Домашнее задание (ht_a)

Арина Булыгина

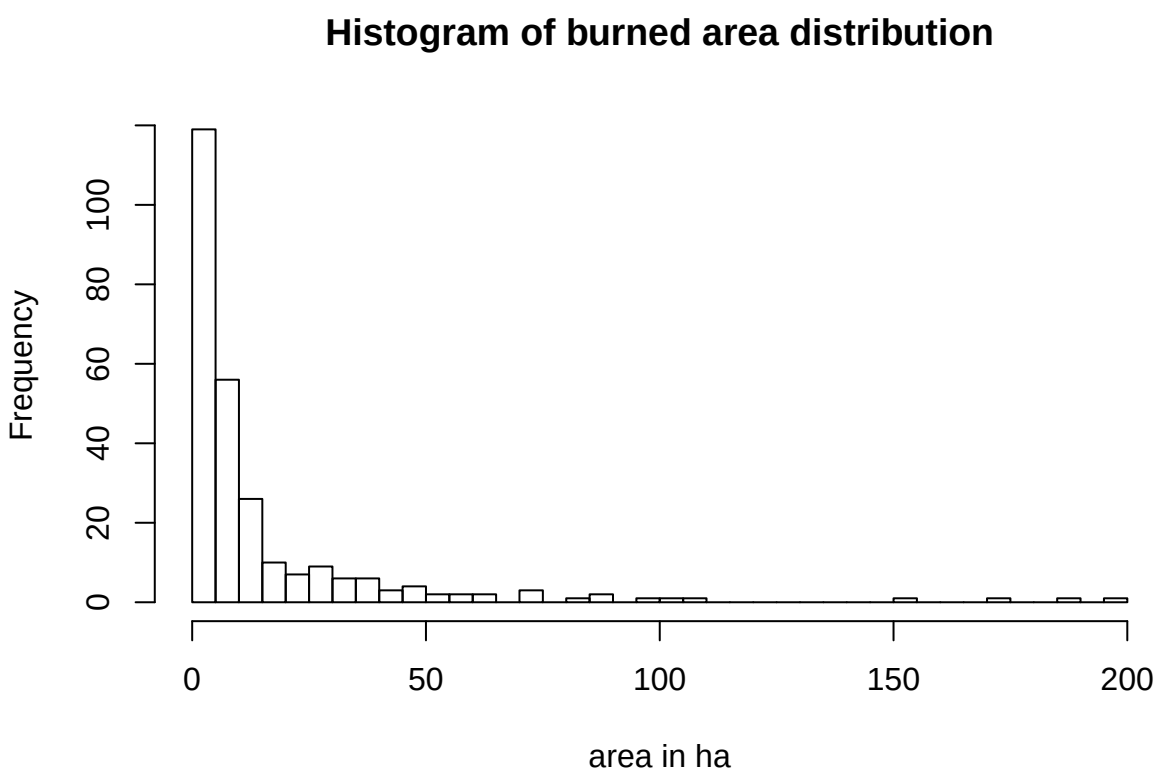
Подключаем необходимые пакеты

№0. Загружаем данные и устанавливаем seed для воспроизводимости.

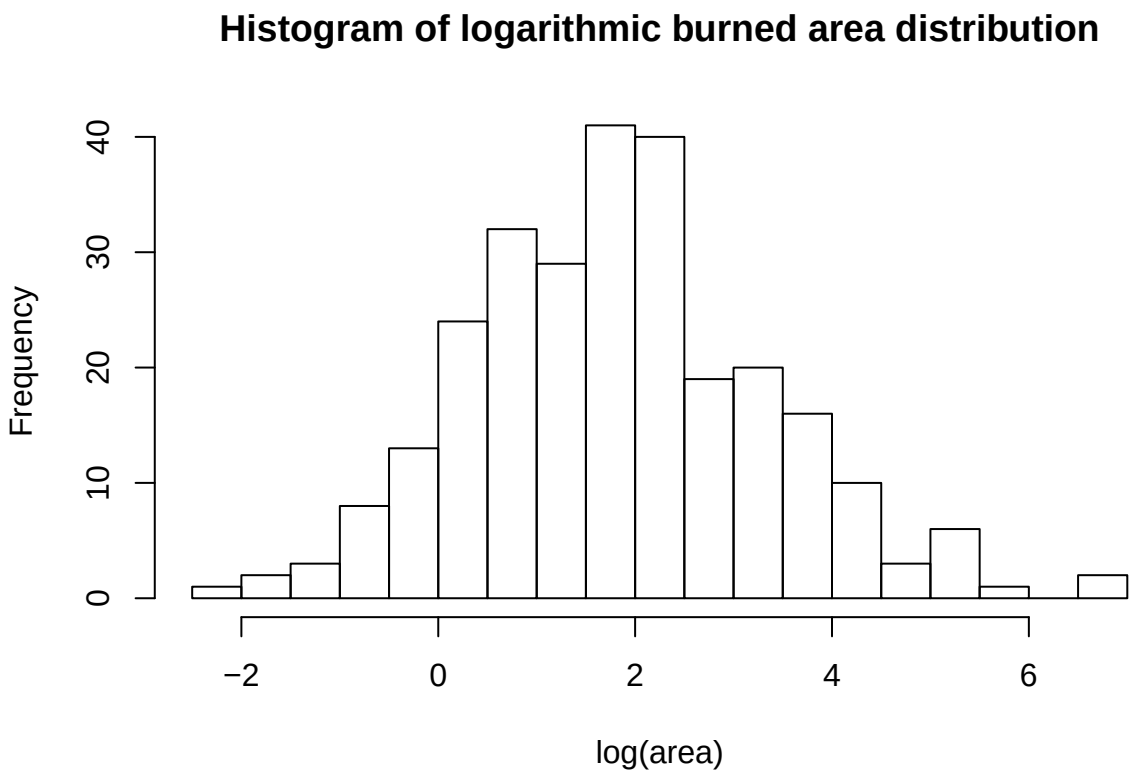
```
data <- read.csv('https://archive.ics.uci.edu/ml/machine-learning-databases/forest-fires/forestfires.csv', header = TRUE)
set.seed(14)
```

№1 Убираем нулевые значения area. Интересно, что она, как видно из гистограммы, имеет распределение похожее на логнормальное с сильно растянутым хвостом, отвечающим за крупные пожары. Логарифмирование превращает распределение в нормальное: Шапиро-Уилк показывает p-value в 0.24, значит на любом вменяемом уровне значимости мы не отвергаем гипотезу о нормальности. Но по заданию используем просто area. На случай, если пригодится, month и day преобразовываем в факторы.

```
data <- data[which(data$area != 0), ]
hist(data$area[which(data$area<200)],breaks=50,xlab = "area in ha", main = paste("Histogram of" , "burned area distribu
```




```
data$month <- as.factor(data$month)
data$day <- as.factor(data$day)
hist(log(data$area), breaks=20, xlab = "log(area)", main = paste("Histogram of" , "logarithmic burned area distribution")
```




```
shapiro.test(log(data$area))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data$area)
## W = 0.99303, p-value = 0.2402
```

№2 Отбираем признаки для модели из содержательных соображений.

Базовые переменные для погодных условий:

temp,RH,wind,rain - выбираем, так как в напрямую показывают погодные условия. Ожидаемые знаки коэффициентов при переменных:

rain <0: ожидаем отрицательную зависимость, так как чем меньше дождя в какой-либо области, тем более высокой будет площадь пожара. Проблема с этой переменной в том, что у неё всего 2 ненулевых значения, а значит она бесполезна и далее в регрессиях она встречаться не будет.

wind >0 : чем выше скорость ветра, тем быстрее и дальше разносится пожар в какой-либо области, значит, площадь пожара будет выше. С другой стороны, сильный ветер в условиях средней влажности может наоборот гасить небольшие источники пламени, что уменьшает потенциальный ущерб, однако эта логика больше применима к общему случаю, когда мы не исключаем 0 в area.

temp >0: чем выше температура, тем больше вероятность возгорания травы и сухостоя, тем больше потенциальных очагов и, следовательно, площадь пожара растёт.

RH <0: больше влажность - больше паров воды в воздухе, значит отсыревает потенциально горючий мусор и ветки, значит меньше вероятность пожара. Понятно, что важна относительность влажности, чтобы говорить о количестве воды, но мы здесь в основно говорим про лето и высокую температуру, так что насыщенность примерно сопоставима, но возможно учёт совместно двух этих параметров в виде произведения даст больше предсказывающей мощности.

Индексы, которые можно дополнительно включить:

FFMC, DMC, ISI, DC - различные комбинации факторов выше, которые тоже в целом говорят о вероятности возгорания и возможной скорости распространения пожара. Тут проблема может возникнуть со смещением из-за ошибок измерения и, вероятно, с мультиколлинеарностью, так как факторы связаны. Ожидаемые знаки коэффициентов:

DMC, DC, FFMC >0 : по методологии FWI, общая идея индексов - растут, когда вероятность пожара или его распространения увеличивается. DMC - про влажность возгораемых веществ в толще земли, растёт с понижением влажности (он самый важный из указанных для нас, так как отвечает именно за распространение, а не возникновение, потому что подсчитывает засуху именно в средних слоях почвы, которые загораются не первыми). DC - про засуху глубинных слоёв (увеличивается с ней), FFMC - про вероятность возгорания(тоже через влажность) (Примечание: индексы имеют лаг измерения, т.е. сегодняшний индекс на самом деле построен по какому-то измерению n дней назад с оценкой испарения. Это сознательный выбор методологов из-за разной волатильности влажности в разных слоях почвы: замерять в глубине можно менее часто, чем на поверхности. Наличие такого лага возможно сильно зашумляет данные по индексам, так что многого ждать не стоит.)

ISI >0: это главная оценка скорости распространения(тоже значимый). Он агрегирует в себе предыдущие индексы, а также отдельно ветер, поэтому с ним вероятнее всего будет сильная мультиколлинеарность, хотя зависимости не линейные, так что посмотрим ниже.

Переменные для контроля:

day: скорее всего по выходным будет бОльшая вероятность пожара из-за повышения числа людей, приехавших отдохнуть. Мусорят, не следят за кострами - вот и боль. Как контрольные факторы можно включить.

month: судя по данным в августе и сентябре больше всего пожаров. Это логично, потому что сезоны засухи и пожаров как раз летом. Логично предположить, что в это время чисто статистически площадь будет больше, но мы тут исследуем влияние именно погодных условий, так что сгодится разве что на контроль.

Описательная статистика по интересующим переменным в коде ниже:

```
data <- data[,-c(1,2)]
res <- stat.desc(data[, -c(1:2)])
round(res, 2)
```

```
##                 FFMC      DMC       DC     ISI    temp      RH    wind
## nbr.val       270.00   270.00   270.00  270.00  270.00  270.00  270.00
## nbr.null        0.00     0.00     0.00    0.00    0.00    0.00    0.00
## nbr.na          0.00     0.00     0.00    0.00    0.00    0.00    0.00
## min            63.50     3.20    15.30    0.80    2.20   15.00    0.40
## max            96.20   291.30   860.60   22.70   33.30   96.00    9.40
## range          32.70   288.10   845.30   21.90   31.10   81.00    9.00
## sum         24579.20 30971.10 154134.10 2477.80 5214.00 11808.00 1110.50
## median         91.70   111.70   665.60    8.40   20.10   41.00    4.00
## mean           91.03   114.71   570.87    9.18   19.31   43.73    4.11
## SE.mean         0.23     3.76    14.00    0.25    0.38    0.92    0.11
## CI.mean.0.95    0.44     7.40    27.56    0.50    0.74    1.81    0.23
## var            13.76  3817.57 52891.37   17.20   38.19  227.41    3.55
## std.dev         3.71    61.79   229.98    4.15    6.18   15.08    1.88
## coef.var        0.04     0.54     0.40    0.45    0.32    0.34    0.46
##                rain    area
## nbr.val      270.00  270.00
## nbr.null     268.00    0.00
## nbr.na         0.00    0.00
## min            0.00    0.09
## max            6.40 1090.84
```

```
## range          6.40 1090.75
## sum            7.80 6642.05
## median         0.00    6.37
## mean           0.03   24.60
## SE.mean        0.02    5.26
## CI.mean.0.95   0.05   10.36
## var            0.16 7482.53
## std.dev        0.40   86.50
## coef.var      13.79    3.52
```

Смотрим на гистограммы распределения и сглаженные плотности.

