ЛАБОРАТОРНАЯ РАБОТА № 1.
Основы анализа данных в статистическом пакете RStudio

Цель работы – получить навыки анализа и обработки данных в RStudio.

Задачи

1. Загрузить данные из файла.
2. Провести предобработку данных.
3. Провести первичный анализ данных.
4. Выявить зависимости между наборами данных. Выполнить:
   - корреляционный анализ;
   - регрессионный анализ;
   - дисперсионный анализ.
5. Представить результаты анализа в табличном и графическом виде.
6. Подготовить отчет по выполненной работе.

1.1. Статистический пакет RStudio

RStudio – среда разработки программ на языке R, которая распространяется под свободной лицензией GNU AGPL v3. RStudio имеет интуитивно понятный графический интерфейс и позволяет работать в операционных системах Linux, Microsoft Windows и MacOS. Последнюю версию программы можно загрузить на сайте https://posit.co. Перед установкой RStudio желательно установить R версии не ниже 3.6.0.

1.2. Загрузка и предобработка данных из CSV-файла

В базовых библиотеках R имеется широкий спектр функций для загрузки внешних данных из файлов и баз данных. Основная функция чтение файлов с табличными данными `read.table()`. На вход функции задаётся название файла и параметры считывания файла. На выходе получаем результат считывания в виде специальной структуры данных `data.frame`.

Полная форма записи `read.table()` представлена далее:

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors, tryLogical = TRUE,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = FALSE,
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

Аргументы функции `read.table()`:

- `file` – обязательный аргумент, имя файла;
- `header` – логический параметр, при значении TRUE считываются имена переменных из файла;
- `sep` – разделитель полей, по умолчанию – пробел;
- `quote` – вид кавычек (двойные или одинарные);
- `dec` – десятичный разделитель в числах (точка или запятая);
- `row.names` – вектор имён строк, представляет собой либо вектор с именами строк итоговой таблицы, либо число – номер столбца исходной таблицы с названиями строк; либо имя столбца считываемой таблицы, где приведены названия строк; если этот параметр не задан, то строки в итоговой таблице будут пронумерованы;
- `col.names` – вектор имён столбцов в итоговой таблице; по умолчанию «V<номер столбца>»;
- `as.is` – либо логический, либо числовой вектор, определяющий столбцы, неконвертируемые в факторы. Проверят нужно ли символьные переменные, не преобразованные в числовые или логические, переводить в факторы;
- `colClasses` – символьный вектор, определяет классы данных в столбцах (символьные, логические, числовые, даты). Возможные значения: `NA` – автоматическая конвертация типов данных, `NULL` – столбец пропускается (данные не преобразовываются), тип данных в который будут переведены элементы столбца, `factor`;

- `na.strings` – символьный вектор, элементы которого при чтении исходной таблицы в файле будут интерпретироваться как `NA`;
- `nrows` – целочисленный аргумент, максимальное число считываемых строк;
- `skip` – положительный целочисленный аргумент, число пропускаемых строк;
- `check.names` – логический аргумент; при значении `TRUE` имена переменных будут проверены на синтаксическую правильность и отсутствие дублирования;
- `fill` – логический аргумент; при значении `TRUE` строки разной длины будут приведены к единой (максимальной) добавлением пустых полей;
- `strip.white` – логический аргумент; используется только если определён разделитель sep, позволяет убирать пробелы перед и после символьных переменных;

Исходные данные для обработки многомерных данных в R хранятся в специальной структуре данных `data.frame`, которая представляет собой таблицу – в строках хранятся наблюдения, а в столбцах – переменные.

Существуют также функция более низкого уровня для считывания файлов `scan()`, а также ряд упрощенных версий функции `read.table()`: `read.csv()`, `read.csv2()`, `read.delim()`, `read.delim2()`.

В качестве исходных данных будет использоваться файл SuperHeroes.csv (https://storage.yandexcloud.net/datalens/SuperHeroes.csv) с информацией о супергероях — имя, пол, раса, издатель комиксов и т.д. Далее представлены первые 8 строк файла SuperHeroes.csv.

```
Name;Gender;Eye color;Race;Hair color;Height;Publisher;Skin
color;Alignment;Weight
Alien;Male;unknown;Xenomorph XX121;No Hair;244.0;Dark Horse
Comics;black;bad;169.0
Killer Frost;Female;blue;Human;Blond;;DC Comics;blue;bad;
Mystique;Female;yellow (without irises);Mutant;Red / Orange;178.0;Marvel
Comics;blue;bad;54.0
Nebula;Female;blue;Luphomoid;No Hair;185.0;Marvel Comics;blue;bad;83.0
Abe Sapien;Male;blue;Icthyo Sapien;No Hair;191.0;Dark Horse
Comics;blue;good;65.0
Dr Manhattan;Male;white;Human / Cosmic;No Hair;;DC Comics;blue;good;
Shadow Lass;Female;black;Talokite;Black;173.0;DC Comics;blue;good;54.0 ...
```

Сохраним файл на диске, например, «D:\Temp\SuperHeroes.csv» и считаем его командой:

```
data = read.table(file = "D:/Temp/SuperHeroes.csv", header = TRUE,
                  sep = ";", quote = "")
```

Обратите внимание, что в названии файла `file = "D:/Temp/SuperHeroes.csv"` вместо привычного разделителя пути «\» необходимо пользоваться символом «/». В первой строке файла содержатся названия переменных `header = TRUE`. Разделители переменных – точка с запятой `sep = ";"`. Строки в данном случае представлены без кавычек `quote = ""`.

Выведем на экран список всех переменных:

```
'data.frame':  734 obs. of  10 variables:
 $ Name      : chr  "Alien" "Killer Frost" "Mystique" "Nebula" ...
 $ Gender    : chr  "Male" "Female" "Female" "Female" ...
 $ Eye.color : chr  "unknown" "blue" "yellow (without irises)" "blue" ...
 $ Race      : chr  "Xenomorph XX121" "Human" "Mutant" "Luphomoid" ...
 $ Hair.color: chr  "No Hair" "Blond" "Red / Orange" "No Hair" ...
 $ Height    : num  244 NA 178 185 191 NA 173 180 183 183 ...
 $ Publisher : chr  "Dark Horse Comics" "DC Comics" "Marvel Comics" "Marvel
Comics" ...
 $ Skin.color: chr  "black" "blue" "blue" "blue" ...
 $ Alignment : chr  "bad" "bad" "bad" "bad" ...
 $ Weight    : num  169 NA 54 83 65 NA 54 181 68 67 ...
```

Как видно из описания, набор данных содержит 8 строковых переменных `(chr)` и 2 – числовых `(num)`. Неизвестные строковые значения `"unknown"` означают, что этих данных нет, но R воспринимает их как еще одно значение в списке других значений. В отличие от строковых, в числовые переменные отсутствующие значения `(NA)` представлены верно.

Проведем предобработку данных средствами R. Во-первых, необходимо во всех строковых переменных заменить `"unknown"` на `NA`. Во-вторых, строковые переменные преобразовать в факторы. Это позволит правильно обрабатывать *отсутствующие/пропущенные/неизвестные* значения.

Пример замены в строковой переменной `Gender` всех значений `"unknown"`, на отсутствующие значения `NA` и преобразование её в дискретную переменную (`factor`) в R выполняется следующими командами.

```
# Замена всех "unknown" в переменной Gender на NA
data$Gender[data$Gender  == "unknown" ] <- NA
# Преобразование переменной в факторы
data$Gender <- as.factor(data$Gender)
```

При большом количестве переменных предпочтительнее эти действия выполнять в цикле.

```
# Замена всех "unknown" в строковых переменных на NA
for (var_name in variable.names(data)) {
  data[[var_name]][ data[[var_name]] == "unknown" ] <- NA
}
# Количество всех переменных
var_count = length(variable.names(data))
# Преобразование всех строковых переменных в факторы
# Кроме первой переменной Name [2:var_count]
for (var_name in variable.names(data)[2:var_count]) {
  if(class(data[[var_name]]) == "character")
    data[[var_name]] <- as.factor(data[[var_name]])
}

# Вывод на экран
str(data)
```

```
'data.frame':  734 obs. of  10 variables:
 $ Name      : chr  "Alien" "Killer Frost" "Mystique" "Nebula" ...
 $ Gender    : Factor w/ 2 levels "Female","Male": 2 1 1 1 2 2 1 2 2 1 ...
 $ Eye.color : Factor w/ 22 levels "amber","black",..: NA 3 20 3 3 17 2 3 3 14
...
 $ Race      : Factor w/ 61 levels "Alien","Alpha",..: 57 23 42 38 32 26 53 42
42 42 ...
 $ Hair.color: Factor w/ 29 levels "Auburn","black",..: 17 6 24 17 17 17 3 7 6
28 ...
 $ Height    : num  244 NA 178 185 191 NA 173 180 183 183 ...
 $ Publisher : Factor w/ 24 levels "ABC Studios",..: 2 3 12 12 2 3 3 12 12 12
...
 $ Skin.color: Factor w/ 16 levels "black","blue",..: 1 2 2 2 2 2 2 2 2 2 ...
 $ Alignment : Factor w/ 3 levels "bad","good","neutral": 1 1 1 1 2 2 2 2 2 3
...
 $ Weight    : num  169 NA 54 83 65 NA 54 181 68 67 ...
```

Таким образом, получаем набор данных, содержащий 734 наблюдения и 9 переменных (табл. 1.1).

