**Задание. Линейная регрессия – цена автомобиля**

Для выполнения задания были загружены данные из файла

“kuiper.tsv” и пропущены первые четыре строки поскольку они не содержат

данных.

install.packages(c('rstudioapi'),repos='https://cran.r-project.org/')

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

df <- read.delim(file='kuiper.tsv', skip= 4, stringsAsFactors=TRUE)

Далее проверяем, что данные прочитаны корреткно

head(df)

В результате получаем

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| row | Price | Mileage | Make | Model | Trim | Type | Cylinder | Liter | Doors | Cruise | Sound | Leather |
| 1 | 17314.1 | 8221 | Buick | Century | Sedan 4D | Sedan | 6 | 3.1 | 4 | 1 | 1 | 1 |
| 2 | 17542.04 | 9135 | Buick | Century | Sedan 4D | Sedan | 6 | 3.1 | 4 | 1 | 1 | 0 |
| 3 | 16218.85 | 13196 | Buick | Century | Sedan 4D | Sedan | 6 | 3.1 | 4 | 1 | 1 | 0 |
| 4 | 16336.91 | 16342 | Buick | Century | Sedan 4D | Sedan | 6 | 3.1 | 4 | 1 | 0 | 0 |
| 5 | 16339.17 | 19832 | Buick | Century | Sedan 4D | Sedan | 6 | 3.1 | 4 | 1 | 0 | 1 |
| 6 | 15709.05 | 22236 | Buick | Century | Sedan 4D | Sedan | 6 | 3.1 | 4 | 1 | 1 | 0 |

Сравнив с оригиналом, убедились, что данные загружены верно.

Далее построили график цены от номера наблюдения

plot(df$Price)

Chart

Description automatically generated

Рис.1. График цены от индекса

И гистограмму

hist(df$Price)

Chart, histogram

Description automatically generated

Рис.2 Гистограмма частоты от цены

Далее посмотрели на характеристики переменных

summary(df)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Price | Mileage | Make | Model | Trim | Type | Cylinder | Liter | Doors | Cruise | Sound | Leather |
| Min. | 8639 | 266 | Buick 80 | AVEO 60 | Sedan 4D 170 | Convertible 50 | 4 | 1.6 | 2 | 0 | 0 | 0 |
| 1stQu. | 14273 | 14624 | Cadillac 80 | Cavalier 60 | Coupe 2D 50 | Coupe 140 | 4 | 2.2 | 4 | 1 | 0 | 0 |
| Median | 18025 | 20914 | Chevrolet 320 | Malibu 60 | LSSedan4D50 | Hatchback 60 | 6 | 2.8 | 4 | 1 | 1 | 1 |
| Mean | 21343 | 19832 | Pontiac 150 | Cobalt 50 | LSCoupe 2D 30 | Sedan 490 | 5.269 | 3.037 | 3.527 | 0.7525 | 0.6791 | 0.7239 |
| 3rdQu. | 26717 | 25213 | SAAB 114 | Ion 50 | LTSedan 4D 30 | Wagon 64 | 6 | 3.8 | 4 | 1 | 1 | 1 |
| Max. | 70755 | 50387 | Saturn 60 | 9\_3HO 40 | AeroSedan 4D 20 | | 8 | 6 | 4 | 1 | 1 | 1 |

table(df$Make)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Buick | Cadillac | Chevrolet | Pontiac | SAAB | Saturn |
| 80 | 80 | 320 | 150 | 114 | 60 |

unique(df$Make)

[1] Buick Cadillac Chevrolet Pontiac SAAB Saturn

Levels: Buick Cadillac Chevrolet Pontiac SAAB Saturn

table(df$Trim)

\*Страшная таблица и поэтому её тут не будет\*

table(df$Type)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Convertible | Coupe | Hatchback | Sedan | Wagon |  |
|  | 50 | 140 | 60 | 490 | 64 |

table(df$Cylinder)

|  |  |  |
| --- | --- | --- |
| 4 | 6 | 8 |
| 394 | 310 | 100 |

table(df$Liter)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.6 | 1.8 | 2 | 2.2 | 2.3 | 2.5 | 2.8 | 3 | 3.1 | 3.4 | 3.5 | 3.6 | 3.8 | 4.6 | 5.7 | 6 |
| 60 | 30 | 60 | 190 | 50 | 4 | 10 | 10 | 10 | 10 | 80 | 30 | 160 | 60 | 20 | 20 |

table(df$Leather)

|  |  |
| --- | --- |
| 0 | 1 |
| 222 | 582 |

table(df$Doors)

|  |  |
| --- | --- |
| 2 | 4 |
| 190 | 614 |

table(df$Cruise)

|  |  |
| --- | --- |
| 0 | 1 |
| 199 | 605 |

table(df$Sound)

|  |  |
| --- | --- |
| 0 | 1 |
| 258 | 546 |

table(df$Make, df$Model)

