Линейная регрессия в R

Варвара Кожухова

6 ноября 2018 г.

Задание

1. Загрузить набор данных для своего варианта, ознакомиться с его содержимым.
2. Построить график корреляционного поля для каждого фактора.
3. Построить уравнение парной линейной регрессии для каждого фактора.
4. Проверить значимость каждого из полученных уравнений регрессии. Показать уравнения регрессии с заданным в варианте доверительным интервалом на графиках.
5. Построить прогнозы по каждому из уравнений парной регрессии для заданных в варианте значений факторов.
6. Построить уравнение множественной линейной регрессии и получить корреляционную матрицу.
7. Построить прогноз по уравнению множественной регрессии для заданных в варианте значений факторов.

Варианты

№ варианта	1	2	3	4	5
Набор данных	airquality	state.x77	Cars93[1]	Cars93	Cars93
y	Ozone	Life Exp[2]	Price	Min.Price	Max.Price
Факторы и их значения для прогноза	Solar.R = 350, Wind = 8,3, Temp = 80	Illiteracy = 1,0, Murder = 12, Income = 4000	Horsepower = 200, RPM = 5200, Passengers = 4	Horsepower = 210, RPM = 5500, Passengers = 4	Horsepower = 220, RPM = 6000, Passengers = 6
№ варианта	6	7	8	9	10
Набор данных	stackloss	longley	longley	LifeCycleSavings	Anscombe[3]
y	stack.loss	GNP	Employed	sr	education
Факторы и их значения для прогноза	Air.Flow = 55, Water.Temp = 20, Acid.Conc = 89	Unemployed = 221, Armed.Forces = 180, Population = 125	GNP.deflator = 102, Armed.Forces = 170, Population = 110	pop15 = 35,5, pop75 = 1,5, dpi = 2500, ddpi = 2,15	income = 3200, young = 347,8, urban = 425

[1] необходимо подключить библиотеку MASS

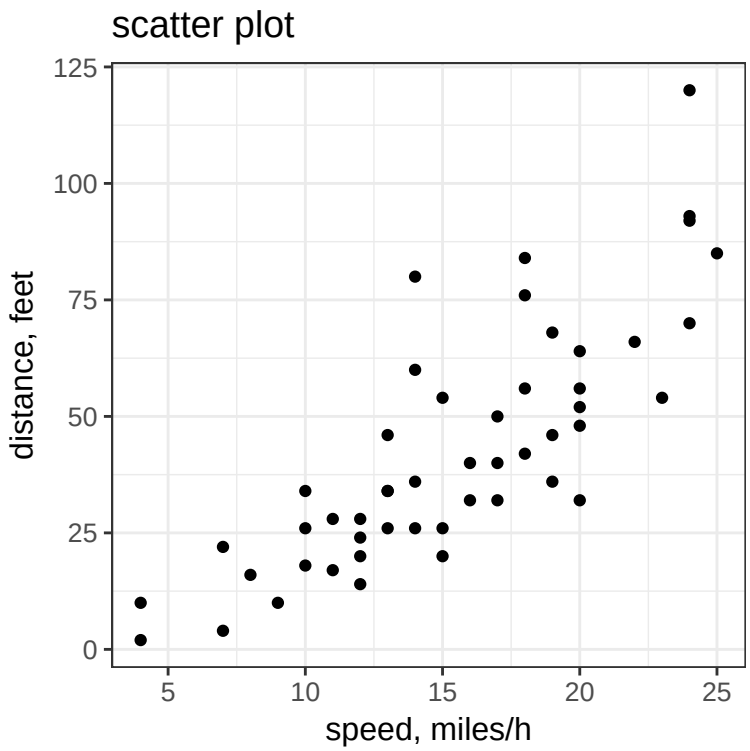
[2] поскольку в названии переменной нельзя использовать символ пробела, данную переменную необходимо переименовать командой colnames, например, в $Life\_Exp$, либо обращаться к ней по номеру столбца [,4].