Таблица 1.1. Описание переменных в наборе данных `SuperHeroes`

Переменная	Тип переменной	Описание
Name	строка	Имя героя
Gender	дискретная	Пол
Eye.color	дискретная	Цвет глаз
Race	дискретная	Раса
Hair.color	дискретная	Цвет волос
Height	непрерывная	Рост (см)
Publisher	дискретная	Издатель комикса
Skin.color	дискретная	Цвет кожи
Alignment	дискретная	Принадлежность группе хороших или плохих героев
Weight	непрерывная	Вес (кг)

В дальнейшем переменная Name не будет участвовать в анализе данных. Будут анализироваться 7 дискретных переменных и 2 непрерывные.

1.3. Первичный анализ данных

После предобработки данных проведем первичный анализ данных. Самый простой способ первичного данных – использовать функцию `summary()`, которая позволит оценить основные характеристики выборки.

```
summary(data)
```

```
     Name              Gender      Eye.color                  Race
 Length:734         Female:200   blue   :225   Human            :208
 Class :character   Male  :505   brown  :126   Mutant           : 63
 Mode  :character   NA's  : 29   green  : 73   God / Eternal    : 14
                                 red    : 46   Cyborg           : 11
                                 black  : 23   Human / Radiation: 11
                                 (Other): 69   (Other)          :123
                                 NA's   :172   NA's             :304

  Hair.color      Height                  Publisher     Skin.color
 Black  :158   Min.   : 15.2   Marvel Comics    :388   green  : 21
 Blond  : 99   1st Qu.:173.0   DC Comics        :215   blue   :  9
 Brown  : 86   Median :183.0   NBC - Heroes     : 19   red    :  9
 No Hair: 75   Mean   :186.7   Dark Horse Comics: 18   white  :  7
 Red    : 51   3rd Qu.:191.0   George Lucas     : 14   grey   :  5
 (Other): 93   Max.   :975.0   (Other)          : 65   (Other): 21
 NA's   :172   NA's   :217     NA's             : 15   NA's   :662

  Alignment       Weight
 bad    :207   Min.   :  2.0
 good   :496   1st Qu.: 61.0
 neutral: 24   Median : 81.0
 NA's   :  7   Mean   :112.3
               3rd Qu.:108.0
               Max.   :900.0
               NA's   :239
               NA's   :239
```

Дискретные и непрерывные переменные анализируются по-разному. Если для непрерывных переменных можно вычислить среднее значение, определить разброс, то для дискретных – это сделать нельзя. В данном случае, все дискретные переменные являются номинальными, то есть по этим переменным невозможно упорядочить наблюдения. Поэтому для дискретных переменных выводятся только частоты.

Сформулируем несколько вопросов:

- Представителей какой расы больше всего?
- Какая студия создала больше всего супергероев?
- Есть ли зависимость роста от принадлежности к лагерю хороших или плохих?

Ответы на первые два вопроса из результатов анализа очевидны:

- Больше всего представителей человеческой расы (`Human`) – 208 героев.
- Больше всего создала героев создала студия `Marvel Comics` – 388 героев.

Для ответа на 3 вопрос необходимо провести корреляционный анализ.

Дополнительно по этим данным можно сказать:

- мужчин (505) среди героев комиксов больше, чем женщин (200);
- преобладают голубоглазые (225) и кареглазые (126) герои;
- герои в основном имеют черные волосы (158);
- их средний рост 186,7 см, но встречаются карлики (15,2 см) и гиганты (975 см);
- информация о цвете кожи у большинства героев не представлена (662), но среди них преобладают представители с зелёной кожей;
- среди героев больше хороших (496), чем плохих (207);
- медианный вес героя 81,0 кг, средний – 112,3 кг, при этом минимальный вес 2 кг, а максимальный – 900 кг.

Рассмотрим распределение непрерывных переменных. Отобразим распределение роста герове в виде гистограммы (рис.1.1).

```
hist(data$Heigh, xlab= "Рост, см", ylab = "Частота", main = "")
```

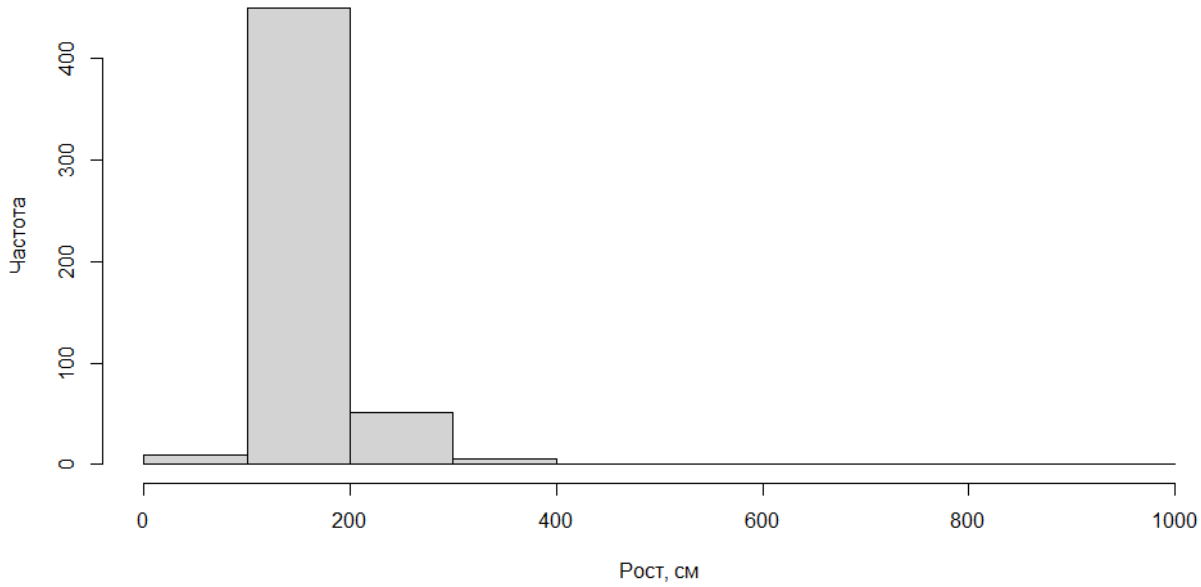

Рисунок 1.1 – Гистограмма распределения роста героев

Видно, что большинство наблюдений не превышает 400 см. Отобразим гистограмму распределение частот роста героев для 400 см (рис.1.2).

```
hist(data$Heigh[data$Heigh<=400], xlab= "Рост, см (<= 400)",
ylab = "Частота", main = "")
```