Как можно увидеть, у многих переменных распределение похоже на нормальное или близкое к нему. Реально же гипотеза о нормальности распределения в тесте Шапиро-Уилка отвергается для всех переменных. Это в общем-то логично, так как мы сильно урезали данные до экстремальных случаев, когда происходило возгарание, то есть убрали потенциально важные для "нормальности" диапазоны значений объясняющих переменных.

Значимые замечания:

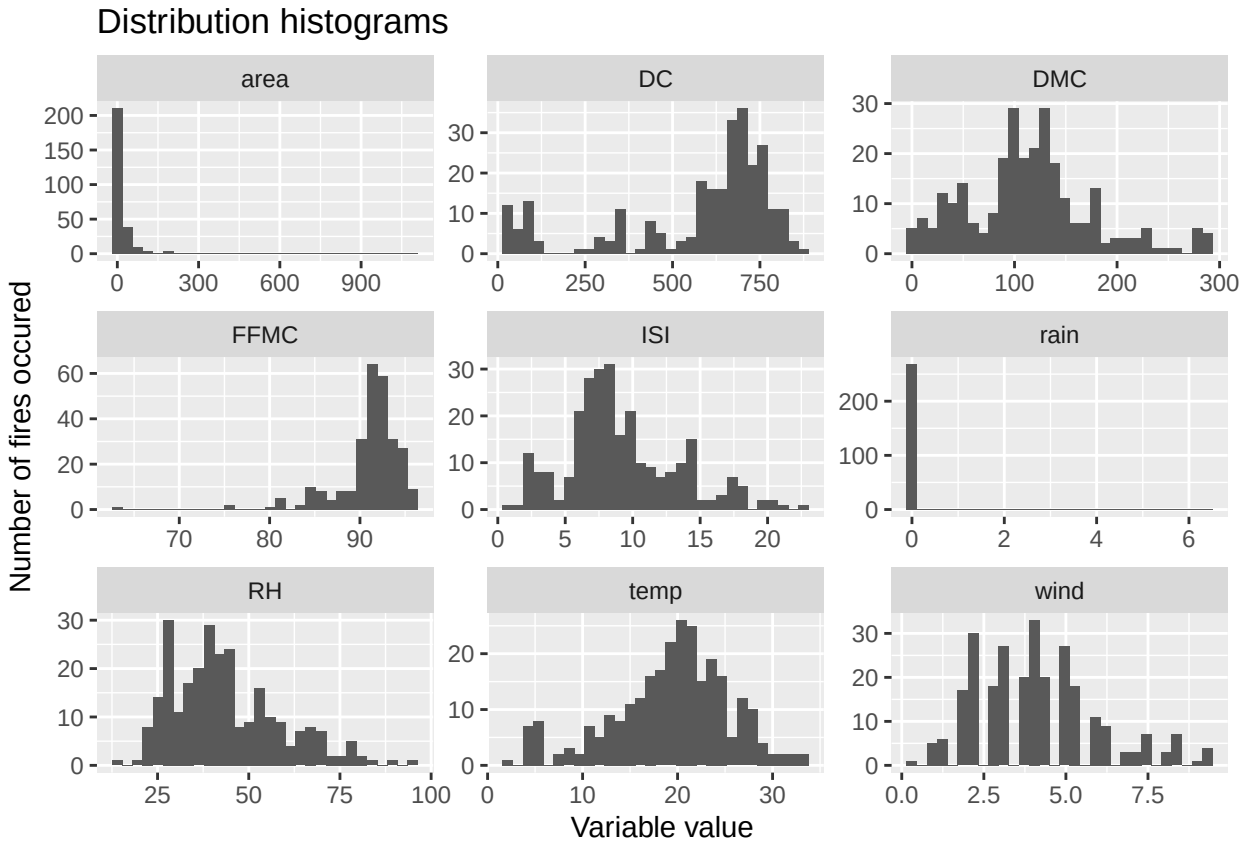
FFMC в силу специфики расчёта, как говорит сайт канадской системы FWI, обычно колеблется в слабом диапазоне в периоды пожаров, и лишь сигнализирует, когда сильные дожди пропитывают землю (он в этих случаях снижает значение), отсюда и такое остроконечное распределение с редкими низкими значениями.

Смещение RH в сторону низких значений обусловлено тем, что у нас здесь данные по случившимся пожарам, а исходя из логики,описанной выше, они как раз случаются при влажности низкой, при прочих равных.

Содержательно инетерсно, что значения температуры сильно разбросаны (2-33). Скорее всего это связано с наличием антропогенного фактора: даже в холодную погоду были пожары, потому что люди жгли костры. Реально для подтверждения этой гипотезы данных нет, но возгорания зимой (dec, feb) случались даже чаще, чем в весенние месяцы. Коли речь о месяцах, упомяну, что, как и отмечалось, большая часть пожаров происходит в районе августа, сентября (примерно 2/3 всех). Возможно стоит отдельно рассмотреть пожары за эти периоды.

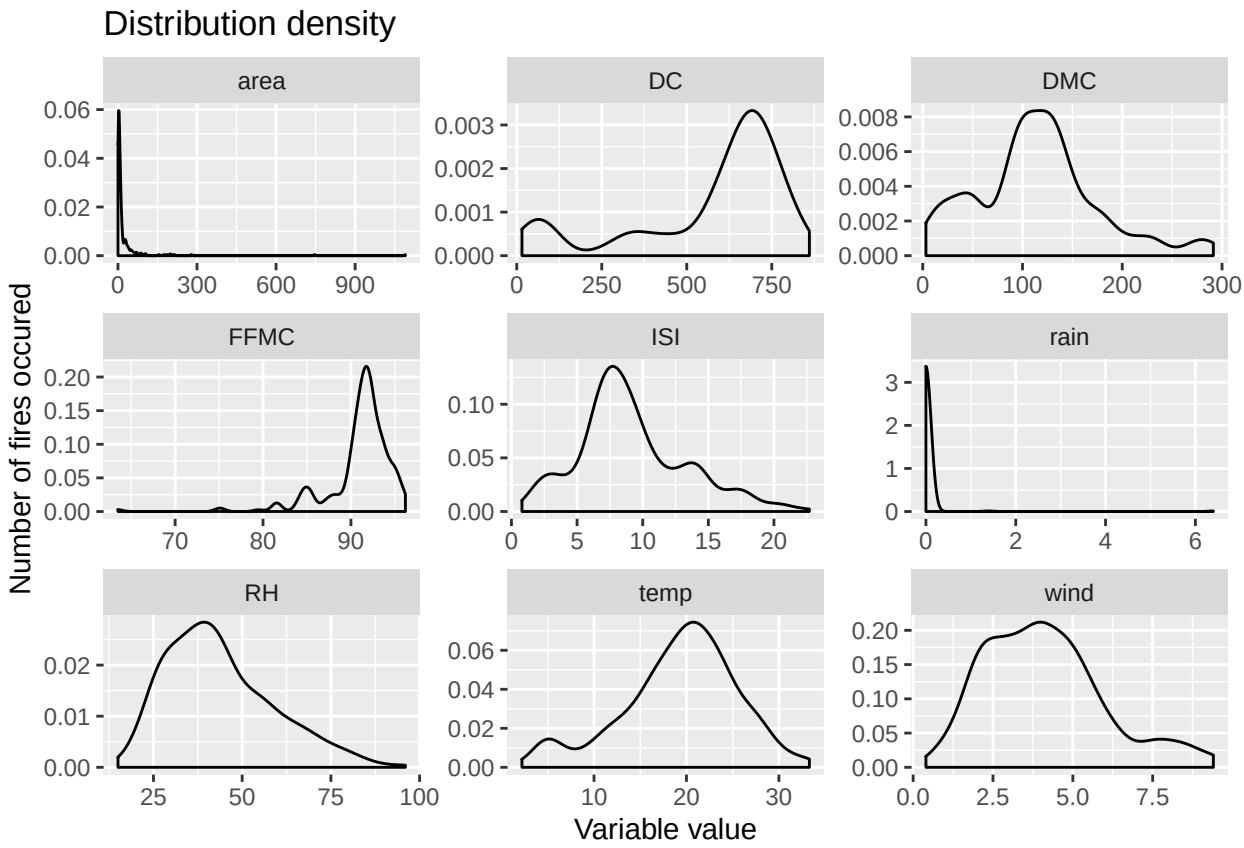
Интересны и данные по дням недели. В целом видно, что в выходные и их окрестности, в частности, в воскресенье, число пожаров больше, и, учитывая, что погода не зависит от дня недели, уже можно говорить о том, что некоторое влияние выходные всё же оказывают, так что в модель всё же стоит включить.

```
#Гистограммы
g <- data %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
g <- g+labs(title = "Distribution histograms", x="Variable value", y="Number of fires occured")
g
```




```
#Плотности
g1 <- data %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_density()

g1 <- g1+labs(title = "Distribution density", x="Variable value", y="Number of fires occured")
g1
```

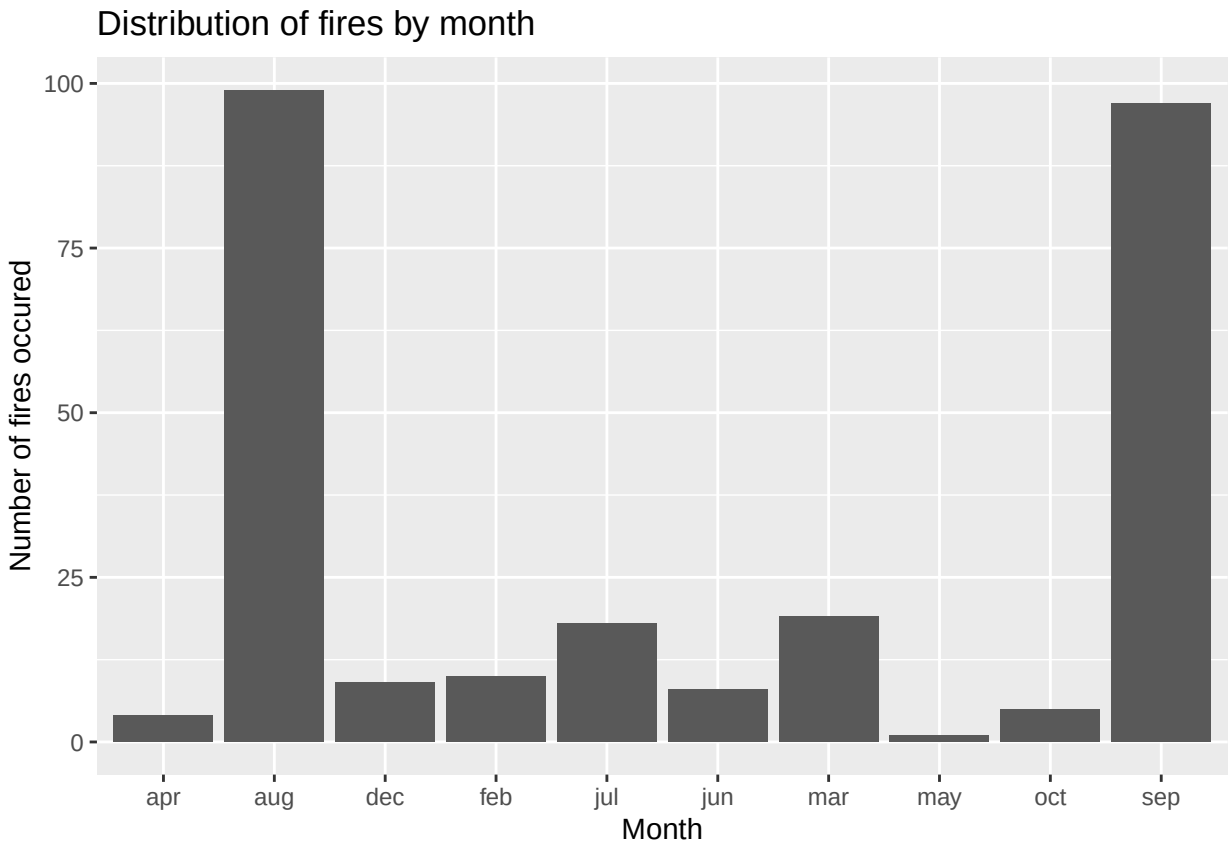



```
#Тест на нормальность
lapply(data[,-c(1,2)], shapiro.test)
```