[3] необходимо подключить библиотеку car

Парная линейная регрессия

Рассмотрим построение парной линейной регрессии на встроенном наборе данных cars. Будем рассматривать зависимость длины тормозного пути (переменная dist) от скорости (переменная speed). Построим график зависимости длины тормозного пути от скорости автомобиля.

```
d <- cars
qplot(data=d, speed, dist) +
  geom_point(aes(x=d$speed, y=d$dist), size = 1) + theme_bw(base_size = 12) +
  xlab("speed, miles/h") + ylab("distance, feet") +
  labs(title = "scatter plot")
```



Оценим модель линейной регрессии длины тормозного пути на скорость автомобиля. Для этого командой lm поместим в переменную model модель линейной регрессии, указав dist в качестве зависимой переменной, и через значок переменную speed в качестве регрессора.

Тип lm представляет собой список из 12 элементов. Посмотрим на коэффициенты уравнения линейной регрессии.

```
model <- lm(data=d, dist~speed) # базовый пакет stats
model$coefficients

## (Intercept)       speed
##  -17.579095    3.932409
```

(Intercept) – это константа в уравнении регрессии, speed – коэффициент регрессии. Таким образом, уравнение регрессии имеет вид:

$$dist_i^m = -17.579 + 3.9324 \cdot speed_i$$

Так же можно посмотреть значения вектора ошибок модели – разницу между реальной длиной тормозного пути dist и полученной по модели $dist_i^m$. Выведем первые 10 значений этого вектора с точностью две цифры после запятой. Более полный набор расчетов по модели можно получить командой summary.

```r
options(digits = 3)
model$residuals[1:10]

##     1     2     3     4     5     6     7     8     9    10
##  3.85 11.85 -5.95 12.05  2.12 -7.81 -3.74  4.26 12.26 -8.68

summary(model) # базовый пакет base

##
## Call:
## lm(formula = dist ~ speed, data = d)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -29.07  -9.53  -2.27   9.21  43.20
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.579      6.758   -2.60    0.012 *
## speed          3.932      0.416    9.46  1.5e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.4 on 48 degrees of freedom
## Multiple R-squared:  0.651,Adjusted R-squared:  0.644
## F-statistic: 89.6 on 1 and 48 DF,  p-value: 1.49e-12
```

Помимо коэффициентов регрессии, R выводит:

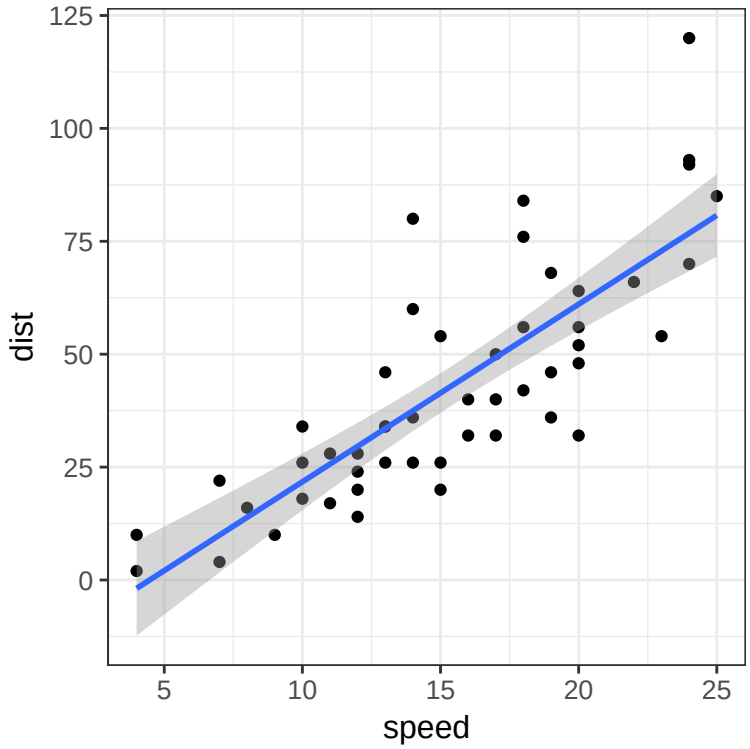
- стандартные ошибки коэффициентов (Std. Error);
- наблюдаемые значения t-критерия при проверке значимости коэффициентов регрессии (t value);
- P-значения для коэффициентов регрессии (P-value).

