МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

**ПО ДИСЦИПЛИНЕ «МЕТОДЫ АНАЛИЗА ДАННЫХ»**

**НА ТЕМУ «Бутстреп-анализ»**

Факультет: АВТФ Преподаватель: Альсова О.К.

Группа: АВТ-010

Студент: Великжанин С.И.

Новосибирск 2023 г.

**Оглавление**

[1. Теоретическая часть 3](#_Toc153781073)

[1.1. Общие сведения 3](#_Toc153781074)

[1.2. Алгоритм метода 3](#_Toc153781075)

[1.3. Достоинства метода 4](#_Toc153781076)

[1.4. Недостатки метода 5](#_Toc153781077)

[1.5. Реализация в R 5](#_Toc153781078)

[2. Практическая часть 7](#_Toc153781079)

[Выводы 13](#_Toc153781080)

[Список литературы 14](#_Toc153781081)

[Приложение А 15](#_Toc153781082)

[Листинг 15](#_Toc153781083)

# Теоретическая часть

# Общие сведения

Помимо статистических методов проверки гипотез существует альтернативный подход, который вместо теории задействует грубую компьютерную силу – ресемплинг. Этот подход объединяет целый класс методов генерации дополнительных выборок из уже имеющихся. При ресемплинге расчет параметров проводится на фактических данных, а теоретическое распределения не используется.

Бутстреппинг основан на том, что зачастую наиболее достоверную информацию о форме распределения случайной величины несёт имеющаяся у нас выборка значений этой случайной величины. Например, выборка наблюдений с двумя пиками в гистограмме не будет хорошо аппроксимироваться кривой нормального распределения, которая имеет только одну вершину. И вместо того, чтобы использовать неадекватную форму распределения случайной величины (например, нормальную), мы используем форму распределения, которую даёт нам выборка.

Для оценки любых параметров можно сформировать тысячи повторных бутстреп-выборок (обычно 500-10 000), каждая из которых содержит 2/3 значений исходной выборки.

# Алгоритм метода

Пусть дана выборка из генеральной совокупности, и требуется оценить параметр θ. Необходимо выбрать количество B псевдовыборок, которые будут формироваться из элементов исходной выборки с возвращением. Для каждой из псевдовыборок вычисляется псевдостатистика .

Псевдостатистики сортируются от меньшей к большей. Квантилями принимаются значения . С их помощью строится доверительный интервал.

Алгоритм нахождения доверительного интервала:

1. Выберите 10 наблюдений из выборки. Отбор осуществляется с возвратом, то есть некоторые наблюдения могут быть выбраны несколько раз, а некоторые могут остаться невыбранными.
2. Вычислите среднее для полученного набора из 10 значений.
3. Повторите шаги 500 - 10 000 раз.
4. Отсортируйте тысячу выборочных средних по возрастанию.
5. Найдите средние, которые представляют собой 2.5 и 97.5 процентили.

В качестве примера предположим, что мы исследуем высоту людей во всем мире. Мы не можем измерить всех людей, а вместо этого выбираем лишь малую часть. Пусть в нашей выборке N людей. Мы можем посчитать среднее значение. Но для того, чтобы рассуждать о доверительном интервале роста населения, нам нужно некоторое представление о вариабельности среднего. В простейшем варианте бутстреппинга мы, используя наши исходные данные о росте N различных людей, составляем новую выборку, также размера N. Это новая выборка взята из исходной случайным образом так, что одно значение может быть выбрано несколько раз (т.е., строя новую выборку, мы каждый раз случайным образом выбираем из N имеющихся значений). У такой выборки будет другое среднее. Сделав такую выборку много раз (500 – 10 000), каждый раз вычисляя среднее, мы получаем гистограмму распределения, которая может ответить на вопросы о доверительном интервале.

# Достоинства метода

Преимущества бутстреппинга проявляются при использовании крайне нелинейных моделей вроде деревьев решений. На разных наборах данных они будут давать весьма несхожие границы, которые будут слабо коррелировать друг с другом. В конечном итоге мы уменьшаем дисперсию путём сочетания нелинейных моделей, которые коррелируют с меньшей вероятностью.

# Недостатки метода

Применить бутстреп-метод без компьютера невозможно, потому что он был создан именно для использования вычислительных мощностей компьютеров. Также недостатками являются неизбежность совпадения элементов размноженных выборок и зависимость от качества датчиков псевдослучайных чисел.

# Реализация в R

В языке R бутстреппинг представлен функцией *boot()*, для использования которой необходимо установить одноименный пакет. Описание функции из официальной документации R:

boot(object, f=coef, labels=names(f(object)), R=999, method=c("case", "residual"), ncores=1, start = FALSE, …)

**object –** регрессионный объект класса "lm", "glm" или "nls". Функция может работать с другими объектами регрессии, которые поддерживают update метод и имеют subset аргумент.

