Линейный регрессионный анализ

Теорема Гаусса-Маркова. Схема регрессионного анализа

Для того чтобы полученные по МНК оценки обладали некоторым полезными статистическими обладали некоторыми полезными статистическими свойствами, необходимо выполнение ряда условий относительно оцениваемой модели, называемых условиями Гаусса-Маркова.

Теорема Гаусса-Маркова

Рассматривается модель парной регрессии, в которой наблюдения Y связаны с X следующей зависимостью:

$$Y_i = b_0 + b_1 X_i + \varepsilon_i.$$

На основе n наблюдений оценивается уравнение регрессии

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_i.$$

Тогда, если данные обладают следующими свойствами:

1. Модель данных правильно специфицирована;
2. Все X_i детерминированы и не все равны между собой;
3. Ошибки не носят систематического характера, то есть $\mathbb{E}(\varepsilon_i) = 0 \ \forall i$;
4. Дисперсия ошибок одинакова и равна некоторой σ^2;
5. Ошибки некоррелированы, то есть $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0 \ \forall i, j$;

— то оценки полученные методом наименьших квадратов оптимальны в классе линейных несмещённых оценок.

Пояснения

- Эффективность оценки означает, что она обладает наименьшей дисперсией.
- Оценка линейна по наблюдениям Y.
- Несмещённость оценки означает, что её математическое ожидание равно истинному значению.

Условия Гаусса-Маркова 1

Спецификация модели

$$y_i = b_0 \cdot x_{i0} + b_1 \cdot x_{i1} + b_2 \cdot x_{i2} + \ldots + b_k \cdot x_{ik} + \varepsilon_i, \quad x_{i0} = 1, \quad i = 1, 2, \ldots, n.$$

1. Если модель линейна по параметрам b_i, то спецификация корректна.
2. Отсутствие коллинеарности.

Условия Гаусса-Маркова 2

Все X_i детерминированы и не все равны между собой.

Если все X_i равны между собой, то $X_i = \bar{X}$, и в уравнении оценки коэффициента наклона прямой в линейной модели в знаменателе будет ноль, из-за чего будет невозможно оценить коэффициенты b_i.

Условия Гаусса-Маркова 3

Ошибки не носят систематического характера:
$\mathbb{E}(\varepsilon_i) = 0 \ \forall i$

Случайный член может быть иногда положительным, иногда отрицательным, но он не должен иметь систематического смещения ни в каком из двух возможных направлений.

Если уравнение регрессии включает постоянный член b_0, то это условие выполняется автоматически, так как постоянный член отражает любую систематическую постоянную составляющую в Y, которую не учитывают объясняющие переменные, входящие в уравнение регрессии.

Условия Гаусса-Маркова 4

Дисперсия ошибок одинакова.

Одинаковость дисперсии ошибок принято называть **гомоскедастичностью**.

Если это условие не выполняется, то коэффициенты регрессии, найденные по методу наименьших квадратов, будут неэффективны

Более эффективные результаты будут получаться путём применения модифицированных методов оценивания (взвешенный МНК или оценка ковариационной матрицы по формуле Уайта или Дэвидсона—Маккинона).

Условия Гаусса-Маркова 5

Ошибки ε_i распределены независимо от ε_j при $i \neq j$.

Это условие предполагает отсутствие систематической связи между значениями случайного члена в любых двух наблюдениях. Если один случайный член велик и положителен в одном направлении, не должно быть систематической тенденции к тому, что он будет таким же великим и положительным (то же можно сказать и о малых, и об отрицательных остатках).

Обычно не выполняется в случае временных рядов!

При невыполнении этого условия оценки, полученные по методу наименьших квадратов, будут неэффективны.

Простая линейная регрессия

```
x <- c(10,20,30,40,50)
y <- c(4,22,44,60,82)
```

На основе $n = 5$ наблюдений строим уравнение регрессии

$$\hat{Y} = \hat{b}_0 + \hat{b}_1 X.$$

которое является оценкой для

$$Y = b_0 + b_1 X + \varepsilon.$$

- X — предиктор
- Y — отклик
- b_0, b_1 — коэффициенты регрессии

$\varepsilon = Y - \hat{Y}$ — ошибка (остаток, погрешность, невязка) равна разности между наблюдаемым и прогнозируемым значениями отклика.