Звездочками или точками в столбце справа R показывает значимость или незначимость коэффициентов: *** – значимы на уровне значимости менее 0.001; ** – значимы на уровне значимости 0.001; * – значимы на уровне значимости 0.01; . – значимы на уровне значимости 0.05 и т.д. Эти обозначения приведены в разделе Signif. codes. Коэффициент детерминации (Multiple R-squared) равен 0.6511; скорректированный коэффициент детерминации (Adjusted R-squared) равен 0.6438. Наблюдаемое значения F-критерия проверки значимости уравнения в целом и P-значение:

F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

Таким образом, уравнение регрессии получилось значимым. Проведем на графике полученную линию регрессии с 95% доверительными интервалами.

```r
qplot(data = d, speed, dist) + stat_smooth(method="lm", level = 0.95) +
  theme_bw(base_size = 12)
```



Рассчитаем также 95% доверительные интервалы для параметров линейной регрессии. Полученные по модели значения можно вывести командой fitted.

```
confint(model, level = 0.95) # базовый пакет stats
```

```
##             2.5 % 97.5 %
## (Intercept) -31.2  -3.99
## speed         3.1   4.77
```

```
options(digits=4)
fitted(model) # базовый пакет stats
```

```
##      1      2      3      4      5      6      7      8      9     10     11     12
## -1.849 -1.849  9.948  9.948 13.880 17.813 21.745 21.745 21.745 25.677 25.677 29.610
##     13     14     15     16     17     18     19     20     21     22     23     24
## 29.610 29.610 29.610 33.542 33.542 33.542 33.542 37.475 37.475 37.475 37.475 41.407
##     25     26     27     28     29     30     31     32     33     34     35     36
## 41.407 41.407 45.339 45.339 49.272 49.272 49.272 53.204 53.204 53.204 53.204 57.137
##     37     38     39     40     41     42     43     44     45     46     47     48
## 57.137 57.137 61.069 61.069 61.069 61.069 61.069 68.934 72.866 76.799 76.799 76.799
##     49     50
## 76.799 80.731
```

Рассчитать необъясненную сумму квадратов отклонений и полную сумму квадратов можно, воспользовавшись функцией deviance и уже известными функциями sum и mean. Для того, чтобы построить прогноз по полученной модели, нужно задать значения регрессора и поместить их в новый data.frame.

```
RSS <- deviance(model) # базовый пакет stats
TSS <- sum((d$dist-mean(d$dist))^2)
RSS; TSS
```

```
## [1] 11354
## [1] 32539
```

```r
# создаем новый набор данных
nd <- data.frame(speed=c(40,60))
# Строим прогноз функцией predict
predict(model,nd)
```

```
##     1     2
## 139.7 218.4
```