**f –** Функция, одним аргументом которой является имя объекта регрессии, который будет применен к обновленному объекту регрессии для вычисления интересующей статистики. По умолчанию coef возвращаются оценки коэффициента регрессии.

**labels –** предоставляет метки для статистики, вычисляемой f.

**R –** количество генераций псевдовыборок. Количество фактически вычисленных псевдовыборок может быть меньше этого значения.

**method –** бустреп-метод.

**start –** следует ли возвращать f начальный значения для каждой бутстреп-итерации? Также start может быть задан числовым вектором начальных значений. По умолчанию оценки из последней итерации начальной загрузки используются в качестве начальных значений для следующей итерации.

**ncores –** числовой аргумент, указывающий количество ядер для параллельной обработки в системах unix. Если меньше или равно 1, параллельная обработка использоваться не будет. *Примечание*: на платформе Windows выдаст предупреждение и установит для этого аргумента значение 1.

С помощью функции *boot.ci()* можно сгенерировать 5 различных типов доверительных интервалов. Описание функции из официальной документации R:

boot.ci(boot.out, conf = 0.95, type = "all",

index = 1:min(2,length(boot.out$t0)), var.t0 = NULL,

var.t = NULL, t0 = NULL, t = NULL, L = NULL,

h = function(t) t, hdot = function(t) rep(1,length(t)),

hinv = function(t) t, …)

**boot.out –** объект класса "boot", содержащий результат вычисления бутстреппинга.

**conf –** скаляр или вектор, содержащий доверительные уровни требуемого интервала(ов).

**type –** вектор символьных строк, представляющих требуемый тип интервалов. Значение должно быть любым подмножеством значений c("norm","basic", "stud", "perc", "bca") или просто "all", вычислющее все пять типов интервалов.

**index –**вектор длиной 1 или 2. Первый элемент index указывает положение нужной переменной boot.out$t0 и соответствующий столбец в boot.out$t. Второй элемент указывает положение дисперсии нужной переменной. Если указаны оба var.t0 и var.t, то второй элемент index (если есть) игнорируется. По умолчанию интересующая переменная находится в позиции 1, а ее дисперсия - в позиции 2 (пока есть 2 позиции boot.out$t0).

**var.t0 –** необязательное значение, которое будет использоваться в качестве оценки дисперсии статистики для нормального приближения и стьюдентизированных интервалов.

**var.t –** это вектор длины boot.out$R дисперсий бутстреповских реплик интересующей переменной. Используется только для студентизированных интервалов.

**t0 –** эмпирическое значение. Значение по умолчанию - boot.out$t0[index[1]].

**t –** вектор длины boot.out$R.

**L –** значения эмпирического влияния интересующей статистики на наблюдаемые данные.

**h –** функция, определяющая преобразование. Интервалы рассчитываются по шкале h(t)и к полученным интервалам применяется обратная функция hinv. Это должна быть функция только одной переменной, а для аргумента вектора она должна возвращать вектор такой же длины, т.е. h(c(t1,t2,t3))должна возвращать c(h(t1),h(t2),h(t3)). По умолчанию используется функция идентификации.

**hdot –** функция одного аргумента, возвращающая производную от h. Это обязательный аргумент, если h указан и требуются нормальные или студентизированные интервалы. Функция используется для аппроксимации дисперсий h(t0) и h(t)использования дельта-метода.

**hinv –** функция, например h, которая возвращает значение, обратное h. Он используется для преобразования интервалов, рассчитанных по шкале, h(t) обратно в исходную шкалу. По умолчанию используется функция идентификации.

# Практическая часть

Для выполнения практической части задания с ресурса Kaggle был взят набор данных о сериалах Netflix, Prime Video, Hulu и Disney+. Цель исследования – проследить динамику оценок сериалов относительно года выпуска. В выборке представлено 5576 записей. Годы премьер сериалов варьируются с 1934 по 2020.



Рис. 1 – Фрагмент набора исходных данных.

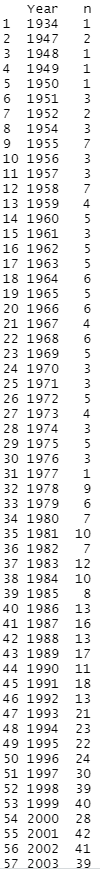
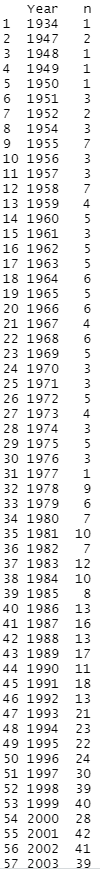
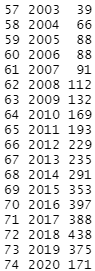
 

Рис. 2 – Количество сериалов в выборке за каждый год.

