 0.54689197
## 2 1.697975 0.4767502 0.51741532
## 3 1.684794 0.3632526 0.08592346
## 4 1.686326 0.5192971 0.56068931
## 5 1.693107 0.3433734 0.38403856
```

#многомерное масштабирование (MDS) матрицы данных.

```
xreg_nor.c <- cmdscale(xreg_nor.dist, k = 2, eig = FALSE)
plot(xreg_nor.c[,1:2], type="n", xlab="Dim.1", ylab="Dim.2")
text(xreg_nor.c[,1:2], labels=abbreviate(xreg_nor[,5],1, method="both.sides"))
```

```
#text(xreg_nor.c[,1:2], labels=xreg_nor[
```



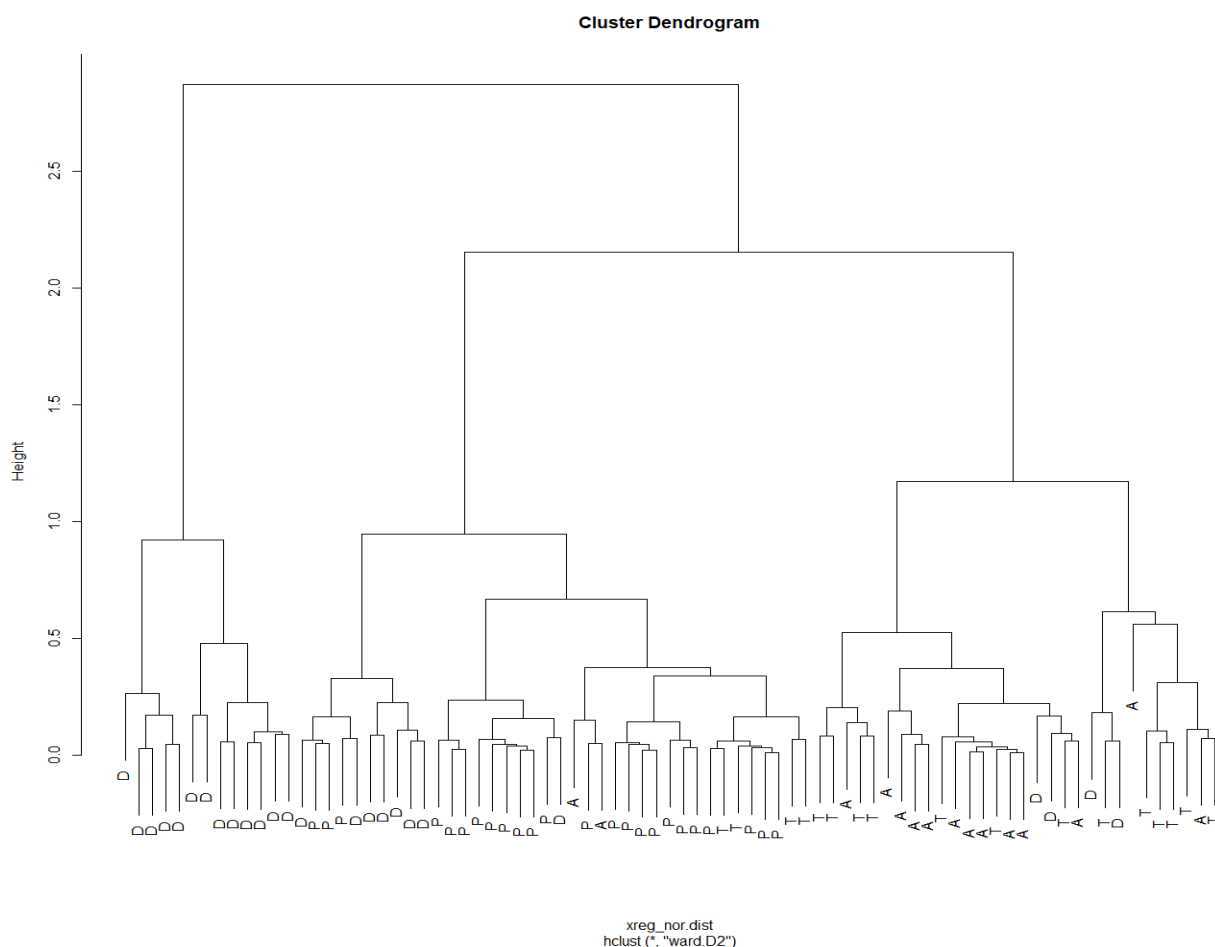
```
,5])
```

Кластерный анализ российских регионов по типу экономики

Иерархическая кластеризация