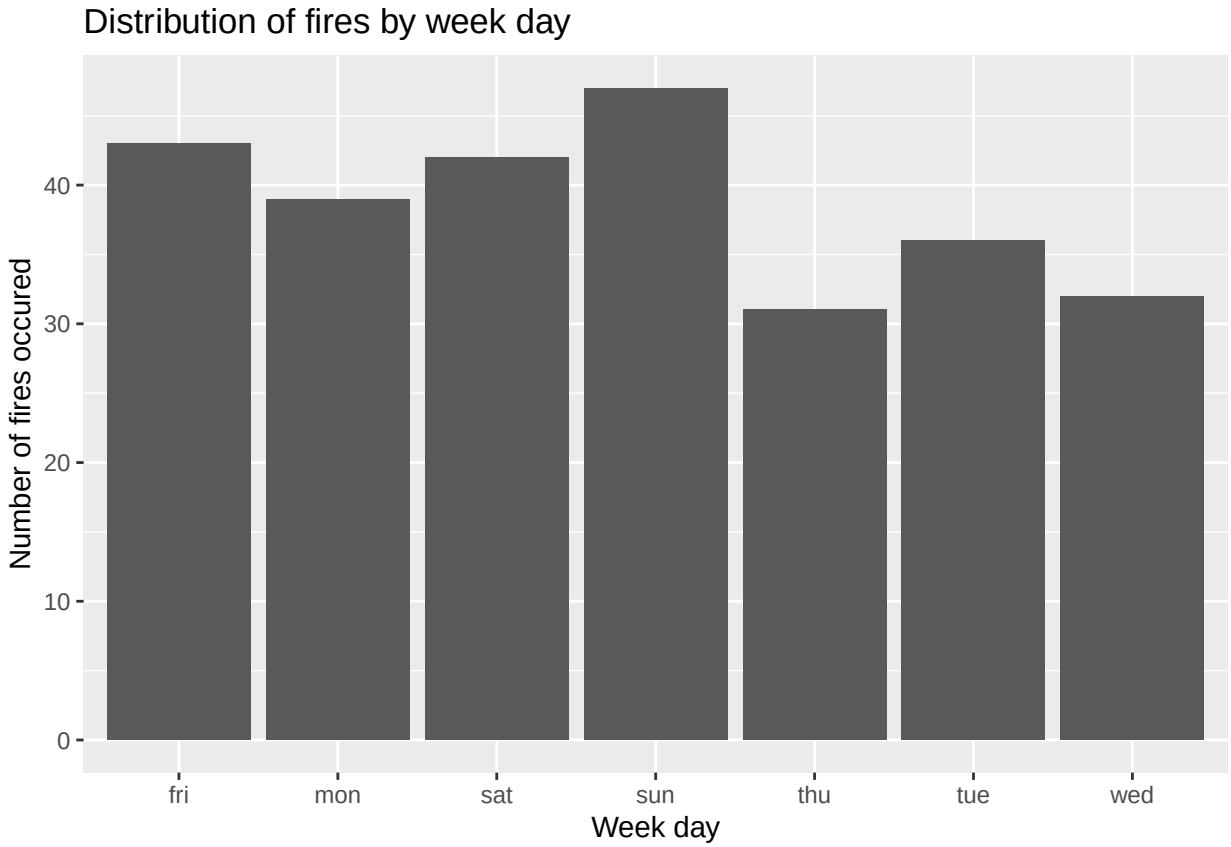
```
## $FFMC
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.78548, p-value < 2.2e-16
##
##
## $DMC
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.96257, p-value = 1.82e-06
##
##
## $DC
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.8118, p-value < 2.2e-16
##
##
```

```
## $ISI
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.95786, p-value = 4.6e-07
##
##
## $temp
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.97212, p-value = 4.02e-05
##
##
## $RH
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.95336, p-value = 1.339e-07
##
##
## $wind
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.95486, p-value = 2.003e-07
##
##
## $rain
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.046407, p-value < 2.2e-16
##
##
## $area
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.23831, p-value < 2.2e-16
```

```r
#Пожары по месяцам
ggplot(data, aes(x = month)) +
  geom_bar() +
  labs(title = "Distribution of fires by month",y="Number of fires occured", x = "Month")
```




```
#Пожары по дням
ggplot(data, aes(x = day)) +
  geom_bar() +
  labs(title = "Distribution of fires by week day",y="Number of fires occured", x = "Week day")
```



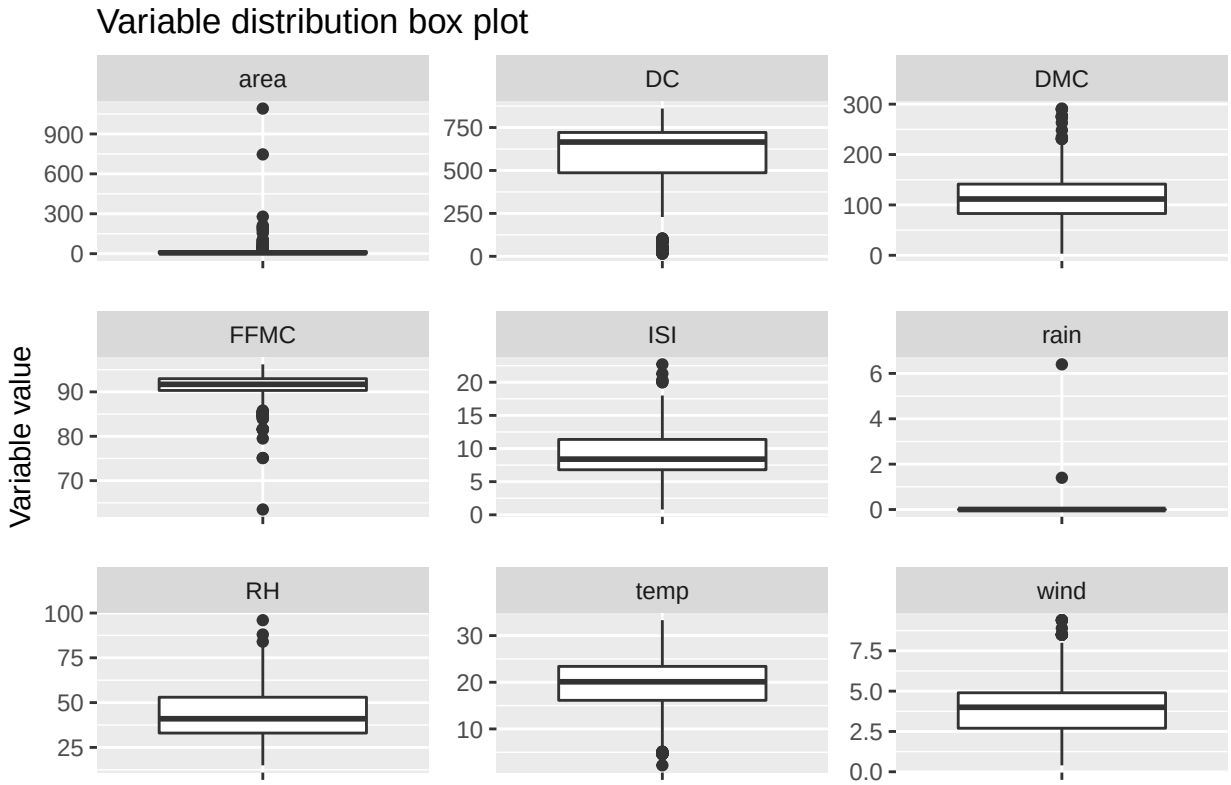

```
#дополнительные графики для прикидки взаимосвязей
#qplot(data=data, temp, area)
#qplot(data=data, day, area)
#qplot(data=data, month, area)
#qplot(data=data, RH, area)
```

Далее смотрим на коробчатые диаграммы, чтобы оценить выбросы и избавиться от них. Самые значительные выбросы у нас наблюдаются по площади пожаров(area), числу осадков(rain) и по значениям индексов,которые будучи агрегированными, тяжело интерпретируются в терминах выбросов, так что на основе их анализа ничего не удаляем.

Если посчитать, то у нас 2 наблюдения с экстремальными по площади пожарами (>600га выжженной земли). Подобные выбросы (при медиане в 6га) сильно исказят наши результаты в случае использования МНК, потому удаляем их. В случае с площадью можно ещё сильнее порезать выборку, например, по границе в 90га (тогда будет удалено ещё 10 наблюдений), однако это будут уже не такие серьёзные выбросы, так что это решение будет принято по ходу подгонки моделей.

По rain для большей однородности данных тоже убераем 2 наблюдения, где rain!=0.

```
#boxplot
g <- data %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(x="",y=value)) +
    facet_wrap(~ key, scales = "free") +
    geom_boxplot()
g <- g+labs(title = "Variable distribution box plot", x="", y="Variable value")
g
```




```
#Считаем выбросы и чистим от них выборку/
summary(data)
```

```
##      month      day         FFMC             DMC              DC
## aug    :99   fri:43   Min.   :63.50   Min.   :  3.2   Min.   : 15.3
## sep    :97   mon:39   1st Qu.:90.33   1st Qu.: 82.9   1st Qu.:486.5
## mar    :19   sat:42   Median :91.70   Median :111.7   Median :665.6
## jul    :18   sun:47   Mean   :91.03   Mean   :114.7   Mean   :570.9
## feb    :10   thu:31   3rd Qu.:92.97   3rd Qu.:141.3   3rd Qu.:721.3
## dec    : 9   tue:36   Max.   :96.20   Max.   :291.3   Max.   :860.6
## (Other):18   wed:32
##       ISI              temp             RH              wind
## Min.   : 0.800   Min.   : 2.20   Min.   :15.00   Min.   :0.400
## 1st Qu.: 6.800   1st Qu.:16.12   1st Qu.:33.00   1st Qu.:2.700
## Median : 8.400   Median :20.10   Median :41.00   Median :4.000
## Mean   : 9.177   Mean   :19.31   Mean   :43.73   Mean   :4.113
## 3rd Qu.:11.375   3rd Qu.:23.40   3rd Qu.:53.00   3rd Qu.:4.900
## Max.   :22.700   Max.   :33.30   Max.   :96.00   Max.   :9.400
##
##       rain              area
## Min.   :0.00000   Min.   :   0.09
## 1st Qu.:0.00000   1st Qu.:   2.14
## Median :0.00000   Median :   6.37
## Mean   :0.02889   Mean   :  24.60
## 3rd Qu.:0.00000   3rd Qu.:  15.42
## Max.   :6.40000   Max.   :1090.84
##
```

```
count(data[which(data$rain>0),])
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     2
```

```
count(data[which(data$area>90),])
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    12
```

```
data <- data[which(data$area<300& data$rain==0),]
```

№3 Для того, чтобы посмотреть на мультиколлинеарность, построим несколько версий линейных моделей и оценим для них VIF и CN. Спецификации пояснены в коде. Логика такого разбиения моделей продиктована порядком выбора переменных: от чистых к агрегированным и далее ещё дополнительный контроль.

Для начала сравниваются результаты по двум выборкам (data1, где выбросы area>90 отрезаны, и data, где выбросы>300 отрезаны), по итогу коэффициенты и значения статистик всё же достаточно сильно меняются, что может как раз говорить о мультиколлинеарности. Но R^2 больше у модели на выборке с большим числом наблюдений(data), поэтому её и оставим.

Для проверки на мультиколлинеарность считаются VIF и CN по каждому набору переменных в модели. По итогу анализа выясняем, что даже в наборе из трёх переменных (RH, temp, wind) матрица уже имеет достаточно маленькие собственные значения (CN=251) и даже, учитывая, что VIF <10, уже можно сказать, что мультиколлинеарность есть. При добавлении индексов CN(16502) увеличивается до огромного значения, так как мы знаем, что индексы зависят от трёх исходных параметров. Ну и с добавлением контроля на дни и месяцы уже даже VIF сигнализируют, что мультиколлинеарность есть.