\*Страшная таблица и поэтому её тут не будет\*

table(df$Liter, df$Cylinder)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 4 | 6 | 8 |
| 1.6 | 60 | 0 | 0 |
| 1.8 | 30 | 0 | 0 |
| 2 | 60 | 0 | 0 |
| 2.2 | 190 | 0 | 0 |
| 2.3 | 50 | 0 | 0 |
| 2.5 | 4 | 0 | 0 |
| 2.8 | 0 | 10 | 0 |
| 3 | 0 | 10 | 0 |
| 3.1 | 0 | 10 | 0 |
| 3.4 | 0 | 10 | 0 |
| 3.5 | 0 | 80 | 0 |
| 3.6 | 0 | 30 | 0 |
| 3.8 | 0 | 160 | 0 |
| 4.6 | 0 | 0 | 60 |
| 5.7 | 0 | 0 | 20 |
| 6 | 0 | 0 | 20 |

table(df$Type, df$Doors)

|  |  |  |
| --- | --- | --- |
|  | 2 | 4 |
| Convertible | 50 | 0 |
| Coupe | 140 | 0 |
| Hatchback | 0 | 60 |
| Sedan | 0 | 490 |
| Wagon | 0 | 64 |

Можно заметить что тип машины и двери однозначно определяются друг от друга и также литры от цилиндров. Далее было проверено ещё несколько комбинаций результат указан комментарием

table(df$Make, df$Model) # неоднозначно

table(df$Liter, df$Cylinder) #однозначно

table(df$Type, df$Doors) #однозначно

table(df$Type, df$Liter) #неоднозначно

table(df$Make, df$Trim) #неоднозначно

table(df$Make, df$Type) #неоднозначно

table(df$Make, df$Doors) #неоднозначно

table(df$Make, df$Sound) #неоднозначно

table(df$Cylinder, df$Doors) #неоднозначно

table(df$Cylinder, df$Type) #неоднозначно

table(df$Doors, df$Liter) #неоднозначно

Я предполагаю, что цена сильнее зависит от литража и цилиндров. Поскольку нужно выбрать одну количественную переменную, то мы берём литры.

reg <- lm(log(Price) ~ Liter, data=df)

Смотрим результаты по нашей регрессии

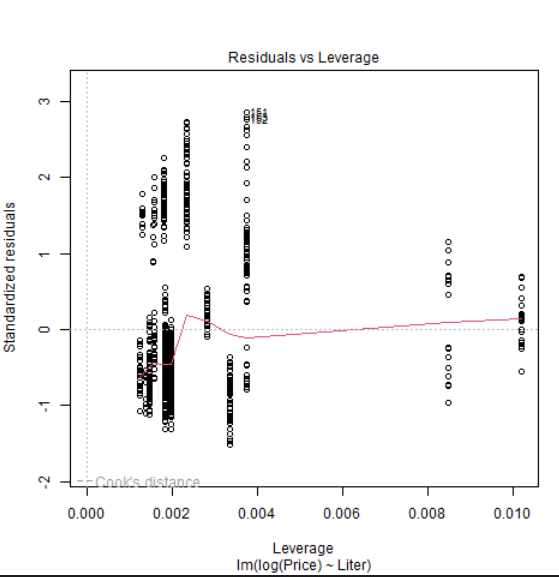
summary(reg)

Text

Description automatically generated

Рис. 3. Основные статистические показатели для нашей регрессии

Как можно видеть на рис.3 мы выбрали достаточно неплохую переменную.



sigma(reg)

deviance(reg)

deviance(lm(log(Price) ~ 1, data = df))

r\_quad <- 1 - deviance(reg) / deviance(lm(log(Price) ~ 1, data=df))

Остаточное квадратное отклонение регрессии равно 0.3312088. Остаточное квадратное отклонение регрессии только от цены 0.4101115. Отклонение подогнанного объекта регрессии равно 87.97883. Отклонение подогнанного объекта регрессии только от цены равно 135.0577. R­­2 регрессии равен 0.348583580045154.

Строим точечную диаграмму логарифма цен от литража и проводим abline по регрессии.

plot(log(Price) ~ Liter, data=df, asp=1)

abline(reg, col=6, lwd=2)

Chart

Description automatically generated

При построении линии из нуля с наклоном 1 мы не видим эту линию на графике.

Построим точечную диаграмму остатков от расчётных значений и добавим туда линию нуля.

plot(resid(reg) ~ fitted(reg), data=df, asp = 1)

abline(h=0, col=4, lwd=4)

grid()

Chart, box and whisker chart

Description automatically generated

Построили таблицу корреляций и точечные диаграммы всех пар количественных переменных.