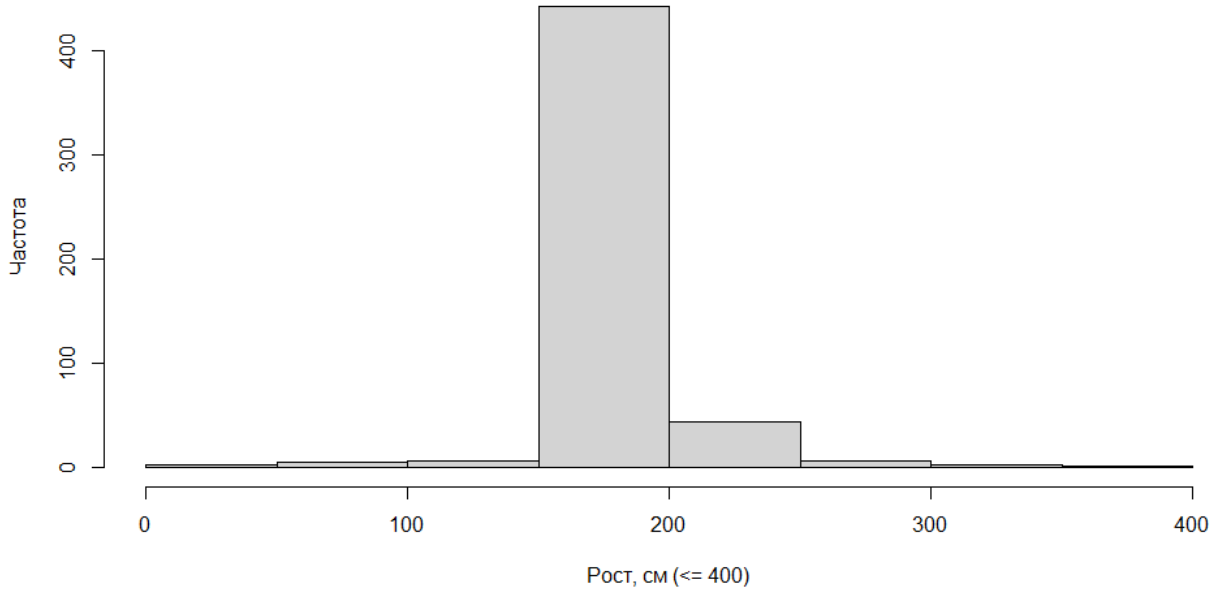

Рисунок 1.2 – Гистограмма распределения роста героев (до 400 см)

Гистограмма в данном случае более детально описывает распределения частот. Видно, что большинство героев имеют рост от 150 до 200 см.

Отобразим распределение веса в виде гистограммы (рис. 1.3).

```
hist(data$Weight, xlab= "Вес, кг", ylab = "Частота", main = "")
```

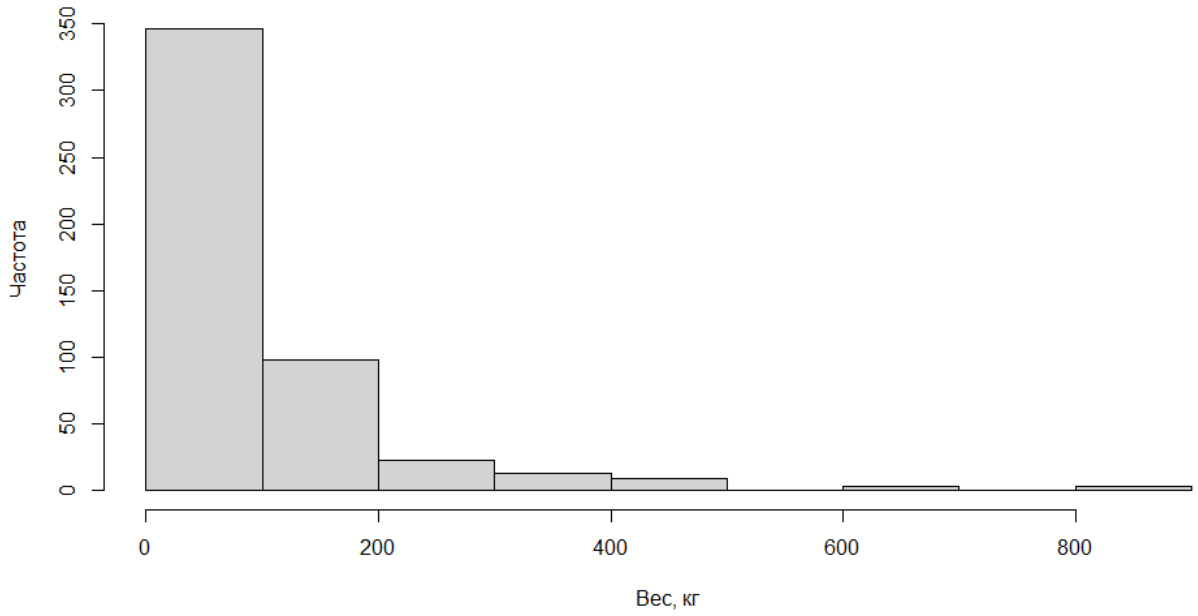

Рисунок 1.2 – Гистограмма распределения веса героев

Видно, что данное распределение имеет выраженную асимметрию. В большинстве случаев желательно, чтобы распределение было нормальным. Один из способов приведения к нормальному виду это преобразование исходных данных методом Бокса-Кокса или логарифмирование (рис. 1.4).

```
hist(log(data$Weight), xlab= "ln(Вес)", ylab = "Частота", main = "")
```

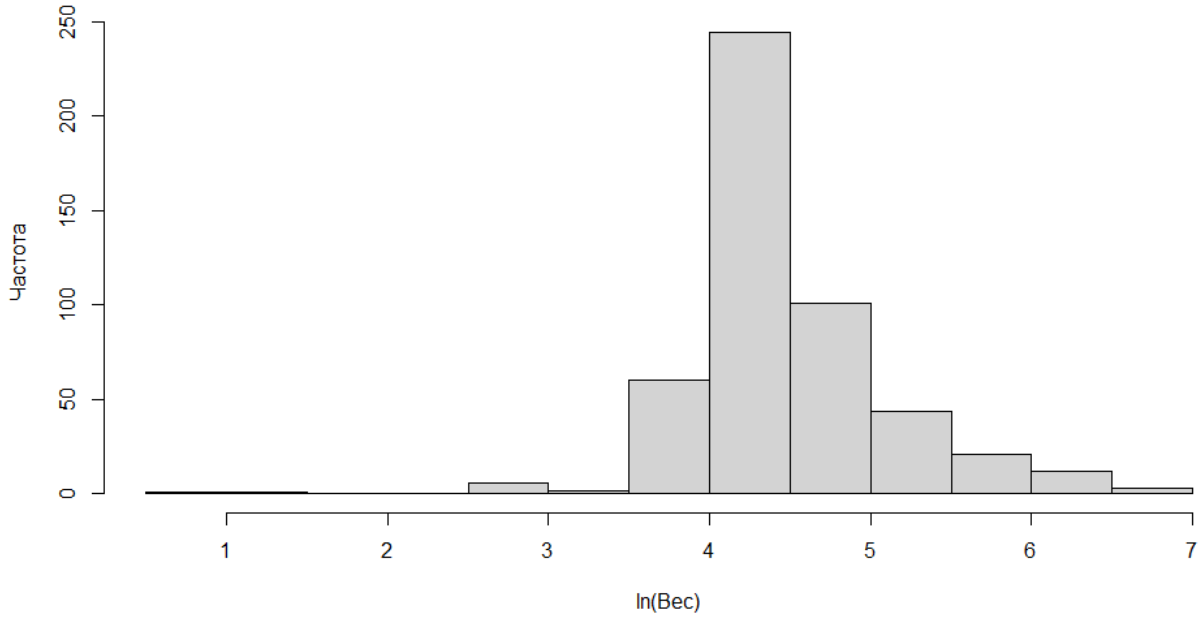

Рисунок 1.4 – Гистограмма распределения логарифма веса героев

Видно, что после преобразования распределение более симметричное.

1.4. Корреляционный анализ данных

Основной задачей корреляционного анализа – выявление связи между случайными величинами и оценка ее тесноты. Связь между непрерывными переменными вычисляют с помощью коэффициента корреляции Пирсона.

Рассмотрим выборку (x_i, y_i), $i = 1, 2, \ldots, n$, полученную из системы случайных величин (ξ, η). Выборочная оценка коэффициента корреляции между случайными величинами ξ и η вычисляется по формуле:

$$\hat{r}_{\xi\eta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}},$$

где $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ – выборочные оценки средних.

Данная оценка коэффициента корреляции называется выборочным коэффициентом корреляции Пирсона. Графики корреляционных зависимостей между случайными величинами ξ и η представлены на рис. 1.5 – 1.9.

