# 

**ОГЛАВЛЕНИЕ**

Введение..................................................................................................2

Основная часть.......................................................................................3

Задача № 1......................................................................................3

Задача № 2....................................................................................13

Задача № 3....................................................................................23

Список литературы...............................................................................30

Код задачи № 1.............................................................................30

Код задачи № 2.............................................................................33

Код задачи № 3.............................................................................38

**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 X1, X2, ..., Xp 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 mean(x) и поделив на среднеквадратическое отклонение σ ~ sqrt(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 R2 в каждой из них не высокий). В случае, если R2 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) R2, 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 R2.

**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**:

# 

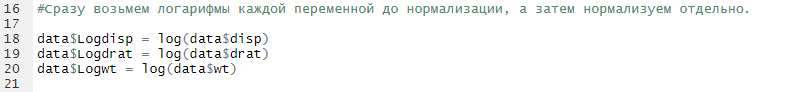
# Все столбцы таблицы содержат числа, причем признаки **vs** и **am** – бинарные.

# 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).

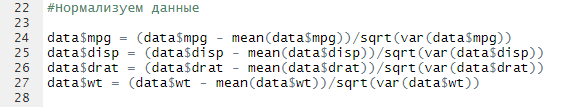
Подключим библиотеку dplyr и прочитаем данные:

# 

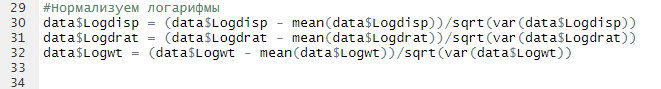
Вычислим **data$Logdisp**, **data$Logdrat**, **data$Logwt** :



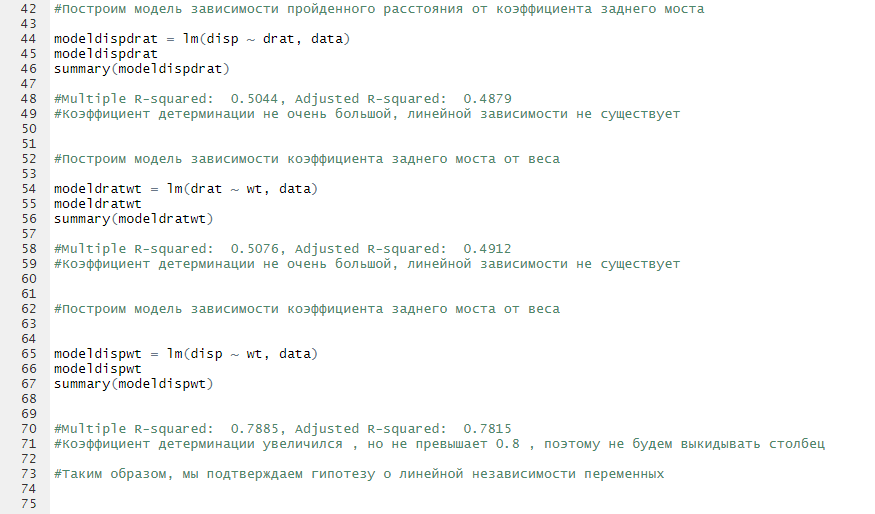
Нормализуем данные, вычтя из каждого столбца среднее значение mean(x) и поделив на среднеквадратическое отклонение σ ~ sqrt(var(x)):



Также нормализуем логарифмы:



**2.**Проверить, что в наборе данных нет линейной зависимости (построить зависимости между переменными, указанными в варианте, и проверить, что R2 в каждой из них не высокий). В случае, если R2 большой, один из таких столбцов можно исключить из рассмотрения.



# **Disp ~ Drat: R^2 = 0.4879 VIF = 1.952682**

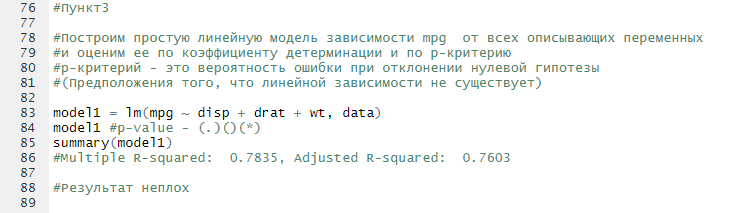
# **Disp ~ Wt : R^2 = 0.7815 VIF = 4.575792**

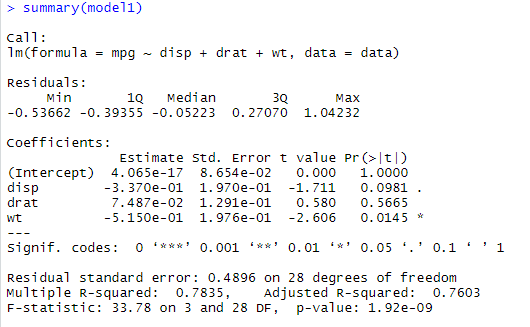
# **Drat ~ Wt : R^2 = 0.4912 VIF = 1.965244**