A picture containing text, electronics

Description automatically generated

Рис. 4. Таблица корреляции и точечные диаграммы

всех пар количественных переменных

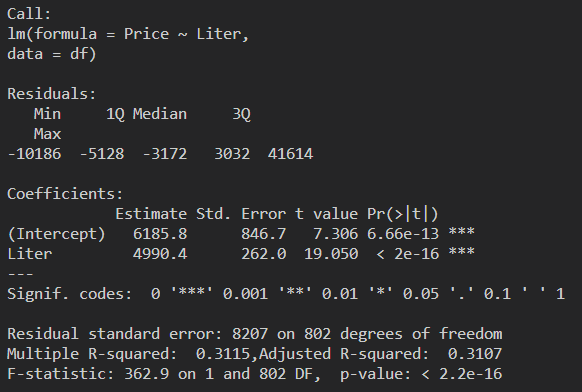
По таблице заметно коррелирует цилиндры, литры, круиз. Понятно по коэффициенту корреляции, по графикам ничего непонятно.

Строим такую же регрессию, но уже без логарифмирования.

regno <- lm(Price ~ Liter, data=df)

summary(regno)

deviance(regno)





Строим регрессию логарифма цены от всех количественных и индикаторных переменных. Вместо количества дверей мы берём индикаторную переменную для 2-ух дверей.

reg\_big <- lm(log(Price) ~ Mileage + Cylinder +

Liter + Doors + Cruise + Sound + Liter, data = df.num)

summary(reg\_big)

Text

Description automatically generated

plot(log(df$Price) ~ fitted(reg\_big))

abline(c(0,1), col =2, lwd =2)

Chart, scatter chart

Description automatically generated

confint(reg\_big)

plot(resid(reg\_big) ~ fitted(reg\_big), col = df$Cylinder, legend("topright", pt.bg = 'blue', lty=1))

abline(h=0, col =2, lwd =2)

|  |  |  |
| --- | --- | --- |
|  | 2.50% | 97.50% |
| (Intercept) | 9.18E+00 | 9.47E+00 |
| Mileage | -9.93E-06 | -4.85E-06 |
| Cylinder | 2.25E-02 | 1.31E-01 |
| Liter | 8.35E-03 | 1.46E-01 |
| Doors | -5.84E-02 | -7.40E-03 |
| Cruise | 2.48E-01 | 3.52E-01 |
| Sound | -1.16E-01 | -2.65E-02 |

Построили точечную диаграмму остатков от расчетных значений, добавьте туда линию нуля. Цвет точек задали в зависимости от Cylinder. То же самое с Make и Type. Почему-то некорректно отработал legend(…). Поправить не удалось.

plot(resid(reg\_big) ~ fitted(reg\_big), col = df$Cylinder)

legend("bottomright", lty = 1, col = df$Cylinder, legend = unique(df$Cylinder))

abline(h=0, col =2, lwd =2)

Chart, scatter chart

Description automatically generated

plot(resid(reg\_big) ~ fitted(reg\_big), col = df$Make)

legend("bottomright", lty = 1, col = df$Make, legend = unique(df$Make))

abline(h=0, col =2, lwd =2)

Chart, scatter chart

Description automatically generated

plot(resid(reg\_big) ~ fitted(reg\_big), col = df$Type)

legend("bottomright", lty = 1, col = df$Type, legend = unique(df$Type))

abline(h=0, col =2, lwd =2)

Chart, scatter chart

Description automatically generated

Удаляя по одной «наименее значимой» переменной построим серию регрессий со всё меньшим числом переменных.

reg\_big <- lm(log(df.num$Price) ~ df.num$Mileage + df.num$Cylinder +

df.num$Doors + df.num$Cruise + df.num$Sound + df.num$Liter + df.num$Leather)

summary(reg\_big)

Text

Description automatically generated

Убираем двери

reg\_big <- lm(log(df.num$Price) ~ df.num$Mileage + df.num$Cylinder +

df.num$Cruise + df.num$Sound + df.num$Liter + df.num$Leather)

summary(reg\_big)

Text

Description automatically generated

Теперь убираем цилиндры

reg\_big <- lm(log(df.num$Price) ~ df.num$Mileage + df.num$Cruise +

df.num$Sound + df.num$Liter + df.num$Leather)

summary(reg\_big)

Text

Description automatically generated

И последняя переменная — это аудиосистема

reg\_big <- lm(log(df.num$Price) ~ df.num$Mileage + df.num$Cruise +

df.num$Liter + df.num$Leather)

summary(reg\_big)

Text

Description automatically generated

Как результат стандартная ошибка увеличилась, R2 уменьшился, F- статистика увеличилась.