



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт кибернетики

Кафедра высшей математики

ОТЧЁТ ПО Практике по получению первичных профессиональных умений
и навыков
(указать вид практики)

Тема практики: Построение предсказания заражения компьютера
вредоносным программным обеспечением «Microsoft Malware Prediction»
(kaggle.com)

приказ университета о направлении на практику
793 – С от 12.02.2019 г.

Отчет представлен к
рассмотрению:

Студент группы
КМБО-01-18

Терехов Т.А.
(расшифровка подписи)
«6» марта 2019г.

Отчет утвержден.
Допущен к защите:

Руководитель практики
от кафедры

Петрусевич Д.А.
(расшифровка подписи)
«6» марта 2019г.

Москва 2019



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

ЗАДАНИЕ НА Практику по получению первичных профессиональных умений и навыков

**Студенту 1 курса учебной группы КМБО-01-18 института кибернетики
Терехову Тимофею Александровичу**

(фамилия, имя и отчество)

Место и время практики: Институт кибернетики, кафедра высшей математики

Время практики: с «16» февраля 2019 по «31» мая 2019

Должность на практике: практикант

1. ЦЕЛЕВАЯ УСТАНОВКА: изучение основ анализа данных и машинного обучения

2. СОДЕРЖАНИЕ ПРАКТИКИ:

2.1 Изучить: литературу и практические примеры по темам: 1) построение линейной регрессии, 2) использование метода главных компонент, 3) поиск и устранение линейной зависимости в данных, 4) основы нормализации данных, 5) методы классификации и кластеризации («решающее дерево», «случайный лес», «k ближайших соседей»), 6) обучение с учителем («градиентный спуск»).

2.2 Практически выполнить: 1) снижение размерности исходных задач при помощи метода главных компонент при возможности; построение линейной регрессии для некоторого параметра, исключение регрессоров, не коррелирующих с объясняемой переменной; решение задачи классификации или кластеризации на основе открытого набора данных с ресурса kaggle.com

2.3 Ознакомиться: с применением метода главных компонент; методов классификации («решающего дерева», «случайного леса»); методов градиентного спуска («градиентным бустингом»); методов кластеризации («k ближайших соседей»).

3. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: построение предсказания заражения компьютера вредоносным программным обеспечением «Microsoft Malware Prediction» (kaggle.com).

4. ОГРАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: построить классификацию на основе нескольких методов и произвести сравнение результатов классификации; сделать выводы о применимости использованных методов; сформировать выводы по результатам задачи из предметной области: какие характеристики компьютера сигнализируют о

том, что он, вероятно, подвергнется атаке вредоносного ПО вскоре, будет заражён вредоносным ПО.

Заведующий кафедрой
высшей математики



Ю.И.Худак

« 16 » февр 2019 г.

СОГЛАСОВАНО

Руководитель практики от кафедры:

«16» февраля 2019 г.


(подпись)

(Петрусеви́ч Д.А.)
(фамилия и инициалы)

Задание получил:

«16» февраля 2019 г.


(подпись)

(Терехов Т.А.)
(фамилия и инициалы)

ИНСТРУКТАЖ ПРОВЕДЕН:

Вид мероприятия	ФИО ответственного, подпись, дата	ФИО студента, подпись, дата
Охрана труда	Петрусеви́ч Д.А.  «16» февраля 2019 г.	Терехов Т.А.  «16» февраля 2019 г.
Техника безопасности	Петрусеви́ч Д.А.  «16» февраля 2019 г.	Терехов Т.А.  «16» февраля 2019 г.
Пожарная безопасность	Петрусеви́ч Д.А.  «16» февраля 2019 г.	Терехов Т.А.  «16» февраля 2019 г.
Правила внутреннего распорядка	Петрусеви́ч Д.А.  «16» февраля 2019 г.	Терехов Т.А.  «16» февраля 2019 г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования



«МИРЭА - Российский технологический университет»

РТУ МИРЭА

РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ Практики по получению
первичных профессиональных умений и навыков

студента Терехова Т.А. 1 курса группы КМБО-01-18 очной формы
обучения, обучающегося по направлению подготовки 01.03.02
«Прикладная математика и информатика»,
профиль «Математическое моделирование и вычислительная математика»

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	16.02.2019	Выбор темы практики/НИР. Пройти инструктаж по технике безопасности.	
1	16.02.2019	Вводная установочная лекция.	
3	02.03.2019	Построение и оценка линейной регрессии с помощью языка R	
5	16.03.2019	Использование метода главных компонент, выделение линейной зависимости в данных	
7	30.03.2019	Методы классификации и кластеризации; построение решающего дерева;	
9	13.04.2019	Концепция бэггинга, «случайный лес»; концепция бустинга; градиентные методы обучения и кластеризации	

17	07.06.2019	Представление отчётных материалов по практике/НИР и их защита. Передача обобщённых материалов на кафедру для архивного хранения.	
	6.03.2020	Зачётная аттестация.	

Содержание практики и планируемые результаты согласованы с руководителем практики от профильной организации.

Согласовано:

Заведующий
кафедрой



/ ФИО / Худак Ю.И.

Руководитель
практики от кафедры



/ ФИО / Петрусевич Д.А.

Обучающийся



/ ФИО / Терехов Т.А.

ОГЛАВЛЕНИЕ

Введение.....	8
Основная часть.....	9
Задача № 1.....	9
Задача № 2.....	18
Задача № 3.....	28
Список литературы.....	34
Код задачи № 1.....	35
Код задачи № 2.....	37
Код задачи № 3.....	42

Введение

Основной первых двух заданий является решение задач регрессионного анализа с применением метода наименьших квадратов (МНК – математический метод, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных) для построения линейных регрессий на заданных наборах, исследования зависимостей между имеющимися переменными и выдвижения предположений на основе проведенных наблюдений.

Анализ данных — это область математики и информатики, занимающаяся построением и исследованием наиболее общих математических методов и вычислительных алгоритмов извлечения знаний из экспериментальных (в широком смысле) данных; процесс исследования, фильтрации, преобразования и моделирования данных с целью извлечения полезной информации и принятия решений.

Регрессионный анализ — статистический метод исследования влияния одной или нескольких независимых переменных X_1, X_2, \dots, X_r на зависимую переменную Y [3]. Независимые переменные также называют регрессорами или предикторами, а зависимые переменные – объясняемыми или критериальными.

Кластеризация — задача группировки множества объектов на подмножества (кластеры) таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо критерию.

Кластерный анализ – это семейство алгоритмов, разработанных для формирования групп таким образом, чтобы члены группы были наиболее похожими друг на друга и не похожими на элементы, не выходящие в группу. Кластер и группа – это синонимы в мире кластерного анализа.

Задачи выполняются с использованием языков программирования Python и R

Основная часть.

Задача № 1.

В задаче № 1 необходимо загрузить данные из указанного набора и произвести следующие действия:

1. Нормализовать данные, вычтя из каждого столбца среднее значение $\text{mean}(x)$ и поделив на среднеквадратическое отклонение $\sigma \sim \sqrt{\text{var}(x)}$, где x – столбец данных.
2. Проверить, что в наборе данных нет линейной зависимости (построить зависимости между переменными, указанными в варианте, и проверить, что R^2 в каждой из них не высокий). В случае, если R^2 большой, один из таких столбцов можно исключить из рассмотрения.
3. Построить линейную модель зависимой переменной от указанных в варианте регрессоров по методу наименьших квадратов (команда `lm` пакета `lmtest` в языке R). Оценить, насколько хороша модель, согласно: 1) R^2 , 2) p -значениям каждого коэффициента.
4. Ввести в модель логарифмы регрессоров. Сравнить модели и выбрать наилучшую.
5. Ввести в модель всевозможные произведения из пар регрессоров, в том числе квадраты регрессоров. Найдите одну или несколько наилучших моделей по доле объяснённого разброса в данных R^2 .

Набор данных (Вариант № 21 (8)):

- Набор данных: `mtcars`.
- Объясняемая переменная: `mpg`.
- Регрессоры: `Disp`, `drat`, `qsec`.

Выполнять указанные действия будем при помощи языка программирования R.

Анализируемый набор **mtcars** содержит данные из журнала «**Motor Trend US**» 1974 года. Данные включают 11 столбцов (численных признаков), среди которых расход топлива и ещё 10 характеристических особенностей для 32 автомобилей моделей 1973 – 1974 годов (всего 32 записи). Задача заключается в обработке указанных ниже столбцов из набора с последующим анализом линейных моделей, построенной для объясняемой переменной через объясняющие (регрессоры). Из набора выделим интересующие нас переменные: **mpg** – пробег миль на галлон – объясняемая переменная

- **disp** – объем двигателя – регрессор
- **drat** – передаточное число заднего моста – количество оборотов задних колес по сравнению с определенной скоростью передачи (чем выше коэффициент, тем медленнее двигатель может работать, все еще позволяя автомобилю достичь заданной скорости) – регрессор
- **wt** – вес автомобиля (в тысячах английских фунтов) – регрессор

Для обработки данных и построения линейных моделей будем использовать библиотеки **dplyr** (функция **mutate_all** для преобразования всех элементов таблицы) и **lmtest** (функция **lm** для построения регрессионной модели).

Для начала загрузим и посмотрим несколько первых записей набора **mtcars**:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

Все столбцы таблицы содержат числа, причем признаки **vs** и **am** – бинарные.

В дальнейшем, чтобы строить разнообразные модели, будут использованы логарифмы от значений регрессоров, поэтому сразу добавим три столбца, **data\$Logdisp**, **data\$Logdrat**, **data\$Logwt**, в таблицу с вычисленными логарифмами (это делается для того, чтобы, когда данные будут стандартизированы, не пришлось исключать отрицательные значения для вычисления логарифмов).

