



GeekBrains

R

Вебинары





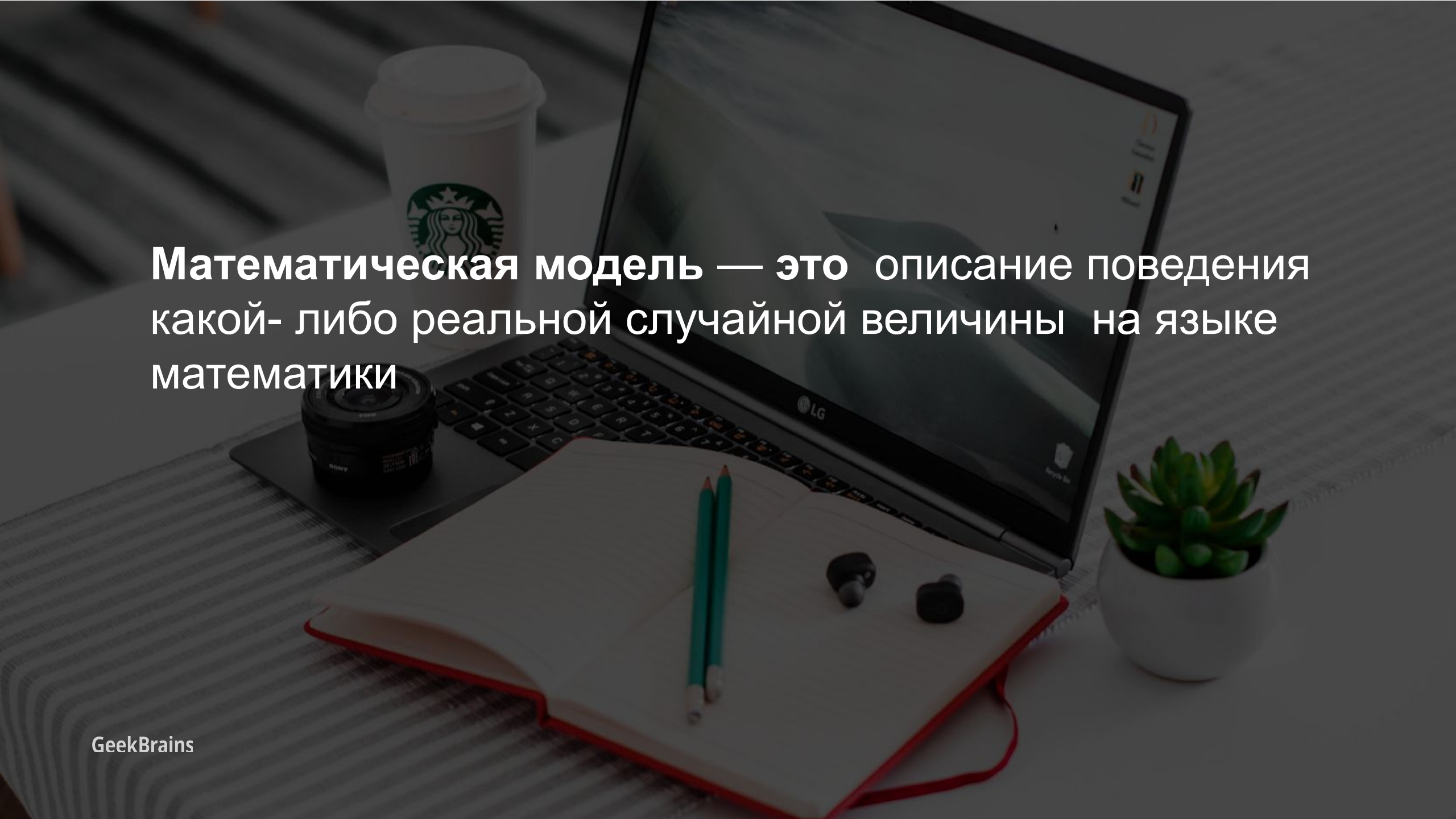
GeekBrains

Урок 1

Язык R для анализа данных

На этом уроке
мы изучим:

1. Линейная регрессия
2. Дисперсионный анализ

A desk setup featuring a laptop, a Starbucks cup, a notebook, pencils, and a succulent. The text is overlaid on the image.

Математическая модель — это описание поведения
какой-либо реальной случайной величины на языке
математики

Линейная регрессия

В основе лежит предположение некой ЛИНЕЙНОЙ зависимости $Y \sim X$, где Y –это зависимая переменная, а X - независимая(-ые)

$$Y = f(X)$$

Для парной линейной регрессии ,т.е. где только одна независимая переменная (признак) X , линейная зависимость будет иметь вид:

