**Файл «cortest.R»**

**df <- mtcars**

**cor.test(x = df$mpg, y = df$hp)** #корреляция между двумя количественными переменными (милли на галлон топлива и количество лошадиных сил), т.к. p<0.05 отклоняем нулевую гипотезу об отсутствии взаимосвязи между переменными, это подтверждается и самим значением коэффициента корреляции =-0.77 (корреляция сильная и при этом отрицательная). По умолчанию вычисляется коэффициент корреляции Пирсона, но при желании его можно поменять на коэффициент Спирмена или Тау-Кендалла (с помощью параметра method).

**fit <- cor.test(x = df$mpg, y = df$hp)**

**cor.test(~ mpg + hp, df)** #тоже самое через формулу

**str(fit)**

**fit$p.value**

**plot(x = df$mpg, y = df$hp)** #построим график зависимости между переменными, видим, что зависимость отрицательная и вначале линейная, а под конец отличается от линейной

**ggplot(df, aes(x = mpg, y = hp, col = factor(cyl)))+**

**geom\_point(size = 5)+**

**facet\_grid(. ~ am)** # диаграмма рассеяния, в которой цветом обозначены значения переменной количество цилиндров

###########################################

Попарный корреляционный анализ

**df <- mtcars**

**df\_numeric <- df[, c(1,3:7)] #выбираем только количественные переменные**

**pairs(df\_numeric)** #попарный корреляционный анализ между парами количественных переменных в виде диаграммы рассеяния

**cor(df\_numeric)** # корреляции между всеми переменными в виде таблицы, в отличии от cor.test выводит только коэффициент корреляции без дополнительной информации

**fit <- corr.test(df\_numeric)** # corr.test используют для попарных корреляций между переменными с возможностью просматривать доп. информацию, например p-value

**fit$r** # сами значения коэффициентов корреляции

**fit$p** # уровень значимости

**fit$adjust** #метод поправки на множественные сравнения

**Файл «simple\_regr.R»**

**df <- mtcars**

**df\_numeric <- df[,c(1,3:7)]**

**fit <- lm(mpg ~ hp, df)** #исследуем зависимость между переменной мили на галлон (mpg) и лошадиные силы (hp) с помощью линейной регрессии (функция lm), здесь mpg – зависимая переменная, а hp – независимая.

**summary(fit)** #получаем более детальное описание с получением описательных статистик по остаткам (блок Residuals), а также статистику по коэффициентам:

*Coefficients:*

*Estimate Std. Error t value Pr(>|t|)*

*(Intercept) 30.09886 1.63392 18.421 < 2e-16 \*\*\**

*hp -0.06823 0.01012 -6.742 1.79e-07 \*\*\**

# здесь Intercept – свободный член, sloap – коэффициент при переменной (угол наклона), Estimate – значение коэффициента, Std.Error – стандартная ошибка, t value – значение t-критерия, Pr(>|t|) – p уровень значимости, который используется для проверки гипотезы отличия от нуля коэффициентов

# и блок

*Residual standard error: 3.863 on 30 degrees of freedom* # кол-во степеней свободы

*Multiple R-squared: 0.6024, Adjusted R-squared: 0.5892* # R2 обычный и скорректированный

*F-statistic: 45.46 on 1 and 30 DF, p-value: 1.788e-07* #значение F-статистики при заданном количестве степеней свободы и уровень значимости

**ggplot(df, aes(hp, mpg))+**

**geom\_point(size = 5)+**

**geom\_smooth(method = "lm")+**

**facet\_grid(.~cyl)** #строим линию тренда со сглаживанием, добавляем в модель facet\_grid(.~cyl) , если хотим построить линию тренда в зависимости от количества цилиндров

**ggplot(df, aes(hp, mpg))+**

**geom\_smooth(method = "lm", se = F)+**

**facet\_grid(.~cyl)** # если убрать geom\_point, то будет построена только линия тренда с доверительными интервалами, но без точек, а если задать параметр se = F, то будет линия без доверительных интервалов

**fitted\_values\_mpg <- data.frame(mpg = df$mpg, fitted = fit$fitted.values )** # записываем датафрейм с реальными значениями переменной и предсказанными (обращение к ним через $fitted.values)

**new\_hp <- data.frame(hp = c(100, 150, 129, 300))** #добавляем новые значения лошадиных сил

**new\_hp$mpg <- predict(fit, new\_hp)** #выполняем предсказание значения mpg для новых значений hp по построенной линейной регрессии

**predict(fit, new\_hp)**

##################################

Регрессионный анализ в случае номинативной независимой переменной

**my\_df <- mtcars**

**my\_df$cyl <- factor(my\_df$cyl, labels = c("four", "six", "eight"))** #делаем cyl номинативной

**fit <- lm(mpg ~ cyl, my\_df)** #линейная регрессия, где независимая переменная cyl номинативная, в качестве Intercept выбирается один из уровней фактора, в нашем случае значения с 4-мя цилиндрами (Intercept здесь это среднее значение в группе), остальные коэффициенты означают изменение среднего значения при переходе с базового уровня (4 цилиндра) к уровню с 6-ю и 8-ю цилиндрами, нулевая гипотеза здесь состоит в предположении, что при переходе между уровнями изменения являются не значимыми.

