Введение в регрессионный анализ: множественная регрессия

Линейные модели на R, осень 2014

Вадим Хайтов, Марина Варфоломеева
Каф. Зоологии беспозвоночных, СПбГУ

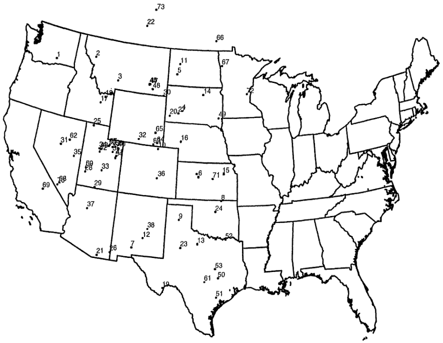
Мы рассмотрим

- Технику подгонки множественных регрессионных моделей
- Технику валидизации множественных регрессионных моделей

Вы сможете

- Подобрать множественную линейную модель
- Протестировать ее состоятельность и валидность
- Дать трактовку результатам

Рабочий пример: Какие факторы определяют распределение функциональных групп растений?



(Пример взят из книги Quinn&Keugh,2002; Оригинальная работа: Paruelo & Lauenroth, 1996)

Считается, что распределение C3 растений регулируется только температуой той местности, где они произрастают. Проверяется гипотеза о связи распределения C3 растений не только со среднегодовой температурой, но и с уровнем осадков

Зависимая перменная

C3 - отностельное обилие травянистых растений, демонстрирующих C_3 путь фотосинтеза

Предикторы

LAT - штрота

LONG - долгота

MAP - среднегодовое количество осадков (мм)

MAT - среднегодвая температура (градусы)

JJAMAP - доля осадков, выпадающих в летние месяцы

DJFMAP - доля осадкв, выпадающих в зимние месяцы

Читаем данные

```r
plant <- read.csv("paruelo.csv")
plant <- plant[,-2]
head(plant)
```

```
##     C3 MAP  MAT JJAMAP DJFMAP   LONG   LAT
## 1 0.65 199 12.4   0.12   0.45 119.55 46.40
## 2 0.65 469  7.5   0.24   0.29 114.27 47.32
## 3 0.76 536  7.2   0.24   0.20 110.78 45.78
## 4 0.75 476  8.2   0.35   0.15 101.87 43.95
## 5 0.33 484  4.8   0.40   0.14 102.82 46.90
## 6 0.03 623 12.0   0.40   0.11  99.38 38.87
```

Можно ли ответить на вопрос таким методом?

```
cor(plant)
```

```
##              C3      MAP       MAT   JJAMAP    DJFMAP     LONG      LAT
## C3      1.00000 -0.06242 -0.511388  0.02301 -0.069231  0.04153  0.66698
## MAP    -0.06242  1.00000  0.355091  0.11226 -0.404512 -0.73369 -0.24651
## MAT    -0.51139  0.35509  1.000000 -0.08077  0.001478 -0.21311 -0.83859
## JJAMAP  0.02301  0.11226 -0.080771  1.00000 -0.791540 -0.49156  0.07417
## DJFMAP -0.06923 -0.40451  0.001478 -0.79154  1.000000  0.77074 -0.06512
## LONG    0.04153 -0.73369 -0.213109 -0.49156  0.770744  1.00000  0.09655
## LAT     0.66698 -0.24651 -0.838590  0.07417 -0.065125  0.09655  1.00000
```

Проблема 1. Взаимосвязь между переменными может находиться под контролем других переменых (частная корреляция).

Пробема 2. Множественные сравнения.

Необходимо учесть все взаимовлияния в одном анализе

Нам предстоит построить множественную регрессионную модель

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \ldots + \beta_p x_{i,p} + \epsilon_i$$

y_i - значение зависимой переменной Y при значении предикторов $X_1 = x_{i,1}$, $X_2 = x_{i,2}$ и т.д.

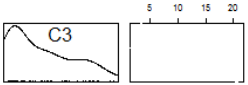
β_0 - свободный член (intercept). Значение Y при $X_1 = X_2 = X_3 = \ldots = X_p = 0$
β_1 - частный угловой коэффициент для зависимости Y от X_1. Показывает насколько единиц изменяется Y при изменении X_1 на одну единицу, при условии, что все остальные предикторы не изменяются.
β_2, β_3, ..., β_p - аналогично

ϵ_i - варьирование Y, не объясняемое данной моделью.

Геометрически, это плоскость в многомерном пространстве

Проводим исследование данных

```
library(car)
scatterplotMatrix(plant[,-2], spread=FALSE)
```



Явные проблемы

1. Распределение зависимой переменной `C3` очень асимметрично
2. Есть сильные корреляции между некоторым предикторами.
3. Возможна пространственная автокоррелированность.