$$y = \beta_0 + \beta_1 * X$$

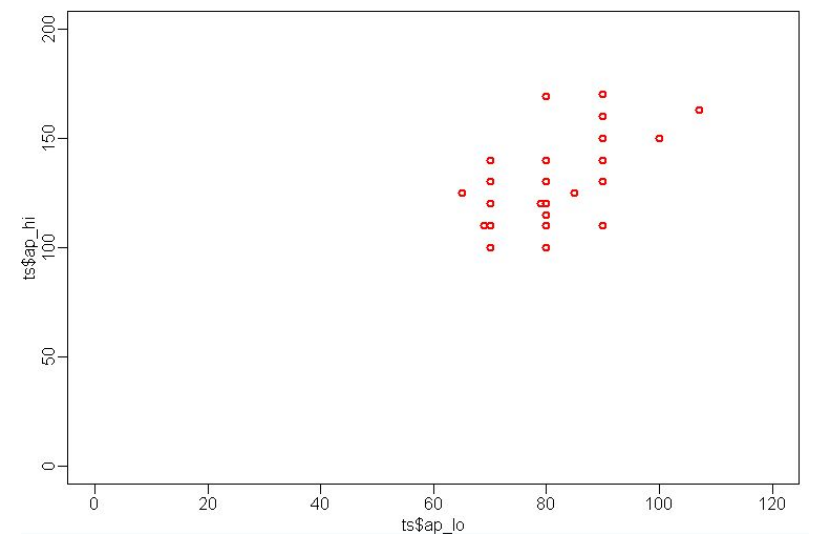
С помощью линейной регрессии попробуем описать зависимость переменной y (верхнее давление) от переменной X , в качестве которой мы будем рассматривать различные колонки и их комбинации из подготовленного набора данных `.tidy_set`

```

> .tidy_set<-tidy_set[tidy_set$ap_hi>tidy_set$ap_lo,]
> head(.tidy_set)
  id   age gender height weight ap_hi ap_lo cholesterol gluc smoke alco active cardio age_years
1  0 18393     2   168    62   110    80          1     1     0     0     1     0         50
2  1 20228     1   156    85   140    90          3     1     0     0     1     1         55
3  2 18857     1   165    64   130    70          3     1     0     0     0     1         51
4  3 17623     2   169    82   150   100          1     1     0     0     1     1         48
5  4 17474     1   156    56   100    60          1     1     0     0     0     0         47
6  8 21914     1   151    67   120    80          2     2     0     0     0     0         60
> dim(.tidy_set)
[1] 68678    14
> set.seed(1)
> ind<-sample(seq(1,nrow(.tidy_set)),100)
> ind
 [1] 18235 25557 39342 62372 13851 61696 64873 45378 43202 4243 14144 12124 47176 26375 52861 34174 49274 68106 26094 53379 64175
[22] 14566 44742 8620 18346 26508 920 26252 59705 23365 33094 41159 33880 12783 56795 45886 54519 7410 49676 28230 56349 44413
[43] 53738 37958 36357 54176 1602 32753 50259 47542 32779 59103 30065 16800 4851 6826 21704 35590 45427 27917 62640 20147 31500
[64] 22808 44659 17704 32834 52578 5781 60055 23264 57592 23785 22899 32680 61208 59296 26754 53325 65898 29817 48877 27439 22318
[85] 51932 13904 48778 8347 16839 9830 16436 4043 44052 60099 53422 54682 31224 28124 55610 41486
> ts<-tidy_set[ind,]
> head(ts)
  id   age gender height weight ap_hi ap_lo cholesterol gluc smoke alco active cardio age_years
18265 26543 19799     1   163    52   110    70          2     1     0     0     0     0         54
25601 37212 17541     1   165    65   115    80          1     1     0     0     1     1         48
39400 57321 23678     1   165    74   120    80          1     1     0     0     1     0         64
62466 90766 18998     1   155    48   100    70          1     1     0     0     0     0         52
13875 20151 20278     1   169    68   169    80          1     1     0     0     0     1         55
61790 89789 21684     1   160    64   120    80          1     1     0     0     1     0         59
> dim(ts)
[1] 100  14
> plot(ts$ap_lo,ts$ap_hi, xlim = c(0,120), ylim=c(0,200),col="red", lwd=2)

```

Из графика видно, что прослеживается линейная зависимость между нижним и верхним давлением пациента



Функции lm() и predict ()

```
> fitm<-lm(ts$ap_hi~ts$ap_lo) #построим парную линейную регрессию, в качестве независимой переменной возьмем нижнее давление пациента
> fitm
```

```
Call:
lm(formula = ts$ap_hi ~ ts$ap_lo)
```

```
Coefficients:
(Intercept)      ts$ap_lo
      16.513         1.347
```

```
> hi_hat<- 16.513 + 1.347*ts$ap_lo
> hi_hat
```

```
[1] 110.803 124.273 124.273 110.803 124.273 124.273 110.803 137.743 124.273 151.213 124.273 122.926 110.803 124.273 104.068 124.273
[17] 124.273 137.743 124.273 110.803 124.273 137.743 137.743 110.803 124.273 124.273 124.273 137.743 109.456 124.273 151.213
[33] 124.273 110.803 124.273 124.273 110.803 124.273 124.273 124.273 124.273 160.642 124.273 124.273 124.273 124.273 137.743 124.273
[49] 124.273 137.743 124.273 137.743 137.743 137.743 124.273 110.803 124.273 131.008 124.273 137.743 124.273 124.273 137.743 124.273
[65] 110.803 124.273 124.273 124.273 124.273 137.743 124.273 137.743 124.273 110.803 137.743 137.743 124.273 124.273 124.273 124.273
[81] 124.273 124.273 110.803 124.273 124.273 124.273 110.803 124.273 124.273 124.273 124.273 124.273 124.273 124.273 137.743 124.273
[97] 110.803 110.803 137.743 110.803
```

Функция predict () упрощает вычисления оценочного параметра, особенно удобно использовать ,когда имеем дело не с одним X

```
> as.numeric(predict(fitm,ts))
```

```
[1] 110.8216 124.2943 124.2943 110.8216 124.2943 124.2943 110.8216 137.7670 124.2943 151.2397 124.2943 122.9470 110.8216 124.2943
[15] 104.0852 124.2943 124.2943 137.7670 124.2943 110.8216 124.2943 137.7670 137.7670 110.8216 124.2943 124.2943 124.2943 124.2943
[29] 137.7670 109.4743 124.2943 151.2397 124.2943 110.8216 124.2943 124.2943 110.8216 124.2943 124.2943 124.2943 124.2943 160.6705
[43] 124.2943 124.2943 124.2943 124.2943 137.7670 124.2943 124.2943 137.7670 124.2943 137.7670 137.7670 137.7670 124.2943 110.8216
[57] 124.2943 131.0306 124.2943 137.7670 124.2943 124.2943 137.7670 124.2943 110.8216 124.2943 124.2943 124.2943 124.2943 137.7670
[71] 124.2943 137.7670 124.2943 110.8216 137.7670 137.7670 124.2943 124.2943 124.2943 124.2943 124.2943 124.2943 110.8216 124.2943
[85] 124.2943 124.2943 110.8216 124.2943 124.2943 124.2943 124.2943 124.2943 124.2943 124.2943 137.7670 124.2943 110.8216 110.8216
[99] 137.7670 110.8216
```

```
> ? predict
```

Причина различий оценочных значений у
кроется в округлении вычислений

```
> signif(fitm$coefficients,10)
(Intercept)    ts$ap_lo
  16.512768    1.347269
> 16.512768+1.347269*ts$ap_lo
 [1] 110.8216 124.2943 124.2943 110.8216 124.2943 124.2943 110.8216 137.7670 124.2943 151.2397 124.2943 122.9470 110.8216 124.2943
[15] 104.0853 124.2943 124.2943 137.7670 124.2943 110.8216 124.2943 137.7670 137.7670 110.8216 124.2943 124.2943 124.2943 124.2943
[29] 137.7670 109.4743 124.2943 151.2397 124.2943 110.8216 124.2943 124.2943 110.8216 124.2943 124.2943 124.2943 124.2943 160.6706
[43] 124.2943 124.2943 124.2943 124.2943 137.7670 124.2943 124.2943 137.7670 124.2943 137.7670 137.7670 137.7670 124.2943 110.8216
[57] 124.2943 131.0306 124.2943 137.7670 124.2943 124.2943 137.7670 124.2943 110.8216 124.2943 124.2943 124.2943 124.2943 137.7670
[71] 124.2943 137.7670 124.2943 110.8216 137.7670 137.7670 124.2943 124.2943 124.2943 124.2943 124.2943 124.2943 110.8216 124.2943
[85] 124.2943 124.2943 110.8216 124.2943 124.2943 124.2943 124.2943 124.2943 124.2943 124.2943 124.2943 137.7670 124.2943 110.8216 110.8216
[99] 137.7670 110.8216
```

FilesPlotsPackagesHelpViewer

←→🏠🖨️📄

🔍

R: Model PredictionsFind in Topic

Model Predictions

Description

`predict` is a generic function for predictions from the results of various model fitting functions. The function invokes particular *methods* which depend on the [class](#) of the first argument.