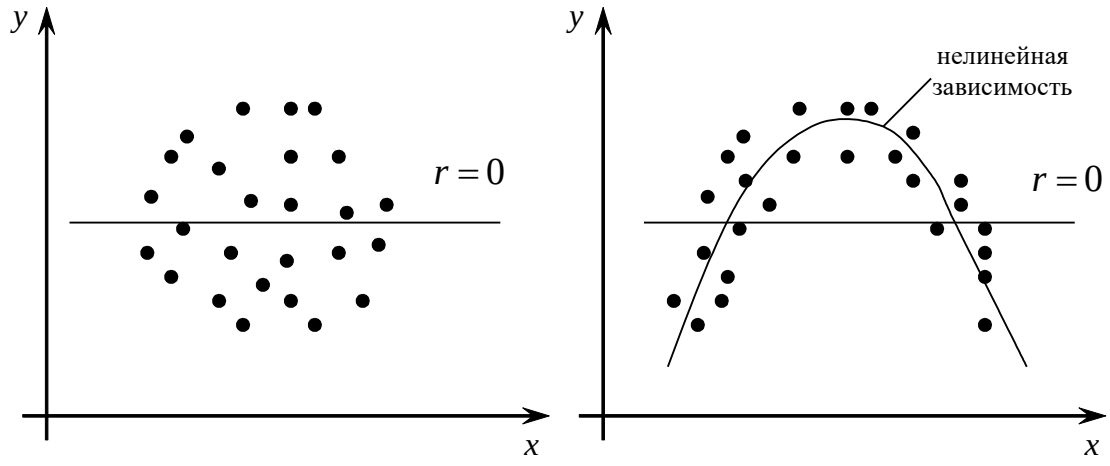

Рисунок 1.5 – Отсутствие линейной корреляционных связей между случайными величинами

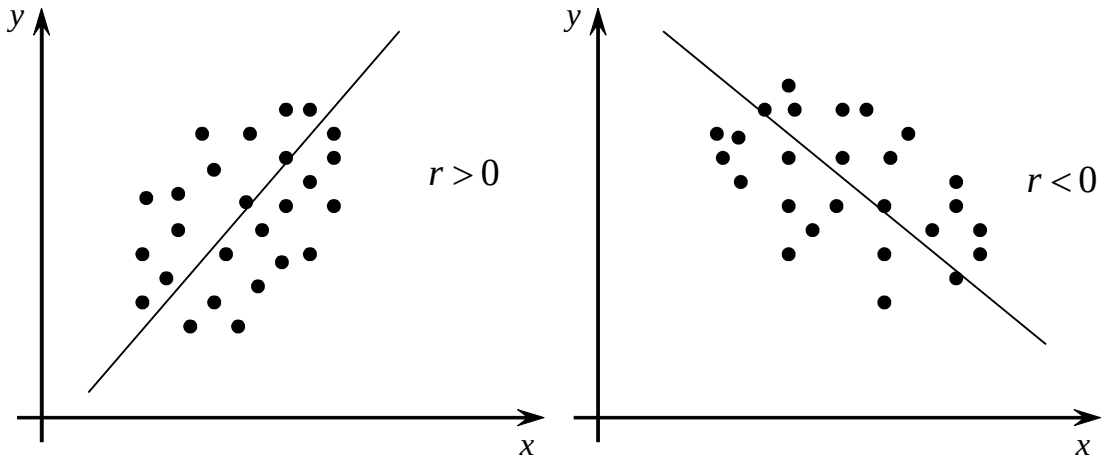

Рисунок 1.6 – Слабая корреляционная связь между случайными величинами

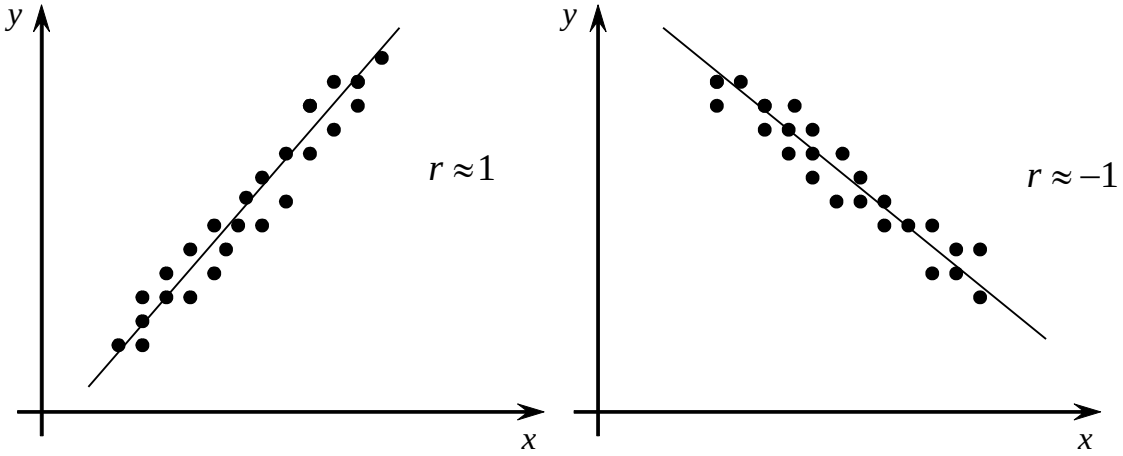

Рисунок 1.7 – Тесная корреляционная связь между случайными величинами

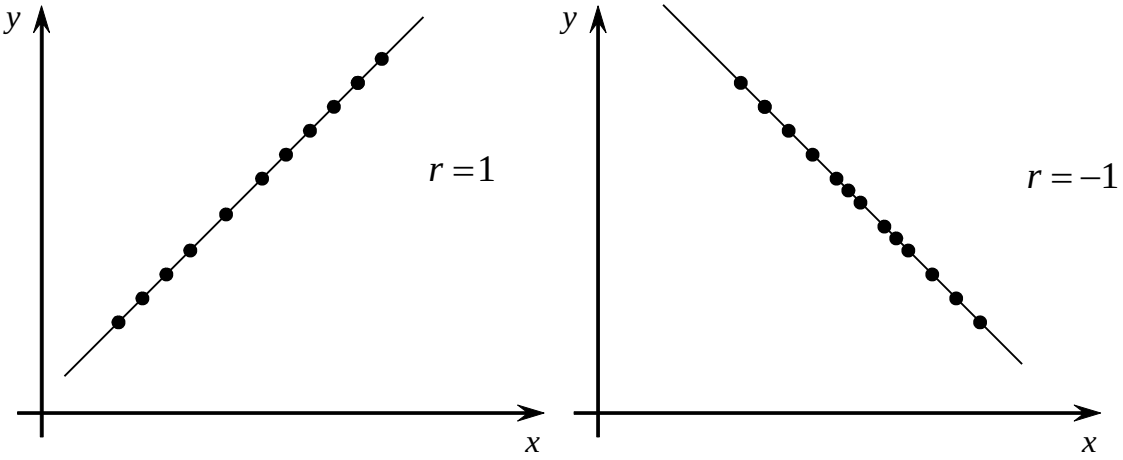

Рисунок 1.8 – Функциональная линейная корреляционная связь между случайными величинами

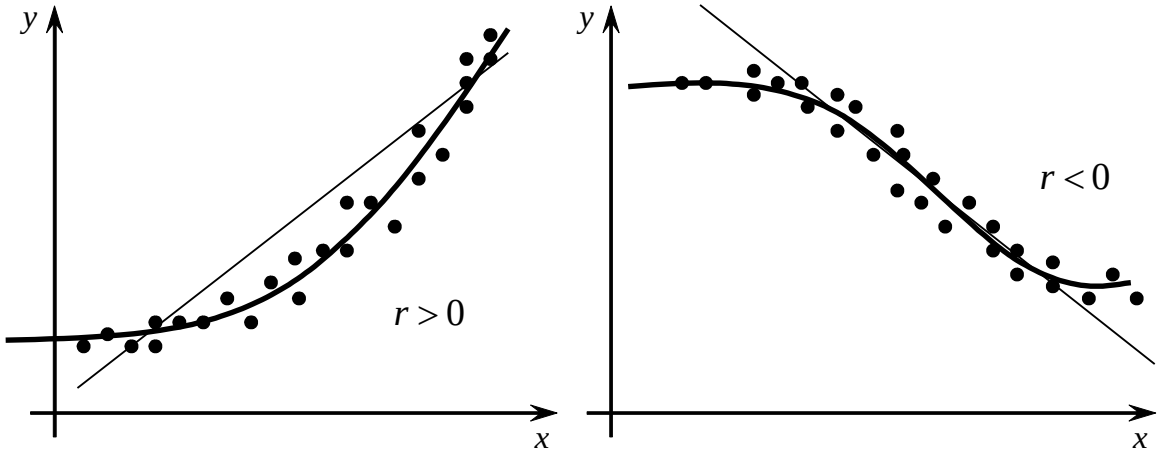

Рисунок 1.9 – Нелинейная связь между случайными величинами

Доверительный интервал коэффициента корреляции оценивают на основе статистики:

$$z = \frac{1}{2}\ln\left(\frac{1+\hat{r}}{1-\hat{r}}\right),$$

которая уже при $n > 10$ имеет нормальное распределение с математическим ожиданием $\mathbf{M}z \approx \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right) + \frac{r}{2(n-1)}$ и дисперсией $\mathbf{D}z \approx \frac{1}{n-3}$. Доверительные границы $z_1 < z < z_2$ с уровнем надежности γ вычисляются по формуле:

$$z_{1,2} = \frac{1}{2}\ln\left(\frac{1+\hat{r}}{1-\hat{r}}\right) - \frac{\hat{r}}{2(n-1)} \mp \frac{u_{(1+\gamma)/2}}{\sqrt{n-3}},$$

где u_β – квантиль стандартного нормального распределения уровня β. Доверительный интервал для коэффициента корреляции имеет вид:

$$\text{th}(z_1) < r < \text{th}(z_2),$$

где $\text{th}(z)$ – гиперболический тангенс, который вычисляется по формуле

$$\text{th}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

Для проверки гипотезы о значимости коэффициента корреляции выдвигают основную гипотезу, которая утверждает, что истинный (теоретический) коэффициент корреляции равен нулю

$$H_0 : r = 0,$$

в то время как альтернативная гипотеза утверждает обратное, а именно, что коэффициент корреляции отличается от нуля

$$H_1 : r \neq 0$$

При выполнении гипотезы H_0 статистика

$$\varphi = \frac{\hat{r}}{\sqrt{1-\hat{r}^2}}\sqrt{n-2} \overset{H_0}{\sim} t : n-2$$

имеет распределение Стьюдента с $n-2$ степенями свободы.