Постоим линейную модель и сразу проверим ее на наличие автокорреляции остатков

Задание. Напишите самостоятельно R код, необходимый для подбора уравнения множественной регрессии и сразу проверьте модель на наличие автокорреляции остатков

Hint 1. Для того, чтобы видеть названия переменных воспользуйтесь функцией `names()`

Hint 2. Подумайте какие предикторы не следует включать в модель в соответствии с гипотезой, поставленной в исследовании.

Hint 3. Проведите тест Дарбина-Уотсона.

Решение

```
model0 <- lm(C3 ~  MAP + MAT + JJAMAP + DJFMAP, data = plant)
durbinWatsonTest (model0, max.lag = 3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1         0.32837         1.251   0.002
##    2         0.19521         1.486   0.034
##    3         0.05101         1.728   0.392
##  Alternative hypothesis: rho[lag] != 0
```

Наличие положиетльных автокорреляций повашает вероятность ошибки I рода!

Возможное решение - нарушить "градиентный" характер материала.

Разделим выборку на две части.

```
plant1 <- plant[order(plant$LAT), ] # Упорядочиваем описания в соответствии с широтой

include <- seq(1, 73, 2) # Отбираем каждое второе описание
exclude <- seq(1, 73) [!(seq(1, 73) %in% include)] # Исключаем из списка отобранные описания

plant_modelling <- plant1[include, ]
plant_testing <- plant1[exclude, ]
```

Строим линейную модель для сокращенного набора данных

```
model1 <- lm((C3)^(1/4) ~  MAP + MAT + JJAMAP + DJFMAP, data = plant_modelling)

# Аналогичная запись
model1 <- lm((C3)^(1/4) ~  .-LONG -LAT, data = plant_modelling)
```

Проверим на автокоррелированность остатков полученную модель

```
durbinWatsonTest(model1, max.lag = 3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       -0.002602         1.967   0.676
##    2        0.076350         1.792   0.464
##    3       -0.137085         2.208   0.468
##  Alternative hypothesis: rho[lag] != 0
```

Смотрим на полученную модель

```
summary(model1)
```

```
##
## Call:
## lm(formula = (C3)^(1/4) ~ . - LONG - LAT, data = plant_modelling)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6828 -0.1460  0.0434  0.1475  0.3863
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.708659   0.444180    3.85  0.00054 ***
## MAP          0.000216   0.000229    0.94  0.35280
## MAT         -0.029890   0.008827   -3.39  0.00189 **
## JJAMAP      -1.743669   0.663465   -2.63  0.01308 *
## DJFMAP      -1.840973   0.905072   -2.03  0.05030 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.231 on 32 degrees of freedom
```

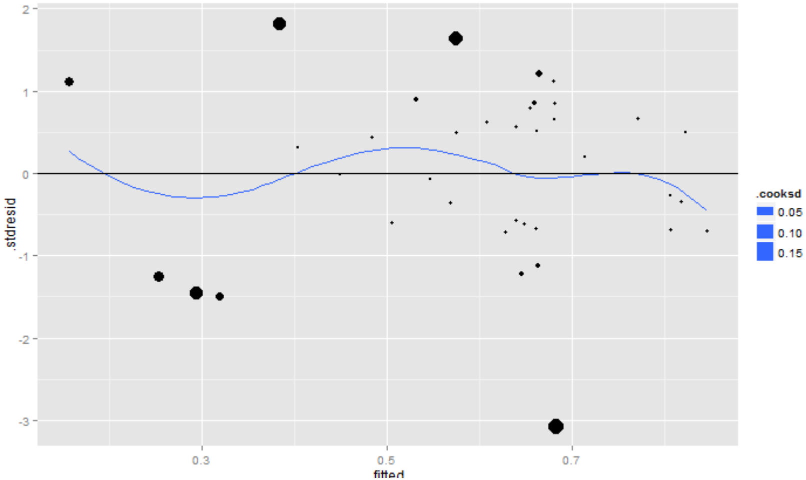
Проверка валидности модели

```
library(ggplot2)
c3_diag <- fortify(model1)
```