Usage

```
predict (object, ...)
```

Arguments

`object` a model object for which prediction is desired.

`...` additional arguments affecting the predictions produced.


```
> plot(seq(1:length(ts$ap_hi)),ts$ap_hi,col="red",type = "l")
> lines(seq(1,length(hi_hat)), hi_hat, col="blue",type="l")
> legend("topleft",c("ts$ap_hi","hi_hat"),col=c(2,3), lty = c(1,1))
> summary(fitm)
```

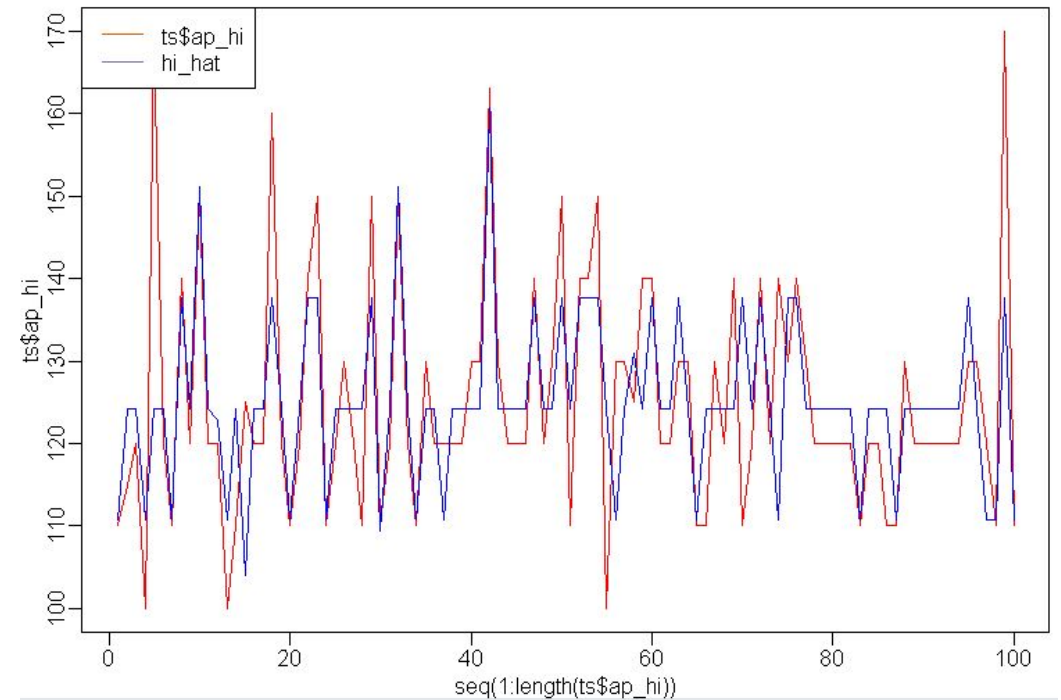
```
Call:
lm(formula = ts$ap_hi ~ ts$ap_lo)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-27.767  -4.294  -4.294   5.706  44.706
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  16.5128    11.5620   1.428   0.156
ts$ap_lo      1.3473     0.1428   9.436 2.03e-15 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 10.35 on 98 degrees of freedom
Multiple R-squared:  0.476,    Adjusted R-squared:  0.4707
F-statistic: 89.03 on 1 and 98 DF,  p-value: 2.032e-15
```



1.Формула

2. Residuals: Распределение остатков

3 Coefficients: Значение коэффициентов для подобранной модели, значимость коэффициентов модели (t – критерий)

4. Residual standard error

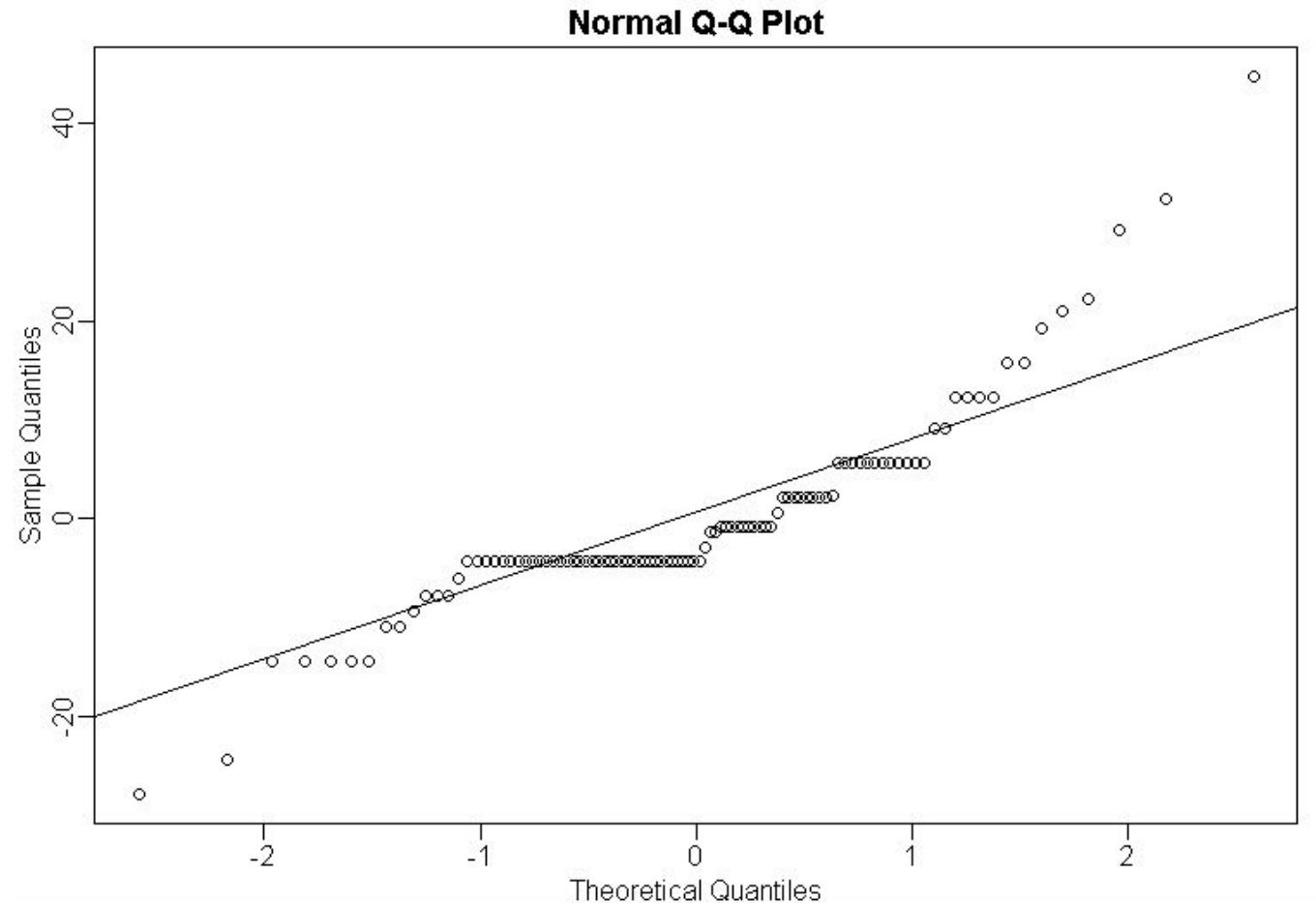
5 Multiple R-squared коэффициент детерминации