Подключим библиотеку dplyr и прочитаем данные:

```
1 library(lmtest)
2 data = mtcars
3
4
```

Вычислим **data\$Logdisp**, **data\$Logdrat**, **data\$Logwt** :

```
16 #Сразу возьмем логарифмы каждой переменной до нормализации, а затем нормализуем отдельно.
17
18 data$Logdisp = log(data$disp)
19 data$Logdrat = log(data$drat)
20 data$Logwt = log(data$wt)
21
```

Нормализуем данные, вычтя из каждого столбца среднее значение $\text{mean}(x)$ и поделив на среднеквадратическое отклонение $\sigma \sim \sqrt{\text{var}(x)}$:

```
22 #Нормализуем данные
23
24 data$mpg = (data$mpg - mean(data$mpg))/sqrt(var(data$mpg))
25 data$disp = (data$disp - mean(data$disp))/sqrt(var(data$disp))
26 data$drat = (data$drat - mean(data$drat))/sqrt(var(data$drat))
27 data$wt = (data$wt - mean(data$wt))/sqrt(var(data$wt))
28
```

Также нормализуем логарифмы:

```
29 #Нормализуем логарифмы
30 data$Logdisp = (data$Logdisp - mean(data$Logdisp))/sqrt(var(data$Logdisp))
31 data$Logdrat = (data$Logdrat - mean(data$Logdrat))/sqrt(var(data$Logdrat))
32 data$Logwt = (data$Logwt - mean(data$Logwt))/sqrt(var(data$Logwt))
33
34
```

2. Проверить, что в наборе данных нет линейной зависимости (построить зависимости между переменными, указанными в варианте, и проверить, что R^2 в каждой из них не высокий). В случае, если R^2 большой, один из таких столбцов можно исключить из рассмотрения.

```

42 #построим модель зависимости пройденного расстояния от коэффициента заднего моста
43
44 modeldispdrat = lm(disp ~ drat, data)
45 modeldispdrat
46 summary(modeldispdrat)
47
48 #Multiple R-squared:  0.5044, Adjusted R-squared:  0.4879
49 #Коэффициент детерминации не очень большой, линейной зависимости не существует
50
51
52 #построим модель зависимости коэффициента заднего моста от веса
53
54 modeldratwt = lm(drat ~ wt, data)
55 modeldratwt
56 summary(modeldratwt)
57
58 #Multiple R-squared:  0.5076, Adjusted R-squared:  0.4912
59 #Коэффициент детерминации не очень большой, линейной зависимости не существует
60
61
62 #построим модель зависимости коэффициента заднего моста от веса
63
64
65 modeldispwt = lm(disp ~ wt, data)
66 modeldispwt
67 summary(modeldispwt)
68
69
70 #Multiple R-squared:  0.7885, Adjusted R-squared:  0.7815
71 #Коэффициент детерминации увеличился , но не превышает 0.8 , поэтому не будем выкидывать столбец
72
73 #Таким образом, мы подтверждаем гипотезу о линейной независимости переменных
74
75

```

Disp ~ Drat: $R^2 = 0.4879$ VIF = 1.952682

Disp ~ Wt : $R^2 = 0.7815$ VIF = 4.575792

Drat ~ Wt : $R^2 = 0.4912$ VIF = 1.965244

В целом можно судить о низком уровне мультиколлинеарности, однако стоит обратить внимание на относительно высокие показатели параметра **VIF** для переменных **disp** и **wt** – возможно в дальнейшем они либо повлияют, либо не повлияют на результативность построенных моделей.

3. Построить линейную модель зависимой переменной от указанных в варианте регрессоров по методу наименьших квадратов (команда `lm` пакета `lmtest` в языке R). Оценить, насколько хороша модель, согласно: 1) R^2 , 2) p-значениям каждого коэффициента.

```

76 #пункт3
77
78 #построим простую линейную модель зависимости mpg от всех описывающих переменных
79 #и оценим ее по коэффициенту детерминации и по p-критерию
80 #p-критерий - это вероятность ошибки при отклонении нулевой гипотезы
81 #(Предположения того, что линейной зависимости не существует)
82
83 model1 = lm(mpg ~ disp + drat + wt, data)
84 model1 #p-value - (.)()(*)
85 summary(model1)
86 #Multiple R-squared:  0.7835, Adjusted R-squared:  0.7603
87
88 #Результат неплох
89

```

```
> summary(model1)

Call:
lm(formula = mpg ~ disp + drat + wt, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-0.53662 -0.39355 -0.05223  0.27070  1.04232

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.065e-17  8.654e-02   0.000  1.0000
disp       -3.370e-01  1.970e-01  -1.711  0.0981 .
drat        7.487e-02  1.291e-01   0.580  0.5665
wt         -5.150e-01  1.976e-01  -2.606  0.0145 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4896 on 28 degrees of freedom
Multiple R-squared:  0.7835,    Adjusted R-squared:  0.7603
F-statistic: 33.78 on 3 and 28 DF,  p-value: 1.92e-09
```

Построенная модель имеет неплохие показатели: получены высокая доля объясненной дисперсии $R^2 = 0.7603$ и вероятность $p\text{-value} = 1.92 \cdot 10^{-9} \ll 0.001$

Однако по значимости переменных можно судить о небольшом влиянии переменных (особенно **drat**). Это также можно увидеть из графика **mpg ~ drat**, построенного ранее – разброс данных относительно велик, а потому переменная drat плохо описывает объясняемую переменную mpg.

4. Ввести в модель логарифмы регрессоров. Сравнить модели и выбрать наилучшую.

```
95 #добавим логарифм от параметра Disp в модель
96 model2 = lm(mpg ~ Logdisp + disp + drat + wt, data)
97 model2 #p-value - (***)(**)(**)
98 summary(model2) #Multiple R-squared:  0.8837, Adjusted R-squared:  0.8665
99
```

```
> summary(model2) #Multiple R-squared:  0.8837, Adjusted R-squared:  0.8665
```

```
Call:
lm(formula = mpg ~ Logdisp + disp + drat + wt, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-0.5251 -0.2597 -0.0251  0.2456  0.5902

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.790e-16  6.460e-02   0.000  1.00000
Logdisp     -1.492e+00  3.094e-01  -4.823  4.91e-05 ***
disp        9.146e-01  2.983e-01   3.066  0.00488 **
drat       -1.105e-01  1.037e-01  -1.066  0.29606
wt         -4.387e-01  1.483e-01  -2.958  0.00637 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3654 on 27 degrees of freedom
Multiple R-squared:  0.8837,    Adjusted R-squared:  0.8665
F-statistic: 51.29 on 4 and 27 DF,  p-value: 3.138e-12
```

$R^2 = 0.8665$, $p\text{-value} = 3.138 \cdot 10^{-12}$

Получили прирост значений параметров R^2 , а также уменьшение **p-value**.

```
101 #добавим логарифм от параметра drat в модель
102 model3 = lm(mpg ~ Logdrat + drat + disp + wt, data)
103 model3 #p-value - ()(.)(*)
104 summary(model3) #Multiple R-squared:  0.7901, Adjusted R-squared:  0.759
105
106
```

```
> summary(model3) #Multiple R-squared:  0.7901, Adjusted R-squared:  0.759
```

Call:

```
lm(formula = mpg ~ Logdrat + drat + disp + wt, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.4715	-0.3783	-0.1480	0.2449	1.0380

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.927e-16	8.678e-02	0.000	1.000
Logdrat	-1.057e+00	1.148e+00	-0.921	0.365
drat	1.126e+00	1.149e+00	0.980	0.336
disp	-3.750e-01	2.018e-01	-1.859	0.074 .
wt	-4.817e-01	2.014e-01	-2.391	0.024 *

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

Residual standard error: 0.4909 on 27 degrees of freedom

Multiple R-squared: 0.7901, Adjusted R-squared: 0.759

F-statistic: 25.41 on 4 and 27 DF, p-value: 8.195e-09

$R^2 = 0.759$, **p-value** = $8.195 \cdot 10^{-9}$

Параметр R^2 стал немного хуже по сравнению с **model 2**, увеличился **p-value**, а также страдает значимость.

```
107 #добавим логарифм от параметра wt в модель
108 model4 = lm(mpg ~ Logwt + wt + disp + drat, data)
109 model4 #p-value - (**)(.)(*)()
110 summary(model4) #Multiple R-squared:  0.8479, Adjusted R-squared:  0.8254
111
```

```
> summary(model4) #Multiple R-squared:  0.8479, Adjusted R-squared:  0.8254
```

Call:

```
lm(formula = mpg ~ Logwt + wt + disp + drat, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.4488	-0.2687	-0.1608	0.1290	0.9797

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-3.710e-16	7.387e-02	0.000	1.00000
Logwt	-1.285e+00	3.801e-01	-3.381	0.00222 **
wt	7.337e-01	4.061e-01	1.807	0.08196 .
disp	-4.067e-01	1.694e-01	-2.401	0.02352 *
drat	-2.454e-02	1.140e-01	-0.215	0.83128

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

Residual standard error: 0.4179 on 27 degrees of freedom

Multiple R-squared: 0.8479, Adjusted R-squared: 0.8254

F-statistic: 37.63 on 4 and 27 DF, p-value: 1.131e-10

$R^2 = 0.8254$ $p\text{-value} = 1.131 \cdot 10^{-10}$

Параметр R^2 стал немного хуже по сравнению с **model 2**, а также получилась не самая лучшая значимость.

В итоге добились большого роста качества модели в сравнении с исходной, где не использовались логарифмы значений переменных.

Самая лучшая получившаяся модель с использованием логарифмов **model 2**.