Таким образом в любой спецификации мы можем говорить о наличии мультиколлинеарности, а значит надо модифицировать функцию потерь, применив модель LASSO, и разобраться по-ковбойски. Прогнав LASSO регрессии по всем 4 спецификациям, получаем интересную картину: в случае с моделью 1 (RH, temp, wind) и моделью 4 (все регрессоры) все коэффициенты 0, то есть просто константа лучше описывает площадь пожара. При спецификации же с днями недели и коэффициентами пожароопасности наконец возникают ненулевые коэффициенты, которые не отличаются знаками от того, что даёт lm. При анализе этих ненулевых коэффициентов и сравнении квадратов остатков делаем вывод, что добавление дней недели не сильно улучшает объясняющие свойства модели, да и содержательно это очень плохие proxi антропогенной причины пожара, так что в итоге отказываемся и от включения дней.

Итогом анализа становится вывод, что надо оценивать модель с переменными ISI, DMC, temp, RH. У этих переменных ненулевые коэффициенты. Вот и оказалось, что DMC и ISI нам всё же важны. Переменная ветра всё же сильно увеличивает мультиколлинеарность, видимо из-за зашумлённости и нелинейного воздействия (форму этого воздействия из логики сложно выявить, а если подгонять полиномами - не ясно, где остановиться), потому добавлять в финальную регрессию не будем, однако стоит заметить, что в тех регрессиях, что были сделаны для пробы, знак при переменной соответствовал ожиданиям(+), хотя в условиях незначимости коэффициента трактовка лишена смысла.

```
set.seed(14)
#Сравниваем модели
data <- data[,-c(10)] #убираем бесполезный rain

#Базовая на всю природу
model1 <- lm(area~RH +temp+wind, data)
```

```r
#Добавляем коэффициенты распространения пожара
model2 <- lm(area~. -month -day, data)
#Дополнительно переменные дня недели и месяца
model3 <- lm(area~. -month, data)
model4 <- lm(area~.,data)
#Итог
stargazer(model1, model2, model3, model4, title="Сравнение объясняющей мощности", type="text",
                                  column.labels=c("Модель 1", "Модель 2","Модель 3", "Модель 4"),
                                  df=FALSE, digits=3)
```

```
##
## Сравнение объясняющей мощности
## ================================================================
##                              Dependent variable:
##                    ------------------------------------
##                                      area
##                    Модель 1 Модель 2 Модель 3 Модель 4
##                      (1)       (2)       (3)       (4)
## ----------------------------------------------------------------
## monthaug                                          60.016*
##                                                  (32.465)
##
## monthdec                                          30.638
##                                                  (27.317)
##
## monthfeb                                          -12.268
##                                                  (21.785)
##
## monthjul                                          44.548
##                                                  (27.887)
##
## monthjun                                          18.563
##                                                  (26.258)
##
## monthmar                                          -11.095
##                                                  (20.371)
##
## monthmay                                          20.137
##                                                  (41.356)
##
## monthoct                                          86.163**
##                                                  (39.654)
##
## monthsep                                          84.558**
##                                                  (36.486)
##
## daymon                              5.014      6.201
##                                    (8.347)    (8.619)
##
## daysat                             12.420    13.812*
##                                    (8.084)    (8.212)
##
## daysun                              6.625      8.572
##                                    (7.962)    (8.114)
```

```
##
## daythu                          -4.833    -3.153
##                                (8.841)   (9.052)
##
## daytue                          12.287    11.772
##                                (8.376)   (8.454)
##
## daywed                           6.676     8.652
##                                (8.670)   (8.868)
##
## FFMC                    0.564    0.773     0.458
##                        (0.974)  (0.989)   (1.036)
##
## DMC                     0.063    0.056    0.178***
##                        (0.055)  (0.055)   (0.067)
##
## DC                     -0.011   -0.008   -0.153***
##                        (0.013)  (0.014)   (0.049)
##
## ISI                    -1.512*  -1.619*    -1.400
##                        (0.816)  (0.844)   (0.894)
##
## RH                  -0.032   -0.038   -0.075   -0.101
##                    (0.172)  (0.191)  (0.199)  (0.240)
##
## temp                0.343    0.571    0.395    0.328
##                    (0.439)  (0.592)  (0.614)  (0.833)
##
## wind                0.087    0.758    0.838    1.219
##                    (1.248)  (1.334)  (1.354)  (1.407)
##
## Constant           12.469  -32.982  -52.648  -14.653
##                   (15.882) (86.568) (88.308) (94.285)
##
## ------------------------------------------------------
## Observations          266      266      266      266
## R2                  0.004    0.025    0.048    0.093
## Adjusted R2        -0.007   -0.001   -0.001    0.011
## Residual Std. Error 36.099   35.997   35.991   35.778
## F Statistic         0.374    0.945    0.977    1.130
## ======================================================
## Note:                      *p<0.1; **p<0.05; ***p<0.01
```

```
data1 <- data[which((data$area)<90),]
#Базовая на всю природу
model11 <- lm(area~RH +temp+wind, data1)
#Добавляем коэффициенты распространения пожара
model21 <- lm(area~. -month -day, data1)
#Дополнительно переменные дня недели и месяца
model31 <- lm(area~. -month, data1)
model41 <- lm(area~.,data1)
#Итог
stargazer(model1, model11, model2, model21, model3, model31, model4, model41, title="Сравнение объясняющей мощно
                                column.labels=c("Модель 1", "Модель 1", "Модель 2","Модель 2","Модель 3", "Модель 3
```

```
                                df=FALSE, digits=3)
```

```
##
## Сравнение объясняющей мощности
## ================================================================================================
##                                          Dependent variable:
##                    ---------------------------------------------------------------------
##                                                  area
##                    Модель 1 Модель 1 Модель 2 Модель 2 Модель 3 Модель 3 Модель 4  Модель 4
##                      (1)       (2)       (3)       (4)       (5)       (6)       (7)       (8)
## ------------------------------------------------------------------------------------------------
## monthaug                                                                          60.016*    8.412
##                                                                                  (32.465)  (15.215)
##
## monthdec                                                                          30.638     10.153
##                                                                                  (27.317)  (12.703)
##
## monthfeb                                                                          -12.268    -7.811
##                                                                                  (21.785)  (10.097)
##
## monthjul                                                                          44.548     0.497
##                                                                                  (27.887)  (13.058)
##
## monthjun                                                                          18.563     -2.860
##                                                                                  (26.258)  (12.230)
##
## monthmar                                                                          -11.095    -9.726
##                                                                                  (20.371)  (9.441)
##
## monthmay                                                                          20.137     16.676
##                                                                                  (41.356)  (19.162)
##
## monthoct                                                                          86.163**   31.553*
##                                                                                  (39.654)  (18.546)
##
## monthsep                                                                          84.558**   20.747
##                                                                                  (36.486)  (17.163)
##
## daymon                                                  5.014     -0.094    6.201     -0.163
##                                                       (8.347)   (3.917)   (8.619)   (4.029)
##
## daysat                                                  12.420    2.597     13.812*   3.724
##                                                       (8.084)   (3.835)   (8.212)   (3.878)
##
## daysun                                                  6.625     3.440     8.572     4.404
##                                                       (7.962)   (3.754)   (8.114)   (3.807)
##
## daythu                                                  -4.833    -2.973    -3.153    -1.637
##                                                       (8.841)   (4.119)   (9.052)   (4.195)
##
## daytue                                                  12.287    1.260     11.772    2.304
##                                                       (8.376)   (3.995)   (8.454)   (4.013)
##
## daywed                                                  6.676     2.051     8.652     3.317
```

```
##                                          (8.670)  (4.073)   (8.868)  (4.137)
##
## FFMC                                      0.564    0.299     0.773    0.331     0.458    0.254
##                                         (0.974)  (0.454)   (0.989)  (0.463)   (1.036)  (0.483)
##
## DMC                                       0.063    0.016     0.056    0.014   0.178***  0.073**
##                                         (0.055)  (0.026)   (0.055)  (0.026)   (0.067)  (0.032)
##
## DC                                       -0.011   -0.007    -0.008   -0.006  -0.153*** -0.058**
##                                         (0.013)  (0.006)   (0.014)  (0.006)   (0.049)  (0.023)
##
## ISI                                      -1.512*   -0.434  -1.619*   -0.423   -1.400    -0.220
##                                         (0.816)  (0.381)   (0.844)  (0.398)   (0.894)  (0.419)
##
## RH                    -0.032   -0.034   -0.038   -0.016   -0.075   -0.051   -0.101    0.018
##                     (0.172)  (0.080)   (0.191)  (0.090)  (0.199)  (0.095)   (0.240)  (0.114)
##
## temp                  0.343    0.098    0.571    0.232    0.395    0.120     0.328    0.457
##                     (0.439)  (0.204)   (0.592)  (0.282)  (0.614)  (0.295)   (0.833)  (0.397)
##
## wind                  0.087    0.867    0.758    1.017    0.838    0.946     1.219    0.946
##                     (1.248)  (0.582)   (1.334)  (0.625)  (1.354)  (0.638)   (1.407)  (0.659)
##
## Constant             12.469   8.068   -32.982  -16.703  -52.648  -17.183   -14.653  -10.000
##                     (15.882) (7.424)   (86.568) (40.301) (88.308) (41.336) (94.285)  (43.933)
##
## ---------------------------------------------------------------------------------------
## Observations          266      256      266      256      266      256       266      256
## R2                  0.004    0.010    0.025    0.020    0.048    0.033     0.093    0.090
## Adjusted R2          -0.007   -0.002   -0.001   -0.007   -0.001   -0.019    0.011    0.004
## Residual Std. Error  36.099   16.616   35.997   16.662   35.991   16.758   35.778    16.567
## F Statistic          0.374    0.855    0.945    0.739    0.977    0.636     1.130    1.048
## =======================================================================================
## Note:                                                  *p<0.1; **p<0.05; ***p<0.01
```

```
#Мультиколлинеарность

#Для модели 1.
vif(model1)
```

```
##       RH     temp     wind
## 1.350192 1.492115 1.126317
```

```
kappa(model1)
```

```
## [1] 251.3799
```

```
#Для модели 2.
vif(model2)
```

```
##     FFMC      DMC       DC      ISI     temp       RH     wind
## 2.678516 2.315281 1.971582 2.353017 2.733039 1.669886 1.293052
```

```
kappa(model2)
```

```
## [1] 16502.51
```

```r
#Для модели 3.
vif(model3)
```

```
##           GVIF Df GVIF^(1/(2*Df))
## day   1.372087  6        1.026712
## FFMC 2.764755  1        1.662755
## DMC   2.330292  1        1.526529
## DC    2.022279  1        1.422068
## ISI   2.514336  1        1.585666
## temp 2.938064  1        1.714078
## RH    1.813841  1        1.346789
## wind 1.333207  1        1.154646
```

```r
kappa(model3)
```

```
## [1] 15743.65
```

```r
#Для модели 4.
vif(model4)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## month 131.427164  9        1.311308
## day     1.738892  6        1.047183
## FFMC    3.069871  1        1.752105
## DMC     3.571631  1        1.889876
## DC     26.864181  1        5.183067
## ISI     2.858577  1        1.690733
## temp    5.476720  1        2.340239
## RH      2.678861  1        1.636723
## wind    1.457025  1        1.207073
```

```r
kappa(model4)
```

```
## [1] 41618.55
```

```r
#LASSO подбираем.

#train.index <- createDataPartition(y = data$area, p = 0.75, list = FALSE)
#data.train <- data[train.index, ]
#data.test <- data[-train.index, ]

#Для модели 1.
x1=as.matrix(data[,7:9]) # матрица регрессоров
lasso1 <- cv.glmnet(x = x1, y=data$area, type.measure = "mse",family = 'gaussian', alpha=1) #оцениваем кросс-валид
lambda1 <- lasso1$lambda.min #запоминаем её
coef1 <- lasso1$glmnet.fit$beta[,lasso1$glmnet.fit$lambda==lambda1] #записываем коэффициенты (без константы)
coef1
```

```
## temp   RH wind
##    0    0    0
```

```r
#Сравниваем точность предсказания lasso и соответствующей обычной модели
mean((data$area-predict(lasso1,s=lambda1,newx=x1))^2) #средний квадрат отклонений (lasso)
```

```
## [1] 1289.036
```

```r
mean((model1$residuals)^2) #средний квадрат отклонений (простая lm)
```

```
## [1] 1283.546
```

```
#Для модели 2.
x2=as.matrix(data[,3:9]) # матрица регрессоров
lasso2 <- cv.glmnet(x = x2, y=data$area, type.measure = "mse",family = 'gaussian', alpha=1) #оцениваем кросс-валид
lambda2 <- lasso2$lambda.min #запоминаем её
coef2 <- lasso2$glmnet.fit$beta[,lasso2$glmnet.fit$lambda==lambda2] #записываем коэффициенты (без константы)
coef2
```

```
##         FFMC          DMC           DC          ISI         temp           RH
##  0.000000000  0.027897865  0.000000000 -0.623159512  0.246293804 -0.008512444
##         wind
##  0.000000000
```

```
#Сравниваем точность предсказания lasso и соответствующей обычной модели
mean((data$area-predict(lasso2,s=lambda2,newx=x2))^2) #средний квадрат отклонений (lasso)
```

```
## [1] 1268.293
```

```
mean((model2$residuals)^2) #средний квадрат отклонений (простая lm)
```

```
## [1] 1256.823
```

```
#Для модели 3.
x3=as.matrix(dummy_cols(data[,2:9])[,-c(1,11)]) # матрица регрессоров (исключили одну бинарную дня, чтобы избежа
lasso3 <- cv.glmnet(x = x3, y=data$area, type.measure = "mse",family = 'gaussian', alpha=1) #оцениваем кросс-валид
lambda3 <- lasso3$lambda.min #запоминаем её
coef3 <- lasso3$glmnet.fit$beta[,lasso3$glmnet.fit$lambda==lambda3] #записываем коэффициенты (без константы)
coef3
```

```
##         FFMC          DMC           DC          ISI         temp           RH
##  0.000000000  0.021972016  0.000000000 -0.369447447  0.095429279 -0.001576942
##         wind      day_fri      day_mon      day_sun      day_thu      day_tue
##  0.000000000 -5.222286083  0.000000000  0.000000000 -7.100831879  0.000000000
##      day_wed
##  0.000000000
```

```
#Сравниваем точность предсказания lasso и соответствующей обычной модели
mean((data$area-predict(lasso3,s=lambda3,newx=x3))^2) #средний квадрат отклонений (lasso)
```

```
## [1] 1255.136
```

```
mean((model3$residuals)^2) #средний квадрат отклонений (простая lm)
```

```
## [1] 1227.204
```

```
#Для модели 4.
x4=as.matrix(dummy_cols(data[,1:9])[,-c(1:2,9,23)]) # матрица регрессоров (исключили по одной бинарной переменной
lasso4 <- cv.glmnet(x = x4, y=data$area, type.measure = "mse",family = 'gaussian', alpha=1) #оцениваем кросс-валид
lambda4 <- lasso4$lambda.min #запоминаем её
coef4 <- lasso4$glmnet.fit$beta[,lasso4$glmnet.fit$lambda==lambda4] #записываем коэффициенты (без константы)
coef4
```

```
##      FFMC       DMC        DC       ISI      temp        RH month_apr month_aug
##         0         0         0         0         0         0         0         0
## month_dec month_feb month_jan month_jul month_jun month_mar month_may month_nov
##         0         0         0         0         0         0         0         0
## month_oct month_sep   day_fri   day_sat   day_sun   day_thu   day_tue   day_wed
##         0         0         0         0         0         0         0         0
```

```
#Сравниваем точность предсказания lasso и соответствующей обычной модели
mean((data$area-predict(lasso4,s=lambda4,newx=x4))^2) #средний квадрат отклонений (lasso)
```

```
## [1] 1289.036
```

```
mean((model4$residuals)^2) #средний квадрат отклонений (простая lm)
```

```
## [1] 1169.356
```