Гипотеза H_0 принимается на уровне значимости α если

$$|\varphi| < t_{1-\frac{\alpha}{2}}(n-2),$$

где $t_{1-\frac{\alpha}{2}}(n-2)$ – квантиль распределения Стьюдента с $n-2$ степенями свободы уровня $1-\frac{\alpha}{2}$.

В статистическом пакете R (пакеты stats) реализованы функции корреляционного анализа данных. Описание этих функций представлено в табл. 1.2.

Таблица 1.2. Описание функций статистического пакета R для проведения корреляционного анализа данных

Функция	Описание функции
`cov(x, y)`	Вычисление выборочной ковариации (ковариационной матрицы)
`cor(x, y)`	Вычисление выборочного коэффициента корреляции (корреляционной матрицы)
`cor.test(x, y, ...)`	Проверка гипотезы о значимости коэффициента корреляции

Вычислим коэффициент корреляции между ростом и весом героев.

```
cor.test(data$Height, data$Weight, use = "pairwise.complete.obs")
```

```
Pearson's product-moment correlation

data:  data$Height and data$Weight
t = 4.3555, df = 488, p-value = 1.619e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1066877 0.2772717
sample estimates:
      cor
0.1934412
```

Выборочный коэффициент корреляции ≈0,193 между ростом и весом по данной оценке не очень высокий, тем не менее он значимо отличается от 0, так как p-значение $1{,}619 \cdot 10^{-5}$ значительно меньше уровня значимости α=0,001. Это

связано с наличием 3 резко выделяющихся наблюдений (выбросов) ростом более 400 см (рис. 1.10), которые смазывают картину.

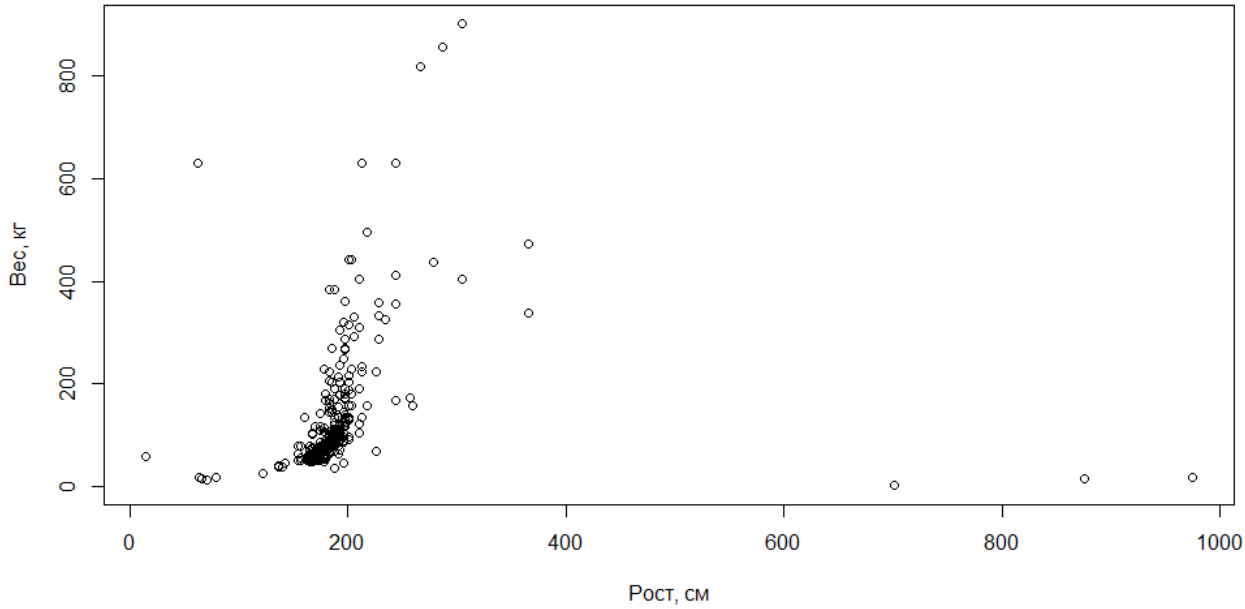

Рисунок 1.10 – Зависимость между весом и ростом

Исключим 3 резко выделяющихся наблюдений ростом более 400 см из рассмотрения, вычислим коэффициент корреляции.

```
cor.test(data$Height[data$Heigh<=400], data$Weight[data$Heigh<=400], use =
"pairwise.complete.obs")
```

```
Pearson's product-moment correlation

data:  data$Height[data$Heigh <= 400] and data$Weight[data$Heigh <= 400]
t = 15.926, df = 485, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5244366 0.6414411
sample estimates:
      cor
0.5859849
```

В данном случае связь между переменными более очевидная выборочный коэффициент корреляции ≈0,586 между ростом и весом, так как p-значение $2{,}2 \cdot 10^{-16}$ значительно меньше уровня значимости $\alpha=0{,}001$.

Построим график зависимости (рис. 1.11)

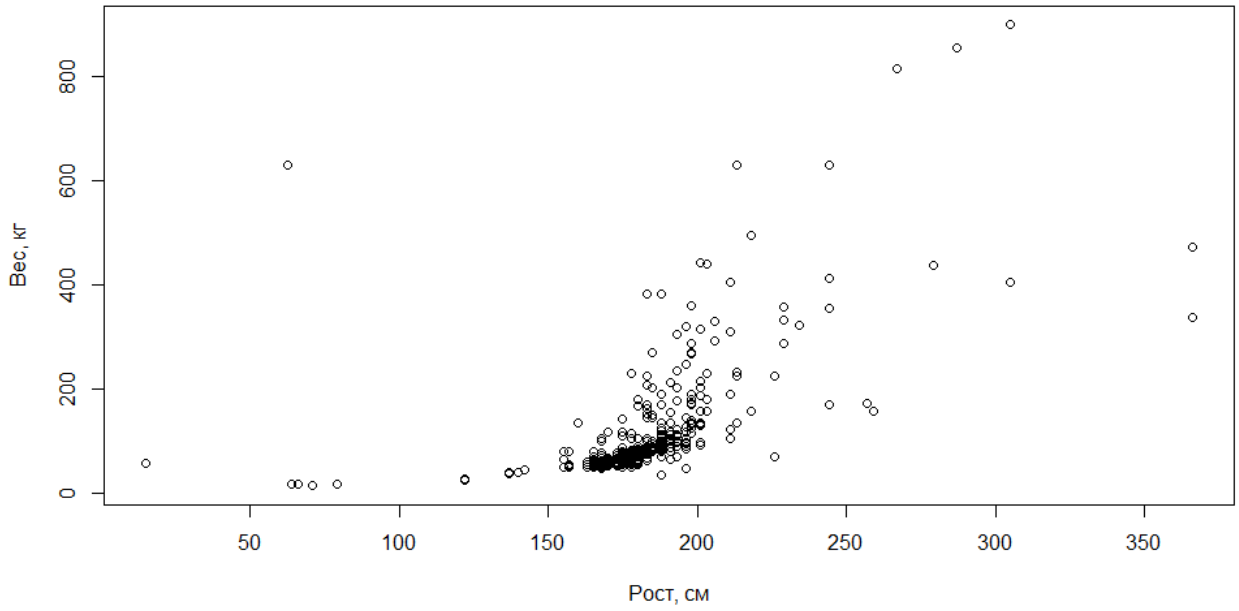

Рисунок 1.11 – Зависимость между весом и ростом (до 400 см)

Таким образом после очистки данных от выбросов были получены более качественные зависимости.