# 

# 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) R2, 2) p-значениям каждого коэффициента.

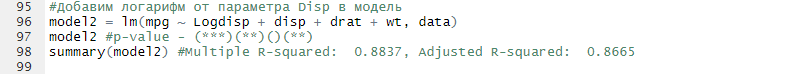
****

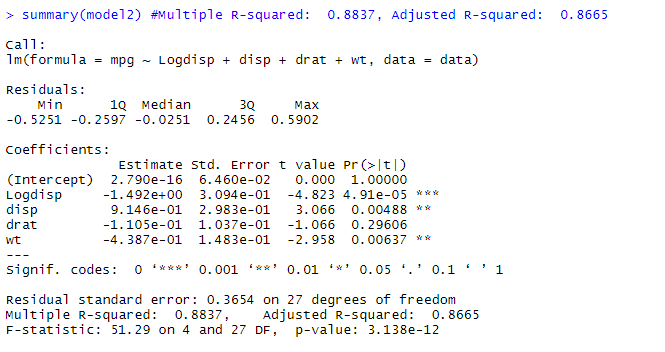
****

Построенная модель имеет неплохие показатели: получены высокая доля объясненной дисперсии **R^2** = 0.7603 и вероятность **p-value** = 1.92 \* 10-9 << 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.

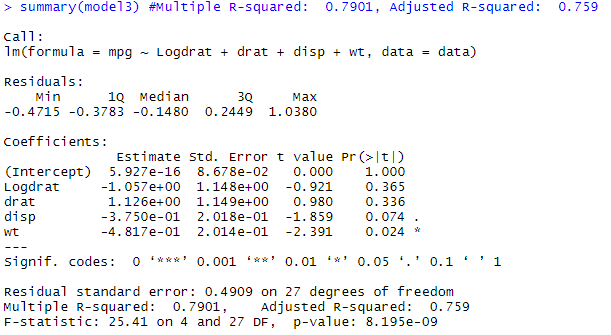
****

****

**R^2** = 0.8665, **p-value** = 3.138\* 10-12

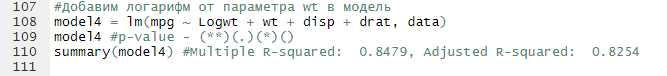
Получили прирост значений параметров **R^2**, а также уменьшение **p-value**.

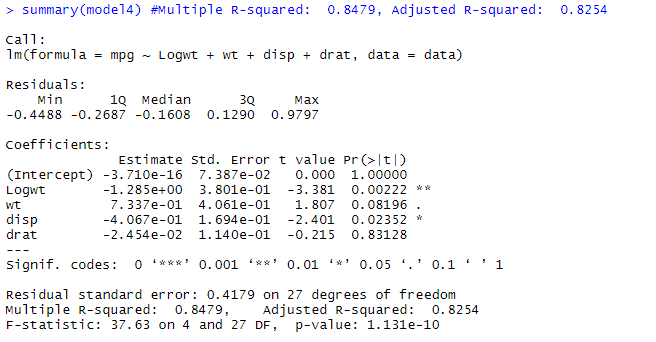
# 



**R^2** = 0.759, **p-value** = 8.195\* 10-9

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





**R^2** = 0.8254 **p-value** = 1.131\* 10-10

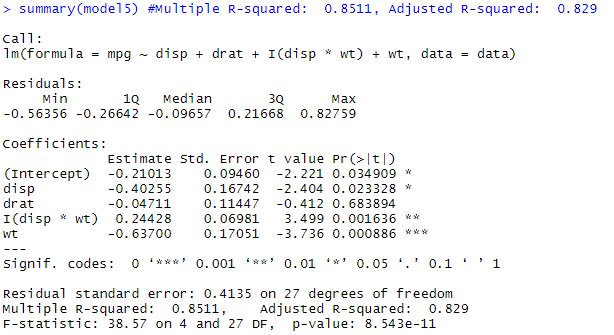
Параметр **R^2** cтал немного хуже по сравнение с **model 2**, а также получилась не самая лучшая значимость.

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 R2.