5. Ввести в модель всевозможные произведения из пар регрессоров, в том числе квадраты регрессоров. Найдите одну или несколько наилучших моделей по доле объяснённого разброса в данных R^2 .

```
119 model5 = lm(mpg ~ disp + drat + I(disp * wt) + wt, data)
120 model5 #p-value - (*)()(**)(***)
121 summary(model5) #Multiple R-squared:  0.8511, Adjusted R-squared:  0.829
122
```

```
> summary(model5) #Multiple R-squared:  0.8511, Adjusted R-squared:  0.829
```

```
call:
lm(formula = mpg ~ disp + drat + I(disp * wt) + wt, data = data)
```

```
Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.56356	-0.26642	-0.09657	0.21668	0.82759

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.21013	0.09460	-2.221	0.034909 *
disp	-0.40255	0.16742	-2.404	0.023328 *
drat	-0.04711	0.11447	-0.412	0.683894
I(disp * wt)	0.24428	0.06981	3.499	0.001636 **
wt	-0.63700	0.17051	-3.736	0.000886 ***

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

```
Residual standard error: 0.4135 on 27 degrees of freedom
Multiple R-squared:  0.8511,    Adjusted R-squared:  0.829
F-statistic: 38.57 on 4 and 27 DF,  p-value: 8.543e-11
```

$R^2 = 0.829$ $p\text{-value} = 8.54 \cdot 10^{-11}$

Неплохая модель, однако, показатель **p-value** значительно увеличился

```
123 model6 = lm(mpg ~ disp + drat + wt + I(disp * drat), data)
124 model6 #p-value - (*)()(**)(***)
125 summary(model6) #Multiple R-squared:  0.84, Adjusted R-squared:  0.8163
126
```



```
> summary(model6) #Multiple R-squared: 0.84, Adjusted R-squared: 0.8163
```

Call:

```
lm(formula = mpg ~ disp + drat + wt + I(disp * drat), data = data)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-0.4527 -0.3400 -0.1223  0.2180  0.8861
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.205082   0.100797  -2.035  0.05182 .
disp          -0.377392   0.172964  -2.182  0.03799 *
drat          -0.001892   0.115734  -0.016  0.98708
wt            -0.547849   0.173349  -3.160  0.00386 **
I(disp * drat) -0.298076   0.096606  -3.085  0.00465 **
```

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

Residual standard error: 0.4287 on 27 degrees of freedom

Multiple R-squared: 0.84, Adjusted R-squared: 0.8163

F-statistic: 35.43 on 4 and 27 DF, p-value: 2.23e-10

$R^2 = 0.8163$ p-value = 2.23×10^{-10}

R^2 и значимость немного упали, но p-value улучшился.

```
127 model7 = lm(mpg ~ disp + drat + wt + I(drat * wt), data)
128 model7 #p-value (*)()(**)(***)
129 summary(model7) #Multiple R-squared: 0.8321, Adjusted R-squared: 0.8072
130
```

```
> summary(model7) #Multiple R-squared: 0.8321, Adjusted R-squared: 0.8072
```

Call:

```
lm(formula = mpg ~ disp + drat + wt + I(drat * wt), data = data)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-0.4279 -0.2825 -0.1449  0.3202  0.9549
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.17229    0.09913  -1.738  0.09360 .
disp          -0.40765    0.17846  -2.284  0.03044 *
drat          -0.04955    0.12404  -0.399  0.69272
wt            -0.54819    0.17762  -3.086  0.00465 **
I(drat * wt)  -0.24963    0.08934  -2.794  0.00945 **
```

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

Residual standard error: 0.4391 on 27 degrees of freedom

Multiple R-squared: 0.8321, Adjusted R-squared: 0.8072

F-statistic: 33.45 on 4 and 27 DF, p-value: 4.228e-10

$R^2 = 0.8072$ p-value = 4.228×10^{-10}

R^2 ,p-value значимость немного упали. Комбинации с **drat дают не самые лучшие результаты, так что попробуем различные варианты исключения этого регрессора.**

```
131 model8 = lm(mpg ~ disp + drat + wt + I(disp^2), data)
132 model8 #p-value (**)(*)(**)(***)
133 summary(model8) #Multiple R-squared: 0.8621, Adjusted R-squared: 0.8417
134
```



```
> summary(model8) #Multiple R-squared:  0.8621, Adjusted R-squared:  0.8417

Call:
lm(formula = mpg ~ disp + drat + wt + I(disp^2), data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-0.61906 -0.23955 -0.07465  0.25728  0.67011

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.34766    0.11315  -3.073  0.004806 **
disp        -0.53366    0.16775  -3.181  0.003668 **
drat        -0.10481    0.11448  -0.916  0.368024
wt          -0.59654    0.16194  -3.684  0.001016 **
I(disp^2)     0.35888    0.09149   3.923  0.000543 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3979 on 27 degrees of freedom
Multiple R-squared:  0.8621,    Adjusted R-squared:  0.8417
F-statistic: 42.2 on 4 and 27 DF,  p-value: 3.058e-11
```

$R^2 = 0.8417$ $p\text{-value} = 3.058 \cdot 10^{-11}$

Видим увеличение R^2 и значимости модели по сравнению с предыдущими.

Пока что лучшая получившаяся модель.

```
140 model10 = lm(mpg ~ disp + drat + wt + I(wt^2), data)
141 model10 #p-value - (*)()(***)(**)
142 summary(model10) #Multiple R-squared:  0.847, Adjusted R-squared:  0.8244
143
```

```
> summary(model10) #Multiple R-squared:  0.847, Adjusted R-squared:  0.8244

Call:
lm(formula = mpg ~ disp + drat + wt + I(wt^2), data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-0.5179 -0.2270 -0.1719  0.1644  0.9255

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.18480    0.09240  -2.000  0.055647 .
disp        -0.37368    0.16898  -2.211  0.035661 *
drat        -0.04025    0.11574  -0.348  0.730678
wt          -0.64784    0.17376  -3.728  0.000904 ***
I(wt^2)       0.19076    0.05699   3.347  0.002414 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4191 on 27 degrees of freedom
Multiple R-squared:  0.847,    Adjusted R-squared:  0.8244
F-statistic: 37.37 on 4 and 27 DF,  p-value: 1.223e-10
```

$R^2 = 0.8244$ $p\text{-value} = 1.223 \cdot 10^{-10}$

R^2 и значимость упали по сравнению с моделью 8. Однако показатель $p\text{-value}$ улучшился.

В итоге, **model 8** оказалась наилучшей среди построенных по показателю R^2 .

Также стоит отметить, что в процессе формирования моделей приходилось исключать регрессор **drat**, который плохо описывает объясняемую переменную **mpg**.

Таким образом можно сделать вывод о том, что рассматриваемая величина **mpg** (число миль на галлон) относительно слабо зависит (плохо выражается) от значения **drat** (передаточного числа заднего моста) для автомобилей.

Задача 2

В этой задаче необходимо проанализировать данные волны мониторинга экономического положения и здоровья населения РФ (данные обследования РМЭЗ НИУ ВШЭ).

Прочитайте данные, выберите столбцы, которые Вам кажутся необходимыми, чтобы описать социально-экономическое положение граждан Российской Федерации.

Минимальный набор параметров: зарплата, пол, семейное положение, наличие высшего образования, возраст, тип населенного пункта, длительность рабочей недели.

Из параметра, отвечающего семейному положению, сделать дамми-переменные (с помощью one-hot-encoding): 1) переменная **wed1** имеет значение 1 в случае, если респондент женат, 0 – в противном случае; 2) **wed2**=1, если респондент разведён или вдовец; 3) **wed3** = 1, если респондент никогда не состоял в браке; 4) если считаете необходимым, введите другие параметры. Следите за мультиколлинеарностью (убедитесь в её отсутствии, оценив вспомогательную регрессию любого параметра (например, зарплату или одного из параметров wed) на эти переменные и используя команду **VIF** для неё).

Из параметра пол сделайте переменную **sex**, имеющую значение 1 для мужчин и равную 0 для женщин.

Из параметра, отвечающего типу населённого пункта, создайте одну дамми-переменную **city_status** со значением 1 для города или областного центра, 0 – в противоположном случае. Введите один параметр **higher_educ**, характеризующий наличие полного высшего образования.

Факторные переменные, «имеющие много значений», такие как: зарплата, длительность рабочей недели и возраст, - необходимо преобразовать в вещественные переменные и нормализовать их: вычесть среднее значение по этой переменной, разделить её значения на стандартное отклонение.

1. Постройте линейную регрессию зарплаты на все параметры, которые Вы выделили из данных мониторинга. Не забудьте оценить коэффициент вздутия дисперсии VIF.
2. Поэкспериментируйте с функциями вещественных параметров: используйте логарифм и степени (хотя бы от 0.1 до 2 с шагом 0.1).
3. Выделите наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённому с помощью построенных зависимостей разбросу adjusted R2 - R2adj.
4. Сделайте вывод о том, какие индивиды получают наибольшую зарплату.
5. Оцените регрессии для подмножества индивидов, указанных в варианте.

Решение

17 Волна. Файл – “r17i_os26b.sav”.

Установим необходимые библиотеки и пакеты для чтения файлов и работы с данными

```
1 # для чтения .sav файлов
2 install.packages("devtools")
3 devtools::install_github("https://github.com/bdemeshev/r1ms")
4
5 # подключение необходимых библиотек
6 library("lmtest")
7 library("r1ms")
8 library("dplyr")
9 library("GGally")
10 library("car")
11 library("sandwich")
12 library("Hmisc")
13
14
```

Составим базу из данных: **mj13.2**, **m_age**, **m_educ**, **status**, **mj6.2**, **m_marst**, **mh5**

```
16 # чтение данных о 17-й волне
17 data <- r1ms_read("C:\\Users\\AmazZing\\Desktop\\practice\\r17i_os26b.sav")
18 glimpse(data)
19
20 # выделяем интересующие нас столбцы
21 data2 = select(data, mj13.2, mh5, m_marst, m_educ, m_age, status, mj6.2)
22 #зарплата - mj13.2, пол - mh5, семейное положение - m_marst, наличие высшего образования - m_educ, возраст - m_age,
23 #тип населенного пункта - status ,длительность рабочей недели - mj6.2
24
```

mj13.2 - среднемесячная зарплата

m_age – возраст

mh5 – пол

m_educ - Образование

status – Тип населённого пункта

mj6.2 - Сколько часов в среднем продолжается Ваша обычная рабочая неделя?

m_marst - Семейные положение мы переделываем в (wed, wed1, wed2, wed3, wed4)

Из параметра, отвечающего семейному положению, сделать дамми-переменные (с помощью one-hot-encoding): 1) переменная **wed1** имеет значение 1 в случае, если респондент женат, 0 – в противном случае; 2) **wed2**=1, если респондент разведён или вдовец; 3) **wed3** = 1, если респондент никогда не состоял в браке; 4) если считаете необходимым, введите другие параметры. Следите за мультиколлинеарностью (убедитесь в её отсутствии, оценив вспомогательную регрессию любого параметра (например, зарплаты или одного из параметров wed) на эти переменные и используя команду **VIF** для неё).