1.5. Регрессионный анализ данных

Основными задачами регрессионного анализа данных являются установление формы зависимости между переменными, оценка параметров функции регрессии по выборке и прогноз значений зависимой переменной.

Входными (или объясняющими) **переменными** будем называть случайные величины ξ_j, $j = 1, 2, \ldots, k$, а случайную величину η – **выходной переменной** (или функцией отклика).

В общем случае задача регрессионного анализа состоит в оценке параметров **модельной функции регрессии**:

$$\hat{y} = \psi(x_1, x_2, \ldots, x_k; b_0, b_1, \ldots, b_p)$$

где $\hat{y}$ – модельное значение переменной η при фиксированных значениях $\xi_j = x_j$, $j = 1, 2, \ldots, k$; $b_0, b_1, \ldots, b_p$ – выборочные значения параметров функции регрессии $\beta_0, \beta_1, \ldots, \beta_p$; $p+1$ – число оцениваемых по выборке параметров.

Задача многомерного линейного регрессионного анализа состоит в оценке параметров множественной линейной регрессионной модели вида:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k .$$

Ошибки аппроксимации функции регрессии (или как ещё их называют **остатки**) имеют вид:

$$e_i = y_i - \hat{y}_i = y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}).$$

Параметры модельных функций регрессии определяются при известном законе распределения остатков. На практике этот закон не всегда известен, а исследователь располагает только ограниченной выборкой:

$$(\mathbf{x}_i, y_i),\ i = 1, 2, \ldots, n,$$

где $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$, n – число наблюдений.

Рассмотрим более подробно метод максимального правдоподобия в случае нормальной распределённых остатков. В матрице $\mathbf{X}$ содержатся данные об измерениях объясняющих переменных ξ_j, $j = 1, 2, \ldots, k$:

$$\mathbf{X} = \begin{array}{c} \begin{matrix} \xi_1 & \xi_2 & \ldots & \xi_k \end{matrix} \\ \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1k} \\ x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nk} \end{pmatrix} \end{array}$$

где x_{ij} - соответствует значению i-го наблюдения по j переменной.

В вектор-строке $\mathbf{Y}$ содержатся данные об измеренных значениях объясняемой переменной η:

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

Обозначим через $\tilde{\mathbf{X}}$ матрицу $\mathbf{X}$ с присоединенным слева единичным вектором:

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{pmatrix}.$$

Вектор-строку этой матрицы обозначим $\tilde{\mathbf{x}}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ik})$, а параметры функции регрессии в виде $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$.

Рассмотрим **метод максимального правдоподобия** для оценки параметров регрессии. Функцию максимального правдоподобия запишем в виде:

$$L(e_1, e_2, \ldots, e_n \mid \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{n} f_\varepsilon(e_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) =$$

$$= \left[\frac{1}{\sqrt{2\pi\sigma^2}}\right]^n \exp\left(-\frac{\sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}))^2}{2\sigma^2}\right) =$$

$$= \left[\frac{1}{\sqrt{2\pi\sigma^2}}\right]^n \exp\left(-\frac{\sum_{i=1}^{n} (y_i - \tilde{\mathbf{x}}_i \cdot \boldsymbol{\beta})(y_i - \tilde{\mathbf{x}}_i \cdot \boldsymbol{\beta})}{2\sigma^2}\right) =$$

$$= \left[\frac{1}{\sqrt{2\pi\sigma^2}}\right]^n \exp\left(-\frac{(\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})^T (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})}{2\sigma^2}\right).$$

Прологарифмируем функцию максимального правдоподобия:

$$LL(e_1, e_2, \ldots, e_n \mid \boldsymbol{\beta}, \sigma^2) = \ln L(e_1, e_2, \ldots, e_n \mid \boldsymbol{\beta}, \sigma^2) = -\frac{n}{2} \ln[2\pi\sigma^2]^n - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})^T (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})$$

Запишем производные этой функции по параметрам $\boldsymbol{\beta}$ и σ^2:

$$\begin{cases} \dfrac{\partial LL}{\partial \boldsymbol{\beta}} = -\dfrac{1}{2\sigma^2} \dfrac{\partial}{\partial \boldsymbol{\beta}} (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})^T (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta}), \\ \dfrac{\partial LL}{\partial \boldsymbol{\beta}} = -\dfrac{n}{2\sigma^2} + \dfrac{1}{2\sigma^4} (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})^T (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta}). \end{cases}$$

Найдем производную $(\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})^T (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})$ по вектору параметров $\boldsymbol{\beta}$:

$$\frac{\partial}{\partial \boldsymbol{\beta}} (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})^T (\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta}) = -2(\mathbf{Y} - \tilde{\mathbf{X}} \cdot \boldsymbol{\beta})^T \tilde{\mathbf{X}}.$$

Таким образом, оценкой вектора-столбца параметров $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$ регрессионной модели будет вектор-столбец $\mathbf{b} = (b_0, b_1, \ldots, b_k)^T$:

$$\mathbf{b} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{Y}.$$

а оценкой дисперсии σ^2 будет $\hat{\sigma}^2$:

$$\hat{\sigma}^2 = \frac{1}{n}\left(\mathbf{Y} - \tilde{\mathbf{X}} \cdot \mathbf{b}\right)^T \left(\mathbf{Y} - \tilde{\mathbf{X}} \cdot \mathbf{b}\right) = \frac{1}{n}\sum_{i=1}^{n} e_i^2 .$$

Важной процедурой является оценка адекватности уравнения регрессии. Одним из критериев адекватности модели является отличие коэффициентов уравнения регрессии b_j от нуля и для проверки этого выдвигаются гипотезы:

$$H_0: \quad \beta_j = 0,$$
$$H_1: \quad \beta_j \neq 0.$$

Проверка значимости предполагает, что коэффициенты регрессии имеют закон распределения Стьюдента.

$$\varphi_j = \frac{b_j}{s_{b_j}} \overset{H_0}{\sim} t : n-k-1$$

где $s_{b_j} = s_e \sqrt{\left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}\right)^{-1}_{jj}}$ - выборочная дисперсия коэффициента регрессии; $s_e^2 = = \frac{1}{n-k-1}\sum_{i=1}^{n} e_i^2$ - выборочная остаточная дисперсия; $\left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}\right)^{-1}_{jj}$ - диагональный элемент матрицы $\left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}\right)^{-1}$. Основная гипотеза H_0 о равенстве коэффициента уравнения регрессии β_j нулю принимается на заданном уровне значимости α, если

$$\left|\varphi_j\right| < t_{1-\frac{\alpha}{2}}(n-k-1),$$

иначе принимается альтернативная гипотеза H_1, т.е. коэффициент β_j значимо отличается от нуля.

Доверительным интервалом для параметра β_j определяется их следующего соотношения:

$$b_j - s_{b_j} \cdot t_{1-\frac{\alpha}{2}}(n-k-1) < \beta_j < b_j + s_{b_j} \cdot t_{1-\frac{\alpha}{2}}(n-k-1)$$

Адекватность уравнения регрессии в целом оценивается по коэффициенту множественной детерминации $\hat{R}^2$:

$$\hat{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

Коэффициент детерминации можно рассматривать как меру качества уравнения регрессии. С увеличением значение коэффициента детерминации увеличивается адекватность модели, при этом его максимальное значение равняется 1. Исправленный коэффициент детерминации позволяет уточнить значение коэффициента детерминации:

$$R^2_{adj} = 1 - \left(1 - \hat{R}^2\right)\frac{n-1}{n-k-1}.$$

Уравнение регрессии неадекватно описывает зависимость, если коэффициент детерминации равняется 0. Проверка этой гипотезы этой гипотезы осуществляется на основе критерия Фишера:

$$\varphi = \frac{\hat{R}^2}{1-\hat{R}^2}\cdot\frac{n-k-1}{k} \overset{H_0}{\sim} F : k, n-k-1$$

имеет распределение Фишера с k и $n-k-1$ степенями свободы. Гипотеза $H_0 : R^2 = 0$ о незначимости коэффициента детерминации принимается на уровне значимости α, если статистика $\varphi < F_{1-\alpha}(n-k, k-1)$, иначе, если $\varphi \geq F_{1-\alpha}(n-k, k-1)$, то принимается гипотеза $H_1 : R^2 > 0$ о значимости коэффициента детерминации (значимом отличии коэффициента детерминации от нуля).

