Московский государственный университет имени М. В. Ломоносова
Факультет Вычислительной математики и кибернетики

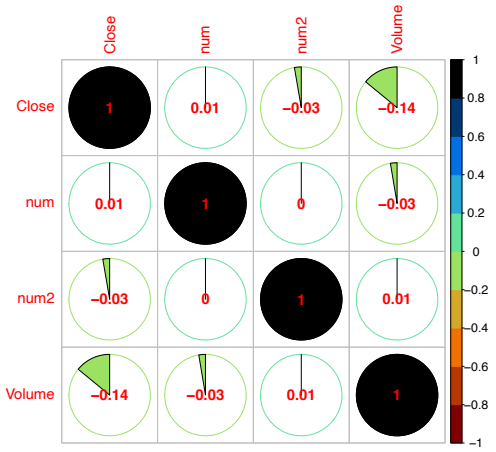
Никонов Максим Викторович
316 группа

2020

Задание 2: Для собственных данных построить доверительные интервалы для модели регрессии и интервал предсказаний.

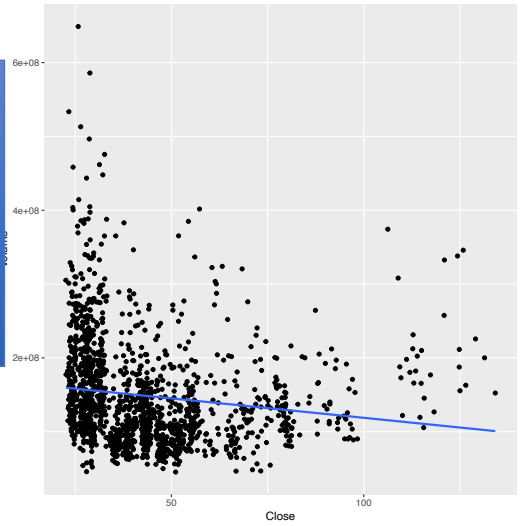
Построим наши данные, чтобы убедиться в том, что модель регрессии слишком грубо аппроксимирует данные. Для этого дополнительно покажем матрицу корреляции

```
state <- as.data.frame(data[,c("Close", "Volume", "num",
"num2")])
M <- cor(state)
col4 <- colorRampPalette(c("#7F0000","#FF7F00", "#7FFF7F",
"#007FFF","#000000"))
corrplot(M, method="pi", col=col4(10), cl.length = 11,
         order = "AOE", addCoef.col = "red")
```

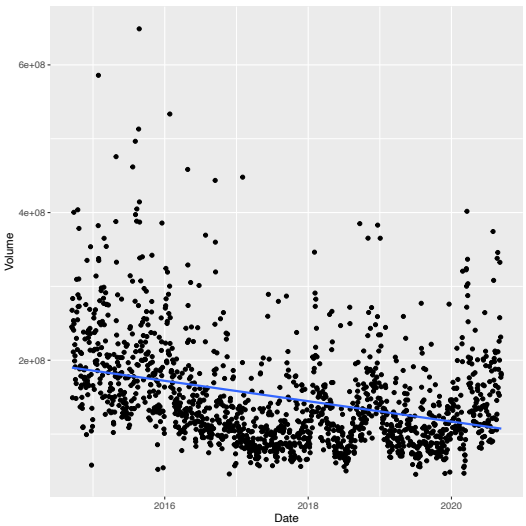


Построим теперь всевозможные регрессионные модели

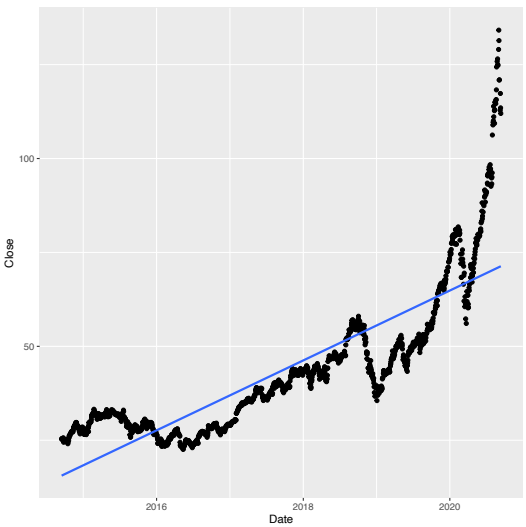
```
ggplot(data, aes(x = data$Close,
data$Volume)) + geom_point() +
geom_smooth(method = "lm", se =
FALSE) +
ylab("Volume") + xlab("Close")
```




```
ggplot(data, aes(x = data$Date,
data$Volume)) + geom_point() +
    geom_smooth(method = "lm", se =
FALSE) +
    ylab("Volume") + xlab("Date")
```




```
ggplot(data, aes(x = data$Date,
data$Close)) + geom_point() +
    geom_smooth(method = "lm", se
= FALSE) +
    ylab("Close") + xlab("Date")
```