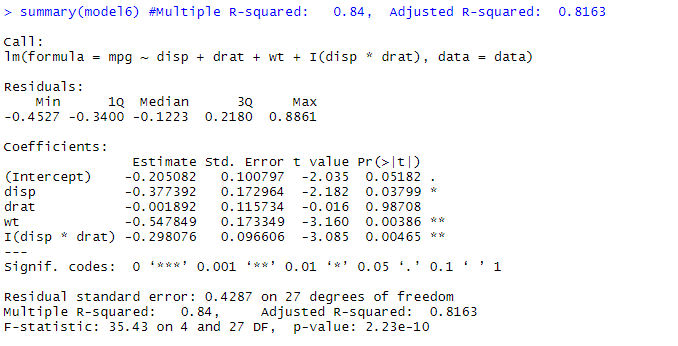




**R^2** = 0.829 **p-value** = 8.54 \* 10-11

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





**R^2** = 0.8163 **p-value** = 2.23 \* 10-10

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

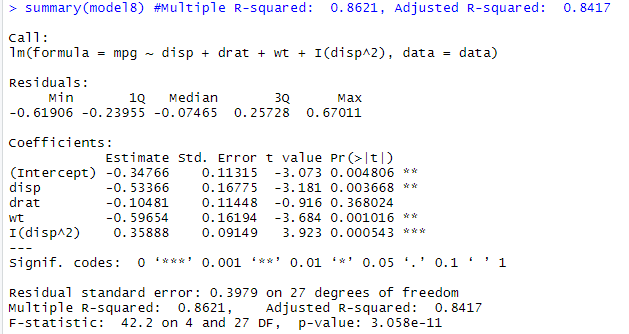


# 

**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.

# 

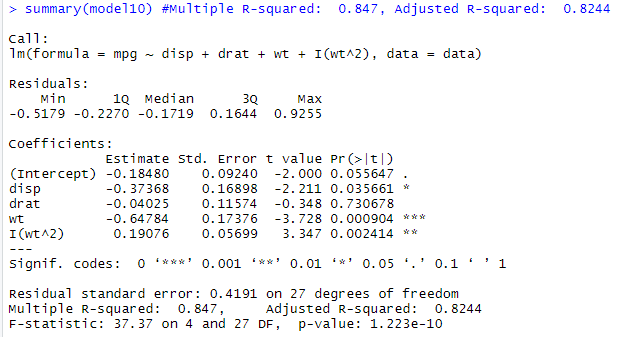


**R^2** = 0.8417 **p-value** = 3.058\* 10-11

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

Пока что лучшая получившаяся модель.





**R^2** = 0.8244 **p-value** = 1.223\* 10-10

**R^2** и значимость упали по сравнение с моделью 8. Однако показатель **p-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 этой̆ 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.

Минимальный̆ 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, зарплаты или одного из параметров wed) на эти переменные и использовав команду **VIF** для неё).

Из параметра пол сделаете 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,** характеризующей 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 этой̆ 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 зависимостей̆ разбросу adjusted R2 - R2adj.

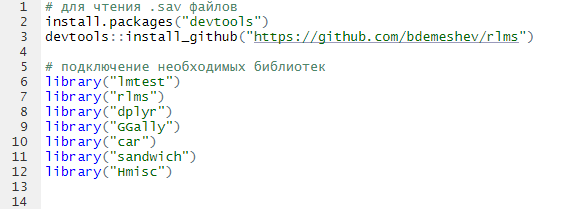
4. Сделайте вывод о том, какие индивиды получают наибольшую зарплату.

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

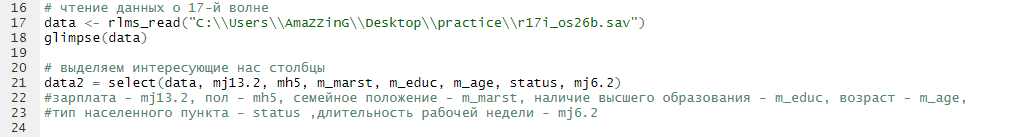
Решение

17 Волна. Файл – “ r17i\_os26b.sav”.

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



Составим базу из данных: **mj13.2**, **m\_age**, **m\_educ**, **status, mj6.2**, **m\_marst**, **mh5**



**mj13.2** - среднемесячная зарплата

**m\_age** – возраст

**mh5** – пол

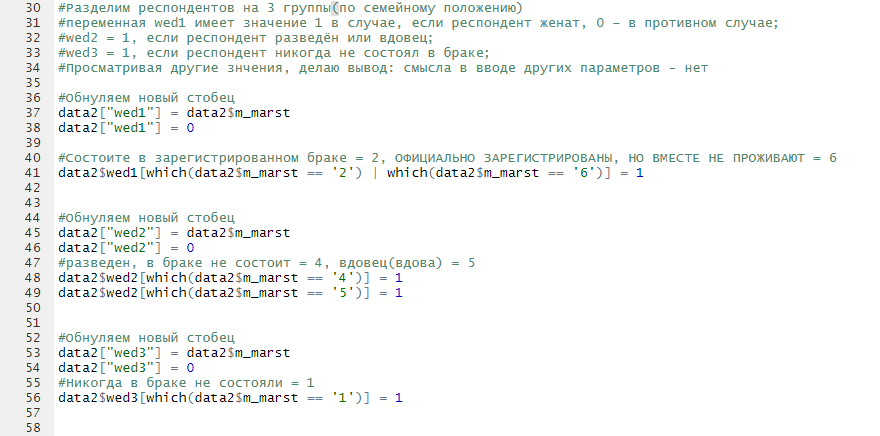
**m\_educ** - Образование

**status** – Тип населённого пункта

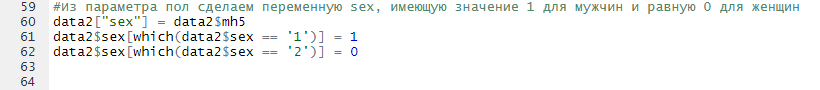
**mj6.2** - Сколько часов в среднем продолжается Ваша обычная рабочая неделя?