```
30 #Разделим респондентов на 3 группы(по семейному положению)
31 #переменная wed1 имеет значение 1 в случае, если респондент женат, 0 – в противном случае;
32 #wed2 = 1, если респондент разведён или вдовец;
33 #wed3 = 1, если респондент никогда не состоял в браке;
34 #Просматривая другие значения, делаю вывод: смысла в вводе других параметров - нет
35
36 #Обнуляем новый столбец
37 data2["wed1"] = data2$m_marst
38 data2["wed1"] = 0
39
40 #Состоите в зарегистрированном браке = 2, ОФИЦИАЛЬНО ЗАРЕГИСТРИРОВАНЫ, НО ВМЕСТЕ НЕ ПРОЖИВАЮТ = 6
41 data2$wed1[which(data2$m_marst == '2') | which(data2$m_marst == '6')] = 1
42
43
44 #Обнуляем новый столбец
45 data2["wed2"] = data2$m_marst
46 data2["wed2"] = 0
47 #разведён, в браке не состоит = 4, вдовец(вдова) = 5
48 data2$wed2[which(data2$m_marst == '4')] = 1
49 data2$wed2[which(data2$m_marst == '5')] = 1
50
51
52 #Обнуляем новый столбец
53 data2["wed3"] = data2$m_marst
54 data2["wed3"] = 0
55 #никогда в браке не состояли = 1
56 data2$wed3[which(data2$m_marst == '1')] = 1
57
58
```

Из параметра пол сделаете переменную **sex**, имеющую значение 1 для мужчин и равную 0 для женщин.

```
59 #Из параметра пол сделаем переменную sex, имеющую значение 1 для мужчин и равную 0 для женщин
60 data2["sex"] = data2$mh5
61 data2$sex[which(data2$sex == '1')] = 1
62 data2$sex[which(data2$sex == '2')] = 0
63
64
```

Из параметра, отвечающего типу населённого пункта, создайте одну дамми-переменную **city_status** со значением 1 для города или областного центра, 0 – в противоположном случае. Введите один параметр **higher_educ**, характеризующей наличие полного высшего образования.

```
65 # Из параметра, отвечающего типу населённого пункта, создайте одну дамми-переменную city_status
66 # со значением 1 для города или областного центра, 0 – в противоположном случае.
67 data2["city_status"] = data2$status
68 data2["city_status"] = 0
69 data2$city_status[which(data2$status == '1')] = 1
70 data2$city_status[which(data2$status == '2')] = 1
71
72 #Введите один параметр higher_educ, характеризующий наличие полного высшего образования
73 data2["higher_educ"] = data2$m_educ
74 data2["higher_educ"] = 0
75 #есть полное высшее образование
76 data2$higher_educ[which(data2$m_educ == '21')] = 1
77 data2$higher_educ[which(data2$m_educ == '22')] = 1
78 data2$higher_educ[which(data2$m_educ == '23')] = 1
79
```

Факторные переменные, «имеющие много значений», такие как: зарплата, длительность рабочей недели и возраст, - необходимо преобразовать в вещественные переменные и нормализовать их: вычесть среднее значение по этой переменной, разделить её значения на стандартное отклонение.

```

87 #Факторные переменные, «имеющие много значений», такие как: зарплата, длительность рабочей недели и возраст
88 #- необходимо преобразовать в вещественные переменные и нормализовать их :
89 # вычесть среднее значение по этой переменной, разделить её значения на стандартное отклонение.
90
91 #Зарплата
92 data2["salary"] = data2$mj13.2
93 data2$salary = as.numeric(data2$salary)
94 data2["salary"] = (data2["salary"] - mean(data2$salary)) / sqrt(var(data2$salary))
95
96 #длительность рабочей недели
97 data2["week_len"] = data2$mj6.2
98 data2$week_len = as.numeric(data2$week_len)
99 data2["week_len"] = (data2["week_len"] - mean(data2$week_len)) / sqrt(var(data2$week_len))
100
101 #возраст
102 data2["age"] = data2$m_age
103 data2$age = as.numeric(data2$age)
104 data2["age"] = (data2["age"] - mean(data2$age)) / sqrt(var(data2$age))
105
106 #Соберем подготовленные данные
107 data3 = select(data2, salary, sex, wed1, wed2, wed3, higher_educ, age, status, week_len)
108 glimpse(data3)
109

```

1. Постройте линейную регрессию зарплаты на все параметры, которые Вы выделили из данных мониторинга. Не забудьте оценить коэффициент вздутия дисперсии VIF.

```

110 #1 Постройте линейную регрессию зарплаты на все параметры, которые вы выделили из данных мониторинга.
111 #Не забудьте оценить коэффициент вздутия дисперсии VIF.
112
113 #построю модель зависимости зарплаты от других факторов
114 model1 = lm(salary ~ sex + wed1 + wed2 + wed3 + higher_educ + age + status + week_len, data3)
115 summary(model1)
116

```

```
> summary(model1)
```

Call:

```
lm(formula = salary ~ sex + wed1 + wed2 + wed3 + higher_educ +
    age + status + week_len, data = data3)
```

Residuals:

```

      Min       1Q   Median       3Q      Max
-1.9390 -0.5009 -0.1669  0.2641 12.9867

```

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.07949	0.03765	2.111	0.0348 *
sex	0.40851	0.03091	13.215	< 2e-16 ***
wed1	NA	NA	NA	NA
wed2	-0.03070	0.04264	-0.720	0.4716
wed3	-0.25584	0.04460	-5.737	1.04e-08 ***
higher_educ	0.53477	0.03338	16.019	< 2e-16 ***
age	-0.10098	0.01677	-6.022	1.88e-09 ***
status	-0.17766	0.01274	-13.944	< 2e-16 ***
week_len	0.16161	0.01494	10.817	< 2e-16 ***

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

Residual standard error: 0.9038 on 3866 degrees of freedom

Multiple R-squared: 0.1847, Adjusted R-squared: 0.1832

F-statistic: 125.1 on 7 and 3866 DF, p-value: < 2.2e-16

Вывод: $R^2 = 0.1832$

Имеем показатель $R^2 = 0.1832$ и низкую значимость регрессоров **wed1** и **wed2**

Построим модель без **wed1** и **wed2**:

```
> summary(model2)

call:
lm(formula = salary ~ sex + wed3 + higher_educ + age + status +
    week_len, data = data3)

Residuals:
    Min       1Q   Median       3Q      Max
-1.9391 -0.4997 -0.1641  0.2678 12.9939

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.07134    0.03591   1.987   0.047 *
sex           0.41414    0.02990  13.849 < 2e-16 ***
wed3        -0.25265    0.04437  -5.694 1.33e-08 ***
higher_educ  0.53527    0.03337  16.039 < 2e-16 ***
age          -0.10284    0.01657  -6.207 5.97e-10 ***
status       -0.17757    0.01274 -13.938 < 2e-16 ***
week_len      0.16152    0.01494  10.812 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9037 on 3867 degrees of freedom
Multiple R-squared:  0.1846,    Adjusted R-squared:  0.1833
F-statistic: 145.9 on 6 and 3867 DF,  p-value: < 2.2e-16
```

$R^2 = 0.1833$

```
> vif(model2)
            sex            wed3 higher_educ            age            status            week_len
1.047006    1.296011    1.047585    1.301681    1.027290    1.058340
```

Исходя из полученных данных, делаем вывод: уровень мультиколлинеарности низкий – регрессоры линейно-независимы.

Все регрессоры замечательно описывают модель, после исключения **wed1**, **wed2**, немного повысился R^2 .

2. Поэкспериментируйте с функциями вещественных параметров: используйте логарифм и степени (хотя бы от 0.1 до 2 с шагом 0.1).

```

9 variables      3874 observations
-----
salary
  n missing distinct      Info      Mean      Gmd      .05      .10      .25      .50      .75      .90      .9
 3874      0         238    0.998 -3.235e-17    0.915    -0.9221    -0.8268    -0.6078    -0.2268    0.2493    1.2017    1.677

lowest : -1.179184 -1.174422 -1.173755 -1.172517 -1.144899, highest:  9.296563 10.248904 11.201244 12.153585 13.105925
-----
sex
  n missing distinct      Info      Sum      Mean      Gmd
 3874      0         2    0.741    1719    0.4437    0.4938
-----
wed1
  n missing distinct      Info      Mean      Gmd
 3874      0         1      0      1      0
Value
Frequency 3874
Proportion 1
-----
wed2
  n missing distinct      Info      Sum      Mean      Gmd
 3874      0         2    0.396    606    0.1564    0.264
-----
wed3
  n missing distinct      Info      Sum      Mean      Gmd
 3874      0         2    0.416    645    0.1665    0.2776
-----
higher_educ
  n missing distinct      Info      Sum      Mean      Gmd
 3874      0         2    0.595    1056    0.2726    0.3967
-----
age
  n missing distinct      Info      Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
 3874      0         64    0.999 1.424e-17    1.147 -1.55541 -1.40040 -0.85787 0.03343 0.76972 1.23475 1.54477

lowest : -2.020435 -1.942931 -1.865426 -1.787922 -1.710418, highest:  2.552322 2.629827 2.707331 2.784835 2.939844
-----
status : ТИП НАСЕЛЕННОГО ПУНКТА
  n missing distinct      Info      Mean      Gmd
 3874      0         4    0.876    2.021    1.209
Value
Frequency 1729 1148 182 815
Proportion 0.446 0.296 0.047 0.210
-----
week_len
  n missing distinct      Info      Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
 3874      0         78    0.916 1.038e-16    0.9392 -1.5028 -0.6788 -0.2669 -0.2669 0.3923 1.3317 2.2051

lowest : -3.233226 -3.150826 -3.068427 -2.986028 -2.903629, highest:  5.336301 5.501099 5.665898 6.325092 6.819488
-----

```

Для логарифмирования вещественных параметров необходимо, чтобы значения были > 0 , поэтому:

```

133 data3["salary"] = data3["salary"] + 1.2
134 data3["age"] = data3["age"] + 2.1
135 data3["week_len"] = data3["week_len"] + 3.3
136

```

```

137 #Строим модель с логарифмами
138 model3 = lm(salary ~ sex + wed3 + higher_educ + log(age) + status + log(week_len), data3)
139 summary(model3)
140

```



```
> summary(model3)
```

Call:
lm(formula = salary ~ sex + wed3 + higher_educ + log(age) + status +
log(week_len), data = data3)

Residuals:

Min	1Q	Median	3Q	Max
-1.8994	-0.5019	-0.1693	0.2788	13.0659

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.82063	0.06295	13.036	< 2e-16	***
sex	0.41743	0.03007	13.880	< 2e-16	***
wed3	-0.20972	0.04764	-4.402	1.1e-05	***
higher_educ	0.52330	0.03354	15.604	< 2e-16	***
log(age)	-0.09773	0.02945	-3.319	0.000913	***
status	-0.17599	0.01282	-13.728	< 2e-16	***
log(week_len)	0.43736	0.04195	10.425	< 2e-16	***

 signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Все регрессоры имеют высокое значение в модели.

Пробую использовать комбинации регрессоров для построения модели.