`Residual = Observed - Predicted`

Этапы линейной регрессии в R

1. Собрать (получить) выборку числовых данных о связи предикторов с откликом.
2. Записать формулу, предположительно связывающую предикторы и отклик. Поместить эту формулу в функцию `lm()`.
3. Проанализировать качество регрессионной модели при помощи `summary()`.
4. По результатам выполнения `lm()` найти коэффициенты регрессии. Записать с их помощью уравнение регрессии $Y = b_0 + b_1 X$.
5. Для прогнозирования новых значений отклика использовать функцию `predict()`. Определить доверительный интервал прогноза.

lm() — функция для подгонки линейных моделей

```
lm(Y ~ model, data)
```

Формула:

```
Y ~ op1 X1 op2 X2 op3 X3 ...
```

представляет собой объект класса `formula`.

`data` — таблица данных

Функция `lm` возвращает линейную регрессионную модель.

Формулы

“Линейная модель” означает линейность относительно коэффициентов регрессии b_i

- $y = b_0 + b_1 x + b_2 x^2$ — линейная модель.
- $y = b_0 x^{b_1}$ — нелинейная модель.

Примеры

- `Y ~ A` — прямая со свободным членом b_0, заданным неявно

$$Y = b_0 + b_1 A$$

- `Y ~ -1 + A` — прямая без свободного члена; будет проходить через (0,0)

$$Y = b_1 A$$

- `Y ~ A + I(A^2)` — полином; внутри функции `I()` можно задавать операции, которые трактуются в обычном для математики смысле

$$Y = b_0 + b_1 A + b_2 A^2$$

Примеры формул

- `Y ~ A + B` — модель 1-го порядка, в которой A и B не взаимодействуют

$$Y = b_0 + b_1 A + b_2 B$$

- `Y ~ A:B` — модель, содержащая только взаимодействие между A и B (1-го порядка)

$$Y = b_0 + b_1 AB$$

- `Y ~ A*B` — полная модель 1-го порядка, учитывающая как A и B, так и взаимодействие между ними (эквивалент: `Y ~ A + B + A:B`)

$$Y = b_0 + b_1 A + b_2 B + b_3 AB$$

- `Y ~ (A + B + C)^2` — модель включает все эффекты 1-го порядка и все взаимодействия вплоть до n-го порядка, где n задается показателем в `(...)^n` (эквивалент: `Y ~ A*B*C - A:B:C`)

$$Y = b_0 + b_1 A + b_2 B + b_3 C + b_4 AB + b_5 AC + b_6 AC$$

```r
lm.y <- lm(y ~ x)
summary(lm.y)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    1    2    3    4    5
##  0.4 -1.0  1.6 -1 ^  ^ ^
##
## Coefficients:
##                Esti                    'r(>|t|)
## (Intercept) -15.8(                      0.0025 **
## x             1.9                      3.84e-05 ***
## ---
## Signif. codes:  0                      *' 0.05 '.' 0.1 ' '
##
## Residual standard error: 1.592 on 3 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.9973
## F-statistic:  1486 on 1 and 3 DF,  p-value: 3.842e-05
```

summary()

- `Residuals` — остатки.
- Оценки коэффициентов модели (Estimate), их стандартные отклонения (Std Error), t-значения и вероятности нулевой гипотезы, что коэффициент равен нулю.
- стандартное отклонение регрессии (Residual standard error).
- коэффициенты детерминации.
- результаты F-теста нулевой гипотезы об одновременном равенстве нулю всех коэффициентов регрессионной модели.

Полезные функции

```
coef(lm.y)   # коэффициенты модели
```

```
## (Intercept)           x
##       -15.80        1.94
```

```
resid(lm.y)  # остатки
```