№4 Оцениваем линейную модель. После анализа мультиколлинеарности осталось 4 регрессора (ISI, DMC, temp, RH). Ещё раз удостоверимся, что при таком наборе мультиколлинеарности почти нет. CN, который стал достаточно малым (33.5). VIF также говорят, что можно считать, что нет значительной мультиколлинеарности. И из-за этого, к слову, особой разницы в коэффициентах между LASSO и lm нет.

Интерпретация коэффициентов(значим коэффициент только при ISI(p-value=0.07,т.е. на 10% уровне значимости считаем, что не равен 0), для остальных особого смысла в интерпретации нет, но возможно с другими se будет другая значимость, потому на всякий и это проинтерпретируем) :

DMC=0.049: как и ожидалось, с ростом индекса влажности потенциально возгораемых слоёв увеличивается площадь сгораемой территории, потому что топливо проще разгорается, когда сухое. Если точно, то при увеличении индекса на 1 площадь увеличивается на 0.049 га при прочих равных.

ISI=-1.112: интерпретация - с ростом индекса распространения пожара на 1 площадь пожара уменьшается на 1.1 га при прочих равных. Влияние индекса оказалось контринтуитивным и не совпадает с ожиданиями, однако оно значимое(на 10% уровне). Надо попытаться объяснить:

1) с поведенческой точки зрения, возможно более низкие при прочих равных значений индекса кажутся пожарным не такими страшными, и потому на борьбу с пожарами бросается недостаточное количество людей, а потому захватывается пожаром большая площадь, чем даже при высоком значении индекса, на которое среагировали должным образом. Для этой гипотезы есть у меня даже косвенное доказательство. Согласно сайту (https://www.malagaweather.com/fwi-txt.htm) значения 10-16 считаются для этого индекса высокими, 16+ - экстремальным. И, если посмотреть на график зависимости только area от ISI (вполне законно, т.к. остальные переменные всё равно не значимы), то видно, что самые большие площади были сожжены при значениях ISI ДО 10, но близких к 10. То есть возможно действительно имела место недооценка возможного распространения. Однако, чтобы точно это замерить, нужно смотреть количество пожарных и прочее..
2) Если смотреть с точки зрения методологии, то индекс считается на основе данных индекса FFMC и измерений скорости ветра. Лаг его расчёта в районе дня, т.е. в нём информация за день. В таком случае проблема может быть в том, что для определённых дней погода сильно менялась и занижала изначально высокое значение индекса. И вообще для такого индекса, имеющего лаг в измерении, лучше брать значение за предыдущий день. Тогда возможно зависимость была бы всё же положительной.

temp=0.384: с ростом температуры на градус цельсия площадь увеличивается на 0.38 га при прочих равных. Вполне соответствует интуиции и ожиданиям о знаке(+).

RH=-0.073: с ростом относительной влажности воздуха на 1% площадь выгоревшая уменьшается на 0.073 га при прочих равных. Тоже соответствует ожиланиям и логике.

Тест на общую незначимость коэффициентов модели даёт F-статистику 1.355 с p-value 0.25, т.е. даже на 10% уровне значимости мы принимаем гипотезу о том, что все коэффициенты равны нулю и модель незначима. Как уже отмечалось, действительно прогнозирование просто константой даёт не сильно большие значения среднего квадратов остатков, но таковы уж данные.

Тест Шапиро-Уилка отвергает гипотезу о нормальности остатков(p-value=10^-16, т.е. на любом вменяемом уровне значимости), поэтому скорее всего нарушаются предположения теоремы Гаусса-Маркова о нормальности случайных ошибок, а потому статистики и доверительные интервалы не корректно считать так, как мы это сделали.

```r
set.seed(14)
#Проверим ещё раз на мультиколлинеарность итоговую модель:
x5=as.matrix(data[,c(4,6:8)]) #матрица регрессоров (ISI,DMC,RH,temp)
kappa(x5)
```

```
## [1] 33.50935
```

```r
#Сама модель
fit <- lm(area~DMC+ISI+temp+RH, data)
summary(fit)
```

```
##
## Call:
## lm(formula = area ~ DMC + ISI + temp + RH, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.577 -16.204 -10.208  -0.962 257.625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.37985   14.10747   1.303   0.1938
## DMC          0.04941    0.04443   1.112   0.2672
## ISI         -1.11199    0.60419  -1.840   0.0668 .
## temp         0.38427    0.54489   0.705   0.4813
## RH          -0.07327    0.18413  -0.398   0.6910
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.87 on 261 degrees of freedom
## Multiple R-squared:  0.02035,    Adjusted R-squared:  0.005336
## F-statistic: 1.355 on 4 and 261 DF,  p-value: 0.2499
```

```r
vif(fit)
```

```
##      DMC      ISI     temp       RH
## 1.546286 1.297239 2.330550 1.567346
```

```r
#LASSO для сравнения
lasso5 <- cv.glmnet(x = x5, y=data$area, type.measure = "mse",family = 'gaussian', alpha=1) #оцениваем кросс-валид
lambda5 <- lasso5$lambda.min #запоминаем её
coef5 <- lasso5$glmnet.fit$beta[,lasso5$glmnet.fit$lambda==lambda5] #записываем коэффициенты (без константы)
coef5
```

```
##         DMC         ISI        temp          RH
##  0.04906792 -1.10457997  0.38227756 -0.07226251
```

```r
mean((data$area-predict(lasso5,s=lambda5,newx=x5))^2) #средний квадрат отклонений (lasso)
```

```
## [1] 1262.805
```

```r
mean((fit$residuals)^2)
```

```
## [1] 1262.804
```

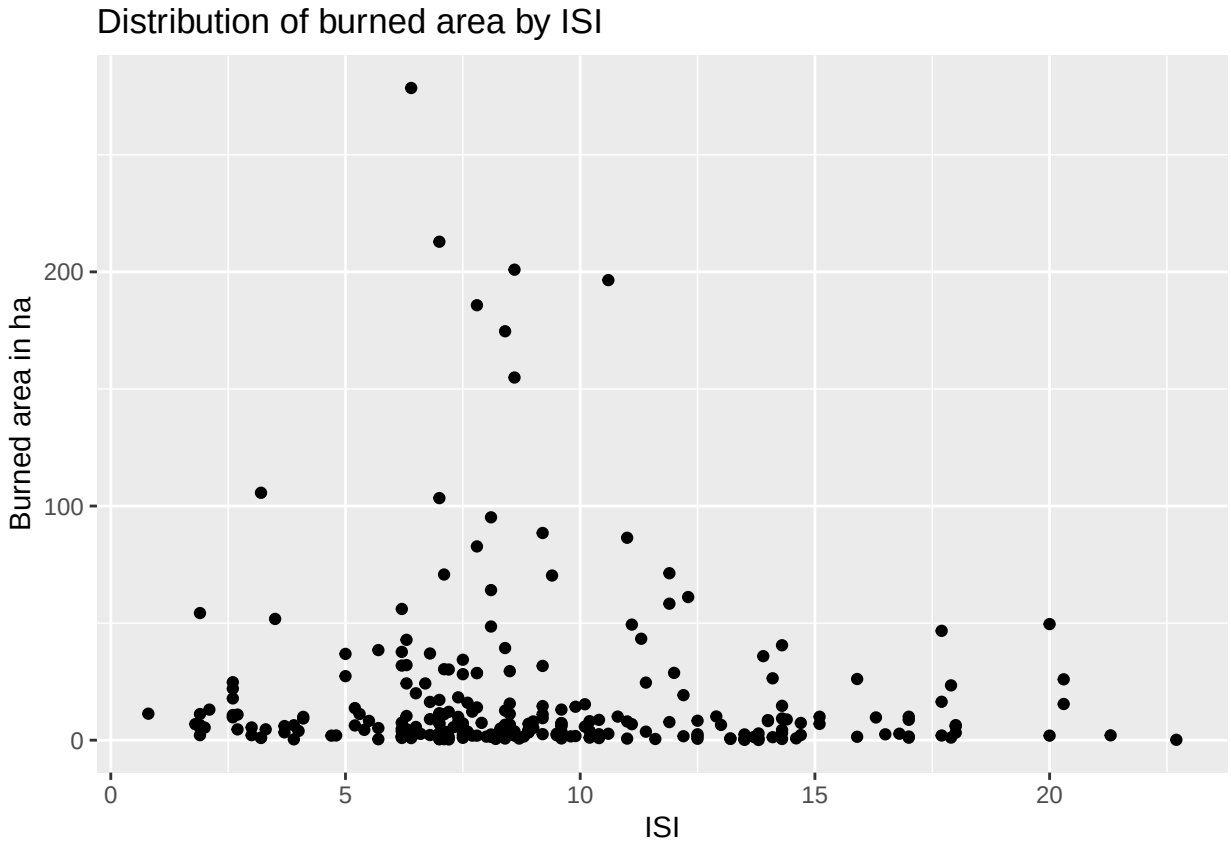
```r
#Для интерпретации

stargazer(fit,title="Зависимость площади пожара от природных факторов", type="text",
```

```
                                column.labels=c("Итоговая модель"),
                                df=FALSE, digits=3)
```

```
##
## Зависимость площади пожара от природных факторов
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                area
##                           Итоговая модель
## ------------------------------------------------
## DMC                          0.049
##                             (0.044)
##
## ISI                          -1.112*
##                             (0.604)
##
## temp                         0.384
##                             (0.545)
##
## RH                           -0.073
##                             (0.184)
##
## Constant                     18.380
##                            (14.107)
##
## ------------------------------------------------
## Observations                   266
## R2                            0.020
## Adjusted R2                    0.005
## Residual Std. Error            35.875
## F Statistic                   1.355
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
#График зависимости переменных для интерпретации
ggplot(data=data, aes(x=ISI, y=area)) +
 geom_point() +
 labs(title="Distribution of burned area by ISI",y="Burned area in ha")
```




```
#Нормальность остатков
shapiro.test(fit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit$residuals
## W = 0.54439, p-value < 2.2e-16
```