```
> summary(model4)
```

Call:
lm(formula = salary ~ sex + wed3 + higher_educ + age + status +
week_len + I(age * week_len) + I(age^2) + I(week_len^2),
data = data3)

Residuals:

Min	1Q	Median	3Q	Max
-1.8541	-0.4903	-0.1605	0.2740	12.9340

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.326275	0.164223	1.987	0.04702	*
sex	0.426897	0.029757	14.346	< 2e-16	***
wed3	-0.090536	0.047336	-1.913	0.05587	.
higher_educ	0.530097	0.033052	16.038	< 2e-16	***
age	0.532577	0.087672	6.075	1.36e-09	***
status	-0.185781	0.012651	-14.686	< 2e-16	***
week_len	0.193625	0.059347	3.263	0.00111	**
I(age * week_len)	-0.010154	0.014696	-0.691	0.48962	
I(age^2)	-0.134081	0.014675	-9.137	< 2e-16	***
I(week_len^2)	-0.002692	0.005936	-0.454	0.65018	

 signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Комбинации с **week_len** и **wed3** имеют низкий приоритет, исключаем их


```
> summary(model5)

call:
lm(formula = salary ~ sex + higher_educ + age + status + week_len +
    I(age^2), data = data3)

Residuals:
    Min       1Q   Median       3Q      Max
-1.8265 -0.4914 -0.1612  0.2744 12.9568

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.34523    0.07934   4.351 1.39e-05 ***
sex            0.42745    0.02964  14.424 < 2e-16 ***
higher_educ    0.53055    0.03304  16.060 < 2e-16 ***
age            0.55631    0.05949   9.351 < 2e-16 ***
status        -0.18554    0.01264 -14.677 < 2e-16 ***
week_len       0.15257    0.01482  10.295 < 2e-16 ***
I(age^2)       -0.14348    0.01348 -10.644 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

После исключения комбинации с **week_len** ничего не изменилась, как и прежде показатели модели относительно хороши

Построим модель на комбинациях логарифмов регрессоров

```
> summary(model6)

call:
lm(formula = salary ~ sex + higher_educ + log(age) + status +
    log(week_len) + I(log(age) * log(week_len)) + I(log(age)^2) +
    I(log(week_len)^2), data = data3)

Residuals:
    Min       1Q   Median       3Q      Max
-1.8903 -0.4988 -0.1652  0.2774 12.9460

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.93697    0.07118  13.163 < 2e-16 ***
sex            0.41134    0.02985  13.781 < 2e-16 ***
higher_educ    0.50267    0.03349  15.010 < 2e-16 ***
log(age)       0.09907    0.06904   1.435 0.151382
status        -0.18065    0.01272 -14.206 < 2e-16 ***
log(week_len)  0.22474    0.06304   3.565 0.000369 ***
I(log(age) * log(week_len)) 0.06125    0.05728   1.069 0.284984
I(log(age)^2)  -0.30140    0.03338  -9.030 < 2e-16 ***
I(log(week_len)^2) 0.11682    0.02825   4.135 3.63e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Теперь снизились показатели модели, а так же влияние комбинации с **log(age)** и **I(log(age) * log(week_len))** построим модель без них

```
> model7 = lm(salary ~ sex + higher_educ + status + log(week_len) + I(log(age)^2) + I(log(week_len)^2), data3)
> summary(model7)

Call:
lm(formula = salary ~ sex + higher_educ + status + log(week_len) +
    I(log(age)^2) + I(log(week_len)^2), data = data3)

Residuals:
    Min       1Q   Median       3Q      Max
-1.9034 -0.5036 -0.1644  0.2750 12.9867

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.88696    0.06248   14.196 < 2e-16 ***
sex             0.40660    0.02993   13.586 < 2e-16 ***
higher_educ     0.52020    0.03337   15.591 < 2e-16 ***
status         -0.17598    0.01271  -13.844 < 2e-16 ***
log(week_len)   0.27686    0.05381    5.146 2.80e-07 ***
I(log(age)^2)  -0.18563    0.02529   -7.341 2.57e-13 ***
I(log(week_len)^2) 0.11347    0.02830    4.009 6.21e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Показатели модели незначительно повысились.

Построим модель с квадратами логарифмов

```
> summary(model8)

Call:
lm(formula = salary ~ sex + higher_educ + age + log(age^2) +
    status + week_len + log(week_len^2), data = data3)

Residuals:
    Min       1Q   Median       3Q      Max
-1.8803 -0.4947 -0.1645  0.2712 12.9231

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.34211    0.08774   15.296 < 2e-16 ***
sex             0.41925    0.02969   14.120 < 2e-16 ***
higher_educ     0.50557    0.03323   15.215 < 2e-16 ***
age            -0.50767    0.04757  -10.671 < 2e-16 ***
log(age^2)      0.39079    0.03942    9.915 < 2e-16 ***
status         -0.18404    0.01266  -14.535 < 2e-16 ***
week_len        0.13129    0.03111    4.220 2.49e-05 ***
log(week_len^2) 0.03368    0.04359    0.773  0.44
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Качество модели относительно высокое, однако значимость регрессора **log(week_len^2)** не высока

3. Выделите наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённой с помощью построенных зависимостей разбросу adjusted R² - R²adj.

1) model5: **R²**=0.2011, (значимость параметров – максимальная ***)

2) model7: **R²**=0.1861, (значимость параметров – максимальная ***)

3) model8: **R²**=0.1985, (значимость параметров – максимальная ***)

В итоге лучшая по показателям **model5**

4. Сделайте вывод о том, какие индивиды получают наибольшую зарплату

```
> summary(model2)
```

```
call:
```

```
lm(formula = salary ~ sex + wed3 + higher_educ + age + status +  
    week_len, data = data3)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max  
-1.9391 -0.4997 -0.1641  0.2678 12.9939
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)  0.07134    0.03591   1.987   0.047 *  
sex           0.41414    0.02990  13.849 < 2e-16 ***  
wed3         -0.25265    0.04437  -5.694 1.33e-08 ***  
higher_educ   0.53527    0.03337  16.039 < 2e-16 ***  
age          -0.10284    0.01657  -6.207 5.97e-10 ***  
status        -0.17757    0.01274 -13.938 < 2e-16 ***  
week_len      0.16152    0.01494  10.812 < 2e-16 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.9037 on 3867 degrees of freedom
```

```
Multiple R-squared:  0.1846,    Adjusted R-squared:  0.1833
```

```
F-statistic: 145.9 on 6 and 3867 DF,  p-value: < 2.2e-16
```

Вывод: наибольшую зарплату получает мужчина (estimate: **sex** > 0), с высшим образованием (estimate: **higher_educ** > 0), преимущественно женатый (estimate: **wed3** < 0), однако этот показатель не вносит наибольший вклад, этот мужчина примерно средних лет (estimate: **age** ~ 0), тем не менее обычно это не городской житель (estimate: **status** < 0), что очень странно, т.к. в реальной ситуации у городского жителя преимущественно большая зарплата. Так же у него наблюдаются переработки (estimate: **duration** > 0). В целом, в модели присутствуют некоторые неточности, но схожести с реальностью присутствуют.

5. Оцените регрессии для подмножества индивидов, указанных в варианте.

1) Городские жители, не состоявшие в браке; 2) разведенные женщины, без высшего образования

1) Данная группа индивидов теряет в зарплате из-за того, что это городские жители (**status** < 0) да и ещё не состоявшие в браке (**wed3** < 0), в итоге у данной группы индивидов зарплата ниже среднего

2) Эта группа индивидов получает относительно низкую з/п из-за отсутствия высшего образования, а регрессия по зарплате говорит о том, что высшее образование играет значительную роль в ее размере, также зарплата снижается из-за женского

пола(**estimate: sex > 0**),то что они разведены, оказывает минимальное значение на их зарплату по модели **m (wed2 ~ 0**

Задача 3

Необходимо провести анализ вашего датасета и сделать обработку данных.

Ответить на следующие вопросы:

1. Сколько в датасете объектов и признаков? Дать описание каждому признаку, если оно есть.
2. Сколько категориальных признаков, какие?
3. Столбец с макимальным количеством уникальных значений категориального признака?
4. Есть ли бинарные признаки?
5. Есть ли пропуски?
6. Сколько объектов с пропусками?
7. Столбец с максимальным количеством пропусков?
8. Есть ли на ваш взгляд выбросы, аномальные значения?
9. Столбец с максимальным средним значением после нормировки признаков через стандартное отклонение?
10. Столбец с целевым признаком?
11. Сколько объектов попадает в тренировочную выборку при использовании `train_test_split` с параметрами `test_size = 0.3`, `random_state = 42`?
12. Сколько признаков достаточно для объяснения 90% дисперсии после применения метода PCA?
13. Какой признак вносит наибольший вклад в первую компоненту?

Для работы с данным набором будем использовать вспомогательные библиотеки, поэтому установим их и загрузим данные:

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [17]: data2 = pd.read_csv('C:/Users/AmaZZinG/Desktop/practice/train.csv',
                             usecols=['MachineIdentifier', 'CountryIdentifier',
                                       'AVProductStatesIdentifier',
                                       'OrganizationIdentifier',
                                       'Census_ProcessorCoreCount',
                                       'HasDetections', 'Census_TotalPhysicalRAM'],
                             nrows=1000)
```

1.Сколько в датасете объектов и признаков? Дать описание каждому признаку, если оно есть.