Интервальный прогноз y_i по данным $\tilde{\mathbf{x}}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ik})$ имеет вид:

$$\hat{y}_i - \delta_i \cdot t_{1-\frac{\alpha}{2}}(n-k-1) < y_i < \hat{y}_i + \delta_i \cdot t_{1-\frac{\alpha}{2}}(n-k-1)$$

где $\delta_i = s_e\sqrt{\tilde{\mathbf{x}}_i\left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)\tilde{\mathbf{x}}_i^T + 1}$ – стандартная ошибка прогноза.

В статистическом пакете R (пакет stats) реализованы функции для проведения регрессионного анализа данных. Описание этих функций представлено в табл. 1.3.

Таблица 1.3. Описание функций статистического пакета R для проведения регрессионного анализа данных

Функция	Описание функции
`lm(formula, data, ...)`	Оценка параметров линейных регрессионных моделей.
`glm(formula, data, ...)`	Оценка параметров обобщенных линейных регрессионных моделей. Допущения обобщенных линейных регрессионных моделях отличаются от допущений классического регрессионного анализа (см. выше)
`predict(object, newdata, ...)`	Прогнозирование на основе линейных регрессионных моделей `object`,

	построенных, например, с помощью функции `lm()`
`formula(x, ...)`	Формула определение вида взаимодействия функции отклика и объясняющих переменных

Построим линейную регрессию между двумя переменными.

```
fit <- lm(Weight ~ Height, data = data);
summary(fit)
```

```
Call:
lm(formula = Weight ~ Height, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-363.97  -46.05  -28.54   -4.48  747.46

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 48.10374   15.42368   3.119  0.00192 **
Height       0.34242    0.07862   4.356 1.62e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 102.6 on 488 degrees of freedom
  (244 пропущенных наблюдений удалены)
Multiple R-squared:  0.03742,  Adjusted R-squared:  0.03545
F-statistic: 18.97 on 1 and 488 DF,  p-value: 1.619e-05
```

Отобразим линейную регрессию между двумя переменными на графике (рис. 1.12).

```
xin <- seq(0, 1000, length=100)
pre <- predict(fit, data.frame(Height=xin), interval="confidence")
plot(data$Height, data$Weight, xlab= "Рост, см", ylab = "Вес, кг")
matplot(xin, pre, type="l", lty=c(1,2,2), add=TRUE)
```

Как видно из графика (рис. 1.12) линейная регрессия между двумя переменными достаточно сильно искажается наличием выбросов.

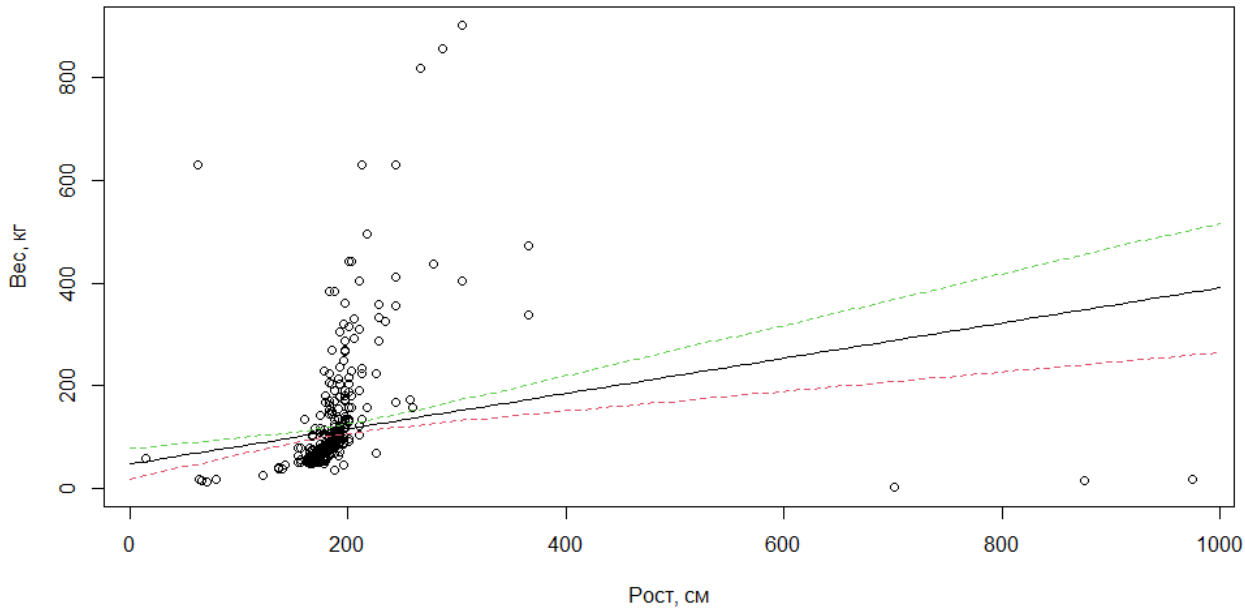

Рисунок 1.12 – Линейная регрессия между весом и ростом

Построим линейную зависимость между ростом и логарифмом веса только для тех, чей рост не превышает 400 см.

```
data1 = data.frame(Weight = data$Weight[data$Height<=400],
                   Height = data$Height[data$Height<=400])


fit <- lm(log(Weight) ~ Height, data = data1);
summary(fit)
```

```
Call:
lm(formula = log(Weight) ~ Height, data = data1)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4522 -0.2225 -0.0891  0.0818  3.7511

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.7513734  0.1343880   13.03   <2e-16 ***
Height      0.0150925  0.0007268   20.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4201 on 485 degrees of freedom
  (244 пропущенных наблюдений удалены)
Multiple R-squared:  0.4706,   Adjusted R-squared:  0.4696
F-statistic: 431.2 on 1 and 485 DF,  p-value: < 2.2e-16
```

И отобразим линейную регрессию на графике (рис. 1.13).

```
xin <- seq(0, 400, length=100)
pre <- predict(fit, data.frame(Height=xin), interval="confidence")
plot(data1$Height, log(data1$Weight), xlab= "Рост, см", ylab = "ln(Вес)")
matplot(xin, pre, type="l", lty=c(1,2,2), add=TRUE)
```

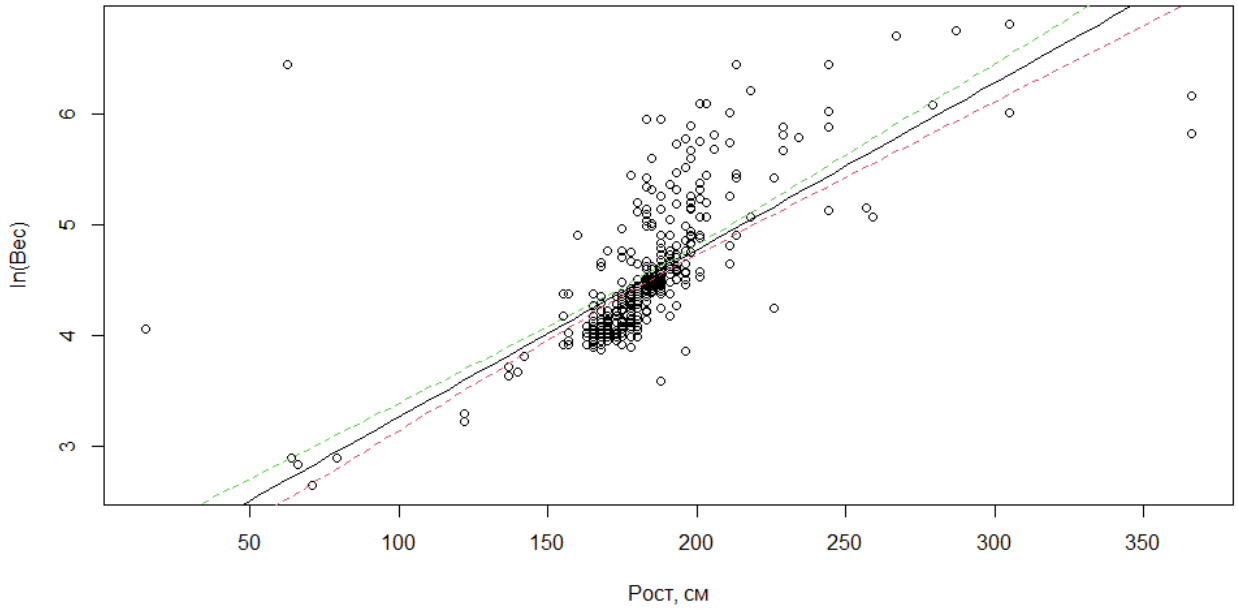

Рисунок 1.13 – Линейная регрессия между логарифмом веса и ростом (до 400 см)

Как видно из представленного на рис. 1.13 графика линейной регрессии, полученное соотношение точнее описывает соотношение между ростом и весом персонажей.