Ещё одним вариантом работы с матрицей различий является кластерный анализ. Существует множество его разновидностей, причём наиболее употребительными являются иерархические методы, которые вместо уже привычных двумерных графиков производят полуторамерные деревья классификации, или дендрограммы.

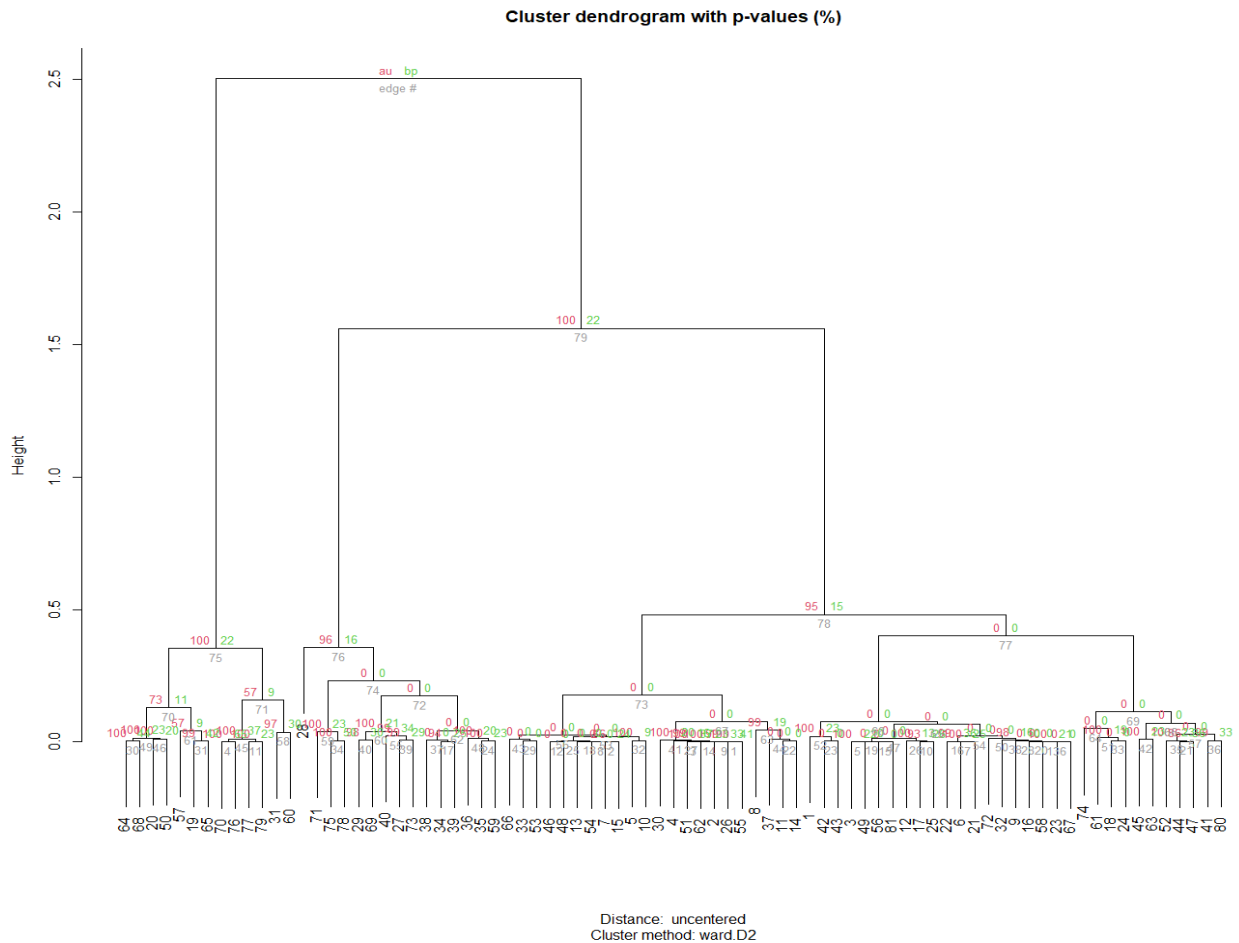
```
#методы агломерации "ward.D", "ward.D2", "single", "complete", "average"
xreg_nor <- xreg_nor[seq(1,nrow(xreg_nor),1),]#выделение подмножества регионов
xreg_nor.dist <- daisy(xreg_nor[,1:4])#вектор несходства
#алгоритм иерархической классификации
xreg_nor.h <- hclust(xreg_nor.dist, method="ward.D2")
plot(xreg_nor.h, labels=abbreviate(xreg_nor[,5],1, method="both.sides"))
```