**m\_marst** - Семейные положение мы переделываем в (wed, wed1, wed2, wed3, wed4)

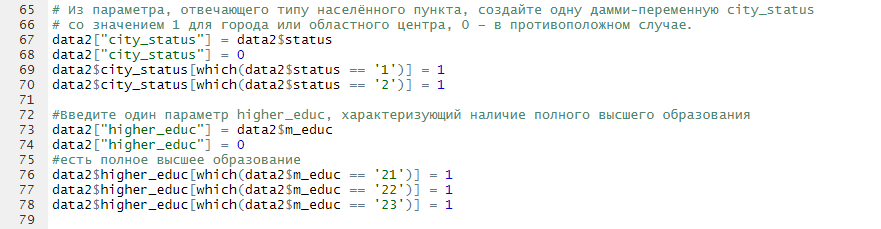
Из параметра, отвечающего семейному положению, сделать дамми-переменные (с помощью one-hot-encoding): 1) переменная **wed1** имеет значение 1 в случае, если респондент женат, 0 – в противном случае; 2) **wed2**=1, если респондент разведён или вдовец; 3) **wed3** = 1, если респондент никогда не состоял 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 использовав команду **VIF** для неё).



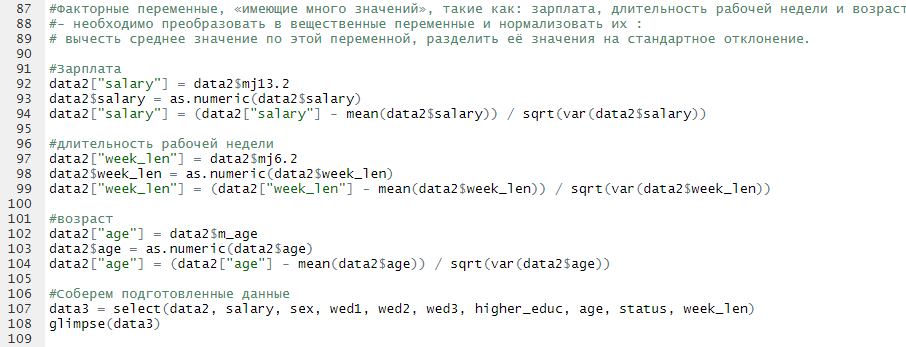
Из параметра пол сделаете переменную sex, имеющую значение 1 для мужчин и равную 0 для женщин.



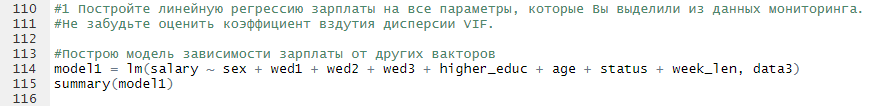
Из параметра, отвечающего типу населённого пункта, создайте одну дамми-переменную **city\_status** со значением 1 для города или областного центра, 0 – в противоположном случае. Введите один параметр **higher\_educ,** характеризующей наличие полного высшего образования.

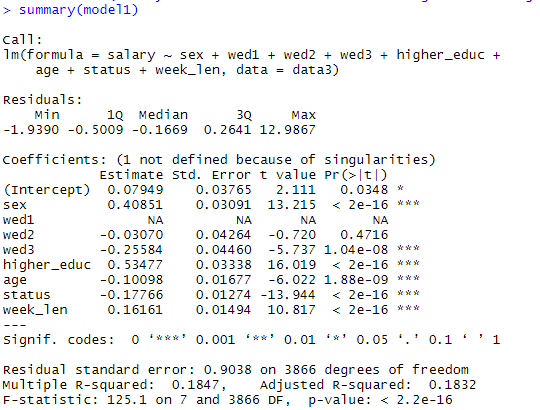


Факторные переменные, «имеющие много значений», такие как: зарплата, длительность рабочей недели и возраст, - необходимо преобразовать в вещественные переменные и нормализовать их: вычесть среднее значение по этой̆ переменной, разделить её значения на стандартное отклонение.