Смотрим на residual plot

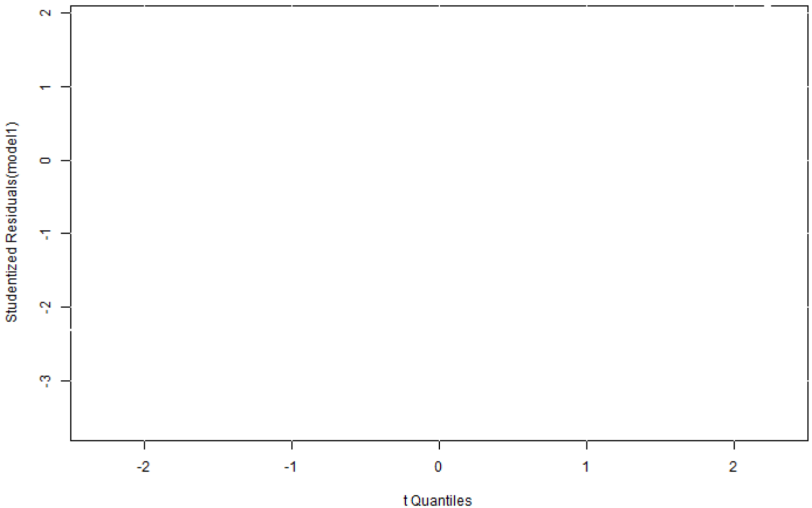
```
pl_resid <- ggplot(c3_diag, aes(x = .fitted, y = .stdresid, size = .cooksd)) +
  geom_point() +
  geom_smooth(se=FALSE) +
  geom_hline(eintercept=0)

pl_resid
```



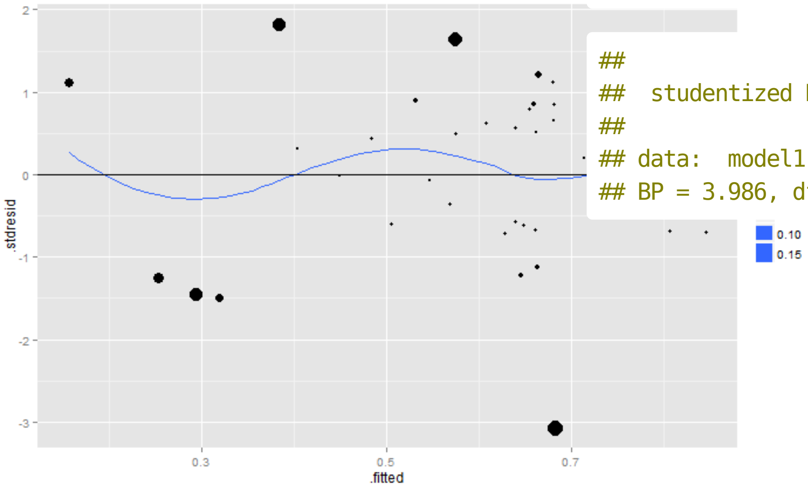
Проверяем на нормальность

```
qqPlot(model1)
```



Проверяем на гетероскедастичность

```
pl_resid
```




```r
library(lmtest)
bptest(model1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 3.986, df = 4, p-value = 0.4079
```

Проверяем на мультиколлинеарность

Мультиколлинеарность - наличие линейной зависимости между независимыми переменными (факторами) регрессионной модели.

При наличии мультиколлинеарности оценки параметров получаются неточными, а значит сложно будет дать интерпретацию влияния тех или иных факторов на объясняемую переменную

Признаки мультиколлинеарности:

- Большие ошибки оценок параметров
- Большинство оценок параметров модели недостоверно, но F критерий всей модели свидетельствует о ее стаистической значимости

Фактор инфляции дисперсии (Variance inflation factor)

```
vif(model1)
```

```
##    MAP    MAT JJAMAP DJFMAP
##  1.809  1.280  3.064  3.758
```

Логика вычисления VIF

1. Строим регрессионную модель

$$x_1 = c_0 + c_2 x_2 + c_2 x_3 + \dots . + c_p x_p$$

1. Находим R^2 для данной модели
2. $VIF = \frac{1}{1-R^2}$

Что делать если мультиколлинеарность выявлена?

Решение № 1. Удалить из модели избыточные предикторы

1. Удалить из модели предикторы с VIF > 5
2. Вновь провести вычисление VIF
3. Возможно, удалить предикторы с VIF > 3
4. Иногда полезно удалить и предикторы с VIF > 2 (Это позволит сократить набор предикторов, но не увлекайтесь!)

Что делать если мультиколлинеарность выявлена?

Решение № 2. Заменить исходные предикторы новыми пермеными, полученными с помощью метода главных компонент

Удалим из модели избыточный предиктор

```
model2 <- update(model1, ~ . -DJFMAP)
vif(model2)
```

```
##    MAP    MAT JJAMAP
##  1.287  1.279  1.030
```

Смотрим на итоги

```
summary(model2)
```

```
##
## Call:
## lm(formula = (C3)^(1/4) ~ MAP + MAT + JJAMAP, data = plant_modelling)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6863 -0.1020  0.0583  0.1695  0.4101
##
## Coefficients:
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.858711   0.157635    5.45 0.0000049 ***
## MAP          0.000466   0.000202    2.31    0.0275 *
## MAT         -0.030486   0.009232   -3.30    0.0023 **
## JJAMAP      -0.643995   0.402453   -1.60    0.1191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.242 on 33 degrees of freedom
## Multiple R-squared:  0.28,   Adjusted R-squared:  0.215
```

Какой из факторов MAT, JJAMAP или DJFMAP оказывает наиболее сильное влияние?

