Дз регрессия

Маслова И

2022-12-24

# Задание 1

## Часть 1

pr <- rmvnorm(n=100, mean = c(85.5, 5.4), sigma = matrix(c(13.25, 2.63, 2.63, 1.45), ncol = 2))  
pr <- as.data.frame(pr)  
cov2cor(matrix(c(13.25, 2.63, 2.63, 1.45), ncol = 2))

## [,1] [,2]  
## [1,] 1.0000000 0.6000173  
## [2,] 0.6000173 1.0000000

summary(lm(pr$V1 ~ pr$V2))

##   
## Call:  
## lm(formula = pr$V1 ~ pr$V2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.1806 -2.3529 0.3597 1.8112 7.6985   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 77.5461 1.4956 51.851 < 2e-16 \*\*\*  
## pr$V2 1.5093 0.2683 5.624 1.76e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.059 on 98 degrees of freedom  
## Multiple R-squared: 0.244, Adjusted R-squared: 0.2363   
## F-statistic: 31.63 on 1 and 98 DF, p-value: 1.761e-07

(Проверка зависимости коэффициента корреляции)

pr2 <- rmvnorm(n=100, mean = c(85.5, 5.4), sigma = matrix(c(13.25, 0.1, 0.1, 1.45), ncol = 2))  
pr2 <- as.data.frame(pr2)  
cor(pr2$V1, pr2$V2)

## [1] -0.07096664

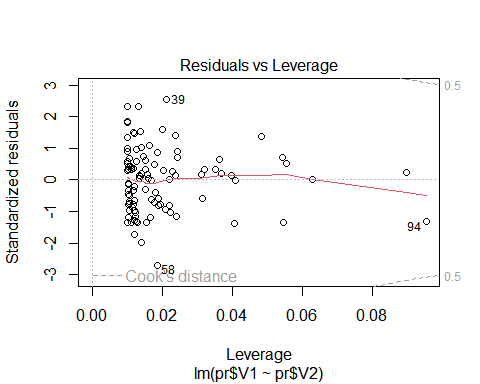
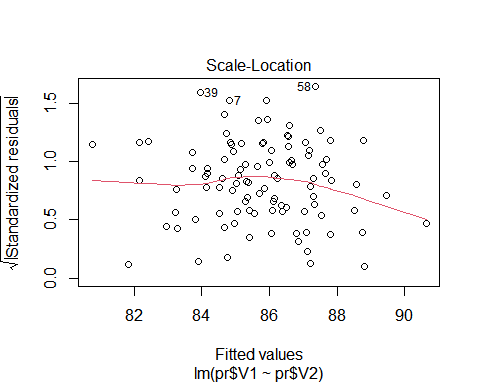
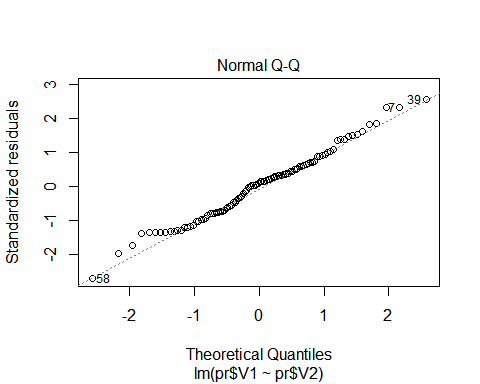
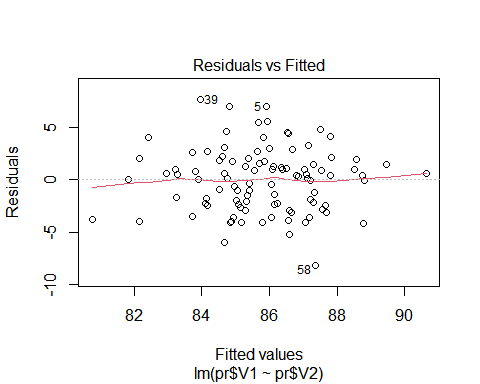
summary(lm(pr2$V1 ~ pr2$V2))