6 F-statistic: Значимость модели в целом (F- критерий)

Residuals:

Одно из важных условий для построения модели линейной регрессии является предположение, что ошибки следуют нормальному распределению

```
> qqnorm(fitm$residuals)  
> qqline(fitm$residuals)
```



Coefficients:

с помощью t-статистики Стьюдента можно проверить значимость коэффициентов построенной модели

```
> summary(fitm)

Call:
lm(formula = ts$ap_hi ~ ts$ap_lo)

Residuals:
    Min       1Q   Median       3Q      Max
-27.767  -4.294  -4.294   5.706  44.706

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  16.5128     11.5620   1.428   0.156
ts$ap_lo      1.3473      0.1428   9.436 2.03e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.35 on 98 degrees of freedom
Multiple R-squared:  0.476,    Adjusted R-squared:  0.4707
F-statistic: 89.03 on 1 and 98 DF,  p-value: 2.032e-15
```

R дает значения t-статистики, p-value и с помощью «*» отмечает наиболее значимые коэффициенты

В строке Signif.codes приведена расшифровка обозначений: например, «***» соответствуют p-value, лежащей между нулем и 0.001

Residual Standard Error:

```
> summary(fitm)

Call:
lm(formula = ts$ap_hi ~ ts$ap_lo)

Residuals:
    Min       1Q   Median       3Q      Max
-27.767  -4.294  -4.294   5.706  44.706

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  16.5128    11.5620   1.428   0.156
ts$ap_lo      1.3473     0.1428   9.436 2.03e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.35 on 98 degrees of freedom
Multiple R-squared:  0.476,    Adjusted R-squared:  0.4707
F-statistic: 89.03 on 1 and 98 DF,  p-value: 2.032e-15
```

RSE вычисляется следующим образом:

```
> rse <- sqrt( sum(residuals(fitm)^2) / fitm$df.residual )
> rse
[1] 10.35432
```

98 степеней свободы = $k - n - 1$, где k – объем выборки, n – число предикторов (в нашем случае 1 предиктор - это нижнее давление)

Multiple R-squared:

```
> summary(fitm)

Call:
lm(formula = ts$ap_hi ~ ts$ap_lo)

Residuals:
    Min       1Q   Median       3Q      Max
-27.767  -4.294  -4.294   5.706  44.706

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  16.5128    11.5620   1.428   0.156
ts$ap_lo      1.3473     0.1428   9.436 2.03e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.35 on 98 degrees of freedom
Multiple R-squared:  0.476,    Adjusted R-squared:  0.4707
F-statistic: 89.03 on 1 and 98 DF,  p-value: 2.032e-15
```

$$R_{\text{adj}}^2 = 1 - (1 - R^2) \frac{k - 1}{k - n - 1}$$

k - число наблюдений

GeekBrains n – число предикторов

R- squared – коэффициент детерминации

Показывает, какую часть изменчивости величины **y** описала построенная модель

Посчитать эту величину можно следующим образом:

```
> Rs<-cor(ts$ap_hi,ts$ap_lo)^2
> Rs
[1] 0.4760263
```

С увеличением числа предикторов коэффициент детерминации растет,

Adjusted R-squared решает эту проблему

Если добавленные новые предикторы не вносят весомого вклада в модель, то этот параметр будет падать, в противном случае расти

```
> R.adj<- 1-((1-Rs)*((100-1)/(100-1-1)))
> R.adj
[1] 0.4706797
```

F-statistic :

позволяет оценить значимость построенной модели в целом

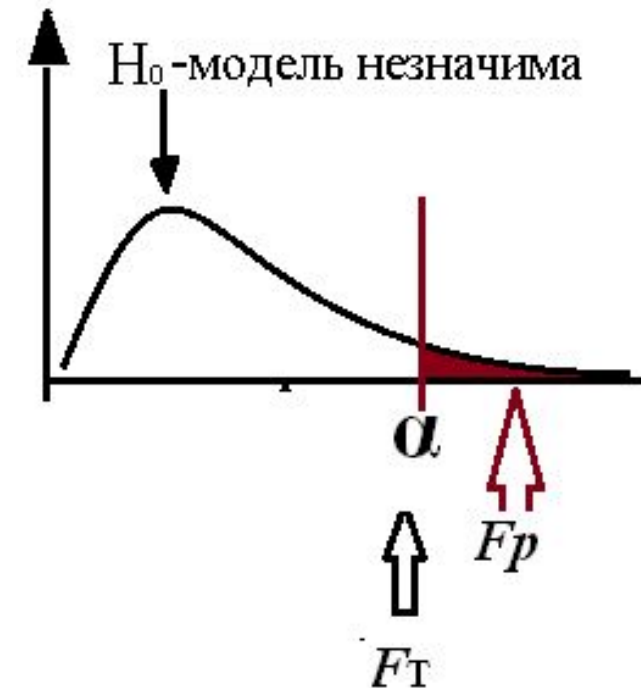
```
> summary(fitm)

Call:
lm(formula = ts$ap_hi ~ ts$ap_lo)

Residuals:
    Min       1Q   Median       3Q      Max
-27.767  -4.294  -4.294   5.706  44.706

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  16.5128    11.5620   1.428   0.156
ts$ap_lo      1.3473     0.1428   9.436 2.03e-15 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.35 on 98 degrees of freedom
Multiple R-squared:  0.476,    Adjusted R-squared:  0.4707
F-statistic: 89.03 on 1 and 98 DF, p-value: 2.032e-15
```




```
> tsn<-ts[,-c(1,2)]
> head(tsn)
      gender height weight ap_hi ap_lo cholesterol gluc smoke alco active cardio age_years
18265      1    163     52   110    70           2    1     0     0      0      0      54
25601      1    165     65   115    80           1    1     0     0      1      1      48
39400      1    165     74   120    80           1    1     0     0      1      0      64
62466      1    155     48   100    70           1    1     0     0      0      0      52
13875      1    169     68   169    80           1    1     0     0      0      1      55
61790      1    160     64   120    80           1    1     0     0      1      0      59
> fit<-lm(tsn$ap_hi~.,data = tsn)
> summary(fit)
```

```
Call:
lm(formula = tsn$ap_hi ~ ., data = tsn)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-23.166  -4.467  -1.165   3.061  41.033
```