Определим размеры нашей таблицы с данными:

```
In [19]: print("data2.shape=",data2.shape)
data2.shape= (1000, 7)
```

В нашей таблице с данными 1000 строки и 4 столбца.
Прежде чем приступить к обработке данных, посмотрим на наши признаки:

```
In [18]: data2.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
MachineIdentifier      1000 non-null object
AVProductStatesIdentifier  997 non-null float64
CountryIdentifier      1000 non-null int64
OrganizationIdentifier  683 non-null float64
Census_ProcessorCoreCount  996 non-null float64
Census_TotalPhysicalRAM  988 non-null float64
HasDetections          1000 non-null int64
dtypes: float64(4), int64(2), object(1)
memory usage: 54.8+ KB
```

Итак, в нашем наборе данных содержится 100000 строк (объектов) и 4 столбца (признака).

Дадим описание каждому признаку:

1. **MachineIdentifier** - категориальный признак, Индивидуальный ID машины
2. **AVProductStatesIdentifier** - вещественный признак, ID для конкретной конфигурации антивирусного программного обеспечения пользователя
3. **CountryIdentifier** - целочисленный признак, ID для страны, в которой находится машина
4. **OrganizationIdentifier** - вещественный признак, ID для организации, которой принадлежит машина, идентификатор организации сопоставляется как с конкретными компаниями, так и с широкими отраслями промышленности
5. **Census_ProcessorCoreCount** - Количество логических ядер в процессоре
6. **Census_TotalPhysicalRAM** - Извлекает физическую оперативную память в МБ
7. **HasDetections** – Обнаружено ли вредоносное ПО на машине

2. Сколько категориальных признаков, какие?

Таким образом, у нас есть 1 категориальный признак – **MachineIdentifier**

3. Столбец с максимальным количеством уникальных значений категориального признака?

В данном случае столбец с максимальным количеством уникальных значений категориального признака (единственный) – **MachineIdentifier**.

4. Есть ли бинарные признаки?

Бинарные признаки в нашем датасете присутствуют в столбце **HasDetections**.

5. Есть ли пропуски?

```
In [20]: np.sum(pd.isnull(data2))
```

```
Out[20]: MachineIdentifier      0
AVProductStatesIdentifier      3
CountryIdentifier              0
OrganizationIdentifier        317
Census_ProcessorCoreCount      4
Census_TotalPhysicalRAM        12
HasDetections                  0
dtype: int64
```

Да, есть, в вещественных столбцах: **AVProductStatesIdentifier**, **OrganizationIdentifier**, **Census_ProcessorCoreCount**, **Census_TotalPhysicalRAM**

6. Сколько объектов с пропусками?

336 объектов с пропусками.

7. Столбец с максимальным количеством пропусков?

Исходя из проделанной операции, можно сделать вывод, что пропусков максимальное количество в столбце **OrganizationIdentifier**

Для дальнейшего анализа, уберём все строки с пропусками.

```
In [25]: data2.dropna(inplace=True)
data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 672 entries, 0 to 999
Data columns (total 7 columns):
MachineIdentifier      672 non-null object
AVProductStatesIdentifier  672 non-null float64
CountryIdentifier      672 non-null int64
OrganizationIdentifier  672 non-null float64
Census_ProcessorCoreCount  672 non-null float64
Census_TotalPhysicalRAM  672 non-null float64
HasDetections          672 non-null int64
dtypes: float64(4), int64(2), object(1)
memory usage: 42.0+ KB
```

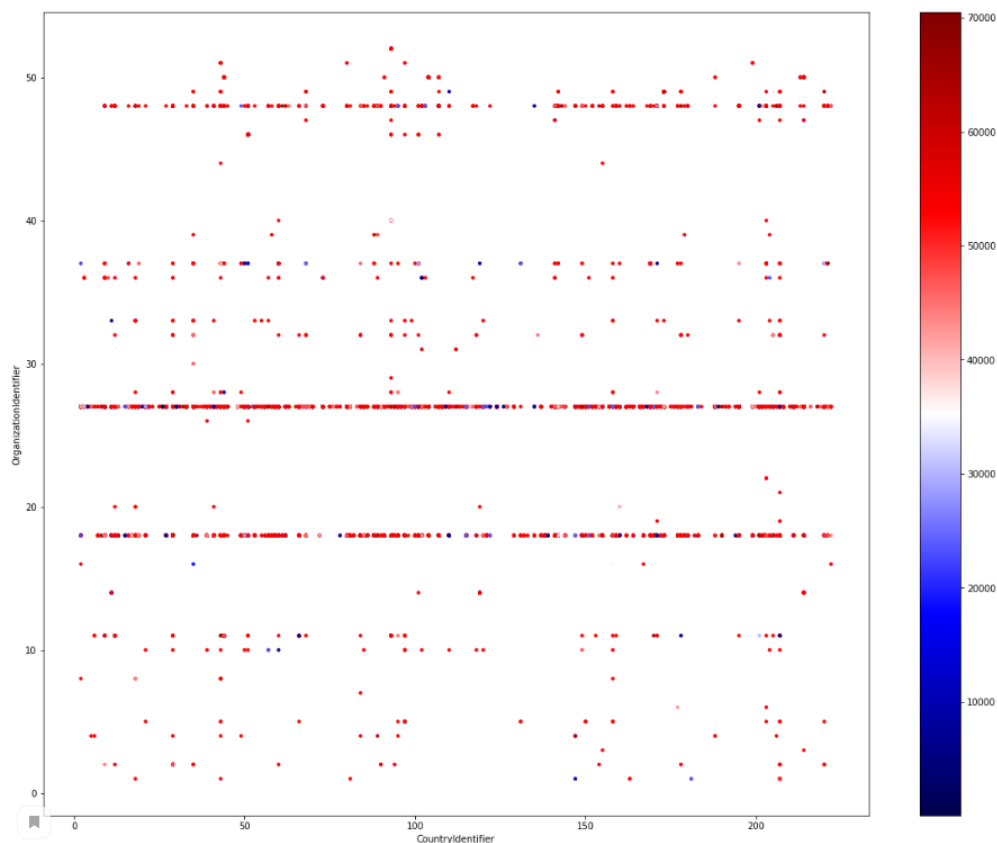
8. Есть ли на ваш взгляд выбросы, аномальные значения?

Для начала обработаем столбец с категориальным признаком.

Проверим наш датасет на наличие аномальных значений с помощью графика:

```
In [54]: plt.figure(figsize=(20,16))
plt.scatter(data2.CountryIdentifier, data2.OrganizationIdentifier, s=9, c=data2.AVProductStatesIdentifier, cmap = 'seismic')
plt.colorbar()
plt.xlabel('CountryIdentifier')
plt.ylabel('OrganizationIdentifier')
```

Out[54]: Text(0, 0.5, 'OrganizationIdentifier')



На мой взгляд, аномальные значения в таблице отсутствуют.

9. Столбец с максимальным средним значением после нормировки признаков через стандартное отклонение?

MachineIdentifier, AVProductStatesIdentifier, CountryIdentifier, OrganizationIdentifier, HasDetections не имеет смысла использовать.

```
In [16]: def maxstd(data2):
        max = 0
        max_name = str()
        for i in data2.columns:
            num = (data2[i] - data2[i].mean()) / data2[i].std()
            data2[i] = num
            if max < num.mean():
                max_name, max = i, num.mean()
        return max_name
maxstd(data2)
```

```
Out[16]: 'Census_TotalPhysicalRAM'
```

После проверки условия, столбцом с максимальным средним после нормировки признаков через стандартное отклонение, является столбец **Census_TotalPhysicalRAM**

10. Столбец с целевым признаком?

Опираясь на условие задачи, можно сделать вывод, что целевой признак - это **HasDetection**

11. Сколько объектов попадает в тренировочную выборку при использовании train_test_split с параметрами test_size = 0.3, random_state = 42?

```
In [26]: target = data2.HasDetections
```

```
In [27]: train = data2
```

```
In [28]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(train, target, test_size = 0.3, random_state = 42)
N_train, _ = X_train.shape
N_test, _ = X_test.shape
print(N_train, N_test)
```

```
470 202
```

Заметим, что в тренировочную выборку попадает 470 объектов.