N4. Bootstrap. Реализация пакета boot и собственная функция. Дальше с помощью пакета смотрим доверительные интервалы эмпирические. Как видно, 95% интервалы (где мы просто отрезали по 250 наблюдений с каждой стороны), всё также говорят о незначимости всех переменных, кроме ISI, т.к. пересекают 0. Это видно и на самодельных интервалах и на пакетной реализации, где ещё дополнительная коррекция делается. На 10% значимой становится ещё и константа.

```
set.seed(14)
#Пакетная реализация
#Для сбора статистики
bs <- function(formula, data, indices) {
  d <- data[indices,]
  fit <- lm(formula, data=d)
  return(coef(fit))
}
results <- boot(data=data, statistic=bs,
   R=10000, formula=area~ DMC + ISI + temp + RH)

#Самодельная функция для линейной регрессии (на вход данные, вектор переменных, число выборок и формула. Bo
custom_bs <- function(data,variables,n,formula) {
```

```
  beta <- data.frame()
for (i in 1:n) {
  index<- sample(1:nrow(data),nrow(data),replace = TRUE )
  sample <- data[index,variables]
  model <- lm(formula=formula, sample)
  beta <- rbind(beta,data.frame(t(model$coefficients)))
}
  return(beta)
}
custom_results <- custom_bs(data,c(4,6:8,10),10000,area~ DMC + ISI + temp + RH)

#Доверительные интервалы из пакета (95%), скорректированные (BCa)
boot.ci(results, type="bca", index=1) #константа
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 1)
##
## Intervals :
## Level       BCa
## 95%   (-12.40,  40.74 )
## Calculations and Intervals on Original Scale
```

```
boot.ci(results, type="bca", index=2) #DMC
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 2)
##
## Intervals :
## Level       BCa
## 95%   (-0.0334,  0.1471 )
## Calculations and Intervals on Original Scale
```

```
boot.ci(results, type="bca", index=3) #ISI
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 3)
##
## Intervals :
## Level       BCa
## 95%   (-2.390, -0.217 )
## Calculations and Intervals on Original Scale
```

```
boot.ci(results, type="bca", index=4) #temp
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
```

```
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 4)
##
## Intervals :
## Level       BCa
## 95%   (-0.5302,  1.9611 )
## Calculations and Intervals on Original Scale
```

```
boot.ci(results, type="bca", index=5) #RH
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 5)
##
## Intervals :
## Level       BCa
## 95%   (-0.3786,  0.4300 )
## Calculations and Intervals on Original Scale
```

```
#Обычные квантильные доверительные интервалы по данным из самодельной функции n-ый
custom_ci <- function(data,n){  #на вход переменная, по которой нужен интервал и уровень значимости в долях.
  quantile(data,c(n,1-n))
}
n=0.01 #делаем 1% интервал
custom_ci(custom_results$X.Intercept.,n)
```

```
##        1%       99%
## -14.42644  48.87860
```

```
custom_ci(custom_results$DMC,n)
```

```
##          1%         99%
## -0.05057548  0.16624171
```

```
custom_ci(custom_results$ISI,n)
```

```
##          1%         99%
## -2.50410645 -0.03507116
```

```
custom_ci(custom_results$temp,n)
```

```
##        1%       99%
## -0.8878925  1.9577898
```

```
custom_ci(custom_results$RH,n)
```

```
##        1%       99%
## -0.4960258  0.4153234
```

№5. Прогноз.

```
nw <- data.frame(DMC = median(data$DMC),ISI = median(data$ISI), temp= median(data$temp), RH = median(data$R
head(nw)
```

```
##      DMC ISI temp RH
## 1 111.45 8.4 20.1 41
```

```
predict(fit, newdata = nw, interval = 'prediction')
```

```
##       fit       lwr      upr
## 1 19.2652 -51.52281 90.05322
```

```
predict(fit, newdata = nw, interval = 'confidence')
```

```
##       fit      lwr      upr
## 1 19.2652 14.70103 23.82938
```

№6. Гетероскедастичность - это ситуация, когда дисперсия случайных ошибок - не константа. Обычно данные зашумлены, поэтому её можно ожидать почти всегда. Но по какой переменной? Да по каждой, если задуматься:

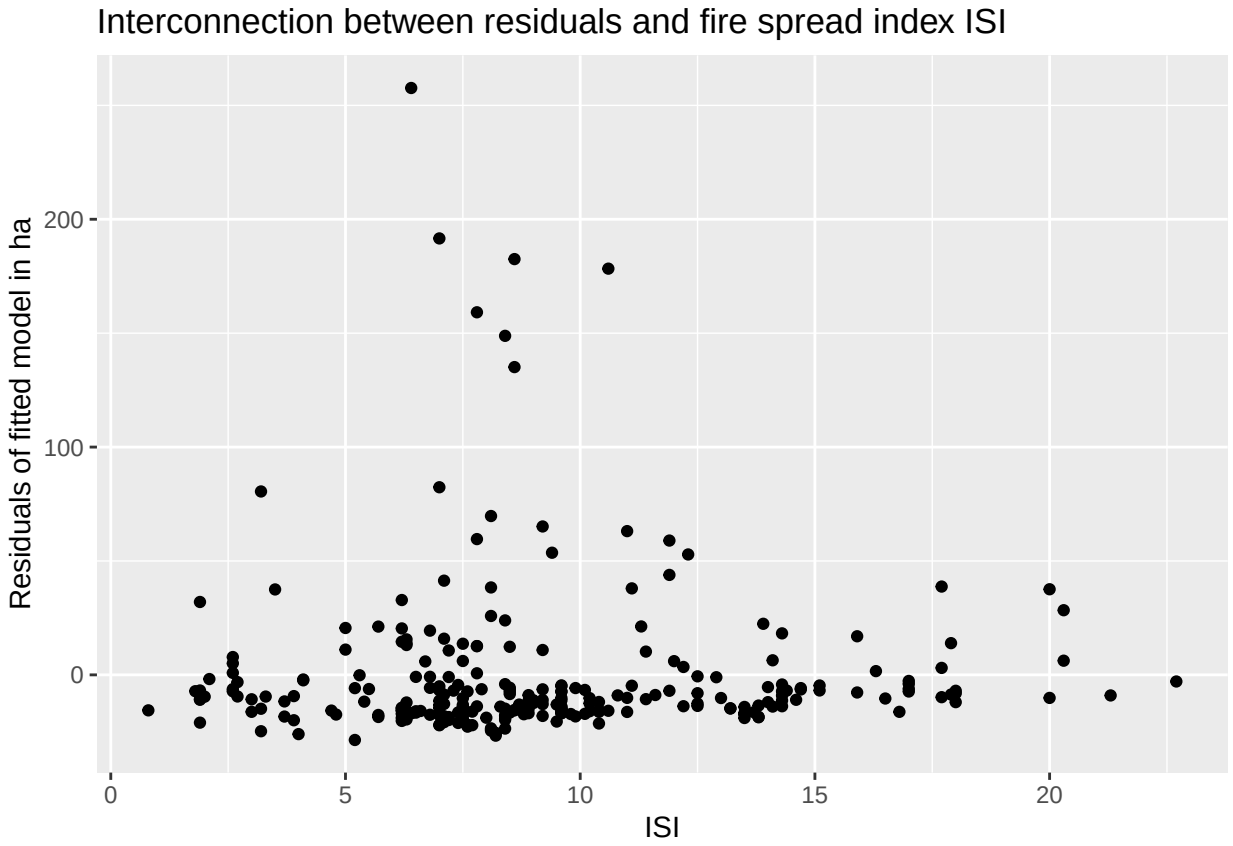
1) temp, т.к. скорее всего на низких температурах у нас пожары в среднем одинаково маленькую площадь затрагивают, небольшой разброс, тогда как при высокой температуре многие другие факторы могут кардинально изменить влияние (тот же ветер), а потому разброс значений площади с ростом температуры будет расти, а следовательно и ошибки тоже.
2) RH, т.к. при высокой влажности воздуха возгорания менее вероятны и наверное будут быстрее затухать, а вот при низкой влажности и высокой особенно температуре и вероятность возгорания увеличивается и скорость распространения (т.к. по сухой, а не влажной от испарения траве проще распространяться огню). Так что тут ожидаем нисходящую дисперсию ошибок.
3) DMC - при низких значениях, как уже уточнялось, этот индекс показывает говорит о том, что горючие материалы влажные, т.е. распространение наверное будет в рамках небольшой области, в то же время большие значения говорят о сухости, т.е. у вас почти хворост, готовый от каждой искры загореться, и тогда уже другие факторы вполне могут вмешиваться и увеличивать волатильности площади пожара. Т.е. ожидаем положительной зависимости разброса случайных ошибок от DMC.
4) ISI- наконец, как уже говорилось ранее, у этого индекса взаимосвязь с area значимая и есть подозрение на эффект от человеческого фактора при значениях в районе 10. То есть опять же для этого индекса распространения можно ожидать в середине (10 находится где-то посередение в доступных данных) всплеск волатильности ошибок.

№7. И тесты, и анализ остатков в общем-то показывает, что гетероскедастичность-таки можно найти по любой переменной. В целом идейные предположения подтвердились: графически для temp и DMC остатки сильнее колеблются при высоких значениях, для RH - при низких, для ISI -посередине.

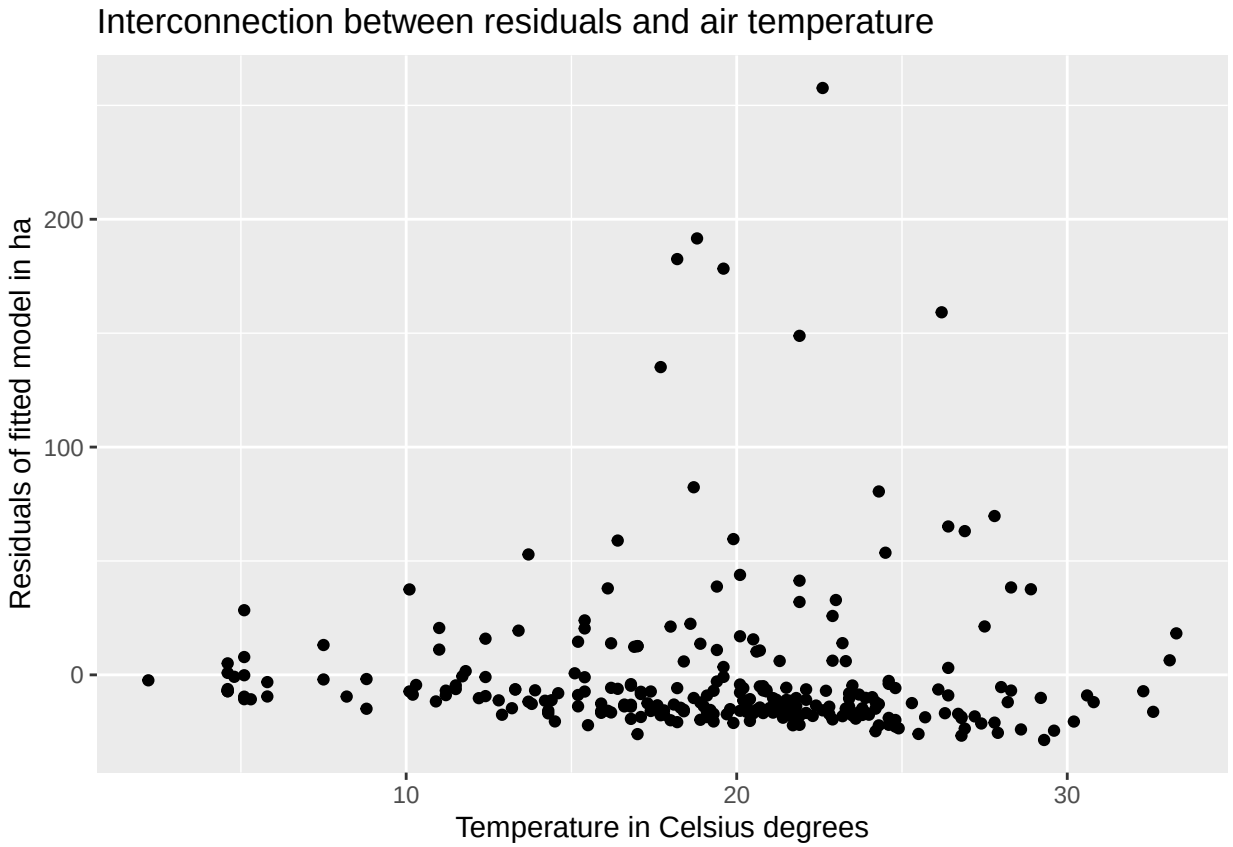
Формальный тест Гольдфельда-Квандта в зависимости от того, какую часть выборки в середине при разбиени удаляем, даёт разные результаты для temp и RH - или есть или нет, но вот для индексов совершенно точно при любом значении изымаемой из центра выборки на 1% уровне значимости отвергается гипотеза для отсутствия гетероскедастичности, так что мы можем утверждать, что она тут есть.