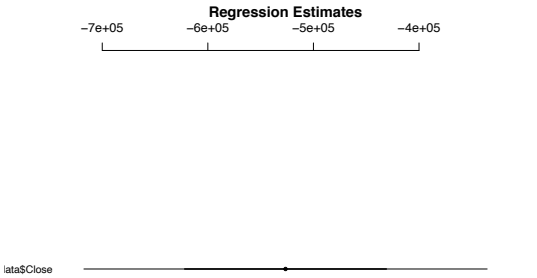


Для полученных данных построим доверительные интервалы, напомним, что Доверительный интервал это случайный интервал для фиксированного не случайного параметра, а предсказательный интервал это интервал, в котором реализуется случайная величина. Он показывает уверенность прогноза или не уверенность.

Для этого предназначена функция **confint , coefplot**, причем параметр для confint по умолчанию выставлен на уровень доверия 95%. Дополнительно зададим уровень доверия 99% через параметр **level**

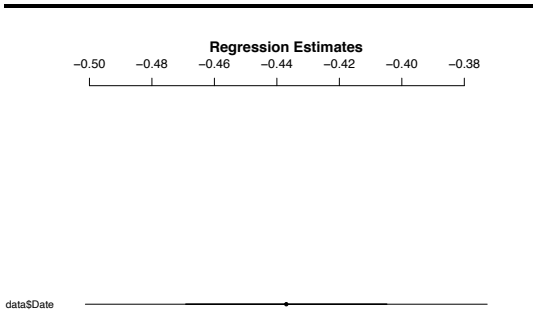
```
model <- lm(data$Volume ~
data$Close, data)
summary(model)
confint(model)
confint(model, level = 0.99)
coefplot(model, parm = -2)
```

```
> confint(model)
                  2.5 %      97.5 %
(Intercept) 162629244.8 180435727.2
data$Close    -713697.8   -339150.6
> confint(model, level = 0.99)
                  0.5 %      99.5 %
(Intercept) 159826219.0 183238753.0
data$Close    -772657.6   -280190.9
```



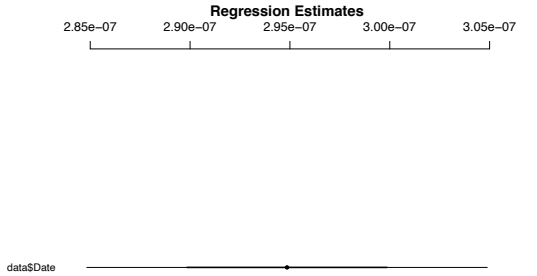

```
model <- lm(data$Volume ~
data$Date, data)
summary(model)
confint(model)
confint(model, level = 0.99)
coefplot(model, parm = -2)
```

```
> confint(model)
                    2.5 %        97.5 %
(Intercept)  7.114074e+08  9.014350e+08
data$Date   -5.000687e-01 -3.739053e-01
> confint(model, level = 0.99)
                    0.5 %        99.5 %
(Intercept)  6.814940e+08  9.313484e+08
data$Date   -5.199288e-01 -3.540451e-01
```




```
model <- lm(data$Close ~
data$Date, data)
summary(model)
confint(model)
confint(model, level = 0.99)
coefplot(model, parm = -2)
```

```
> confint(model)
                    2.5 %        97.5 %
(Intercept) -4.151895e+02 -3.855671e+02
data$Date    2.850223e-07  3.046892e-07
> confint(model, level = 0.99)
                    0.5 %        99.5 %
(Intercept) -4.198525e+02 -3.809040e+02
data$Date    2.819264e-07  3.077851e-07
```