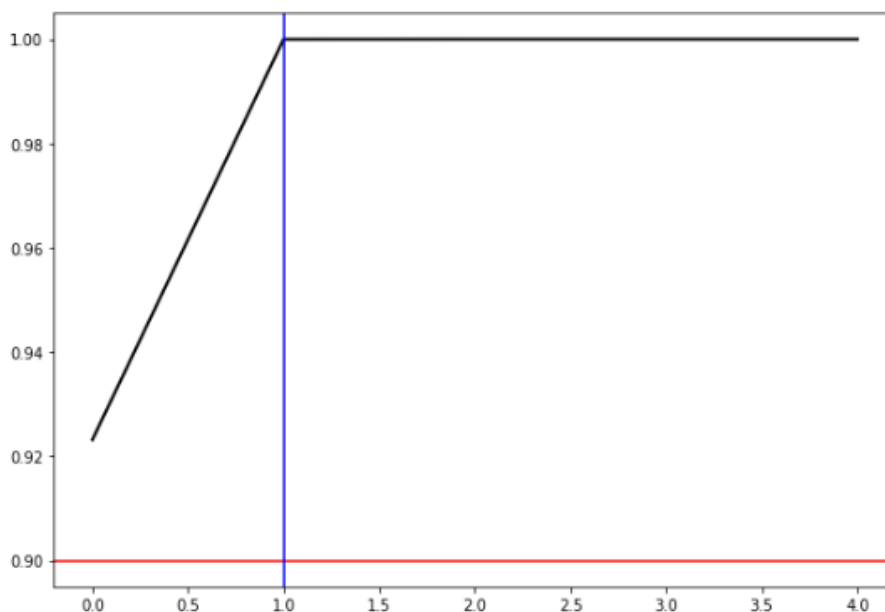
12. Сколько признаков достаточно для объяснения 90% дисперсии после применения метода PCA?


```
In [113]: from sklearn.decomposition import PCA
%matplotlib inline
import matplotlib.pyplot as plt
pca = PCA()
pca.fit(X_train)
X_pca = pca.transform(X_train)
for i, component in enumerate(pca.components_):
    print("{} component: {}% of initial variance".format(i + 1,
        round(100 * pca.explained_variance_ratio_[i], 2)))
    print(" + ".join("%.3f x %s" % (value, name)
        for value, name in zip(component, train.columns)))

1 component: 92.31% of initial variance
-1.000 x AVProductStatesIdentifier + -0.000 x CountryIdentifier + 0.000 x OrganizationIdentifier + -0.000 x Census_ProcessorCoreCount + -0.006 x Census_TotalPhysicalRAM
2 component: 7.68% of initial variance
-0.006 x AVProductStatesIdentifier + -0.000 x CountryIdentifier + -0.000 x OrganizationIdentifier + 0.000 x Census_ProcessorCoreCount + 1.000 x Census_TotalPhysicalRAM
3 component: 0.0% of initial variance
-0.000 x AVProductStatesIdentifier + 1.000 x CountryIdentifier + 0.004 x OrganizationIdentifier + 0.001 x Census_ProcessorCoreCount + 0.000 x Census_TotalPhysicalRAM
4 component: 0.0% of initial variance
0.000 x AVProductStatesIdentifier + -0.004 x CountryIdentifier + 1.000 x OrganizationIdentifier + 0.016 x Census_ProcessorCoreCount + 0.000 x Census_TotalPhysicalRAM
5 component: 0.0% of initial variance
-0.000 x AVProductStatesIdentifier + -0.001 x CountryIdentifier + -0.016 x OrganizationIdentifier + 1.000 x Census_ProcessorCoreCount + -0.000 x Census_TotalPhysicalRAM
```

```
In [104]: plt.figure(figsize=(10,7))
plt.plot(np.cumsum(pca.explained_variance_ratio_), color='k', lw=2)
plt.axhline(0.9, c='r')
plt.axvline(1, c='b')
```

Out[104]: <matplotlib.lines.Line2D at 0x258c2f2bd08>



Чтобы описать 90% дисперсии после применения метода PCA, достаточно 1 признака.

13. Какой признак вносит наибольший вклад в первую компоненту?

По полученным данным, можно понять, что наибольший вклад в 1 компоненту вносит признак **AVProductStatesIdentifier**

Вывод:

Была произведена первичная обработка данных, исключены признаки **MachineIdentifier** и **HasDetections**. Также были выделены тестовые и тренировочные выборки. С помощью метода главных компонент было выявлено, что для объяснения 90 процентов дисперсии достаточно лишь 1 признака, а также было замечено, что наибольший вклад в первую компоненту вносит признак **AVProductStatesIdentifier**.

Список литературы.

1. Роберт И. Кабаков R в действии: Анализ и визуализация данных на языке R: [Электронный ресурс] - 2014. URL: <http://kek.ksu.ru/eos/WM/kabacoff2014ru.pdf>
2. Баженов Д. О задачах классификации: [Электронный ресурс]. URL: <http://bazhenov.me/blog/2012/06/05/classification.html>
3. Алексей Орлов @Lexho Как работает метод главных компонент (PCA) на простом примере: [Электронный ресурс]. URL: <https://habr.com/ru/post/304214/>
4. Микаел Григорян @temujin R - значит регрессия - 2018. URL: <https://habr.com/ru/post/350668/>

Код номера 1

```
library(lmtest)

data = mtcars

#Пункт 1

#Нормализуем данные для того, чтобы набор данных был схож с нормальным распределением:
#Вычитаем среднее значение и делим результат на его среднееквадратичное отклонение
#Входные данные:
#mpg - объясняемая переменная. Мили/(US) галлон.
#Disp - объясняющая переменная. Перемещение.
#drat - объясняющая переменная. Коэффициент заднего моста.
#wt - объясняющая переменная. Вес.
#Сразу возьмем логарифмы каждой переменной до нормализации, а затем нормализуем отдельно.
data$Logdisp = log(data$disp)
data$Logdrat = log(data$drat)
data$Logwt = log(data$wt)
#Нормализуем данные
data$mpg = (data$mpg - mean(data$mpg))/sqrt(var(data$mpg))
data$disp = (data$disp - mean(data$disp))/sqrt(var(data$disp))
data$drat = (data$drat - mean(data$drat))/sqrt(var(data$drat))
data$wt = (data$wt - mean(data$wt))/sqrt(var(data$wt))
#Нормализуем логарифмы
data$Logdisp = (data$Logdisp - mean(data$Logdisp))/sqrt(var(data$Logdisp))
data$Logdrat = (data$Logdrat - mean(data$Logdrat))/sqrt(var(data$Logdrat))
data$Logwt = (data$Logwt - mean(data$Logwt))/sqrt(var(data$Logwt))

#Пункт 2

#Проверим гипотезу о линейной независимости наших переменных
#Для этого построим линейную регрессию между параметрами
#Оценивать будем по показателю Multiple\Adjusted R-squared (Коэффициент детерминации)
#Это значение показывает сколько процентов данных мы смогли описать той или иной моделью
#Построим модель зависимости пройденного расстояния от коэффициента заднего моста
modeldispdrat = lm(disp ~ drat, data)
modeldispdrat
summary(modeldispdrat)
#Multiple R-squared: 0.5044, Adjusted R-squared: 0.4879
#Коэффициент детерминации не очень большой, линейной зависимости не существует
#Построим модель зависимости коэффициента заднего моста от веса
```

```

modeldratwt = lm(drat ~ wt, data)
modeldratwt
summary(modeldratwt)
#Multiple R-squared:  0.5076,  Adjusted R-squared:  0.4912
#Коэффициент детерминации не очень большой, линейной зависимости не существует
#Построим модель зависимости коэффициента заднего моста от веса
modeldispwt = lm(disp ~ wt, data)
modeldispwt
summary(modeldispwt)
#Multiple R-squared:  0.7885,  Adjusted R-squared:  0.7815
#Коэффициент детерминации увеличился , но не превышает 0.8 , поэтому не будем выкидывать столбец
#Таким образом, мы подтверждаем гипотезу о линейной независимости переменных
#Пункт3
#Построим простую линейную модель зависимости mpg  от всех описывающих переменных
#и оценим ее по коэффициенту детерминации и по р-критерию
#р-критерий - это вероятность ошибки при отклонении нулевой гипотезы
#(Предположения того, что линейной зависимости не существует)
model1 = lm(mpg ~ disp + drat + wt, data)
model1 #p-value - (.)()(*)
summary(model1)
#Multiple R-squared:  0.7835,  Adjusted R-squared:  0.7603
#Результат неплох
#Пункт 4
#Введем в модель логарифмы
#Чтобы избежать взятия логарифмов от отрицательных чисел мы взяли их заранее
#Добавим логарифм от параметра Disp в модель
model2 = lm(mpg ~ Logdisp + disp + drat + wt, data)
model2 #p-value - (***)(**)(**)
summary(model2) #Multiple R-squared:  0.8837,      Adjusted R-squared:  0.8665
#Добавим логарифм от параметра drat в модель
model3 = lm(mpg ~ Logdrat + drat + disp + wt, data)
model3 #p-value - ()()(.*())
summary(model3) #Multiple R-squared:  0.7901,      Adjusted R-squared:  0.759
#Добавим логарифм от параметра wt в модель
model4 = lm(mpg ~ Logwt + wt + disp + drat, data)
model4 #p-value - (***)(.*)(*)()

```

```

summary(model4) #Multiple R-squared:  0.8479,      Adjusted R-squared:  0.8254
#Лучший результат получился у модели model2
#Пункт 5
#Ввести в модель всевозможные произведения из пар регрессоров, в том числе квадраты регрессоров
#Найти одну или несколько наилучших моделей по доле объясненного разброса в данных R^2
model5 = lm(mpg ~ disp + drat + I(disp * wt) + wt, data)
model5 #p-value - (*)()(**)(***)
summary(model5) #Multiple R-squared:  0.8511,      Adjusted R-squared:  0.829
model6 = lm(mpg ~ disp + drat + wt + I(disp * drat), data)
model6 #p-value - (*)()(**)(**)
summary(model6) #Multiple R-squared:  0.84, Adjusted R-squared:  0.8163
model7 = lm(mpg ~ disp + drat + wt + I(drat * wt), data)
model7 #p-value (*)()(**)(**)
summary(model7) #Multiple R-squared:  0.8321,      Adjusted R-squared:  0.8072
model8 = lm(mpg ~ disp + drat + wt + I(disp^2), data)
model8 #p-value (**)(**)(***)
summary(model8) #Multiple R-squared:  0.8621,      Adjusted R-squared:  0.8417
model9 = lm(mpg ~ disp + drat + wt + I(drat^2), data)
model9 #p-value - (.)()(*)()
summary(model9) #Multiple R-squared:  0.791, Adjusted R-squared:  0.76
model10 = lm(mpg ~ disp + drat + wt + I(wt^2), data)
model10 #p-value - (*)()(***)(**)
summary(model10) #Multiple R-squared:  0.847,      Adjusted R-squared:  0.8244
#model8 оказалась моделью с самыми хорошими показателями

```

Код номера 2

```

# для чтения .sav файлов
install.packages("devtools")
devtools::install_github("https://github.com/bdemeshev/rlms")
# подключение необходимых библиотек
library("lmtest")
library("rlms")
library("dplyr")
library("GGally")