Итог: гетероскедастичность есть в модели.

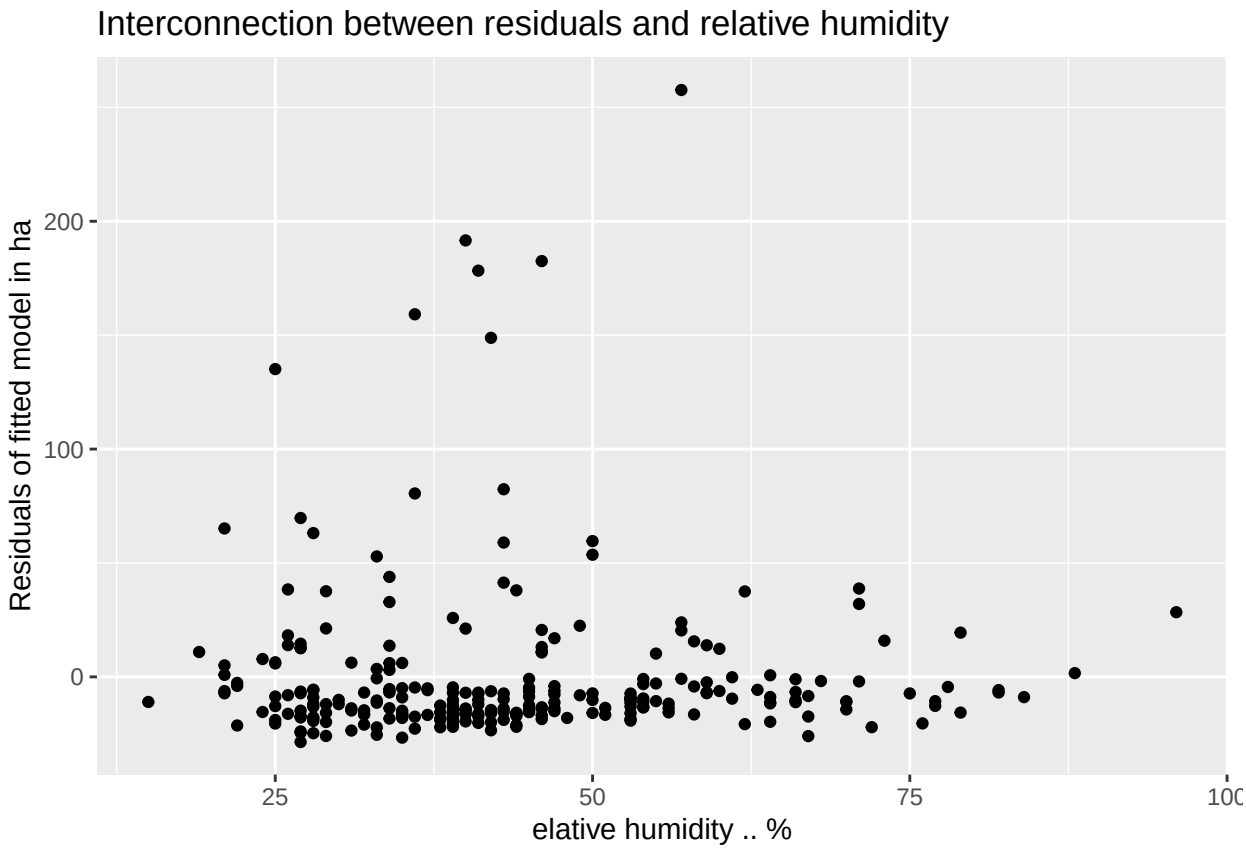
```
#Графики
ggplot(data=data, aes(x=ISI, y=fit$residuals)) +
 geom_point() +
 labs(title="Interconnection between residuals and fire spread index ISI",y="Residuals of fitted model in ha")
```



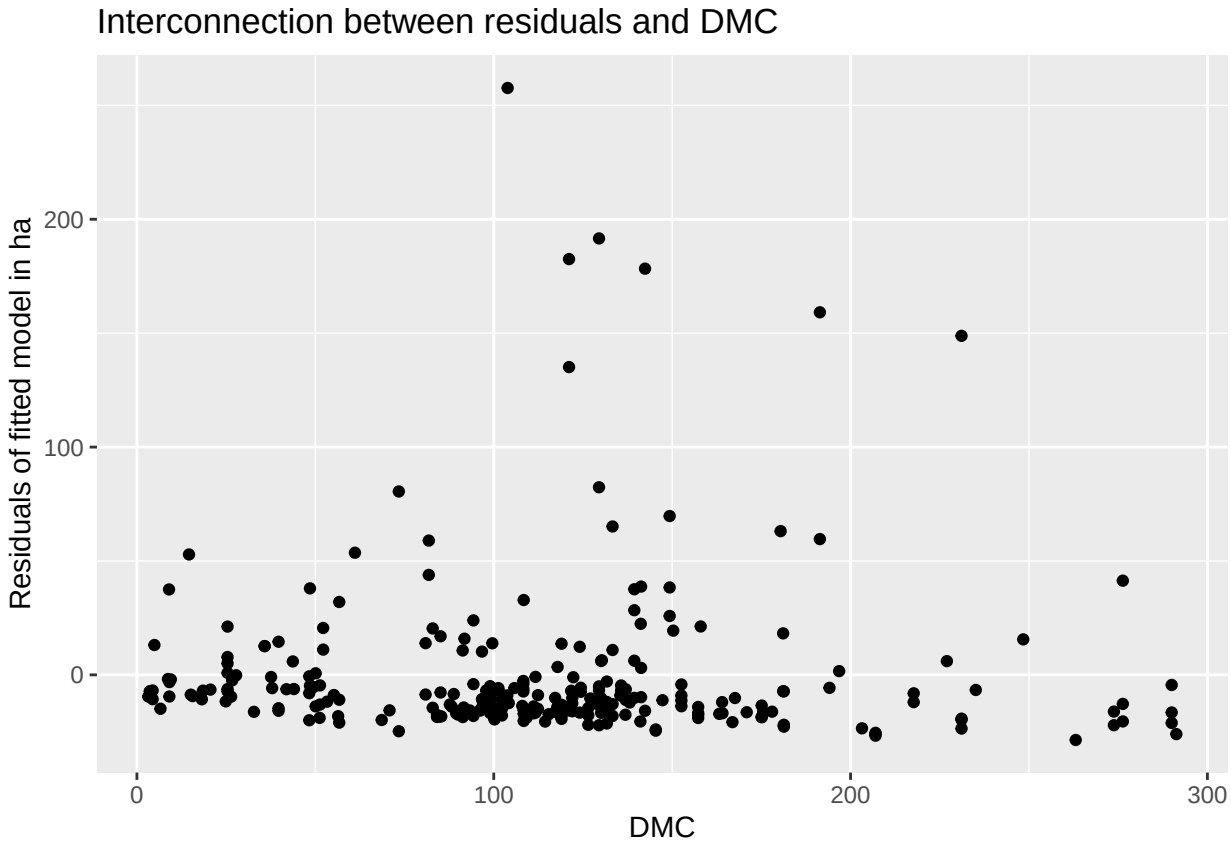

```
ggplot(data=data, aes(x=temp, y=fit$residuals)) +
 geom_point() +
 labs(title="Interconnection between residuals and air temperature",y="Residuals of fitted model in ha", x="Temperature
```




```
ggplot(data=data, aes(x=RH, y=fit$residuals)) +
 geom_point() +
 labs(title="Interconnection between residuals and relative humidity",y="Residuals of fitted model in ha",x="elative humi
```




```
ggplot(data=data, aes(x=DMC, y=fit$residuals)) +
 geom_point() +
 labs(title="Interconnection between residuals and DMC",y="Residuals of fitted model in ha")
```




```
#Гольдфельд-Квандт

m=0.1#Какую часть выборки вырезаем из центра
#Для temp
gqtest(fit, fraction=m, alternative = c("greater"),
  order.by = data$temp)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 1.3662, df1 = 115, df2 = 114, p-value = 0.04828
## alternative hypothesis: variance increases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("less"),
  order.by = data$temp)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 1.3662, df1 = 115, df2 = 114, p-value = 0.9517
## alternative hypothesis: variance decreases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("two.sided"),
  order.by = data$temp)
```

```
##
##  Goldfeld-Quandt test
```

```
##
## data:  fit
## GQ = 1.3662, df1 = 115, df2 = 114, p-value = 0.09656
## alternative hypothesis: variance changes from segment 1 to 2
```

```
#Для RH
gqtest(fit, fraction=m, alternative = c("greater"),
  order.by = data$RH)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 1.4661, df1 = 115, df2 = 114, p-value = 0.0209
## alternative hypothesis: variance increases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("less"),
  order.by = data$RH)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 1.4661, df1 = 115, df2 = 114, p-value = 0.9791
## alternative hypothesis: variance decreases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("two.sided"),
  order.by = data$RH)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 1.4661, df1 = 115, df2 = 114, p-value = 0.0418
## alternative hypothesis: variance changes from segment 1 to 2
```

```
#Для ISI
gqtest(fit, fraction=m, alternative = c("greater"),
  order.by = data$ISI)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 0.40741, df1 = 115, df2 = 114, p-value = 1
## alternative hypothesis: variance increases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("less"),
  order.by = data$ISI)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 0.40741, df1 = 115, df2 = 114, p-value = 1.179e-06
## alternative hypothesis: variance decreases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("two.sided"),
  order.by = data$ISI)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 0.40741, df1 = 115, df2 = 114, p-value = 2.358e-06
## alternative hypothesis: variance changes from segment 1 to 2
```

```
#Для  DMC
gqtest(fit, fraction=m, alternative = c("greater"),
  order.by = data$DMC)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 2.0642, df1 = 115, df2 = 114, p-value = 6.515e-05
## alternative hypothesis: variance increases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("less"),
  order.by = data$DMC)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 2.0642, df1 = 115, df2 = 114, p-value = 0.9999
## alternative hypothesis: variance decreases from segment 1 to 2
```

```
gqtest(fit, fraction=m, alternative = c("two.sided"),
  order.by = data$DMC)
```

```
##
##  Goldfeld-Quandt test
##
## data:  fit
## GQ = 2.0642, df1 = 115, df2 = 114, p-value = 0.0001303
## alternative hypothesis: variance changes from segment 1 to 2
```

№8 Взвешенный МНК (частный случай ОМНК) оцениваем, исходя из предположения о диагональности ковариационной матрицы случайных ошибок и не фиксированной дисперсии каждой ошибки. Чтобы оценить то, что стоит на диагонали, надо ввести какие-то предпосылки относительно того, от каких параметров наши ошибки/остатки зависят. По результатам теста Гольдфельда Квандта из пункта 7 можно предположить, что зависят они от ISI и DMC (можно было бы не париться и просто взять регрессию e~y, но для интереса сделаем так). Тогда посчитаем регрессию остатков на них, возьмём оценённые значения остатков и используем их квадраты как оценочные значения дисперсий, которые стоят на диагонали матрицы ковариации. А дальше дело техники: просто подставим их в матричное уравнение для коэффициентов модели.