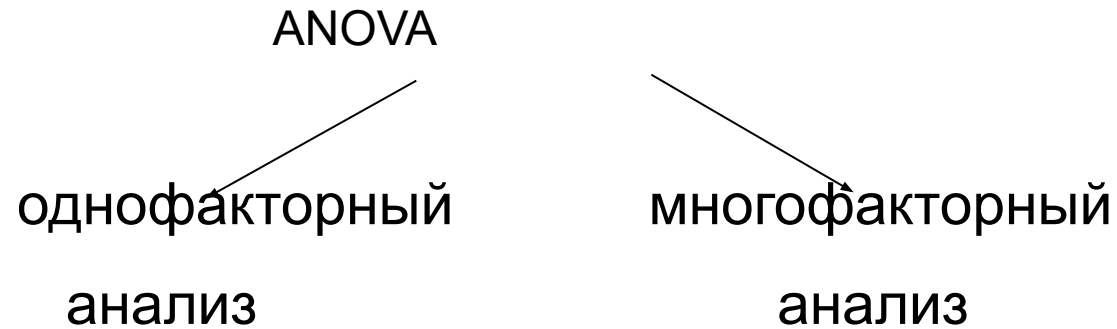
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 33.89768   29.04593   1.167   0.2463
gender      -0.94719    2.63943  -0.359   0.7206
height      -0.15278    0.16360  -0.934   0.3529
weight       0.14070    0.07647   1.840   0.0691 .
ap_lo        1.25183    0.16371   7.647 2.43e-11 ***
cholesterol -0.59561    2.30262  -0.259   0.7965
gluc         0.55254    2.48630   0.222   0.8246
smoke       -6.31950    4.56285  -1.385   0.1696
alco        -6.05406    7.56574  -0.800   0.4258
active      -1.70883    2.50435  -0.682   0.4968
cardio       4.23484    2.30739   1.835   0.0698 .
age_years    0.12601    0.16535   0.762   0.4481
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 10.12 on 88 degrees of freedom
Multiple R-squared:  0.5507,    Adjusted R-squared:  0.4945
F-statistic: 9.805 on 11 and 88 DF,  p-value: 2.379e-11
```

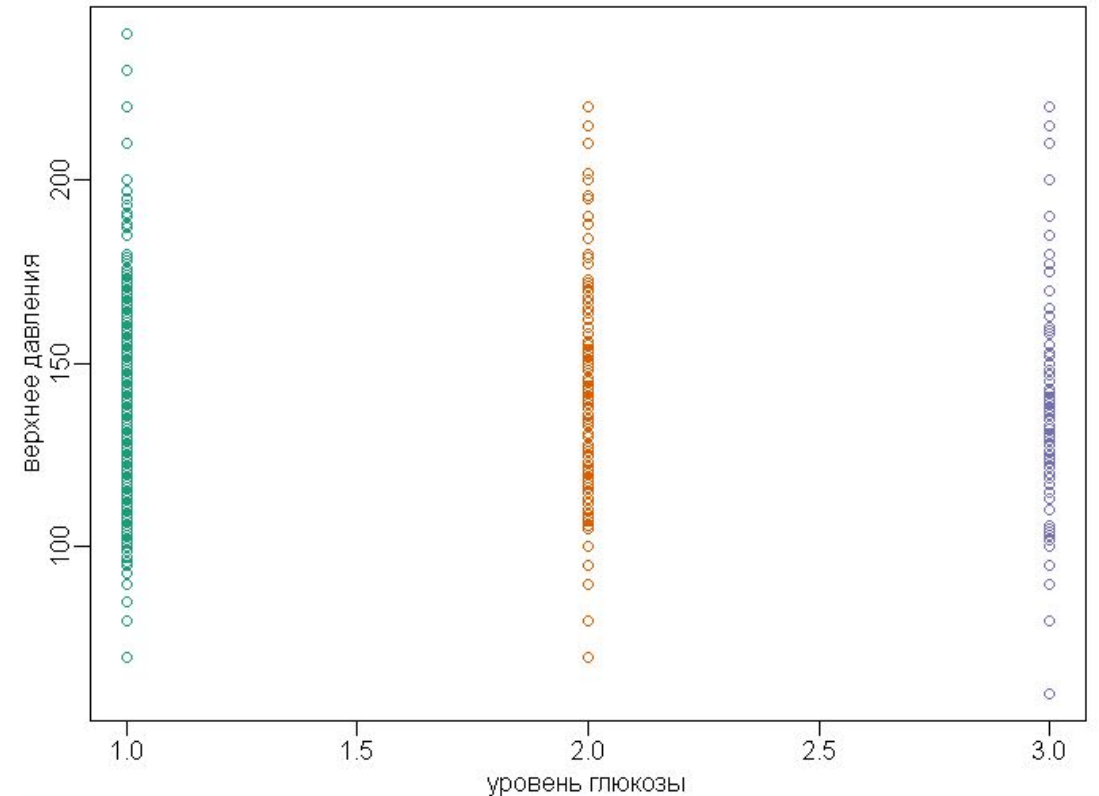
ANOVA

Дисперсионный анализ (ANOVA- analysis of variance) используется, когда мы хотим выяснить влияние одного или нескольких факторов (качественных переменных) на количественную переменную (**отклик**)



Используем, чтобы избежать множественных сравнений, которые приводят к росту вероятности ошибки первого рода

При использовании поправок на множественные сравнения, растет вероятность ошибки второго рода с ужесточением уровня значимости



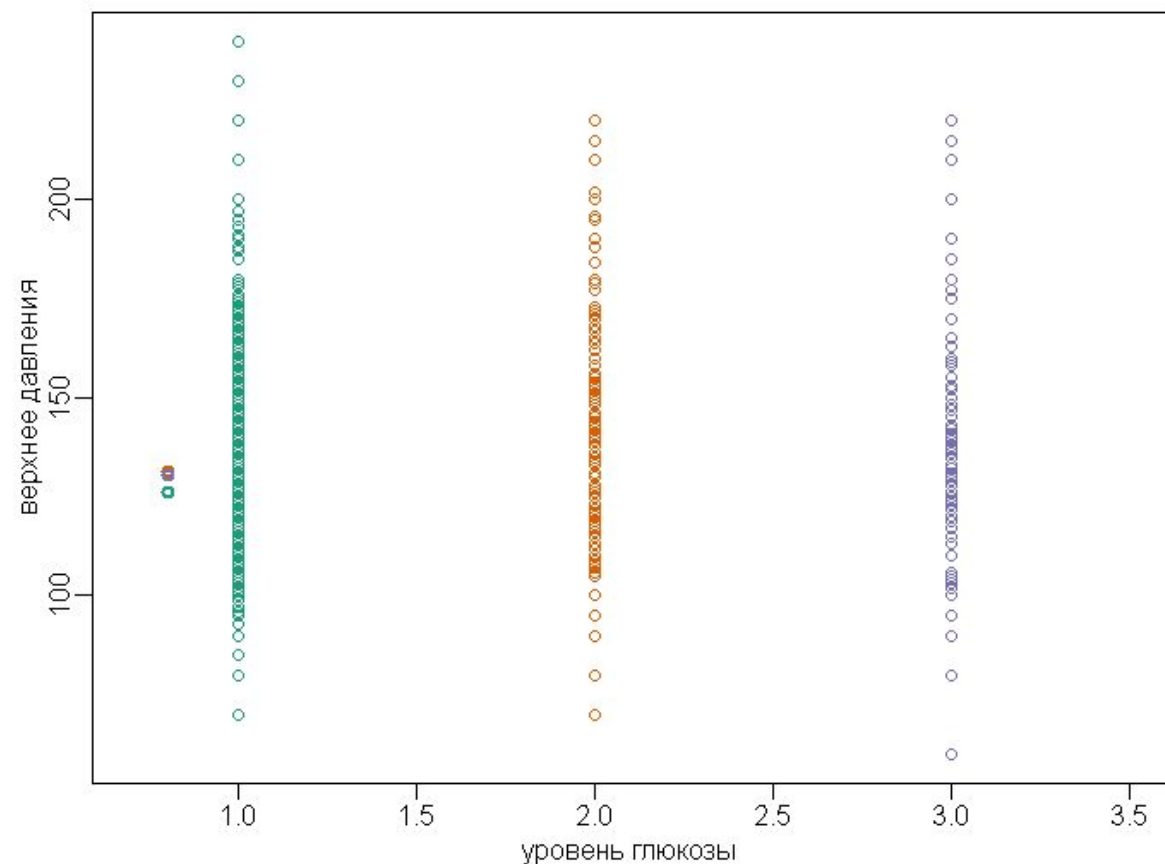
Задача сводится к сравнению средних арифметических по подгруппам

В ANOVA мы имеем дело с

- объясненной (факторной дисперсией, межгрупповой)
- необъясненной(внутригрупповой)

На данном рисунке видно ,что межгрупповая дисперсия очень мала, что предполагает, что данный фактор не оказывает влияния на числовую переменную

С помощью ANOVA мы можем проверить наше предположение