```

```

library("car")

library("sandwich")

library("Hmisc")

# чтение данных о 17-й волне

data <- rlsms_read("C:\\Users\\AmaZZinG\\Desktop\\practice\\r17i_os26b.sav")

glimpse(data)

# выделяем интересующие нас столбцы

data2 = select(data, mj13.2, mh5, m_marst, m_educ, m_age, status, mj6.2)

#зарплата - mj13.2, пол - mh5, семейное положение - m_marst, наличие высшего образования - m_educ, возраст
- m_age,

#тип населенного пункта - status ,длительность рабочей недели - mj6.2

# отбрасываем пустые поля

data2 = na.omit(data2)

glimpse(data2)

#Разделим респондентов на 3 группы(по семейному положению)

#переменная wed1 имеет значение 1 в случае, если респондент женат, 0 – в противном случае;

#wed2 = 1, если респондент разведён или вдовец;

#wed3 = 1, если респондент никогда не состоял в браке;

#Просматривая другие значения, делаю вывод: смысла в вводе других параметров - нет

#Обнуляем новый столбец

data2["wed1"] = data2$m_marst

data2["wed1"] = 0

#Состоите в зарегистрированном браке = 2, ОФИЦИАЛЬНО ЗАРЕГИСТРИРОВАННЫ, НО ВМЕСТЕ НЕ
ПРОЖИВАЮТ = 6

data2$wed1[which(data2$m_marst == '2') | which(data2$m_marst == '6')] = 1

#Обнуляем новый столбец

data2["wed2"] = data2$m_marst

data2["wed2"] = 0

#разведен, в браке не состоит = 4, вдовец(вдова) = 5

data2$wed2[which(data2$m_marst == '4')] = 1

data2$wed2[which(data2$m_marst == '5')] = 1

#Обнуляем новый столбец

data2["wed3"] = data2$m_marst

data2["wed3"] = 0

#Никогда в браке не состояли = 1

data2$wed3[which(data2$m_marst == '1')] = 1

```

```

#Из параметра пол сделаем переменную sex, имеющую значение 1 для мужчин и равную 0 для женщин
data2["sex"] = data2$mh5
data2$sex[which(data2$sex == '1')] = 1
data2$sex[which(data2$sex == '2')] = 0

# Из параметра, отвечающего типу населённого пункта, создайте одну дамми-переменную city_status
# со значением 1 для города или областного центра, 0 – в противоположном случае.
data2["city_status"] = data2$status
data2["city_status"] = 0
data2$city_status[which(data2$status == '1')] = 1
data2$city_status[which(data2$status == '2')] = 1

#Введите один параметр higher_educ, характеризующий наличие полного высшего образования
data2["higher_educ"] = data2$m_educ
data2["higher_educ"] = 0

#есть полное высшее образование
data2$higher_educ[which(data2$m_educ == '21')] = 1
data2$higher_educ[which(data2$m_educ == '22')] = 1
data2$higher_educ[which(data2$m_educ == '23')] = 1
data2$wed1 = as.numeric(data2$wed1)
data2$wed2 = as.numeric(data2$wed2)
data2$wed3 = as.numeric(data2$wed3)
data2$sex = as.numeric(data2$sex)
data2$city_status = as.numeric(data2$city_status)
data2$higher_educ = as.numeric(data2$higher_educ)

#Факторные переменные, «имеющие много значений», такие как: зарплата, длительность рабочей недели и
возраст,
#- необходимо преобразовать в вещественные переменные и нормализовать их :
# вычесть среднее значение по этой переменной, разделить её значения на стандартное отклонение.

#Зарплата
data2["salary"] = data2$mj13.2
data2$salary = as.numeric(data2$salary)
data2["salary"] = (data2["salary"] - mean(data2$salary)) / sqrt(var(data2$salary))

#длительность рабочей недели
data2["week_len"] = data2$mj6.2
data2$week_len = as.numeric(data2$week_len)
data2["week_len"] = (data2["week_len"] - mean(data2$week_len)) / sqrt(var(data2$week_len))

#возраст

```

```

data2["age"] = data2$m_age
data2$age = as.numeric(data2$age)
data2["age"] = (data2["age"] - mean(data2$age)) / sqrt(var(data2$age))

#Соберем подготовленные данные
data3 = select(data2, salary, sex, wed1, wed2, wed3, higher_educ, age, status, week_len)
glimpse(data3)

#1 Постройте линейную регрессию зарплаты на все параметры, которые Вы выделили из данных мониторинга.

#Не забудьте оценить коэффициент вздутия дисперсии VIF.

#Построю модель зависимости зарплаты от других факторов
model1 = lm(salary ~ sex + wed1 + wed2 + wed3 + higher_educ + age + status + week_len, data3)
summary(model1)

#Все регрессоры, кроме wed1 и wed2, хорошо описывают данные(по 3 - *)

#Строю модель без wed1 и wed2
model2 = lm(salary ~ sex + wed3 + higher_educ + age + status + week_len, data3)
vif(model2)

#уровень мультиколлинеарности низкий
summary(model2)

#2 Поэкспериментируйте с функциями вещественных параметров: используйте логарифм и степени (хотя бы от 0.1 до 2 с шагом 0.1)
describe(data3)

#для логорифмирование необходимо, чтобы значения были > 0
data3["salary"] = data3["salary"] + 1.2
data3["age"] = data3["age"] + 2.1
data3["week_len"] = data3["week_len"] + 3.3

#Строим модель с логарифмами
model3 = lm(salary ~ sex + wed3 + higher_educ + log(age) + status + log(week_len), data3)
summary(model3)

#Все регрессоры имеют высокое значение в модели

#Попробую использовать комбинации регрессоров для построения модели
model4 = lm(salary ~ sex + wed3 + higher_educ + age + status + week_len + I(age * week_len) + I(age^2) + I(week_len^2), data3)
summary(model4)

#комбинации с week_len и wed3 имеют низкий приоритет, исключаем их
model5 = lm(salary ~ sex + higher_educ + age + status + week_len + I(age^2), data3)
summary(model5)

```


#После исключения комбинации с week_len ничего не изменилась, как и прежде показатели модели относительно хороши

#Построю модель на комбинациях логарифмов регрессоров

```
model6 = lm(salary ~ sex + higher_educ + log(age) + status + log(week_len) + I(log(age) * log(week_len)) + I(log(age)^2) + I(log(week_len)^2), data3)
```

```
summary(model6)
```

#Теперь снизились показатели модели, а так же влияние комбинации с log(age) и I(log(age) * log(week_len)) построим модель без них

```
model7 = lm(salary ~ sex + higher_educ + status + log(week_len) + I(log(age)^2) + I(log(week_len)^2), data3)
```

```
summary(model7)
```

#Показатели модели незначительно повысились

#Построим модель с квадратами логарифмов

```
model8 = lm(salary ~ sex + higher_educ + age + log(age^2)
```

```
+ status + week_len + log(week_len^2), data3)
```

```
summary(model8)
```

#Качество модели относительно высокое, однако значимость регрессора log(week_len^2) не высока

#3 Выделите наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённой с помощью построенных зависимостей разбросу adjusted R² - R².adj.

#1)model5: R²=0.2011 ,(значимость параметров – максимальная ***)

#2)model7: R²=0.1861 ,(значимость параметров – максимальная ***)

#3)model8 : R²=0.1985, (значимость параметров – максимальная ***)

В итоге лучшая по показателям model5

#4 Сделайте вывод о том, какие индивиды получают наибольшую зарплату

#4 Сделайте вывод о том, какие индивиды получают наибольшую зарплату

```
summary(model2)
```

#Вывод:наибольшую зарплату получает мужчина (estimate: sex > 0), с высшим образованием (estimate: higher_educ > 0),

#преимущественно женатый (estimate: wed3 < 0), однако этот показатель не вносит наибольший вклад,

#этот мужчина примерно средних лет (estimate: age ~ 0), тем не менее обычно это не городской житель (estimate: status < 0),

#что очень странно, т.к. в реальной ситуации у городского жителя преимущественно большая зарплата.

#Так же у него наблюдаются переработки (estimate: duration > 0). В целом, в модели присутствуют некоторые неточности,

#но схожести с реальностью присутствуют.

#5 1)Городские жители, не состоявшие в браке; 2)разведенные женщины, без высшего образования

#1)Данная группа индивидов теряет в зарплате из-за того, что это городские жители (status < 0) да и ещё несостоявшие в браке (wed3 < 0) ,

#в итоге у данной группы индивидов зарплата ниже среднего

#2)Эта группа индивидов получает относительно низкую з/п из-за отсутствия высшего образования, а регрессия по зарплате говорит о том,

#что высшее образование играет значительную роль в ее размере, также зарплата снижается из-за женского пола(estimate: sex > 0),

#то что они разведены, оказывает минимальное значение на их зарплату по модели m (wed2 ~ 0)

Код номера 3

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data2 =
pd.read_csv('C:/Users/AmaZZinG/Desktop/practice/train.csv',usecols=['MachineIdentifier','CountryIdentifier','AVProductStatesIdentifier','OrganizationIdentifier','Census_ProcessorCoreCount','HasDetections','Census_TotalPhysicalRAM'], nrows = 1000)

print("data2.shape=",data2.shape)

data2.info()

np.sum(pd.isnull(data2))

data2.dropna(inplace=True)

data2.info()

plt.figure(figsize=(20,16))

plt.scatter(data2.CountryIdentifier, data2.OrganizationIdentifier, s=9, c=data2.AVProductStatesIdentifier, cmap =
'seismic')

plt.colorbar()

plt.xlabel('CountryIdentifier')

plt.ylabel('OrganizationIdentifier')

def maxstd(data2):
    max = 0
    max_name = str()
    for i in data2.columns:
        num = (data2[i] - data2[i].mean()) / data2[i].std()
        data2[i] = num
        if max < num.mean():
            max_name, max = i, num.mean()
    return max_name
```

```
maxstd(data2)

target = data2.HasDetections

train = data2

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(train, target, test_size = 0.3, random_state = 42)

N_train, _ = X_train.shape

N_test, _ = X_test.shape

print (N_train, N_test)
```

```
from sklearn.decomposition import PCA

%matplotlib inline

import matplotlib.pyplot as plt

pca = PCA()

pca.fit(X_train)

X_pca = pca.transform(X_train)

for i, component in enumerate(pca.components_):

    print("{} component: {}% of initial variance".format(i + 1,

        round(100 * pca.explained_variance_ratio_[i], 2)))

    print(" + ".join("%.3f x %s" % (value, name)

        for value, name in zip(component, train.columns)))

plt.figure(figsize=(10,7))

plt.plot(np.cumsum(pca.explained_variance_ratio_), color='k', lw=2)

plt.axhline(0.9, c='r')

plt.axvline(1, c='b')
```