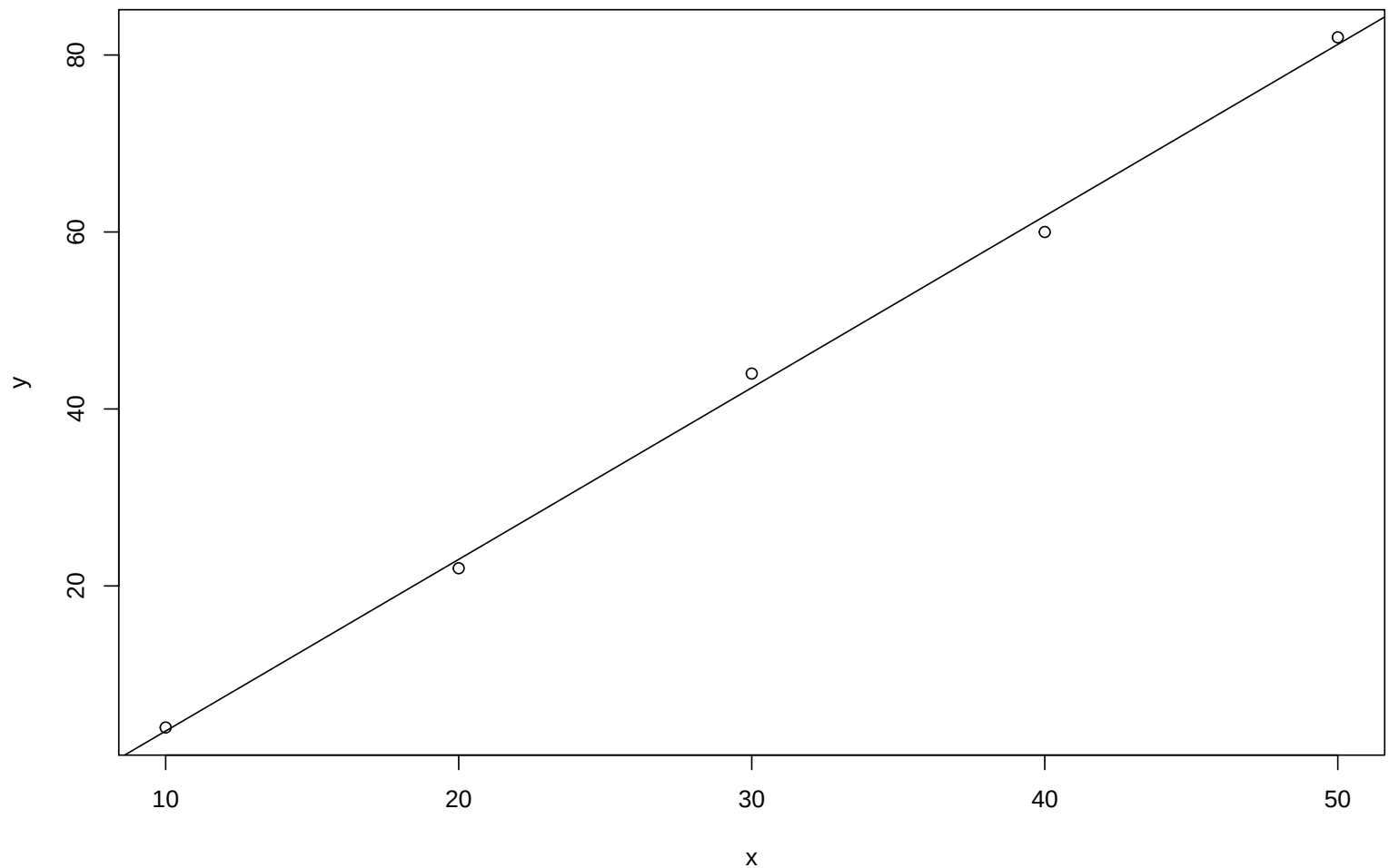
```
##   1   2   3   4   5
##  0.4 -1.0  1.6 -1.8  0.8
```

```
fitted(lm.y) # подогнанные значения Y
```

```
##    1    2    3    4    5
##  3.6 23.0 42.4 61.8 81.2
```

```
yh = -15.8 + 1.94*xh
```

```
plot(x, y)
abline(lm.y)
```