```
#Иерархическая кластеризация выполняется для заданных данных
# p-значения вычисляются для каждого из кластеров.
#method.dist="uncentered", "correlation", "abscor", manhattan, euclidean
xreg_nor.pv <- pvclust(t(xreg_nor[,1:4]),method.dist="uncentered",
method.hclust="ward.D2", nboot=1000,parallel=TRUE)

## Creating a temporary cluster...done:
## socket cluster with 5 nodes on host 'localhost'
## Multiscale bootstrap... Done.

plot(xreg_nor.pv, print.num=TRUE)
```



##печатаать кластеры с высокими p-значениями

```
xreg_nor.pp <- pvpick(xreg_nor.pv)
```

```
xreg_nor.pp
```

```
## $clusters
```

```
## $clusters[[1]]
```

```
## [1] "19" "20" "31" "50" "57" "60" "64" "65" "68" "70" "76" "77" "79"
```

```
##
```

```
## $clusters[[2]]
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14"
```

```
"15"
```

```
## [16] "16" "17" "18" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30" "32"
```

```
"33"
```

```
## [31] "34" "35" "36" "37" "38" "39" "40" "41" "42" "43" "44" "45" "46" "47"
```

```
"48"
```

```
## [46] "49" "51" "52" "53" "54" "55" "56" "58" "59" "61" "62" "63" "66" "67"
```

```
"69"
```

```
## [61] "71" "72" "73" "74" "75" "78" "80" "81"
```

```
##
```

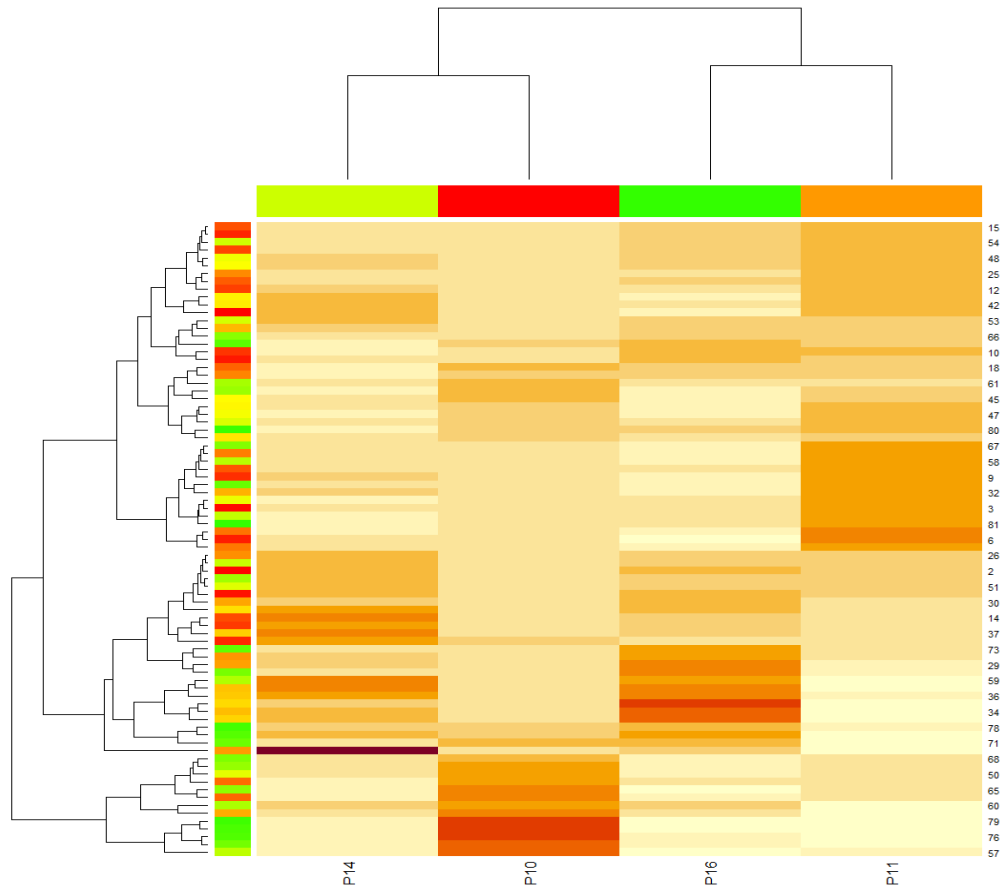
```
##
```

```
## $edges
```

```
## [1] 75 79
```

#Иерархическая кластеризация регионов с тепловой картой

```
xM = as.matrix(xreg_nor[,1:4])
rc <- rainbow(nrow(xM), start = 0, end = .3)
cc <- rainbow(ncol(xM), start = 0, end = .3)
hv <- heatmap(xM, scale = "col", RowSideColors = rc,
ColSideColors = cc, margins = c(10,10),
cexCol = 1.5, cexRow = 1)
```



*#метод k средних Мак-Кина, в котором априори задается
#искомое число конечных кластеров, обозначаемое как "k".*

#Расчет кластеров регионов и их параметров

```
df.stand <- xreg_nor[,1:4]
(clus <- kmeans(df.stand, centers = 4))

## K-means clustering with 4 clusters of sizes 26, 36, 7, 12
##
## Cluster means:
##      P10      P11      P14      P16
## 1 0.04184832 0.25359993 0.18090880 0.5236429
## 2 0.03740828 0.57127684 0.06823664 0.3230782
## 3 0.69916727 0.06201858 0.03544624 0.2033679
## 4 0.35672645 0.31842375 0.03869516 0.2861546
##
```

```

## Clustering vector:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 2 1 2 1 2 2 2 1 2 2 1 2 2 1 2 2 2 4 4 4 2 2 2 4 2 1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
## 1 1 1 1 3 2 2 1 1 1 1 1 1 1 2 2 2 2 4 2 2 2 2 4 1 2
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
## 2 2 1 2 3 2 1 3 4 1 4 4 4 2 2 4 1 3 4 2 1 1 1 3 3 1
## 79 80 81
## 3 2 2
##
## Within cluster sum of squares by cluster:
## [1] 1.1572742 0.6978739 0.2152071 0.3299380
## (between_SS / total_SS = 74.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
##      "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"