Видим, что за 1934-1979 годы число сериалов в выборке может опускаться до 1, а за 2008-2019 достигает не менее 112 за год, поэтому возьмем для оценки сериалы 1986-2020 годов (не менее 11 сериалов за год), а затем сравним с изначальной выборкой.

В качестве меры качества сериала будем использовать рейтинг от ресурса IMDb, так как большая часть сериалов, представленная в выборке, не имеет рейтинга от Rotten Tomatoes.

С помощью бустреппинга сгенерируем псевдовыборки. Бутстреппинг будет применяться в совокупности с функцией *cor.test()*, по умолчанию использующей метод Пирсона. Найдем корреляцию между годом и рейтингом, заодно проверив различия в результатах при 500 и 10 000 итерациях.

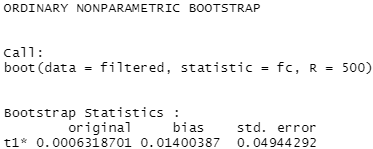


Рис. 3 – Краткие результаты бутстреппинга (500 итераций).

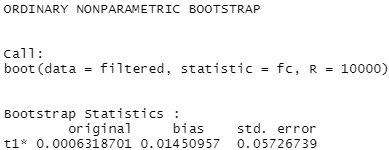


Рис. 4 – Краткие результаты бутстреппинга (10 000 итераций).

При увеличении числа итераций в 20 раз системное отклонение (bias) и стандартная ошибка (std. error) увеличились. Не изменилось лишь t1 – это p-значение. В данном случае оно меньше 0.0007, а значит, парная связь «год-рейтинг» является статистически значимой.

Построим график зависимости рейтинга сериала от года премьеры. На графике обозначим линию регрессии.

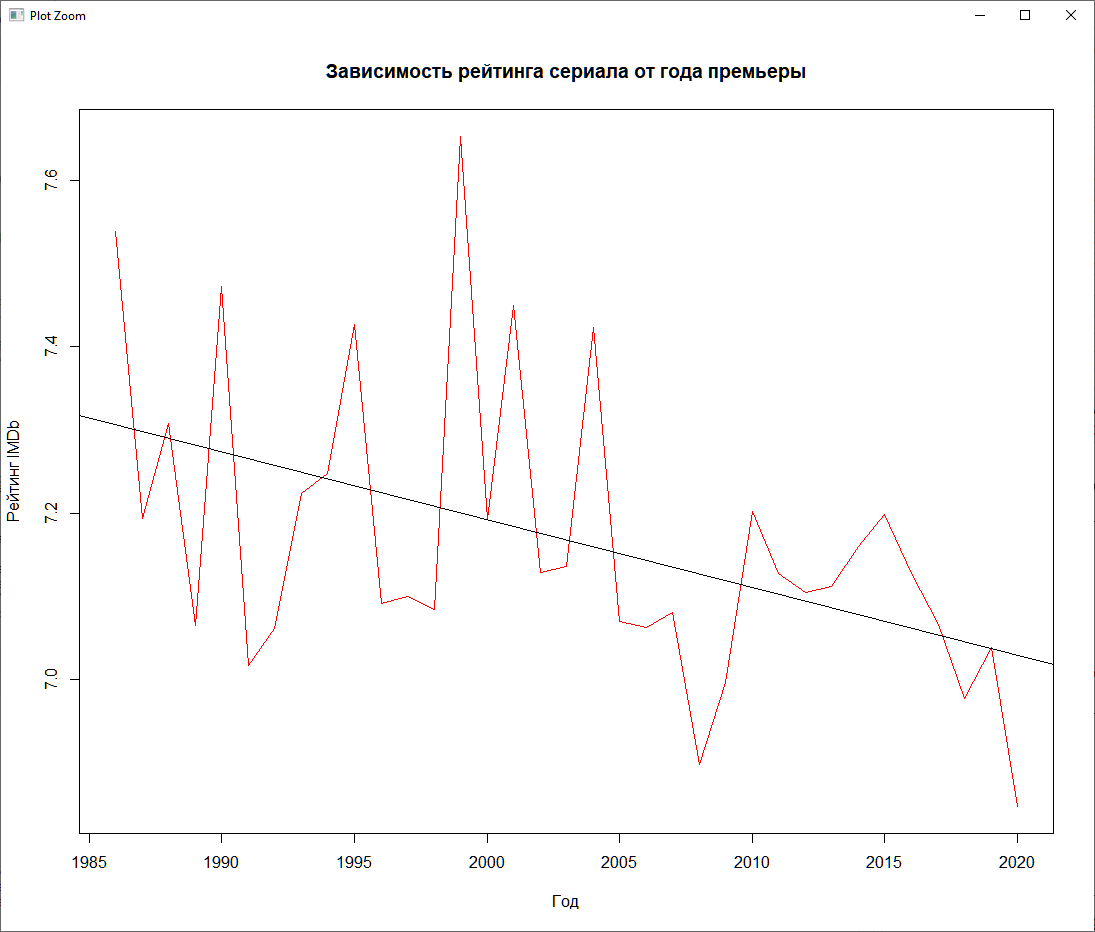


Рис. 5 – Зависимость рейтинга сериала от года премьеры (выборка 1986-2020).