```
> plot(.tidy_set$gluc,.tidy_set$ap_hi,cex=1, col=.tidy_set$gluc,  
+       xlab="уровень глюкозы", ylab = "верхнее давления", xlim = c(0.7,3.5))  
> points(rep(0.8,3),c(m1,m2,m3), col=c(1,2,3), lwd=2)
```


Три важных условия при проведении дисперсионного анализа:

1. Случайность и независимость измерений - **ОБЯЗАТЕЛЬНОЕ**
2. Переменная-отклик следует нормальному распределению в группах
3. Гомоскедастичность дисперсий

При невыполнении 2 или 3 условия дисперсионного анализа растет вероятность принять значимые факторы за незначимые

Но ко 2 условиюanova менее чувствителен

Если одно из этих условий не выполняется, следует это отметить в конечном результате исследования

Сбалансированные и несбалансированные данные

Если данные имеют разное количество наблюдений в группе, то мы имеем дело с несбалансированными данными

```
> #работаем с выборкой
> set.seed(1)
> ind<-sample(seq(1,nrow(.tidy_set)),100)
> ind
 [1] 18235 25557 39342 62372 13851 61696 64873 45378 43202 4243 14144 12124 47176 26375 52861 34174 49274 68106 26094 53379 64175 14566
[23] 44742 8620 18346 26508 920 26252 59705 23365 33094 41159 33880 12783 56795 45886 54519 7410 49676 28230 56349 44413 53738 37958
[45] 36357 54176 1602 32753 50259 47542 32779 59103 30065 16800 4851 6826 21704 35590 45427 27917 62640 20147 31500 22808 44659 17704
[67] 32834 52578 5781 60055 23264 57592 23785 22899 32680 61208 59296 26754 53325 65898 29817 48877 27439 22318 51932 13904 48778 8347
[89] 16839 9830 16436 4043 44052 60099 53422 54682 31224 28124 55610 41486
> ts<-tidy_set[ind,]
> head(ts)
      id   age gender height weight ap_hi ap_lo cholesterol gluc smoke alco active cardio age_years
18265 26543 19799      1    163     52   110    70           2     1     0     0         0         0         54
25601 37212 17541      1    165     65   115    80           1     1     0     0         1         1         48
39400 57321 23678      1    165     74   120    80           1     1     0     0         1         0         64
62466 90766 18998      1    155     48   100    70           1     1     0     0         0         0         52
13875 20151 20278      1    169     68   169    80           1     1     0     0         0         1         55
61790 89789 21684      1    160     64   120    80           1     1     0     0         1         0         59
> table(ts$gluc)

 1   2   3 
90   5   5 
> table(ts$gender,ts$gluc)

      1   2   3 
1  59   4   3 
2  31   1   2
```

- Если размеры выборок одинаковые, то неоднородность дисперсий слабо влияет на результат. Несбалансированные данные особенно важны при неоднородности дисперсий
- Слабые отклонения от нормальности не сильно влияют на результат. В однофакторном дисперсионном анализе при больших объемах выборок можно пренебречь этим условием

Если данные имеют дисбаланс, то ANOVA становится более чувствительным к нарушениям условий его применения. Нарушение условий ведет к росту вероятности ошибки первого рода



ВЫВОД : стараемся делать одинаковые выборки

Задача: проведем исследование влияния 2-х факторов: пол и уровень глюкозы на верхнее давление

1.Берем выборки. Соблюдаем условие случайности и независимости