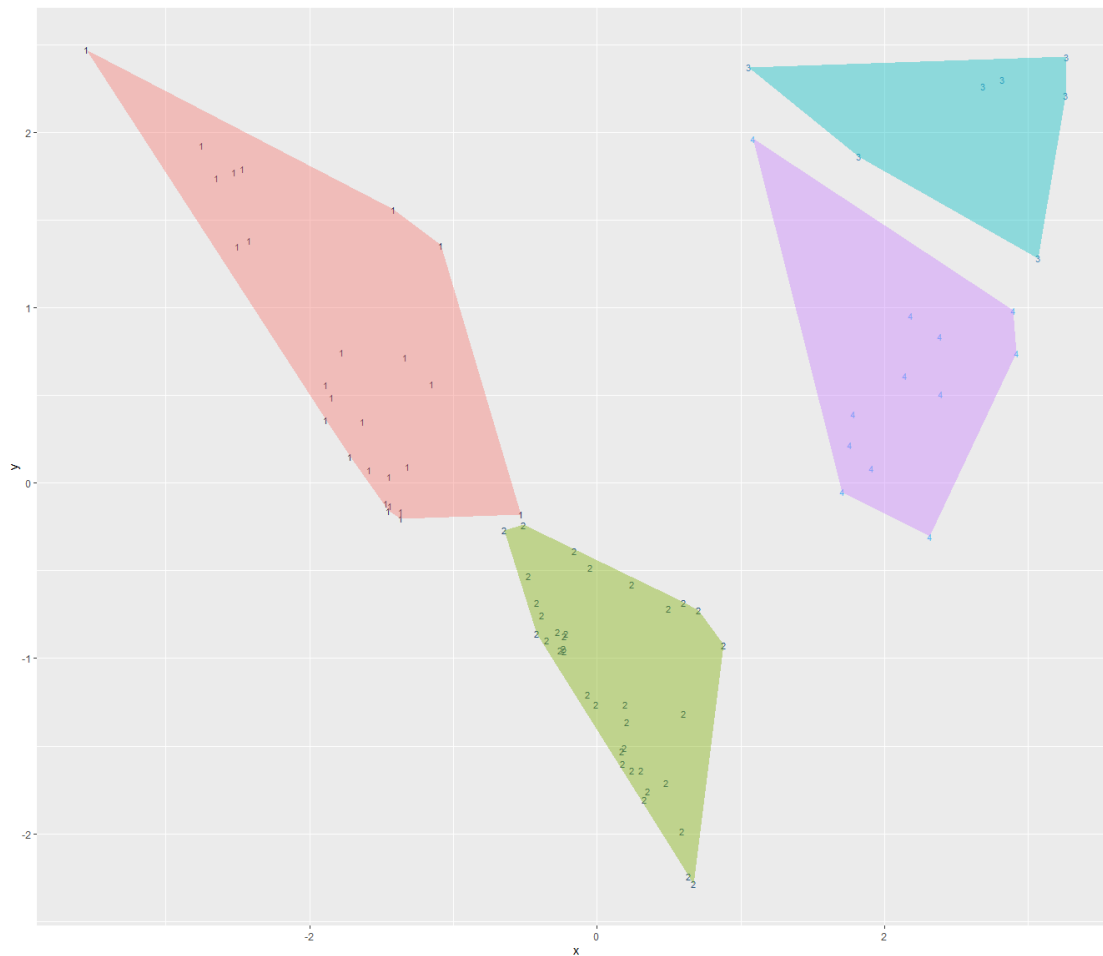
#Расчет матрицы внутрикластерных и межкластерных расстояний
n <- dim(df.stand)[[1]]
euc.dist <- as.matrix(dist(df.stand))
dist = melt(euc.dist)
df.stand$cluster <- clus$cluster
pairs <- data.frame(dist = dist,
  ca = as.vector(outer(1:n, 1:n, function(a, b)
    df.stand[a, 'cluster'])),
  cb = as.vector(outer(1:n, 1:n, function(a, b)
    df.stand[b, 'cluster'])))
dcast(pairs, ca ~ cb, value.var='dist.value', mean)

##   ca      1      2      3      4
## 1  1 0.2608027 0.4257579 0.8011970 0.4814663
## 2  2 0.4257579 0.1767062 0.8627888 0.4414384
## 3  3 0.8011970 0.8627888 0.2042172 0.4758884
## 4  4 0.4814663 0.4414384 0.4758884 0.2005801