##   
## Call:  
## lm(formula = pr2$V1 ~ pr2$V2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.2260 -3.0787 -0.1748 2.1830 9.7388   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 86.7465 1.6198 53.553 <2e-16 \*\*\*  
## pr2$V2 -0.2086 0.2962 -0.704 0.483   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.592 on 98 degrees of freedom  
## Multiple R-squared: 0.005036, Adjusted R-squared: -0.005116   
## F-statistic: 0.4961 on 1 and 98 DF, p-value: 0.4829

Полученные нами оценки связаны с выборочными характеристиками. Мат ожидание в наибольшей степени влияет на оценку параметра b (Intercept). Увеличение дисперсии - дает более широкий разброс значений получаемым нами оценкам. Коэффициент корреляции - в основном влияет на оценку параметра a.

## Часть 2

plot(lm(pr$V1 ~ pr$V2))



f <- lm(pr$V1 ~ pr$V2)  
res1 <- residuals(f,type="response")  
shapiro.test(res1)

##   
## Shapiro-Wilk normality test  
##   
## data: res1  
## W = 0.98636, p-value = 0.396

Основываясь на полученных результатах (p-value, график) можно сделать вывод, что остатки распределены нормально.

## Часть 3

pr$V3 <- runif(100, min = 0, max = 100)  
summary(lm(pr$V1 ~ pr$V2 + pr$V3))

##   
## Call:  
## lm(formula = pr$V1 ~ pr$V2 + pr$V3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.1726 -2.3678 0.3597 1.7998 7.7165   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.753e+01 1.557e+00 49.797 < 2e-16 \*\*\*  
## pr$V2 1.509e+00 2.699e-01 5.591 2.08e-07 \*\*\*  
## pr$V3 4.261e-04 9.895e-03 0.043 0.966   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.075 on 97 degrees of freedom  
## Multiple R-squared: 0.244, Adjusted R-squared: 0.2285   
## F-statistic: 15.66 on 2 and 97 DF, p-value: 1.28e-06

Коэф детерминации и модефицированный коэф детерминации меняют свои значения на сотые доли в зависимости от добавленных значений.

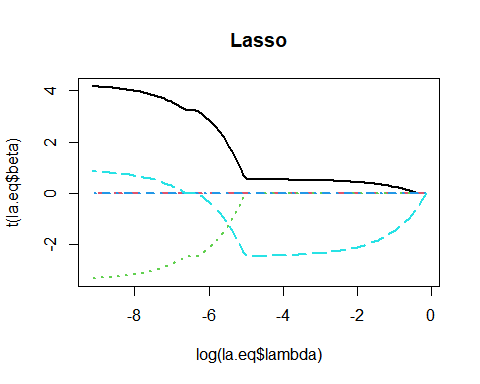
# Задание 2

## Часть 1

n <- 100  
x1 <- rnorm(n, mean = 0.3)  
x2 <- x1 \* (1+runif(n, min = 0, max = 0.005))  
x3 <- x1 + x2 \* +rnorm(n, sd= 0.3)   
x4 <- x3 \*(1+ runif(n, min = 0, max = 0.002))  
x5 <- x4 - x1 \*(1+ runif(n, min = 0, max = 0.2))  
y <- x1 + 5\*x2 +3\*x3 - 8\*x4 + 2\* x5 + rnorm(n, sd=1)  
  
summary(lm(y~x1+x2+x3+x4 +x5))

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.2616 -0.5995 0.1026 0.6560 1.9278   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -0.07482 0.10618 -0.705 0.483  
## x1 17.59521 64.37284 0.273 0.785  
## x2 -13.71437 64.41425 -0.213 0.832  
## x3 -154.55658 168.43672 -0.918 0.361  
## x4 151.43213 168.75577 0.897 0.372  
## x5 0.47669 1.82031 0.262 0.794  
##   
## Residual standard error: 0.9789 on 94 degrees of freedom  
## Multiple R-squared: 0.5286, Adjusted R-squared: 0.5035   
## F-statistic: 21.08 on 5 and 94 DF, p-value: 4.455e-14