```
tidy_set <- dat %>% filter((ap_lo < 200 & ap_lo > 20) & (ap_hi < 300 & ap_hi > 40))
.tidy_set <- tidy_set[tidy_set$ap_hi > tidy_set$ap_lo,]
head(.tidy_set)
set.seed(1)

s1.g1 <- sample(.tidy_set$ap_hi[.tidy_set$gluc == 1 & .tidy_set$gender == 1], 20)
s2.g1 <- sample(.tidy_set$ap_hi[.tidy_set$gluc == 1 & .tidy_set$gender == 2], 20)
s2.g1

s1.g2 <- sample(.tidy_set$ap_hi[.tidy_set$gluc == 2 & .tidy_set$gender == 1], 20)
s2.g2 <- sample(.tidy_set$ap_hi[.tidy_set$gluc == 2 & .tidy_set$gender == 2], 20)

s1.g3 <- sample(.tidy_set$ap_hi[.tidy_set$gluc == 3 & .tidy_set$gender == 1], 20)
s2.g3 <- sample(.tidy_set$ap_hi[.tidy_set$gluc == 3 & .tidy_set$gender == 2], 20)
```

На этом этапе цель : построить дата фрейм, как показано справа

[illegible]

```
> head(anovaframe,25)
      sam_s gender.new gluc.new
1      110          1         1
2      120          1         1
3      100          1         1
4      120          1         1
5      120          1         1
6      120          1         1
7      120          1         1
8      130          1         1
9      120          1         1
10     110          1         1
11     140          1         1
12     120          1         1
13     120          1         1
14     120          1         1
15     140          1         1
16     120          1         1
17     160          1         1
18     120          1         1
19     100          1         1
20     110          1         1
21     120          2         1
22     120          2         1
23     130          2         1
24     120          2         1
25     120          2         1
```

```
> table(anovaframe$gender.new,anovaframe$gluc.new)
```

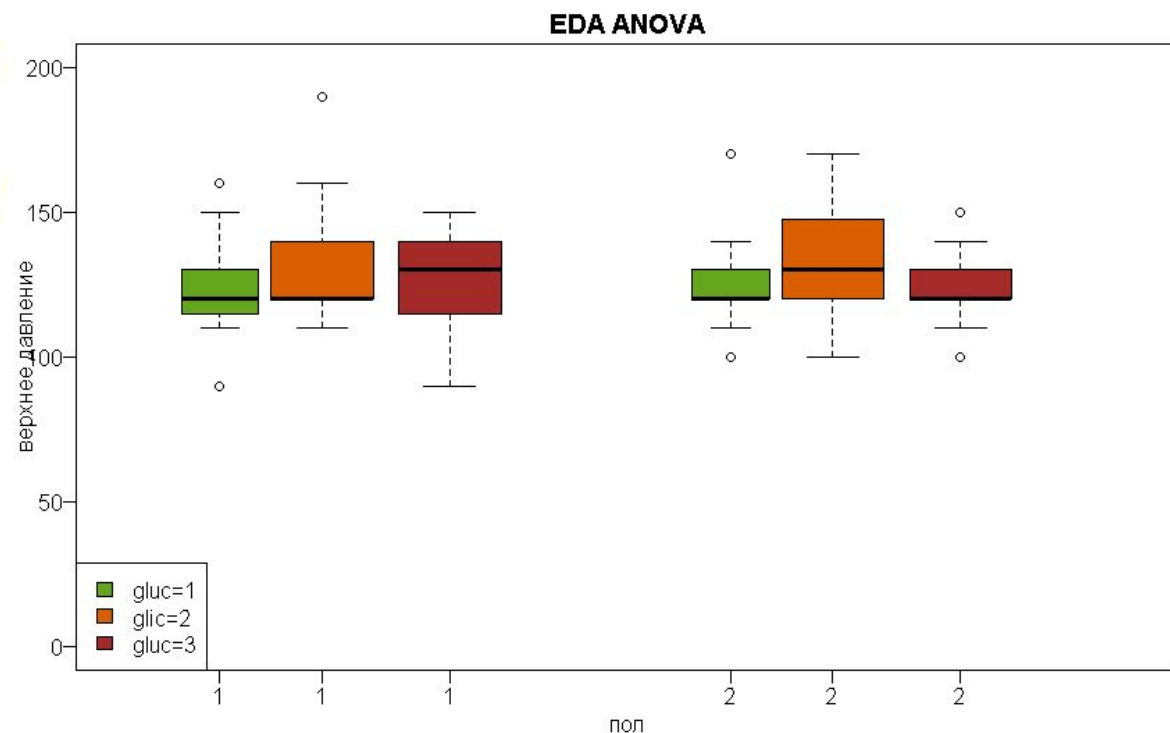
	1	2	3
1	20	20	20
2	20	20	20

2. Разведочный анализ

2.1 Для наглядности представим данные графически

```
boxplot(sam_s ~ gender.new, data = anovaframe,  
        boxwex = 0.15, at = 1:2-0.3,  
        subset = gluc.new == "1", col = "5",  
        main = "EDA ANOVA",  
        xlab = "пол",  
        ylab = "верхнее давление",  
        xlim = c(0.5, 2.5), ylim = c(0, 200))  
boxplot(sam_s ~ gender.new, data = anovaframe, add = TRUE,  
        boxwex = 0.2, at = 1:2-0.1 ,  
        subset = gluc.new == "2", col = "2")  
boxplot(sam_s ~ gender.new, data = anovaframe, add = TRUE,  
        boxwex = 0.2, at = 1:2 + 0.15,  
        subset = gluc.new == "3", col = "brown")  
legend("bottomleft", c("gluc=1", "gluc=2", "gluc=3"),  
      fill = c("5", "2","brown"))
```

Видим небольшую неоднородность дисперсий



2.2 Помимо визуальной оценки однородности дисперсий (п.2.1) проверим гомоскедастичность с помощью специальных критериев

Распространенные критерии:

Критерий	Функция	Условия применения
F- критерий	<code>var.test()</code>	1)Для сравнения 2-х дисперсий 2)Возможен разный объем выборок
Критерий Бартлетта	<code>bartlett.test()</code>	1) Для множественных сравнений 2) Объемы выборки могут быть различны, но не менее 3 3) Должно соблюдаться условие нормальности. Тест очень чувствительный к нарушению этого условия
Критерий Кохрена	<code>cochran.test()</code> <code>package "outliers"</code>	1) Для множественных сравнений 2) Одинаковый объем выборок
Критерий Левенэ	<code>leveneTest ()</code> <code>package "car"</code>	Аналог критерий Бартлетта, но считается менее чувствительным к нарушению условия нормальности

Воспользуемся критерием Бартлетта.

Условие нормальности соблюдается хорошо