Ограничения линейной регрессии

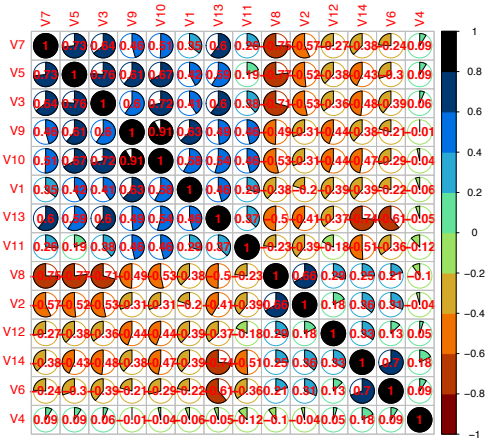
- Линейность: зависимая переменная может линейно аппроксимировать независимые переменные
- Нормальность распределения Y и ε
- Отсутствие избытка влиятельных наблюдений
- Гомоскадастичность распределения остатков
- Отсутсвие мультиколлинеарности

Для наглядности линейно регрессионной модели и доверительных интервалов была взята база **Housing Dataset** опубликованной в статье данных о недвижимости в Бостоне в 1978 г

```
BD <-
read.table("https://raw.githubusercontent.com/rasbt/python-
machine-learning-book-2nd-
edition/master/code/ch10/housing.data.txt")
```

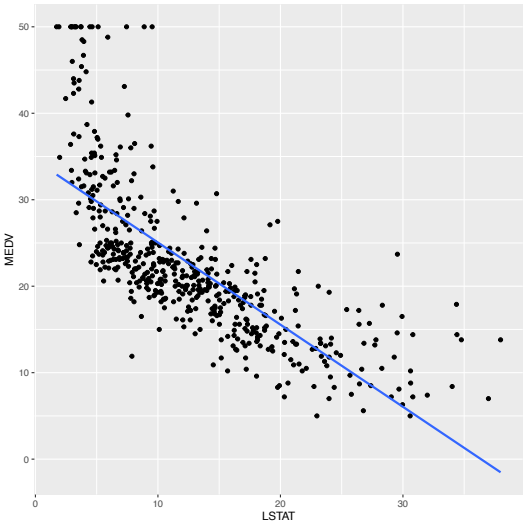
Снова выведем таблицу корреляций и определим зависимые и независимые переменные

```
state <- as.data.frame(BD)
M <- cor(state)
corrplot(M, method="pi",
col=col4(10), cl.length = 11,
         order = "AOE",
addCoef.col = "red")
```



Построим

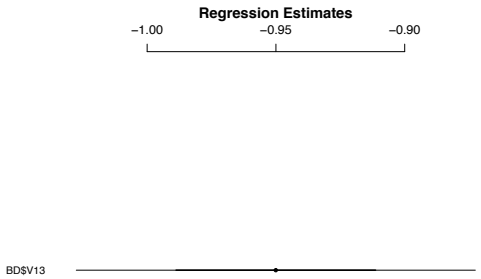
```
gplot(BD, aes(x = BD$V13, BD$V14))
+ geom_point() +
    geom_smooth(method = "lm", se
= FALSE) +
    ylab("MEDV") + xlab("LSTAT")
```



А теперь и доверительный интервал.

```
model <- lm(BD$V14 ~ BD$V13,
data)
summary(model)
confint(model)
confint(model, level = 0.99)
coefplot(model, parm = -2)
```

```
> confint(model)
                2.5 %     97.5 %
(Intercept) 33.448457 35.6592247
BD$V13      -1.026148 -0.8739505
> confint(model, level = 0.99)
                0.5 %     99.5 %
(Intercept) 33.099101 36.0085810
BD$V13      -1.050199 -0.8498995
```



Задание 3: Модифицировать функцию regr() для расчета возраста Вселенной с помощью бутстрепа.

Сначала для базы **hubble** переведем y - относительную скорость движения любых двух галактик и $x = y/h \mid h$ – постоянная Хаббла в соответсвенно км/год и км

```
hubble$x <- hubble$x*3.09e19
hubble$y <- hubble$y*60^2*24*365.25
```

Зададим изменённую фунцию **regr**

```
regr <- function(data, indices)
{
    BD <- data[indices, ]
    fit <- lm(x ~ -1 + y, data = BD)
    return(summary(fit)$coefficients[1])
}
```

Теперь найдем границы через **boot** с новой функцией **regr**

```
results <- boot(data = hubble, statistic = regr, R = 1000)
results
h.lower <- quantile(results$t, 0.025)
h.upper <- quantile(results$t, 0.975)
c(h.lower, h.upper)
boot.ci(results, type = "bca")
```

Получим

```
> c(h.lower, h.upper)
       2.5%       97.5%
10894068308 13577431085
> boot.ci(results, type = "bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca")

Intervals :
Level       BCa
95%   (10805802002, 13507914453 )
Calculations and Intervals on Original Scale
>
```

Верхняя граница - 13.5
Нижняя граница – 10.8

Это показывает, что возраст вселенной составляет примерно 12.1 млрд лет. Однако по последним данным возраст вселенной составляет 13.7 млрд лет.