Для этого надо "уравнять" шкалы, всех предикторов, то есть

стандартизировать их

Задание.

Нпишите R-код, который позволяет стандартизировать шкалу предиктора.

Стандаризизируйе, например, вектор `MAT`

Решение

```
MAT_stand <- (plant_modelling$MAT - mean(plant_modelling$MAT))/sd(plant_modelling$MAT)
```

Можно использовать функцию `scale()`

```
model2_scaled <- lm((C3)^(1/4) ~ scale(MAP) + scale(MAT) + scale(JJAMAP), data = plant_modell
```

Какой фактор оказывает наиболее сильное влияние на долю C3-растенний?

```
summary(model2_scaled)
```

```
##
## Call:
## lm(formula = (C3)^(1/4) ~ scale(MAP) + scale(MAT) + scale(JJAMAP),
##     data = plant_modelling)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6863 -0.1020  0.0583  0.1695  0.4101
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.5979     0.0398   15.03  2.5e-16 ***
## scale(MAP)      0.1055     0.0457    2.31   0.0275 *
## scale(MAT)     -0.1506     0.0456   -3.30   0.0023 **
## scale(JJAMAP)  -0.0655     0.0409   -1.60   0.1191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.243 on 33 degrees of freedom
```

Если модель хорошая, то она должна хорошо предсказывать

Мы рассмотрим самый простой случай кросс-валидации

```
predicted_C3_model1 <- predict(model1, newdata=plant_testing)
cor(predicted_C3_model1, plant_testing$C3)
```

```
predicted_C3_model2 <- predict(model2, newdata=plant_testing)
cor(predicted_C3_model2, plant_testing$C3)
```

```
## [1] 0.398
## [1] 0.4067
```

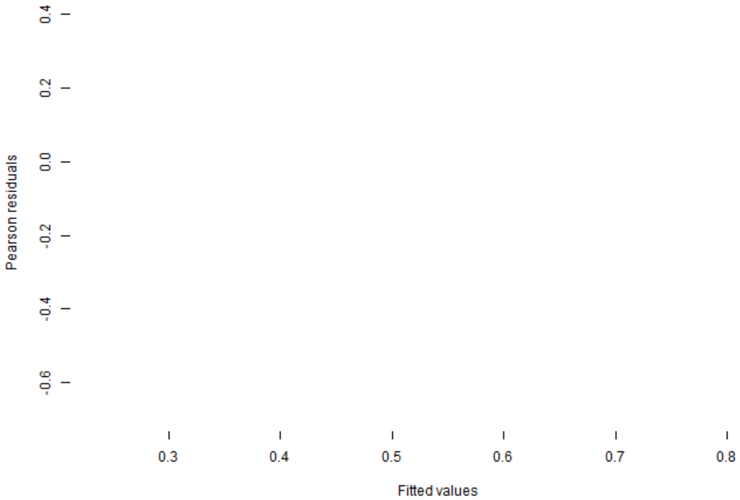
Оцениваем валидность финальной модели

```
durbinWatsonTest(model2)
bptest(model2)
vif(model2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1          -0.075         2.116   0.972
##  Alternative hypothesis: rho != 0
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 2.145, df = 3, p-value = 0.5428
##
##    MAP    MAT JJAMAP
##  1.287  1.279  1.030
```

Оцениваем валидность финальной модели

```
residualPlot(model2)
```



Summary

- При построении множественной регрессии важно, помимо проверки прочих условий применимости, проверить модель на наличие мультиколлинеарности
- Если модель построена на основе стандартизированнх значений предикторов, то можно сравнивать влияние этих предикторов.
- Кросс-валидация позволяет оценить степень работоспособности модели.

Что почитать

- Кабаков Р.И. R в действии. Анализ и визуализация данных на языке R. М.: ДМК Пресс, 2014.
- Quinn G.P., Keough M.J. (2002) Experimental design and data analysis for biologists, pp. 92-98, 111-130
- Diez D. M., Barr C. D., Cetinkaya-Rundel M. (2014) Open Intro to Statistics., pp. 354-367.
- Logan M. (2010) Biostatistical Design and Analysis Using R. A Practical Guide, pp. 170-173, 208-211