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

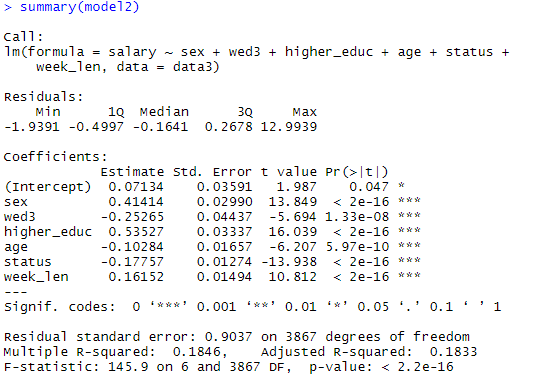




Вывод: **R^2** = 0.1832

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

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



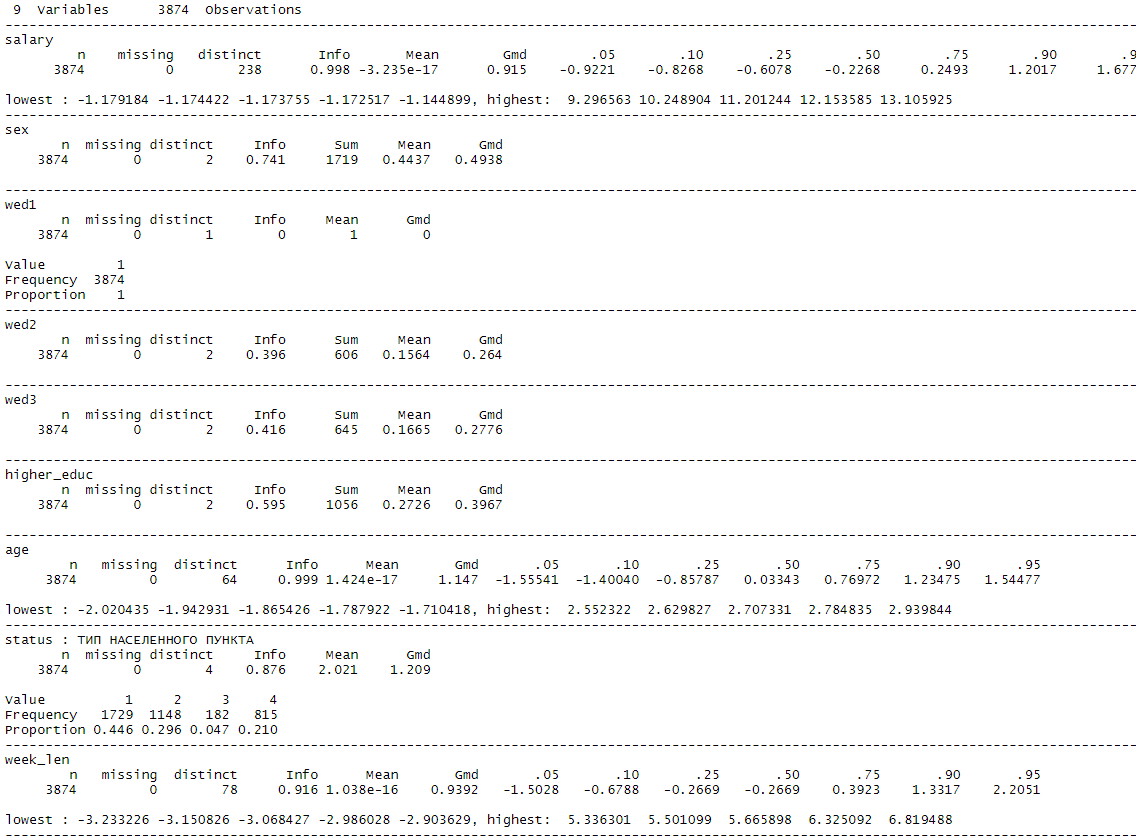
**R^2** = 0.1833



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.**

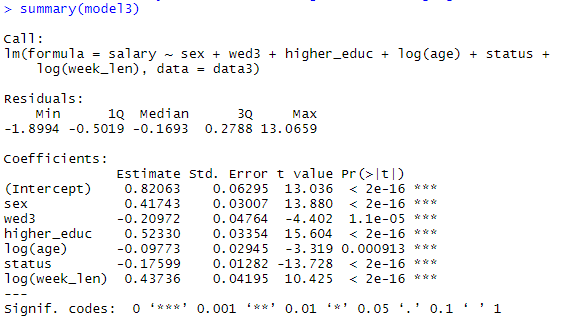
2. Поэкспериментируйте с функциями вещественных параметров: используйте логарифм и степени (хотя бы от 0.1 до 2 с шагом 0.1).



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

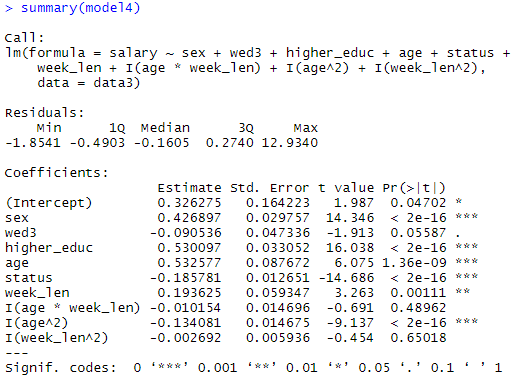




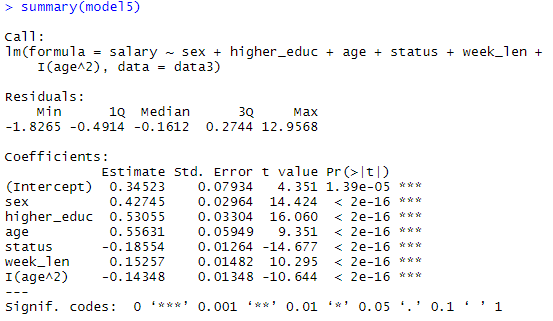


Все регрессоры имеют высокое значение в модели.