#Визуализация кластеров регионов в пространстве двух главных компонент
c.pca <- prcomp(df.stand, center = TRUE, scale = TRUE)
d <- data.frame(x=c.pca$x[, 1], y=c.pca$x[, 2])
d$cluster <- clus$cluster
h <- do.call(rbind, lapply(unique(clus$cluster),
  function(c) { f <- subset(d, cluster==c); f[chull(f),]}))

ggplot() + geom_text(data=d, aes(label=cluster, x=x, y=y,
  color=cluster), size=3) +
  geom_polygon(data=h, aes(x=x, y=y, group=cluster,
  fill=as.factor(cluster)), alpha=0.4, linetype=0) +
  theme(legend.position = "none")

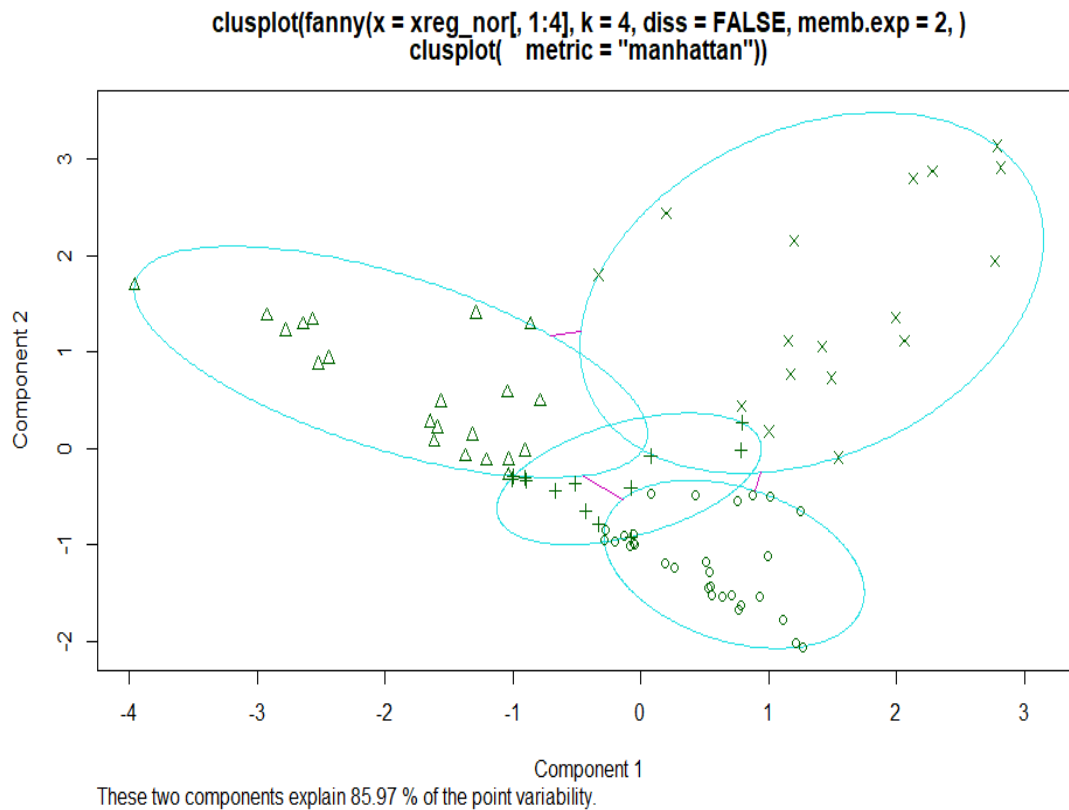
```



Нечеткая классификация: fuzzy-методы

Эти методы основаны на идее того, что каждый объект может принадлежать сразу нескольким кластерам, но с разной силой.

```
#Вычисляет нечеткую кластеризацию регионов
#Априори задается число кластеров k=4.
xreg_nor.f <- fanny(xreg_nor[,1:4],4,diss = FALSE,memb.exp = 2,
metric ="manhattan")
plot(xreg_nor.f, which=1)
```



```
RegionsRU <- data.frame(reg=xreg_nor[,5], xreg_nor.f$membership)
#Функция принадлежности регионов выделяемым кластерам
#Ближайшая четкая кластеризация регионов и.т.д
print(xreg_nor.f)
```