Множественная линейная регрессия

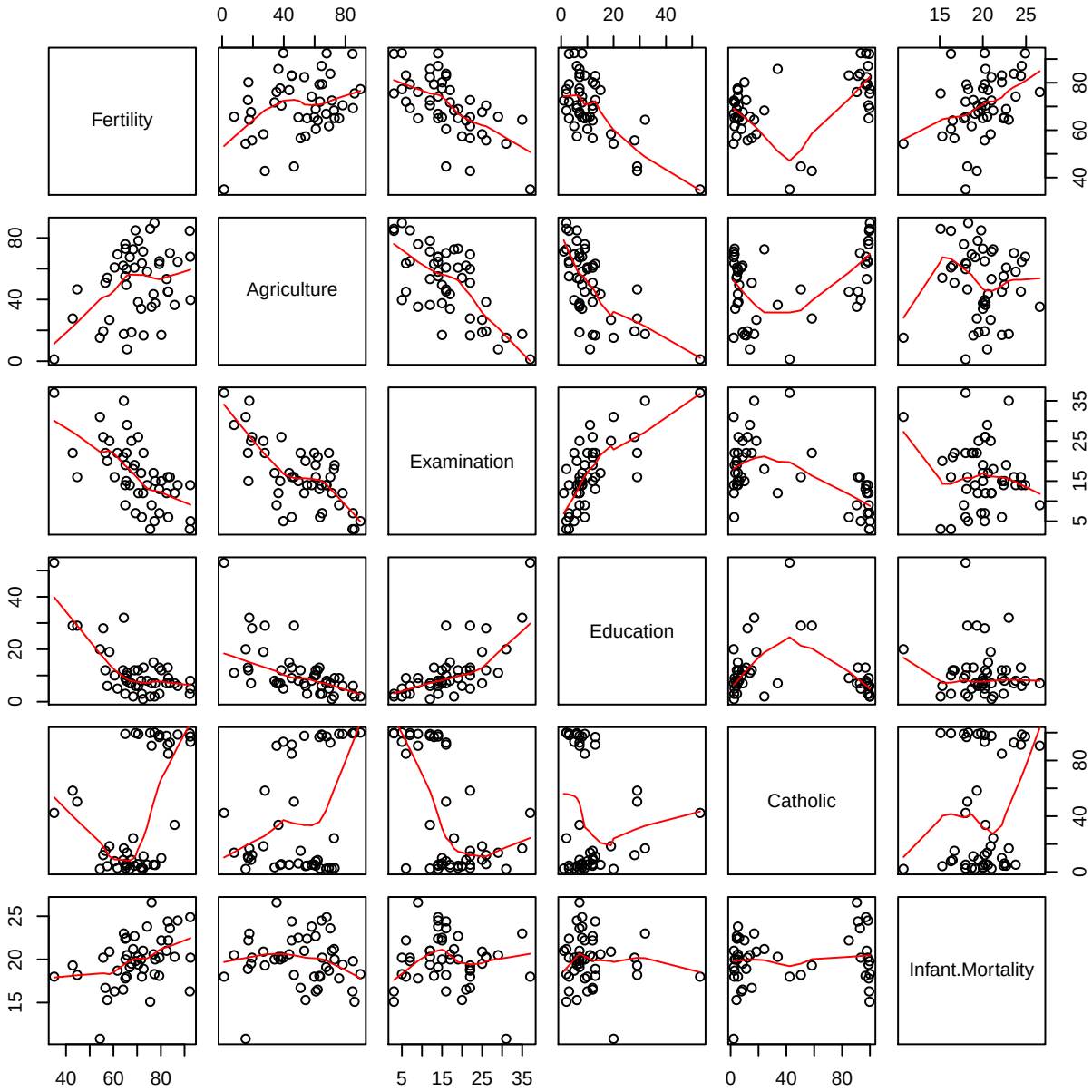
Рассмотрим встроенный набор данных по социально-экономическим показателям в 47 провинциях Швейцарии в 1888 г. Этот набор данных содержит 6 переменных по 47 наблюдений, каждая из которых измеряется в процентах (help(swiss)): Fertility – рождаемость; Agriculture – Examination – Education – Catholic – Infant.Mortality – Посмотрим на этот набор данных.

```r
t <- swiss # встроенный набор данных по Швейцарии
glimpse(t)
```

```
## Observations: 47
## Variables: 6
## $ Fertility        <dbl> 80.2, 83.1, 92.5, 85.8, 76.9, 76.1, 83.8, 92.4, 82.4, 82.9, ...
## $ Agriculture      <dbl> 17.0, 45.1, 39.7, 36.5, 43.5, 35.3, 70.2, 67.8, 53.3, 45.2, ...
## $ Examination      <int> 15, 6, 5, 12, 17, 9, 16, 14, 12, 16, 14, 21, 14, 19, 22, 18,...
## $ Education        <int> 12, 9, 5, 7, 15, 7, 7, 8, 7, 13, 6, 12, 7, 12, 5, 2, 8, 28, ...
## $ Catholic         <dbl> 9.96, 84.84, 93.40, 33.77, 5.16, 90.57, 92.85, 97.16, 97.67,...
## $ Infant.Mortality <dbl> 22.2, 22.2, 20.2, 20.3, 20.6, 26.6, 23.6, 24.9, 21.0, 24.4, ...
```