**Файл «multiple regression.R»**

# Множественная линейная регрессия

#Будем рассматривать встроенный датасет swiss, который содержит информацию о рождаемости в разных регионах Швейцарии в 1888 году

# числовые предикторы

fit <- lm(Fertility ~ Examination + Catholic, data = swiss) #рождаемость в зависимости от оценки призывника и принадлежности к католической религии

summary(fit) # значимо предсказывает рождаемость только переменная Examination, причём чем она выше, тем ниже рождаемость

fit2 <- lm(Fertility ~ Examination\*Catholic, data = swiss) # знак \* также как и в дисперсионном анализе задает взаимодействие переменных

summary(fit2)

confint(fit2) #доверительный интервал для оценки коэффициентов. Только переменная Examination не содержит в 95% доверительном интервале 0, а значит предсказывает зависимую переменную на значимом уровне

# категориальные предикторы

hist(swiss$Catholic, col = 'red') #построим гистограмму числа католиков, по которой видно, что в некоторых областях католиков очень мало, а в некоторых очень много, поэтому будем создавать категориальную переменную с двумя значенями

swiss$religious <- ifelse(swiss$Catholic > 60, 'Lots', 'Few') #создаём новую переменную, принимающую значение Lots, если католиков >60% и Few - иначе

swiss$religious <- as.factor(swiss$religious) #делаем её категориальной

fit3 <- lm(Fertility ~ Examination + religious, data = swiss) #строим модель с количественной и категориальной зависимыми переменными

summary(fit3) # т.к. одна из переменных категориальная, то Intercept теперь задаёт среднее предсказанное значение зависимой переменной для первого уровня категориальной переменной (Few) при том, что все непрерывные переменные равны 0 (среднее значение рождаемости в регионах с низким уровнем католицизма и при Examination=0). Коэффициент при Examination задаёт влияние непрерывной переменной только на одном уровне, т.е. при низком уровне католицизма. religiousLots задаёт изменение предсказанного среднего значение зависимой переменной при переходе от регионов с низким уровнем католицизма к регионам с высоким его уровнем.

fit4 <- lm(Fertility ~ religious\*Examination, data = swiss) #добавим в модель взаимодействие переменных

summary(fit4) # здесь добавляется religiousLots:Examination, которое указывает на то, как влияет физическая подготовка на рождаемость в регионах с высоким уровнем католицизма. Видно, что Examination по-разному влияет на уровень рождаемости в регионах с большим и малым числом католиков (положительно и отрицательно соответственно)

# plots

ggplot(swiss, aes(x = Examination, y = Fertility)) +

geom\_point() #зависимость между рождаемостью и физ.подготовкой почти линейная и отрицательная

ggplot(swiss, aes(x = Examination, y = Fertility)) +

geom\_point() +

geom\_smooth() #строим линию тренда

ggplot(swiss, aes(x = Examination, y = Fertility)) +

geom\_point() +

geom\_smooth(method = 'lm') #линейная модель

ggplot(swiss, aes(x = Examination, y = Fertility, col = religious)) + #добавляем религию

geom\_point() #

ggplot(swiss, aes(x = Examination, y = Fertility, col = religious)) +

geom\_point() +

geom\_smooth() #тренд с доверительными интервалами

ggplot(swiss, aes(x = Examination, y = Fertility, col = religious)) +

geom\_point() +

geom\_smooth(method = 'lm') #линейная модель

#

fit5 <- lm(Fertility ~ religious\*Infant.Mortality\*Examination, data = swiss) #модель с категориальной и двумя непрерывными переменными

summary(fit5) # Коэффициенты Intercept, Infant.Mortality, Examination и Infant.Mortality:Examination относятся к показателям рождаемости в регионах с низким уровнем католицизма. Остальные – для регионов с высоким уровнем. Интерпретация коэффициентов аналогична предыдущим примерам с категориальными признаками.

**Памятка по интерпретации результатов регрессионного анализа с категориальными и непрерывными переменными**

**Модель для примера:**

*DV*~ *IV\_numeric* \* *IV\_categorical*

*IV\_categorical* - фактор с двумя уровнями (Level1 и Level2)

**Коэффициенты:**

**Intercept** — предсказанное значение *DV*для первого уровня *IV\_categorical* с учётом того, что *IV\_numeric* равна нулю.

**IV\_numeric** — насколько изменяется предсказанное значение *DV*при увеличении *IV\_numeric* на одну единицу в группе, соответствующей первому уровню*IV\_categorical*

**IV\_categoricalLevel2** — насколько изменяется предсказанное значение *DV* при переходе от первого уровня *IV\_categorical* ко второму уровню. С учётом того, что *IV\_numeric* равна нулю.

**IV\_numeric:IV\_categoricalLevel2** — насколько сильнее (или слабее) изменяется предсказанное значение *DV*при увеличении *IV\_numeric* на одну единицу в группе, соответствующей второму уровню *IV\_categorical*, по сравнению с первым уровнем.

**Как предсказывать значения в новом датасете на основе полученных коэффициентов**

1). Предположим у нас есть новый объект, про который мы знаем, что он принадлежит к группе, соответствующей *IV\_categorical (Level1)* и измеренный у него *IV\_numeric* составил **10**: *Предсказанное значение DV* = **Intercept** + **10** \* **IV\_numeric**

2). Предположим у нас есть новый объект, про который мы знаем, что он принадлежит к группе, соответствующей *IV\_categorical (Level2)* и измеренный у него *IV\_numeric* составил **6**: *Предсказанное значение DV* = **Intercept** + **IV\_categoricalLevel2** + **6** \* (**IV\_numeric** + **+IV\_numeric:IV\_categoricalLevel2**)