```
## Fuzzy Clustering object of class 'fanny' :
## m.ship.expon.      2
## objective      6.790778
## tolerance      1e-15
## iterations      329
## converged       1
## maxit          500
## n              81
## Membership coefficients (in %, rounded):
##      [,1] [,2] [,3] [,4]
## 1      32  25  30  13
## 2      15  40  38   7
## 3      62  14  17   7
## 4      14  42  37   7
## 5      23  32  37   8
## 6      46  19  22  13
## 7      41  22  30   7
## 8      19  33  31  17
## 9      53  17  20   9
## 10     31  27  34   8
## 11     16  39  36   8
## 12     47  20  26   6
```

## 13	35	25	33	7
## 14	17	39	35	10
## 15	43	21	29	6
## 16	61	14	17	7
## 17	58	16	21	6
## 18	21	27	29	23
## 19	11	13	13	63
## 20	14	16	17	53
## 21	48	18	21	13
## 22	51	17	20	11
## 23	59	15	18	8
## 24	23	27	31	19
## 25	59	15	20	6
## 26	16	38	40	6
## 27	15	43	33	10
## 28	20	32	29	19
## 29	16	39	31	13
## 30	14	42	37	7
## 31	12	15	15	58
## 32	57	16	20	8
## 33	18	35	41	6
## 34	18	35	29	18
## 35	18	36	29	17
## 36	17	38	30	15
## 37	18	37	33	12
## 38	18	36	30	16
## 39	19	34	29	18
## 40	14	43	33	9
## 41	34	24	31	11
## 42	44	22	27	8
## 43	41	22	27	9
## 44	37	21	26	17
## 45	24	23	26	27
## 46	34	25	34	7
## 47	39	20	24	17
## 48	30	27	37	7
## 49	60	15	19	7
## 50	14	17	17	53
## 51	16	38	39	7
## 52	36	22	27	15
## 53	20	33	40	7
## 54	40	22	31	7
## 55	15	39	40	6
## 56	57	16	20	7
## 57	14	16	16	55
## 58	56	16	19	10
## 59	17	36	29	17
## 60	15	22	20	44
## 61	20	25	27	28
## 62	15	39	39	6
## 63	27	21	24	29
## 64	16	18	19	48

```

## 65 13 14 15 58
## 66 23 30 39 8
## 67 56 16 19 9
## 68 17 19 20 44
## 69 17 36 31 16
## 70 13 15 15 57
## 71 16 27 24 33
## 72 52 16 20 11
## 73 17 39 34 11
## 74 19 33 35 12
## 75 16 35 29 21
## 76 14 15 15 56
## 77 15 17 17 52
## 78 16 34 28 22
## 79 15 17 17 50
## 80 32 24 31 13
## 81 54 17 21 8
## Fuzzyness coefficients:
## dunn_coeff normalized
## 0.32229972 0.09639963
## Closest hard clustering:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 1 2 1 2 3 1 1 2 1 3 2 1 1 2 1 1 1 3 4 4 1 1 1 3 1 3
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
## 2 2 2 2 4 1 3 2 2 2 2 2 2 2 1 1 1 1 4 1 1 3 1 4 3 1
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
## 3 1 3 1 4 1 2 4 4 3 4 4 4 3 1 4 2 4 4 1 2 3 2 4 4 2
## 79 80 81
## 4 1 1
##
## Available components:
## [1] "membership" "coeff" "memb.exp" "clustering" "k.crisp"
## [6] "objective" "convergence" "diss" "call" "silinfo"
## [11] "data"

```

Задача классификации российских регионов с учителем.

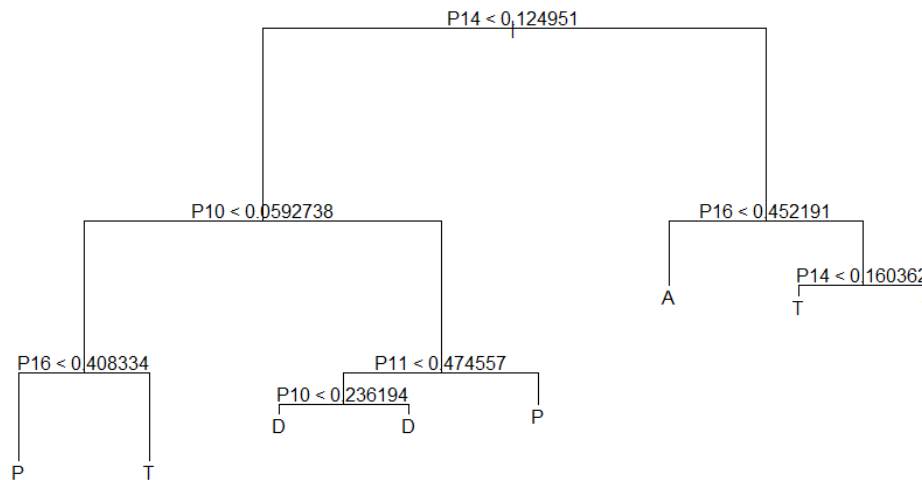
#ДЕРЕВО РЕШЕНИЙ

*#здесь выясняется, какие именно показатели могут быть использованы для
#разделения объектов на заранее заданные группы. В результате строится #ключ, в
котором на каждой ступени объекты делятся на две группы: разделения #объектов
#дерево решений для регионов строиться с использованием полной обучающей
#выборки (т.е. все 81 регион априори отнесены к тому или иному классу)*

```

xreg_nor.tree <- tree(xreg_nor[,5] ~ ., xreg_nor[, -5])
plot(xreg_nor.tree)
text(xreg_nor.tree)

```



#Задание обучающей и проверочной последовательностей

#Половина регионов относится к обучающей последовательности, остальные к проверочной

```
xreg_nor.train <- xreg_nor[seq(1,nrow(xreg_nor),2),]
xreg_nor.unknown <- xreg_nor[-seq(1,nrow(xreg_nor),2),]
```

#Алгоритм Random Forest

```
set.seed(17)
```

#алгоритм случайного леса Бреймана реализуется на тестовой последовательности (создание модели)