Пробую использовать комбинации регрессоров для построения модели.

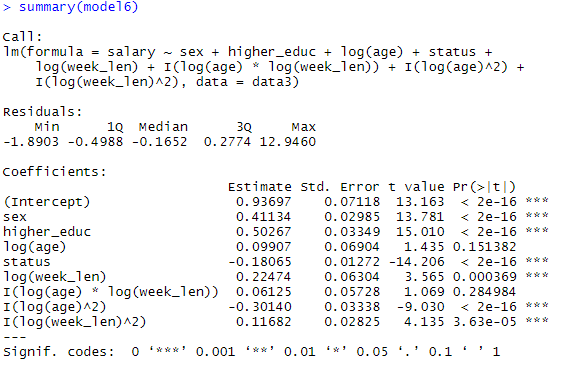


Комбинации с **week\_len** и **wed3** имеют низкий приоритет, исключаем их

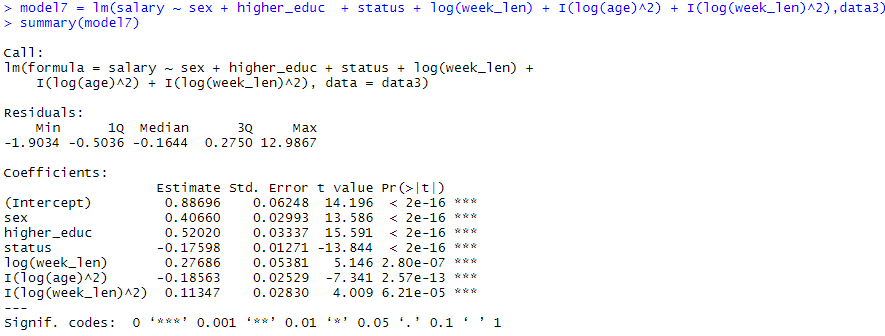


После исключения комбинации с **week\_len** ничего не изменилась, как и прежде показатели модели относительно хороши

Построим модель на комбинациях логарифмов регрессоров

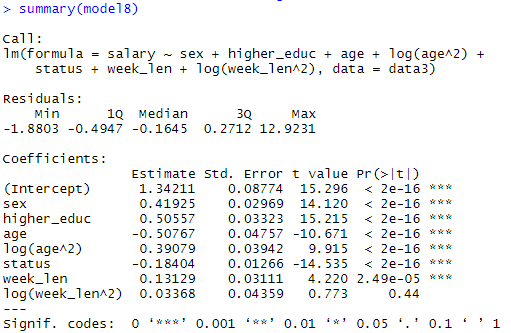


Теперь снизились показатели модели, а так же влияние комбинации с **log(age)** и **I(log(age) \* log(week\_len))** построим модель без них



Показатели модели незначительно повысились.

Построим модель с квадратами логарифмов



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 объяснённому с помощью построенных зависимостей̆ разбросу adjusted R2 - R2adj.

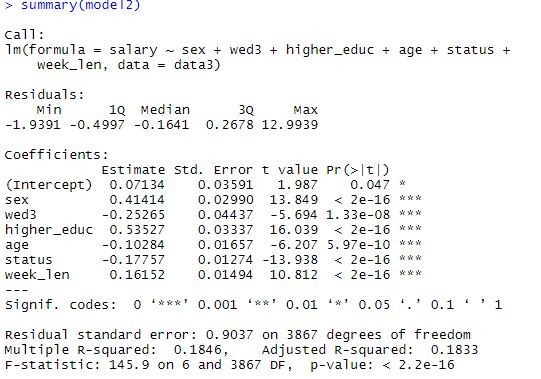
1)model5: **R^2**=0.2011 ,(значимость параметров – максимальная \*\*\*)

2)model7: **R^2**=0.1861 ,(значимость параметров – максимальная \*\*\*)

3)model8 : **R^2**=0.1985, (значимость параметров – максимальная \*\*\*)

В итоге лучшая по показателям **model5**

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



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**

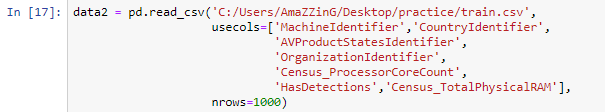
Необходимо провести анализ вашего датасета и сделать обработку данных.

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:

****

****

**1.Сколько в датасете объектов и признаков? Дать описание каждому**

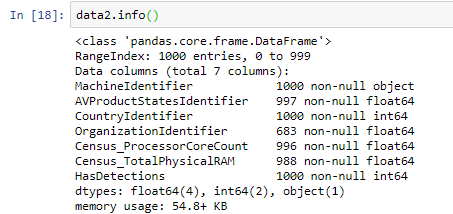
**признаку, если оно есть.**

Определим размеры нашей таблицы с данными:

******

В нашей таблице с данными 1000 строки и 4 столбца.