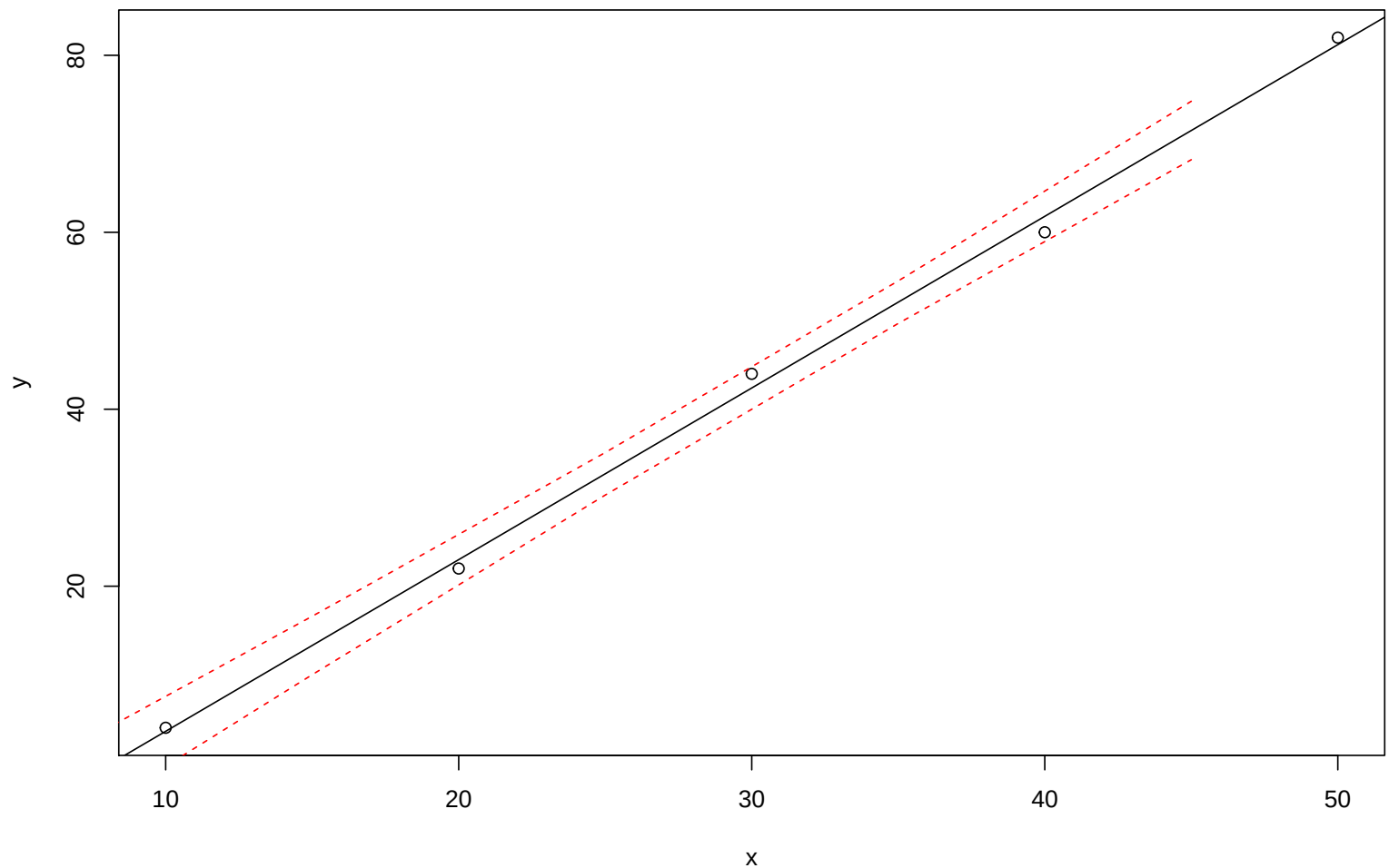
Прогнозирование и доверительный интервал

```
predict(model, data.frame(новые данные),
                level = 0.95, interval = "confidence")
```

```
x.new = c(5,15,25,35,45)
preds <- predict(lm.y, data.frame(x = x.new),
                 level = 0.95, interval = "confidence")
```

```
##    fit         lwr       upr
## 1 -6.1 -10.700809 -1.499191
## 2 13.3   9.997813 16.602187
## 3 32.7  30.297306 35.102694
## 4 52.1  49.697306 54.502694
## 5 71.5  68.197813 74.802187
```

```r
plot(x, y)
abline(lm.y)
lines(x.new, preds[ ,3], lty = 'dashed', col = 'red')
lines(x.new, preds[ ,2], lty = 'dashed', col = 'red')
```