```
xreg_nor.rf <- randomForest(xreg_nor.train[,5] ~ ., data=xreg_nor.train[,1:4])
```

#классификация по построенной модели регионов проверочной последовательности

```
xreg_nor.rfp <- predict(xreg_nor.rf, xreg_nor.unknown[,1:4])
```

```
xreg_nor.rfp
```

```
##  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52
##  A  A  P  A  T  P  A  P  D  D  P  D  A  A  T  P  A  T  T  A  A  P  P  T  D  P
## 54 56 58 60 62 64 66 68 70 72 74 76 78 80
##  P  P  P  D  A  D  T  D  D  P  T  D  T  P
## Levels: A D P T
```

#Априорная классификация регионов на проверочной последовательности

```
xreg_nor.unknown[,5]
```

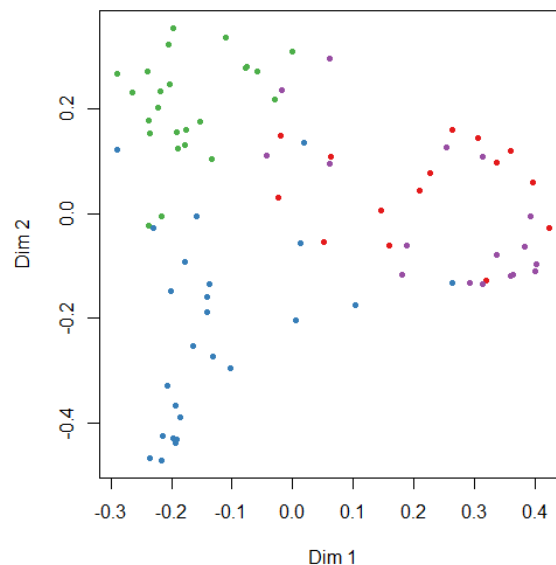
```
## [1] T A P A T P A P D D P D A A T P T T T A P P P T D D P P P D A D D D D D
## [39] D D
## Levels: A D P T
```

#матрица соответствия регионов их фактическим классам на проверочной последовательности

```
table(xreg_nor.rfp, xreg_nor.unknown[,5])
```

```
##
## xreg_nor.rfp  A  D  P  T
##           A  7  0  1  2
##           D  0  9  0  0
##           P  0  3 10  0
##           T  0  3  0  5
```

#визуализация работы алгоритма случайного леса на всей выборке регионов
 set.seed(17)
 xreg_nor.urf <- randomForest(xreg_nor[, -5])
 MDSplot(xreg_nor.urf, xreg_nor[, 5])



#Support Vector Machines

```
xreg_nor.svm <- svm(xreg_nor.train[, 5] ~ ., data = xreg_nor.train[, 1:4])
xreg_nor.svm.p <- predict(xreg_nor.svm, xreg_nor.unknown[, 1:4])
table(xreg_nor.svm.p, xreg_nor.unknown[, 5])
```

```
##
## xreg_nor.svm.p  A  D  P  T
##           A  6  0  1  1
##           D  0  9  0  0
##           P  0  4 10  2
##           T  1  2  0  4
```

Наивный байесовский классификатор

Наивный байесовский метод является одним из самых простых, но наиболее мощных алгоритмов классификации, основанных на теореме Байеса с предположением независимости среди предикторов. Наивная байесовская модель проста в построении и особенно полезна для очень больших наборов данных.

```

#Формирование обучающей и тестовой последовательности
set.seed(4948493)
reg_sample<-sample(1:nrow(xreg_nor),size=nrow(xreg_nor)*.7)
reg_train<-xreg_nor[reg_sample,] #Выбор 70% строк
reg_test<- xreg_nor[-reg_sample,] #Выбор 30% строк

xreg_ftrain <-reg_train[,5]
xreg_ftest <-reg_test[,5]
reg_train <-reg_train[,-5]
reg_test <-reg_test[,-5]

#Наивный байесовский классификатор
m <- naiveBayes(reg_train,xreg_ftrain,laplace = 1)

#Прогноз кластеров регионов на тестовой последовательности
test_pred <- predict(m,reg_test)
test_pred

## [1] T P P P D P T A T A P D D D D A D D D T D D D T D
## Levels: A D P T

#Фактические кластеры регионов на тестовой последовательности
xreg_ftest

## [1] T P T P D P A T A A P P D D D A D D D T D D D D D
## Levels: A D P T

#Оценка эффективности алгоритма

CrossTable(test_pred, xreg_ftest,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('predicted', 'actual'))

```

```
##      Cell Contents
## |-----|
## |                N |
## |      N / Table Total |
## |-----|
##
##
## Total Observations in Table:  25
##
##
##      predicted | actual
##      predicted |      A      D      P      T | Row Total |
## -----|-----|-----|-----|-----|
##      A      |      2      0      0      1 |      3 |
##      |      0.080      0.000      0.000      0.040 |
## -----|-----|-----|-----|-----|
##      D      |      0     11      1      0 |     12 |
##      |      0.000      0.440      0.040      0.000 |
## -----|-----|-----|-----|-----|
##      P      |      0      0      4      1 |      5 |
##      |      0.000      0.000      0.160      0.040 |
## -----|-----|-----|-----|-----|
##      T      |      2      1      0      2 |      5 |
##      |      0.080      0.040      0.000      0.080 |
## -----|-----|-----|-----|-----|
## Column Total |      4     12      5      4 |     25 |
## -----|-----|-----|-----|-----|
##
##
```