Встроенный пакет graphics содержит функцию pairs, позволяющую получить все возможные диаграммы рассеяния на одном графике, а также выполнить их сглаживание с помощью опции panel.smooth:

```r
pairs(swiss, panel = panel.smooth)
```



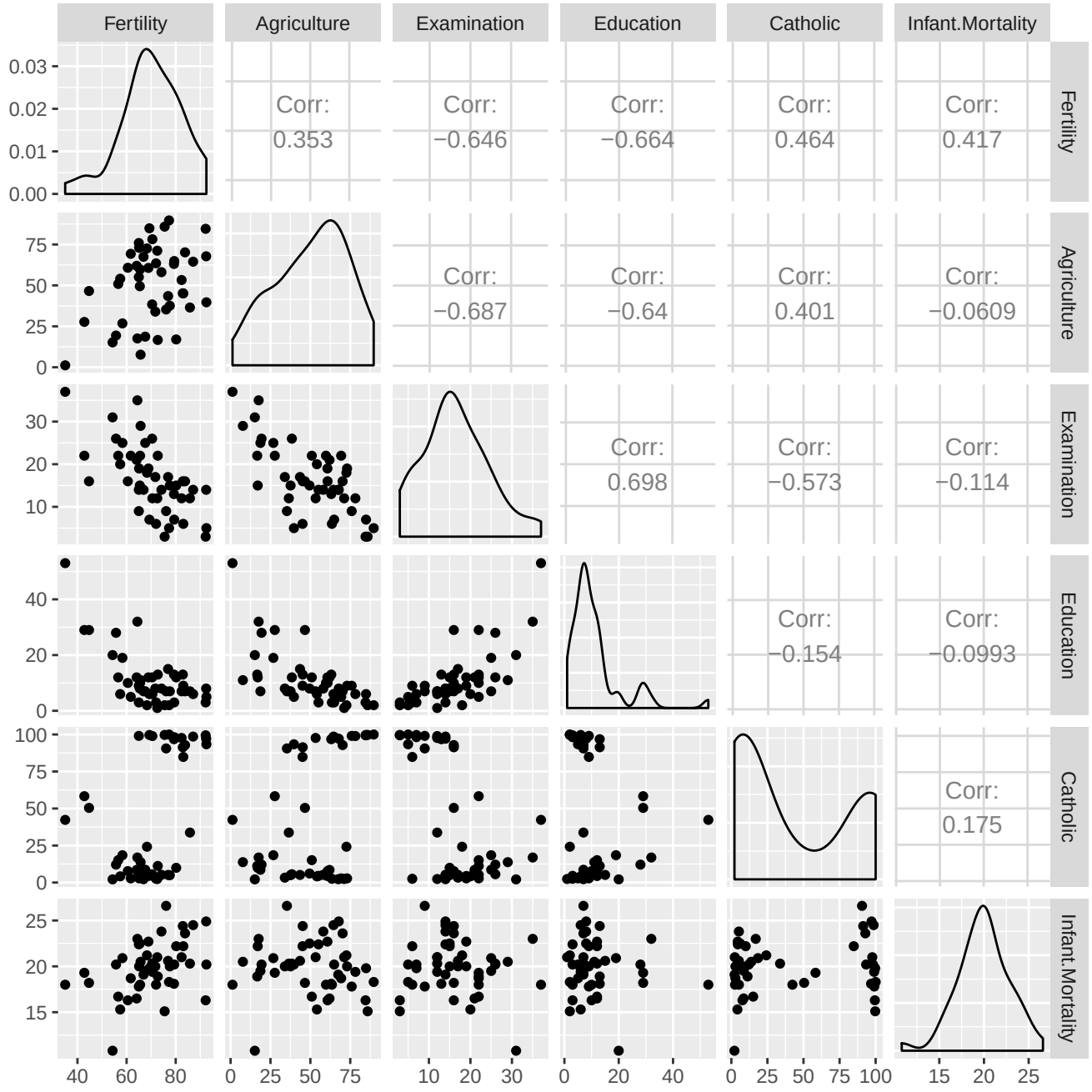
Функция cor позволяет как вычислить корреляцию между двумя выборками, так и получить корреляционную матрицу для всех переменных из набора данных.

```
options(digits = 5)
cor(swiss)
```

```
##                  Fertility Agriculture Examination Education Catholic Infant.Mortality
## Fertility          1.00000    0.353079    -0.64588 -0.663789  0.46368         0.416556
## Agriculture        0.35308    1.000000    -0.68654 -0.639523  0.40110        -0.060859
## Examination       -0.64588   -0.686542     1.00000  0.698415 -0.57274        -0.114022
## Education         -0.66379   -0.639523     0.69842  1.000000 -0.15386        -0.099322
## Catholic           0.46368    0.401095    -0.57274 -0.153859  1.00000         0.175496
## Infant.Mortality   0.41656   -0.060859    -0.11402 -0.099322  0.17550         1.000000
```