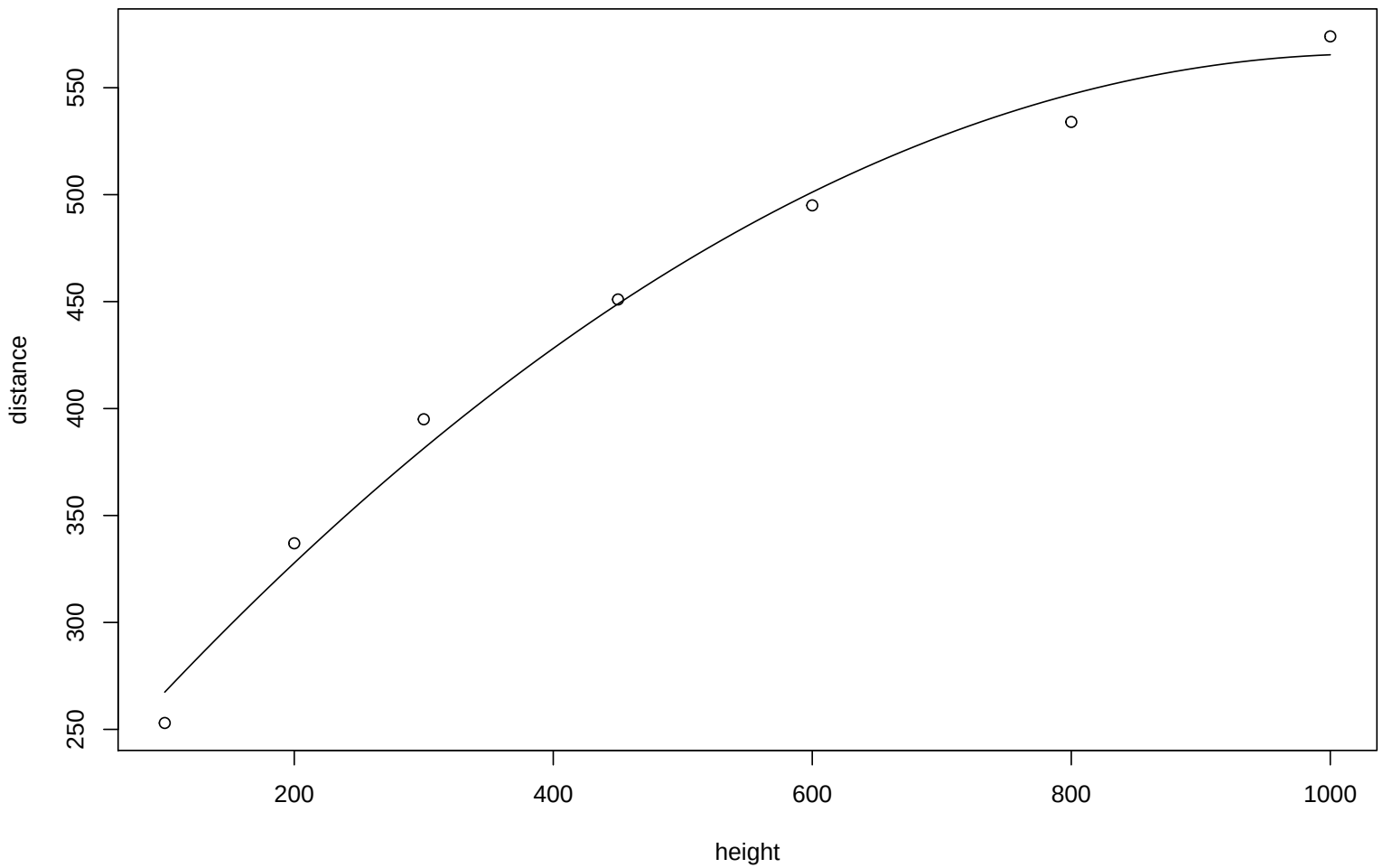


Квадратичная аппроксимация

```
# Данные Галилея
height = c(100, 200, 300, 450, 600, 800, 1000)
distance = c(253, 337, 395, 451, 495, 534, 574)
# Модель в форме квадратичного полинома
lm.r = lm(distance ~ height + I(height^2))
# Создадим высоты для прогноза
newh = seq(100, 1000, 10)
# Вычислим расстояния для каждой из новых высот
fit = 200.211950 + 0.706182*newh - 0.000341*newh^2
```

Квадратичная аппроксимация: график

```
plot(height, distance)  # исходные данные
lines(newh, fit, lty=1) # показать результаты подгонки
```



Дополнительные материалы

- R Tutorial. Multiple Linear Regression
- Ramsey F., Schafer D. Statistical Sleuth, 3rd edition, 2013). Пакет Sleuth3 содержит данные Галилея.