```
table(predict(m, reg_test),xreg_ftest)
```

```
##      xreg_ftest
##      A  D  P  T
##      A  2  0  0  1
##      D  0 11  1  0
##      P  0  0  4  1
##      T  2  1  0  2
```

#Нейросетевой классификатор

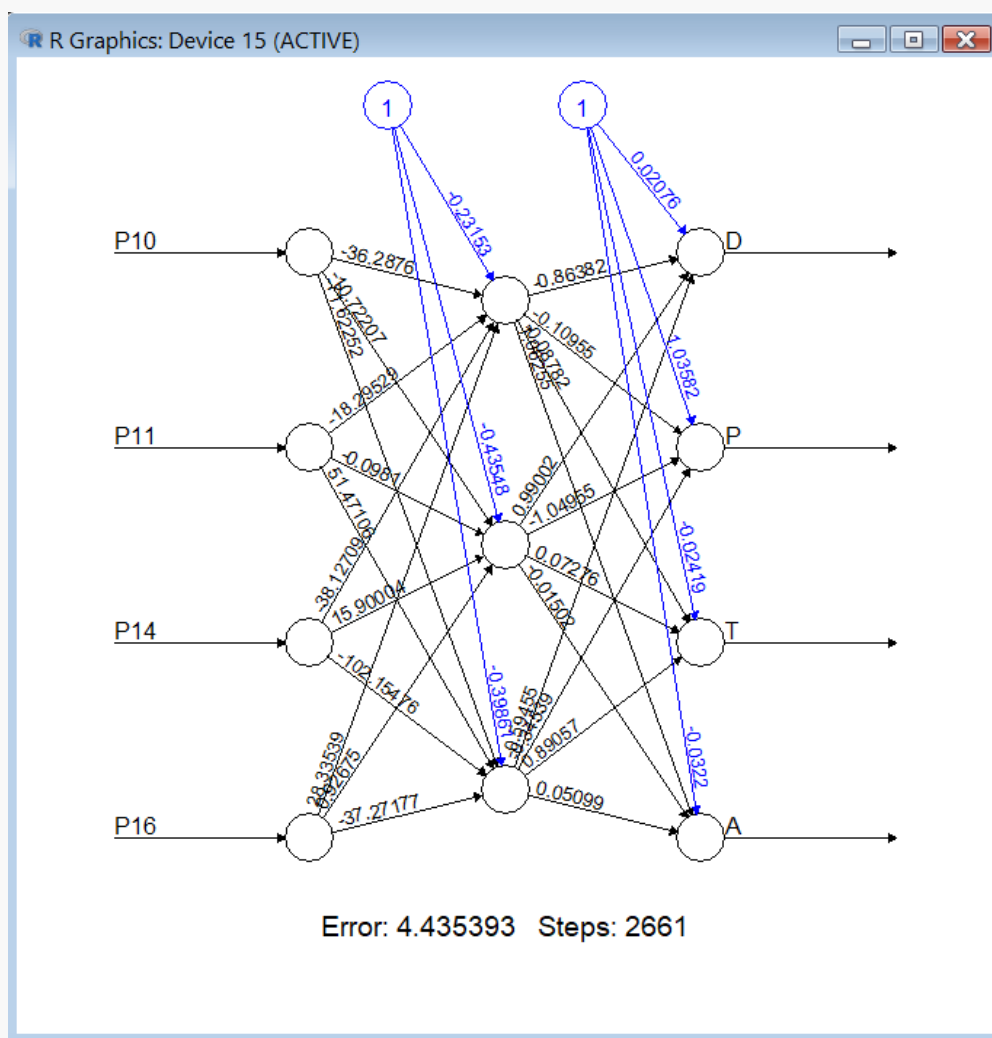
#Обучение нейронной сети

```
set.seed(1348)
```

```
net.reg <- neuralnet(xreg_ftrain ~ P10 + P11 + P14 + P16,reg_train, hidden = 3)
```

#net.reg\$result.matrix

#Визуализация обученной сети
`plot(net.reg)`



#Проверка сети на тестовой последовательности

```
net.regtest <- compute(net.reg, reg_test)$net.result
pred = c("A", "D", "P", "T")[apply(net.regtest, 1, which.max)]
(table(Факт = xreg_ftest, Прогноз = pred))
```

```
##      Прогноз
## Факт  A  D  P  T
##   A   3  0  0  1
##   D   0 11  1  0
##   P   0  2  3  0
##   T   1  0  0  3
```

```
Acc = mean(pred == xreg_ftest)
paste("Точность=", round(100*Acc, 2), "%", sep = "")
```

```
## [1] "Точность=80%"
```