X <- matrix(c(x1, x2, x3, x4, x5), ncol = 5)  
la.eq <- glmnet(X,y, family = 'gaussian', intercept = F, alpha = 1)  
matplot(log(la.eq$lambda), t(la.eq$beta), type = 'l', main='Lasso', lwd=2)



la.eq <- glmnet(X,y, family = 'gaussian', intercept = F, alpha = 1, , lambda = 0.2)  
predict(la.eq, X, type = "coefficients")

## 6 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) .   
## V1 0.3836555  
## V2 .   
## V3 .   
## V4 .   
## V5 -1.9288312

Коэффициенты сильно изменились. Пятый коэффициент принимает отрицательные значения; Intercept, третий и четвертый - обнулиись; первый и второй - сильно приблизились к нулю.

## Часть 2

Активная клеточная масса, костная масса и вес являются сильно коррелирующими переменными( коэф корреляции принимает значение +- 0.9). Первый столбец в нашем примере соответствует значениям активной клеточной массы в килограммах, второй - костной массе в килограммах, третий - вес в килограммах.

dop <- rmvnorm(n=100, mean = c(24.3, 3.3), sigma = matrix(c(5, 1.7, 1.7, 0.7), ncol = 2))  
dop <- as.data.frame(dop)  
dop$ves <- 2\*dop$V1 + 10\*dop$V2 +rnorm(n, mean = 5, sd = 5)   
cov2cor(matrix(c(5, 1.7, 1.7, 0.7), ncol = 2))

## [,1] [,2]  
## [1,] 1.0000000 0.9086882  
## [2,] 0.9086882 1.0000000

cor(dop$ves, dop$V2)

## [1] 0.9138285

cor(dop$ves, dop$V1)

## [1] 0.8660678

Посмотрим полученные коэффициенты:

summary(lm(ves ~ V1 + V2, dop))

##   
## Call:  
## lm(formula = ves ~ V1 + V2, data = dop)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.6219 -2.9139 -0.4416 2.2052 14.0750   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.037 9.934 1.312 0.193   
## V1 1.405 0.589 2.385 0.019 \*   
## V2 12.032 1.571 7.656 1.43e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.184 on 97 degrees of freedom  
## Multiple R-squared: 0.8442, Adjusted R-squared: 0.841   
## F-statistic: 262.8 on 2 and 97 DF, p-value: < 2.2e-16

# Задание 3

Sep <- data.frame(Neu = rnorm(201, mean = 80, sd = 5) %>% round(0), Ly = rnorm(201, mean = 20, sd = 5)%>%round(0))  
Sep$p <- ifelse(Sep$Neu/Sep$Ly < 3, 0, (Sep$Neu/Sep$Ly - 3)/6 )  
Sep$p <- ifelse(Sep$Neu/Sep$Ly > 9, 1, Sep$p ) %>% round(2)

mod <- glm(p ~ Neu + Ly, Sep, family = gaussian)  
summary(glm(p ~ Neu + Ly, Sep, family = gaussian))

##   
## Call:  
## glm(formula = p ~ Neu + Ly, family = gaussian, data = Sep)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.10121 -0.05654 -0.03122 0.01466 0.33648   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.533620 0.106432 5.014 1.18e-06 \*\*\*  
## Neu 0.005629 0.001306 4.311 2.56e-05 \*\*\*  
## Ly -0.037823 0.001230 -30.752 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.008799674)  
##   
## Null deviance: 10.1993 on 200 degrees of freedom  
## Residual deviance: 1.7423 on 198 degrees of freedom  
## AIC: -375.95  
##   
## Number of Fisher Scoring iterations: 2

var <- data.frame(Neu = 90, Ly = 15)  
predict(mod, var)

## 1   
## 0.4728651

При подсчете вероятности вручную при показателях Neu = 90, Ly = 15: p = 0.5 Наша модель предсказывает вероятность близкую к данной: p = 0.48 (при округлении)