Прежде чем приступать к обработке данных, посмотрим на наши признаки:



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

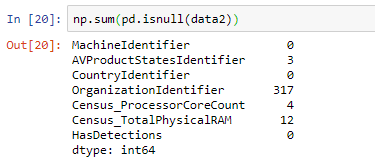
**3.Столбец с максимальным количеством уникальных значений категориального признака?**

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?**

****

Да, есть, в вещественных столбцах: **AVProductStatesIdentifier**, **OrganizationIdentifier, Census\_ProcessorCoreCount, Census\_TotalPhysicalRAM**

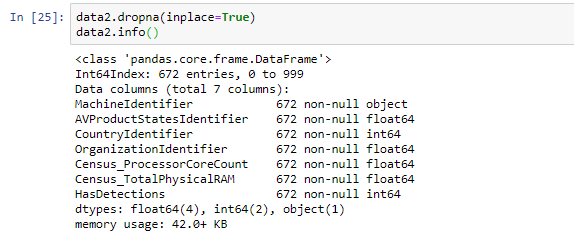
**6. Сколько объектов с пропусками?**

336 объектов с пропусками.

**7. Столбец с максимальным количеством пропусков?**

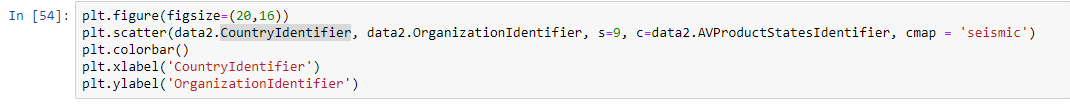
Исходя из проделанной операции, можно сделать вывод, что пропусков максимальное количество в столбце **OrganizationIdentifier**

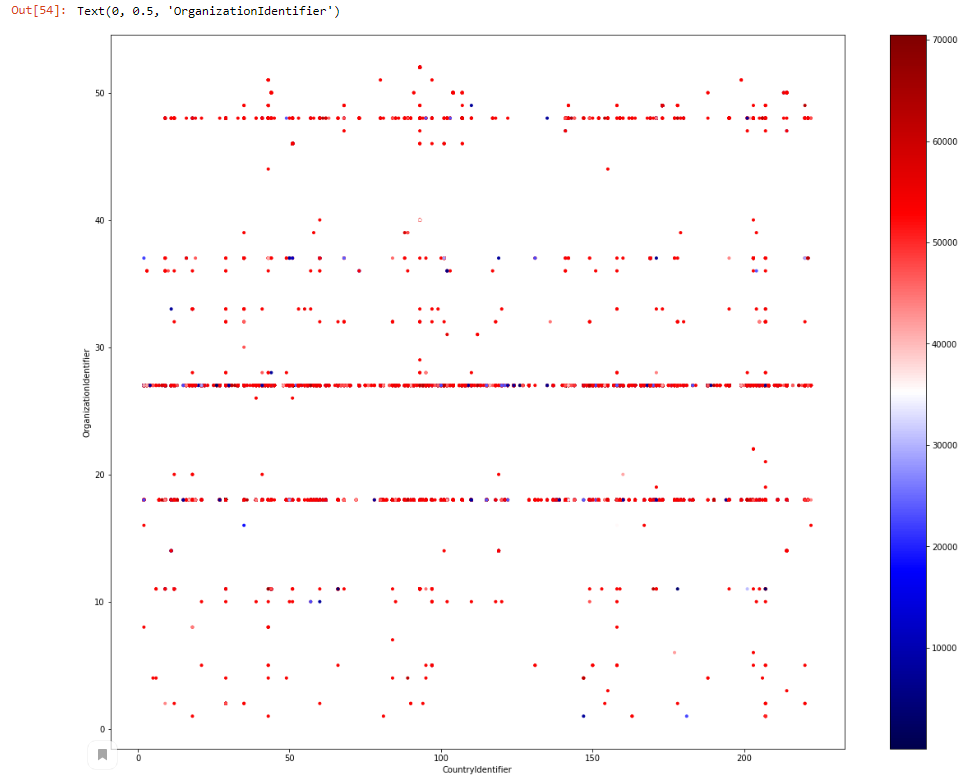
Для дальнейшего анализа, уберём все строки с пропусками.

****

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

Для начала обработаем столбец с категориальным признаком.