Важным количественным показателем здесь является **F-критерий**

```
F-statistic: 373.1 on 1 and 23 DF,  p-value: 1.032e-15
```

Так же как и в дисперсионном анализе, в регрессионном анализе F-критерий используется для сравнения дисперсий

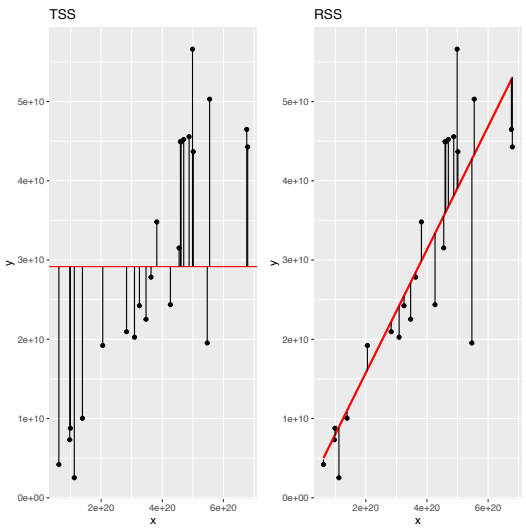
$$F = \frac{\frac{TSS-RSS}{p}}{\frac{RSS}{n-p-1}}$$

где, TSS – общая сумма квадрантов, а RSS - сумма остатков квадрантов. n – объем выборки, p – число параметров модели
Продемонстрируем геометрическую интерпретацию

```
p1 = ggplot(hubble, aes(x, y)) +
geom_point() +

geom_hline(aes(yintercept=mean(hubbl
e$y)), color = "red") +
  geom_segment(aes(x = x, y = y,
xend = x, yend = mean(hubble$y))) +
  ggtitle("TSS")

p2 = ggplot(hubble, aes(x, y)) +
geom_point() +
  geom_smooth(method = "lm", se =
FALSE, color = "red") +
  geom_segment(aes(x = x, y = y,
xend = x, yend = fit)) +
  ggtitle("RSS")
grid.arrange(p1, p2, ncol = 2)
```



F-критерий в этом примере составил 373.1, что гораздо больше 1. Вероятность получить такое высокое значение при отсутствии связи между x и y очень мала (P = 1.032e-15). Соответственно, мы можем сказать, что в целом полученная модель хорошо описывает имеющиеся данные.

Задание 4: Изучить коэффициент детерминации (R-squared и adjusted R-squared) как критерий качества аппроксимации моделью.

```
Multiple R-squared:  0.9419, Adjusted R-squared:  0.9394
```

R-squared – коэффициент детерминации

$$R^2 = 1 - \frac{RSS}{TSS}$$

Чем ближе значение коэффициента детерминации к 1, тем точнее модель описывает данные.Дело в том, что значение R^2 всегда будет возрастать при увеличении числа предикторов в модели, даже если некоторые из этих предикторов не имеют связи с зависимой переменной. Простой коэффициент детерминации будет отдавать предпочтение переобученным моделям, что крайне негативно. Поэтому имеем скорректированный коэффициент детерминации или **Adjusted R-squared**

$$R^2_{adj} = R^2 - (1 - R^2)\frac{p}{n - p - 1}$$

Очевидно, что чем больше количество p, тем больше отнимем от R-squared
Однако для наших данных не критично и модель достаточно точно описывает данные

Используемые packages. ggplot2, dplyr, car, arm, corrplot, gamair, boot, gridExtra
Тестирование. Не требует
Неразрешенные вопросы. Нет
Новые функции. confit, coefplot, boot, quantile, fitted
Статус компиляции. ОК. Данные из протокола:

```
^[[1m^[[7m%^[[27m^[[1m^[[0m
^M ^M^[]7;file://MBP-
Nikon.Dlink/Users/Nikon/Desktop/CMC%20MSU/MC/5%20sem/R/tz12^G^M^[[0m^[[27m^[[24m^[[JNikon@
MBP-Nikon tz12 % ^[[K^[[?2004hR --no-save <task.r^[[19Dexit
^[[15D^[[?2004l^M^M

                        Script done on Tue Nov 24 17:19:45 2020
```