Ниже приведено сравнение моделей со стандартными ошибками и значимостью. В итоге разница между моделями значительна (между исходной и Взвешенный МНК1): во-первых, стала значима переменная температуры, которая сменила знак, во-вторых, значима стала константа. Также важно отметить, что теперь гипотеза о значимости уравнения не отвергается, как это было с первоначальной моделью.

Метаморфозы с температурой, которая теперь имеет контринтутивное влияние(отрицательное), могут

быть свидетельством того, что предпосылка об устройстве матрицы ковариации не верна (на это же намекает слишком высокий R^2). На всякий случай была оценена ещё и модель взвешенного МНК 2 (в таблице последняя), сделанная из предположения, что остатки зависят от оценённых в изначальной модели y: e~y. Коэффициенты этой модели поддерживают положительное влияние temp на площадь пожара, при этом переменная значимая. Также (на уровне значимости 10%) отвергается гипотеза о том, что всё уравнение целиком не значимо.

На основе этого анализа можно сделать выводы, что много переменных не учтено (раз константа значимая), а также что стоит более реалистичные предположения наложить на матрицу ковариации случайных ошибок.

```
#Оцениваем связь ошибок и переменных? и строим на их основе диагональную матрицу ковариации W:
W <- diag(abs(lm(fit$residuals^2~data$DMC+data$ISI)$fitted.values))#возьмём оценки по модулю, иначе будет отрица
datfeel <- data
datfeel$intercept <- rep(1,266)
X <- as.matrix(datfeel[,c(11,4,6:8)]) #Матрица регрессоров с добавленной константой
Y <- as.matrix(data[,10]) #зависимая переменная
beta <- solve(t(X)%*%solve(W)%*%X)%*%t(X)%*%solve(W)%*%Y #Оценка взвешенного МНК

#Пакетная реализация
weights <- 1/abs(lm(fit$residuals^2~data$DMC+data$ISI)$fitted.values)
WOLS <- lm(area~DMC+ISI+temp+RH, data, weights=weights )
summary(WOLS) #Если сравнить коэффициенты с оцененными самостоятельно, то они будут одинаковыми- истинна
```

```
##
## Call:
## lm(formula = area ~ DMC + ISI + temp + RH, data = data, weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2559 -0.4542 -0.2950 -0.0440  6.8267
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 48.74664   10.55356   4.619 6.06e-06 ***
## DMC          0.01136    0.04405   0.258   0.7967
## ISI         -0.73121    0.34178  -2.139   0.0333 *
## temp        -0.88136    0.41109  -2.144   0.0330 *
## RH          -0.19854    0.15325  -1.296   0.1963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 261 degrees of freedom
## Multiple R-squared:  0.207,  Adjusted R-squared:  0.1949
## F-statistic: 17.03 on 4 and 261 DF,  p-value: 2.002e-12
```

```
#Другое предположение о связи e с переменными: e~y
weights1 <- 1/abs(lm(fit$residuals^2~fit$fitted.values)$fitted.values)
WOLS1 <- lm(area~DMC+ISI+temp+RH, data, weights=weights1 )
summary(WOLS1)
```

```
##
## Call:
## lm(formula = area ~ DMC + ISI + temp + RH, data = data, weights = weights1)
##
## Weighted Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.7691 -0.4359 -0.3268 -0.0139  6.3192
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.40764    7.07791   1.894   0.0593 .
## DMC          0.01172    0.04065   0.288   0.7734
## ISI         -0.76049    0.35976  -2.114   0.0355 *
## temp         0.60259    0.34010   1.772   0.0776 .
## RH          -0.04565    0.10112  -0.451   0.6520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.025 on 261 degrees of freedom
## Multiple R-squared:  0.03026,    Adjusted R-squared:  0.0154
## F-statistic: 2.036 on 4 and 261 DF,  p-value: 0.08976
```

```
#Сравнение
stargazer(fit,WOLS,WOLS1,title = "Сравнение обычного МНК и взвешенного",type="text",
                                  column.labels=c("МНК", "Взвешенный МНК 1", "Взвешенный МНК 2"),
                                  df=FALSE, digits=3)
```

```
##
## Сравнение обычного МНК и взвешенного
## ================================================================
##                                Dependent variable:
##                    ---------------------------------------
##                                       area
##                       МНК     Взвешенный МНК 1 Взвешенный МНК 2
##                       (1)          (2)              (3)
## ----------------------------------------------------------------
## DMC                    0.049        0.011            0.012
##                       (0.044)      (0.044)          (0.041)
##
## ISI                   -1.112*      -0.731**         -0.760**
##                       (0.604)      (0.342)          (0.360)
##
## temp                   0.384       -0.881**          0.603*
##                       (0.545)      (0.411)          (0.340)
##
## RH                    -0.073       -0.199           -0.046
##                       (0.184)      (0.153)          (0.101)
##
## Constant              18.380      48.747***         13.408*
##                       (14.107)     (10.554)         (7.078)
##
## ----------------------------------------------------------------
## Observations            266          266              266
## R2                     0.020        0.207            0.030
## Adjusted R2            0.005        0.195            0.015
## Residual Std. Error   35.875        1.055            1.025
## F Statistic            1.355       17.034***         2.036*
## ================================================================
## Note:                              *p<0.1; **p<0.05; ***p<0.01
```

№9. HC0. Основная идея ошибок Уайта в том, что мы предполагаем ковариационную матрицу диагональной со значениями дисперсий ошибок равных квадратам остатков модели (то есть не переоцениваем эти остатки с дополнительными предположениями, а в лоб принимаем их дисперсиями). А дальше с такой оценкой считаем матрицу ковариации коэффициентов, как если бы мы делали это через ОМНК. Таким образом коэффициенты в модели не меняются, но их стандартные ошибки строятся в предположении о другом виде ковариационной матрицы случайных ошибок.

HC3 отличаются только тем, что там предположение о матрице ковариации другое: через h_ij.

Ниже процедура оценки HC0 и HC3 и статистики. Видно, что значимости добавилось только коэффициенту при индексе ISI в сравнении с обычными МНК se.

```
#HC0 самостоятельно для данной модели

W1 <- diag(fit$residuals^2) #ковариационная матрица в рамках предположений HC0
datfeel <- data
datfeel$intercept <- rep(1,266)
X <- as.matrix(datfeel[,c(11,4,6:8)]) #Матрица регрессоров с добавленной константой
Y <- as.matrix(data[,10]) #зависимая переменная
Var_beta <- solve(t(X)%*%X)%*%t(X)%*%W1%*%X%*%solve(t(X)%*%X) #ковариационная матрица коэффициенто
HC0 <- sqrt(diag(Var_beta)) #Итоговые стандартные ошибки

#HC0 из пакета
cse = function(reg) {
  rob = sqrt(diag(vcovHC(reg, type = "HC0")))
  return(rob)
}
cse(fit)
```

```
## (Intercept)         DMC         ISI        temp          RH
## 12.71251333  0.04422958  0.52376356  0.58392561  0.18749921
```

```
#Статистика для HC0
coeftest(fit, vcov = vcovHC(fit, type = "HC0"))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.379847  12.712513  1.4458  0.14943
## DMC          0.049406   0.044230  1.1170  0.26501
## ISI         -1.111994   0.523764 -2.1231  0.03469 *
## temp         0.384272   0.583926  0.6581  0.51106
## RH          -0.073269   0.187499 -0.3908  0.69629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#HC3 из пакета
cse1 = function(reg) {
  rob = sqrt(diag(vcovHC(reg, type = "HC3")))
  return(rob)
}
#Представление модели в таблице с ошибками HC0 и HC3
stargazer(fit, fit,fit,
          se=list(HC0,cse(fit),cse1(fit)),
          title="Сравнение стандартных ошибок в разных формах", type="text",
          column.labels=c("HC0 своё","HC0","HC3"), report = "vcstp*",
```

```
       df=FALSE, digits=3)
```

```
##
## Сравнение стандартных ошибок в разных формах
## ================================================================
##                          Dependent variable:
##                     ---------------------------
##                                  area
##                     HC0 своё      HC0         HC3
##                       (1)         (2)         (3)
## ----------------------------------------------------------------
## DMC                   0.049       0.049       0.049
##                      (0.044)     (0.044)     (0.045)
##                      t = 1.117   t = 1.117   t = 1.090
##                      p = 0.264   p = 0.264   p = 0.276
##
## ISI                  -1.112      -1.112      -1.112
##                      (0.524)     (0.524)     (0.538)
##                     t = -2.123  t = -2.123  t = -2.065
##                     p = 0.034** p = 0.034** p = 0.039**
##
## temp                  0.384       0.384       0.384
##                      (0.584)     (0.584)     (0.598)
##                      t = 0.658   t = 0.658   t = 0.643
##                      p = 0.511   p = 0.511   p = 0.521
##
## RH                   -0.073      -0.073      -0.073
##                      (0.187)     (0.187)     (0.192)
##                     t = -0.391  t = -0.391  t = -0.382
##                      p = 0.696   p = 0.696   p = 0.703
##
## Constant             18.380      18.380      18.380
##                                 (12.713)    (12.987)
##                      t = 1.446   t = 1.446   t = 1.415
##                      p = 0.149   p = 0.149   p = 0.158
##
## ----------------------------------------------------------------
## Observations           266         266         266
## R2                    0.020       0.020       0.020
## Adjusted R2           0.005       0.005       0.005
## Residual Std. Error   35.875      35.875      35.875
## F Statistic           1.355       1.355       1.355
## ================================================================
## Note:                         *p<0.1; **p<0.05; ***p<0.01
```

№9 PCA. Строятся просто в пакете. Идейно мы смешиваем все переменные так, чтобы в итоговых получилась наибольшая вариации, при этом все они линейно-независимы. Так устраняется мультиколлинеарность. Чисто теоретически можно эти главные компоненты ко всем отброшенным из-за мультиколлинеарности переменным применить, но оставим так. Объясняемые первыми двумя компонентами доли дисперсии соответственно 0.5038 и 0.2638 (суммарно больше 3/4).

Модель по двум первым компонентам не сильно улучшила предсказания(R^2 всё также низок), и при этом уничтожила интерпретацию (так как эти взвешенные переменные содержательно мало что значат), к тому же коэффициенты перед векторами незначимы.

```r
set.seed(14)
#выделяем главные компоненты
dfpca <- prcomp(data[c(4,6:8)], center = TRUE,scale. = TRUE)

#Записываем 1 и 2-ю компоненты
x1 <- dfpca$x[,1]
x2 <- dfpca$x[,2]
#Доли дисперсии через summary смотрим
summary(dfpca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4
## Standard deviation     1.4196 1.0272 0.8161 0.51325
## Proportion of Variance 0.5038 0.2638 0.1665 0.06586
## Cumulative Proportion  0.5038 0.7676 0.9341 1.00000
```

```r
#модель
model <- lm(data$area ~ x1+x2)
summary(model)
```

```
##
## Call:
## lm(formula = data$area ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.740 -16.026 -11.753  -3.443 261.089
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.0148     2.2118   8.145 1.54e-14 ***
## x1            1.0502     1.5610   0.673    0.502
## x2           -0.4343     2.1572  -0.201    0.841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.07 on 263 degrees of freedom
## Multiple R-squared:  0.001872,   Adjusted R-squared:  -0.005719
## F-statistic: 0.2466 on 2 and 263 DF,  p-value: 0.7817
```

```r
stargazer(model,title = "Регрессия на главные компоненты",type="text",
                                  column.labels=c("МНК"),
                                  df=FALSE, digits=3)
```

```
##
## Регрессия на главные компоненты
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                area
##                                МНК
## ------------------------------------------------
## x1                              1.050
##                                (1.561)
##
```

```
## x2                          -0.434
##                            (2.157)
##
## Constant                    18.015***
##                            (2.212)
##
## ------------------------------------------------
## Observations                  266
## R2                           0.002
## Adjusted R2                  -0.006
## Residual Std. Error          36.074
## F Statistic                  0.247
## ================================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```