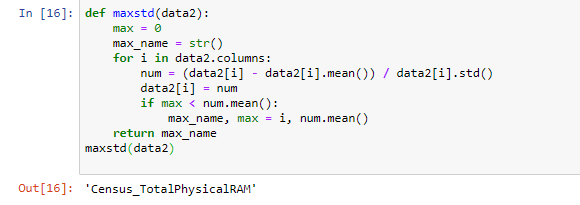
Проверим наш датасет на наличие аномальных значений с помощью графика:  
******



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 смысл использовать.**

****

После проверки условия, столбцом с максимальным средним после

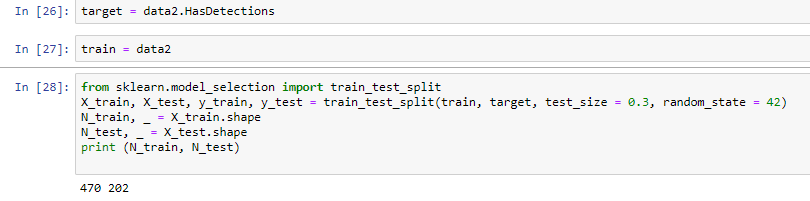
нормировки признаков через стандартное отклонение, является столбец

**Census\_TotalPhysicalRAM**

**10. Столбец с целевым признаком**?

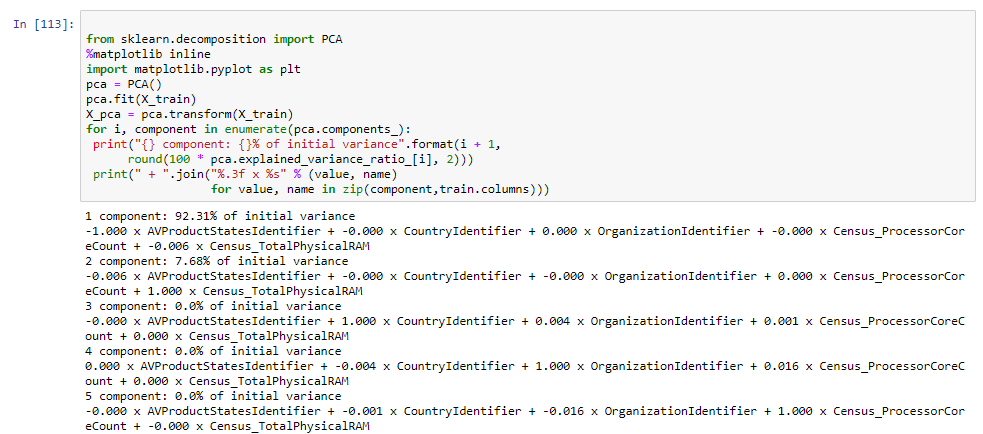
Опираясь на условие задачи, можно сделать вывод, что целевой признак - это **HasDetection**

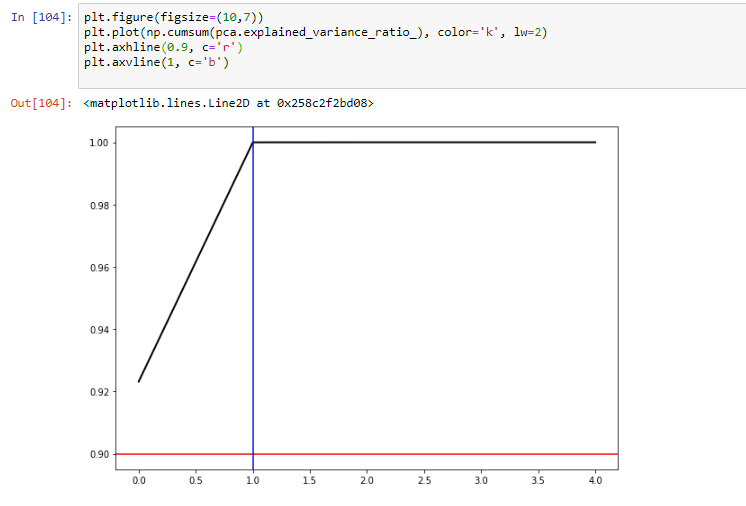
**11. Сколько объектов попадает в тренировочную выборку при использовании train\_test\_split с параметрами test\_size = 0.3, random\_state = 42?**

****

Заметим, что в тренировочную выборку попадает 470 объектов.

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





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 @Lexxo 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 среднеквадратичное 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 p-критерию

#p-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 <- rlms\_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 знчения, делаю вывод: смысла в вводе других параметров - нет

#Обнуляем новый стобец

data2["wed1"] = data2$m\_marst

data2["wed1"] = 0

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

data2$wed1[which(data2$m\_marst == '2') | which(data2$m\_marst == '6')] = 1

#Обнуляем новый стобец

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 стобец

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 вакторов

model1 = lm(salary ~ sex + wed1 + wed2 + wed3 + higher\_educ + age + status + week\_len, data3)

summary(model1)

#все регрессоры, кроме wed1 и wed2, хорошо описывают даные(по 3 - \*)

#Строю модель без wed1 и wed2

model2 = lm(salary ~ sex + wed3 + higher\_educ + age + status + week\_len, data3)

vif(model2)

#уровень мультиколлиниарности 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 значеие 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 объяснённому с помощью построенных зависимостей разбросу adjusted R^2 - R^2.adj.

#1)model5: R^2=0.2011 ,(значимость параметров – максимальная \*\*\*)

#2)model7: R^2=0.1861 ,(значимость параметров – максимальная \*\*\*)

#3)model8 : R^2=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 data3.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')