Существует еще одна функция, позволяющая получить корреляционную матрицу, диаграммы рассеяния и сглаженные распределения одновременно.

```
library("GGally")
ggpairs(t) # функция из пакета GGally
```



Чтобы оценить регрессию рождаемости на остальные переменные, можно воспользоваться уже знакомой функцией lm, а регрессоры перечислить через знак «плюс»:

```
model2 <- lm(data=t, Fertility~Agriculture+Education+Catholic)
```

В данном случае регрессорами стали % занятых в с/х, % католического населения и % имеющих образование выше начального.

Получить оценки коэффициентов уравнения регрессии, а также проверить основные гипотезы поможет функция summary:

```
summary(model2)

##
## Call:
## lm(formula = Fertility ~ Agriculture + Education + Catholic,
```

```
##     data = t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.178  -6.548   1.379   5.822  14.840
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 86.22502    4.73472  18.211  < 2e-16 ***
## Agriculture -0.20304    0.07115  -2.854  0.00662 **
## Education   -1.07215    0.15580  -6.881 1.91e-08 ***
## Catholic     0.14520    0.03015   4.817 1.84e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.728 on 43 degrees of freedom
## Multiple R-squared:  0.6423,Adjusted R-squared:  0.6173
## F-statistic: 25.73 on 3 and 43 DF,  p-value: 1.089e-09
```

Построим прогноз по аналогии с парной линейной регрессией. Отличие заключается лишь в том, что в наборе данных необходимо указать значения каждого фактора.

```
# создаем новый набор данных
nd2 <- data.frame(Agriculture=0.5, Catholic=0.5, Education=20)
predict(model2,nd2)

##    1
## 64.8
```

Построение прогноза по нескольким точкам выполняется с помощью векторов значений.

```
# создаем новый набор данных
nd2 <- data.frame(Agriculture=c(0.5,0.8), Catholic=c(0.5, 0.65), Education=c(20, 25))
# прогнозируем
predict(model2,nd2)

##    1    2
## 64.8 59.4
```