1.6. Дисперсионный анализ данных

Ответим на поставленный выше вопрос:

- Есть ли зависимость роста от принадлежности к лагерю хороших или плохих?

На этот вопрос можно ответить с помощью дисперсионного анализа. Дисперсионный анализ позволяет выявить зависимость между дискретной и переменной переменными.

В статистическом пакете R (пакет stats) реализованы функции для проведения дисперсионного анализа данных. Описание этих функций представлено в табл. 1.4.

Таблица 1.4. Описание функций статистического пакета R для проведения дисперсионного анализа данных

Функция	Описание функции
`aov(formula, data, ...)`	Оценка параметров моделей дисперсионного анализа.
`anova(object, data, ...)`	Вычисление таблиц дисперсионного анализа.
`boxplot (formula, data, ...)`	Построение графиков размаха

Проведём дисперсионный анализ между ростом и принадлежностью группе хороших или плохих персонажей.

```
fit = aov(Height ~ Alignment, data = data)
summary(fit)
```

```
             Df  Sum Sq Mean Sq F value  Pr(>F)
Alignment     2   46558   23279   6.726 0.00131 **
Residuals   509 1761609    3461
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
222 пропущенных наблюдений удалены
             Df  Sum Sq Mean Sq F value  Pr(>F)
Alignment     2   46558   23279   6.726 0.00131 **
Residuals   509 1761609    3461
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
222 пропущенных наблюдений удалены
```

Как видно из представленных выше результатов p-значение ≈0,013 меньше, чем 0,01, но больше 0,001. Таким образом, есть основания полагать, что существует зависимость между ростом и принадлежностью группе хороших или плохих персонажей

Отобразим результаты дисперсионного анализа на графике размахов (рис. 1.14).

```
boxplot(Height ~ Alignment, data = data,  ylim = c(0,400),
        xlab= "Принадлежность группе", ylab = "Рост, см")
```

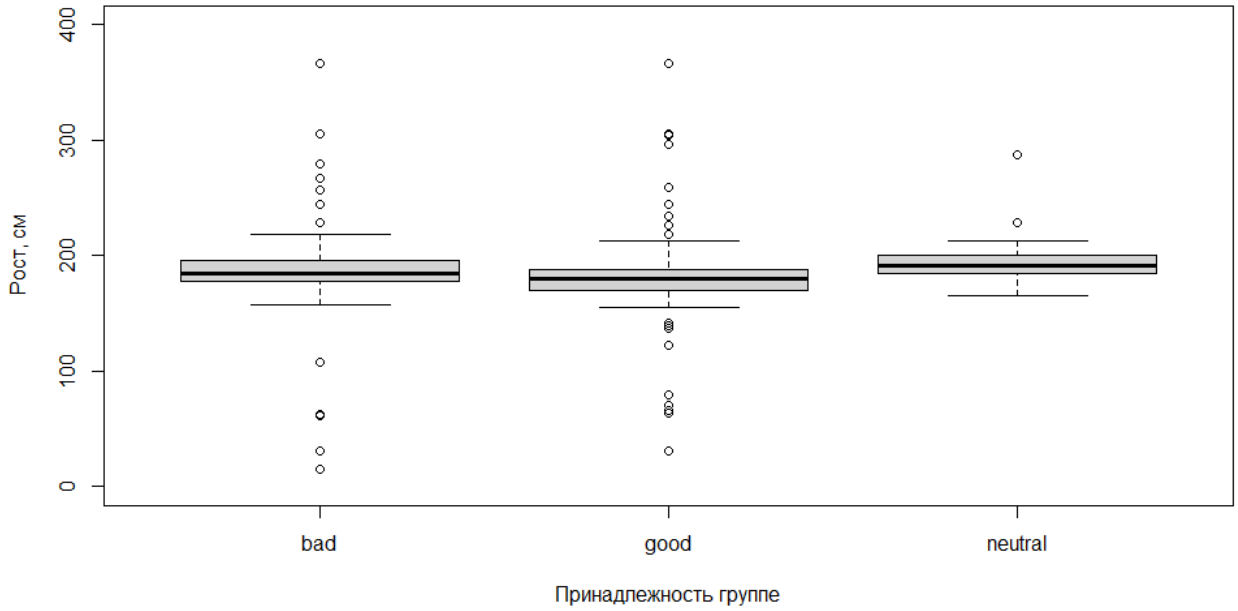

Рисунок 1.14 – График зависимости между ростом (до 400 см) и принадлежности к группе хороших или плохих персонажей

Как видно из представленного рис. 1.14 средний рост хороших персонажей чуть ниже, чем плохих или нейтральных.

1.7. Задание на лабораторную работу

1. Подключиться в https://www.kaggle.com/ и скачать данные в формате *.csv в соответствии со своим вариантом (номер варианта соответствует номеру по списку в учебной группе).
2. Загрузить и описать данные в формате *.csv.
3. Провести исследование зависимостей и сделать логические выводы по результатам анализа данных.
4. Подготовить отчёт по выполненной работе в формате Word/PDF.

1.8. Варианты заданий

1. Latest Global Temperatures (https://www.kaggle.com/datasets/csafrit2/latest-global-temperatures)
2. Finance companies in India (https://www.kaggle.com/datasets/anuragbantu/finance-companies-in-india)
3. Black Friday Sales EDA(https://www.kaggle.com/datasets/pranavuikey/black-friday-sales-eda)
4. Football Transfers from 1992/93 to 2021/22

seasons(https://www.kaggle.com/datasets/cbhavik/football-transfers-from-199293-to-202122-seasons)

5. World, Region, Country GDP/GDP per capita (https://www.kaggle.com/datasets/tmishinev/world-country-gdp-19602021)
6. Air Pollution Level (https://www.kaggle.com/datasets/totoro29/air-pollution-level)
7. Stack Overflow Developer Survey 2011-2022 (https://www.kaggle.com/datasets/yasirabdaali/stack-overflow-developer-survey-20112022)
8. Financing Healthcare (https://www.kaggle.com/datasets/programmerrdai/financing-healthcare)
9. Law School Admissions Bar Passage (https://www.kaggle.com/datasets/danofer/law-school-admissions-bar-passage)
10. Vending Machine Sales (https://www.kaggle.com/datasets/awesomeasingh/vending-machine-sales)
11. Global Companies dataset (https://www.kaggle.com/datasets/narayan63/global-companies-dataset)
12. Smoke Detection Dataset (https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset)
13. Netflix Data: Cleaning, Analysis and Visualization (https://www.kaggle.com/datasets/ariyoomotade/netflix-data-cleaning-analysis-and-visualization)
14. World Population by Countries Dataset (1960-2021) (https://www.kaggle.com/datasets/kaggleashwin/population-dataset)
15. Online Retails Sale Dataset (https://www.kaggle.com/datasets/rohitmahulkar/online-retails-sale-dataset)
16. Spotify unpopular songs (https://www.kaggle.com/datasets/estienneggx/spotify-unpopular-songs)
17. Chocolate Bar Ratings (https://www.kaggle.com/datasets/evangower/chocolate-bar-ratings)
18. Airbnb Open Data (https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata)
19. IMDB Movies (https://www.kaggle.com/datasets/totoro29/imdb-movies)
20. Youtube Statistics (https://www.kaggle.com/datasets/advaypatil/youtube-statistics)

Контрольные вопросы

1. Какую зависимость называют регрессионной? В чем отличие регрессионной зависимости от функциональной?
2. Как формулируется задача регрессионного анализа? Из каких соображений выбирается форма регрессионной зависимости?
3. Какой вид имеет линейная регрессионная модель? Как называются переменные, представленные в модели?
4. Какой метод используется для оценки параметров уравнения регрессии? Запишите формулы для ММП-оценок парной регрессии.
5. Как оценивается качество построенного уравнения регрессии? Приведите формулу для расчёта коэффициента детерминации.
6. Как производится проверка значимости построенного уравнения

регрессии? Какой критерий при этом используется?

7. Запишите линейную регрессионную модель с k независимыми переменными. Как выглядит система уравнений множественной линейной регрессии в матричной форме?
8. Из каких соображений получается система нормальных уравнений для определения оценок параметров уравнения регрессии?
9. Запишите в матричной форме систему нормальных уравнений.
10. Как проводится дисперсионный анализ для определения значимости уравнения множественной регрессии?
11. Как проверяется значимость коэффициентов уравнения регрессии?
12. Приведите формулы для расчёта доверительного интервала прогнозного значения в случае индивидуальных значений зависимой переменной. В чем отличие случая построения прогноза для функции регрессии?