```
> bartlett.test(list(s1.g1,s2.g1,s1.g2,s2.g2,s1.g3,s2.g3))
```

```
Bartlett test of homogeneity of variances
```

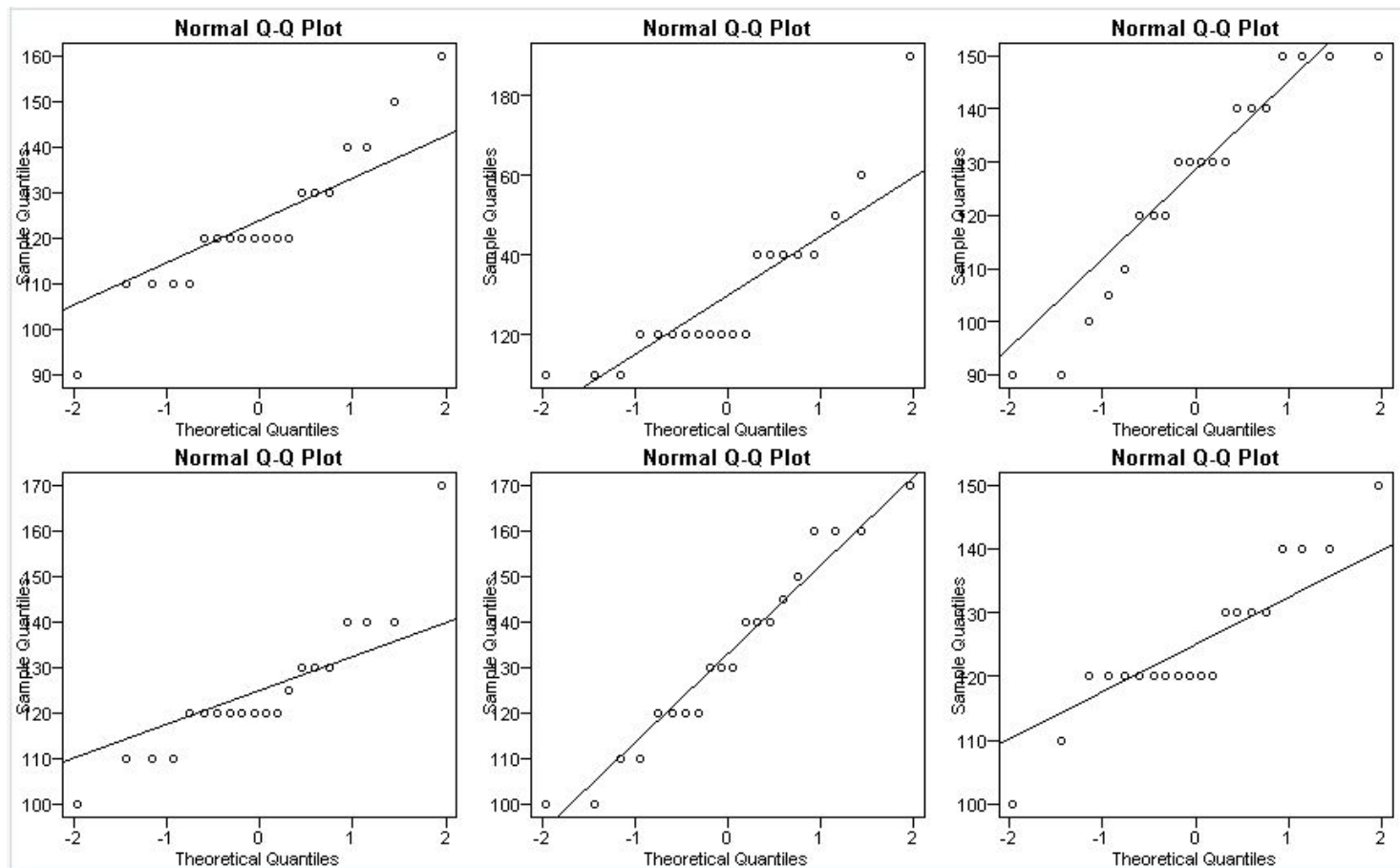
```
data: list(s1.g1, s2.g1, s1.g2, s2.g2, s1.g3, s2.g3)  
Bartlett's K-squared = 8.4374, df = 5, p-value = 0.1337
```

Принимаем нулевую гипотезу на уровне значимости 0.05 . Статистически значимых различий между дисперсиями выборок нет

Все условия соблюдены. Теперь можно приступить непосредственно к самому дисперсионному анализу

2.3. Проверим предположение о нормальности распределений с помощью qq-графика

```
mypar(2,3)
qqnorm(s1.g1)
qqline(s1.g1)
qqnorm(s1.g2)
qqline(s1.g2)
qqnorm(s1.g3)
qqline(s1.g3)
qqnorm(s2.g1)
qqline(s2.g1)
qqnorm(s2.g2)
qqline(s2.g2)
qqnorm(s2.g3)
qqline(s2.g3)
```



Есть совсем небольшие отклонения. Нас это усаивает. Тем более ,что мы имеем одинаковые объемы выборок

Сбалансированные данные НЕ влияют на порядок включения факторов в модель

```
> summary(aov(sam_s~gender.new+gluc.new+gender.new:gluc.new, data= anovaframe))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender.new	1	17	16.87	0.055	0.814
gluc.new	1	45	45.00	0.148	0.701
gender.new:gluc.new	1	31	31.25	0.103	0.749
Residuals	116	35290	304.22		

```
> summary(aov(sam_s~gluc.new+gender.new+gluc.new:gender.new, data= anovaframe))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gluc.new	1	45	45.00	0.148	0.701
gender.new	1	17	16.87	0.055	0.814
gluc.new:gender.new	1	31	31.25	0.103	0.749
Residuals	116	35290	304.22		

Чтобы не прописывать эффект взаимодействия, факторы в модели указывать с помощью знака «*»

```
> summary(aov(sam_s~gender.new*gluc.new, data= anovaframe))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender.new	1	17	16.87	0.055	0.814
gluc.new	1	45	45.00	0.148	0.701
gender.new:gluc.new	1	31	31.25	0.103	0.749
Residuals	116	35290	304.22		

Интерпретация результата

```
> summary(aov(sam_s~gender.new*gluc.new, data= anovaframe))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender.new	1	17	16.87	0.055	0.814
gluc.new	1	45	45.00	0.148	0.701
gender.new:gluc.new	1	31	31.25	0.103	0.749
Residuals	116	35290	304.22		

Взаимодействие факторов «уровень глюкозы» и «пол» , а также сами факторы не оказывают значимого эффекта на давление пациента на уровне значимости 0.05

Итоги:

1. Научились подбирать линейную модель с помощью функции `lm()`
2. Интерпретировать результат: оценивать значимость коэффициентов (критерий t) и самой модели в целом (критерий F)
3. Рассмотрели чем отличаются параметры R и R_{adj}
4. Рассмотрели условия для проведения ANOVA
5. Составили общую схему действий для ANOVA при сбалансированных и несбалансированных данных
6. Изучили различные методы в R проверки выборок на однородность дисперсий
7. Интерпретировали конечный результат статистического анализа