Линия регрессии показывает устойчивое падение среднего рейтинга сериалов. На момент 2020 года средний рейтинг сериалов поставил антирекорд, опустившись до минимальной отметки. Последний раз схожая ситуация наблюдалась лишь в 2008 году.

Построим аналогичный график по полной выборке.

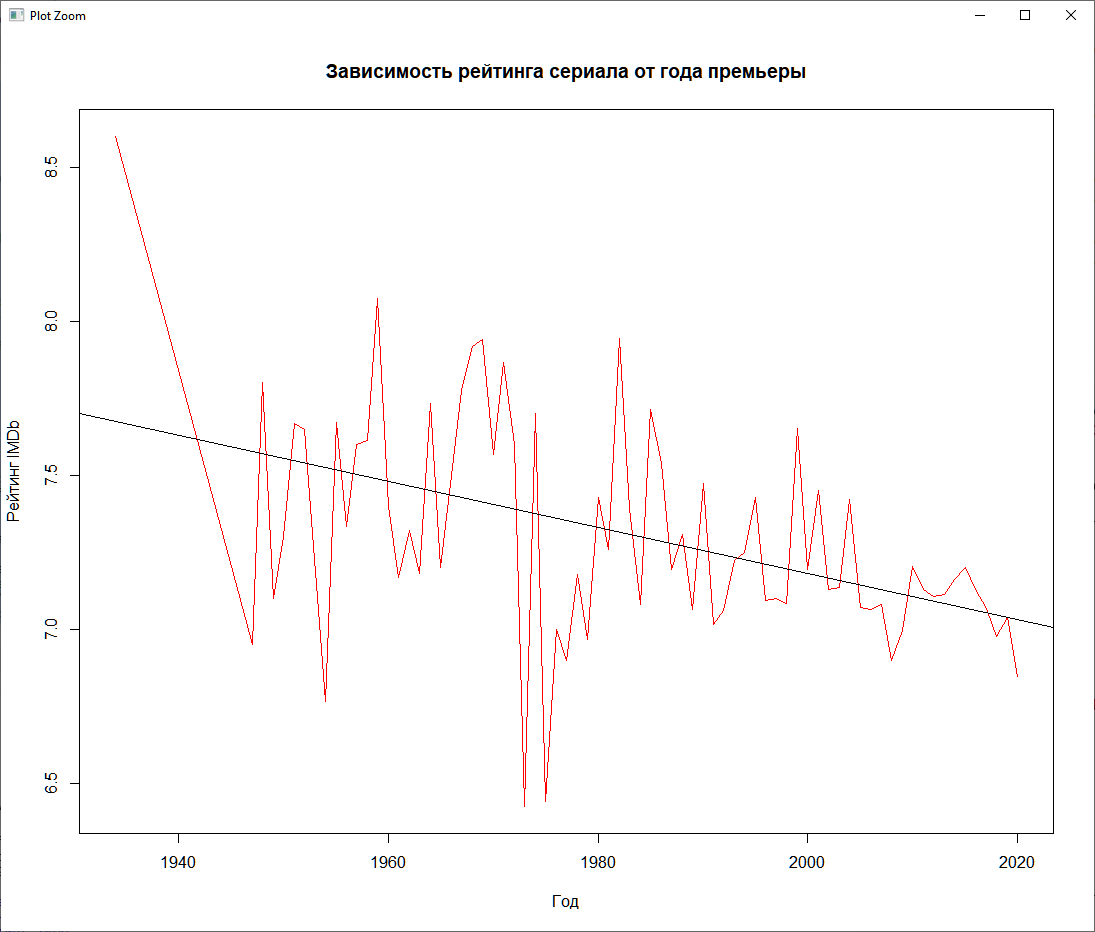


Рис. 6 – Зависимость рейтинга сериала от года премьеры (полная выборка).

Как видно из рис. 7, в середине 1970-х годов сериалы в среднем имели рекордно плохой рейтинг за всю историю. В начале XX века – рекордно хороший. Начало 1980-х – 2000-х годов имеет в среднем одинаковые. Построим аналогичный график с выборкой сериалов этих лет.

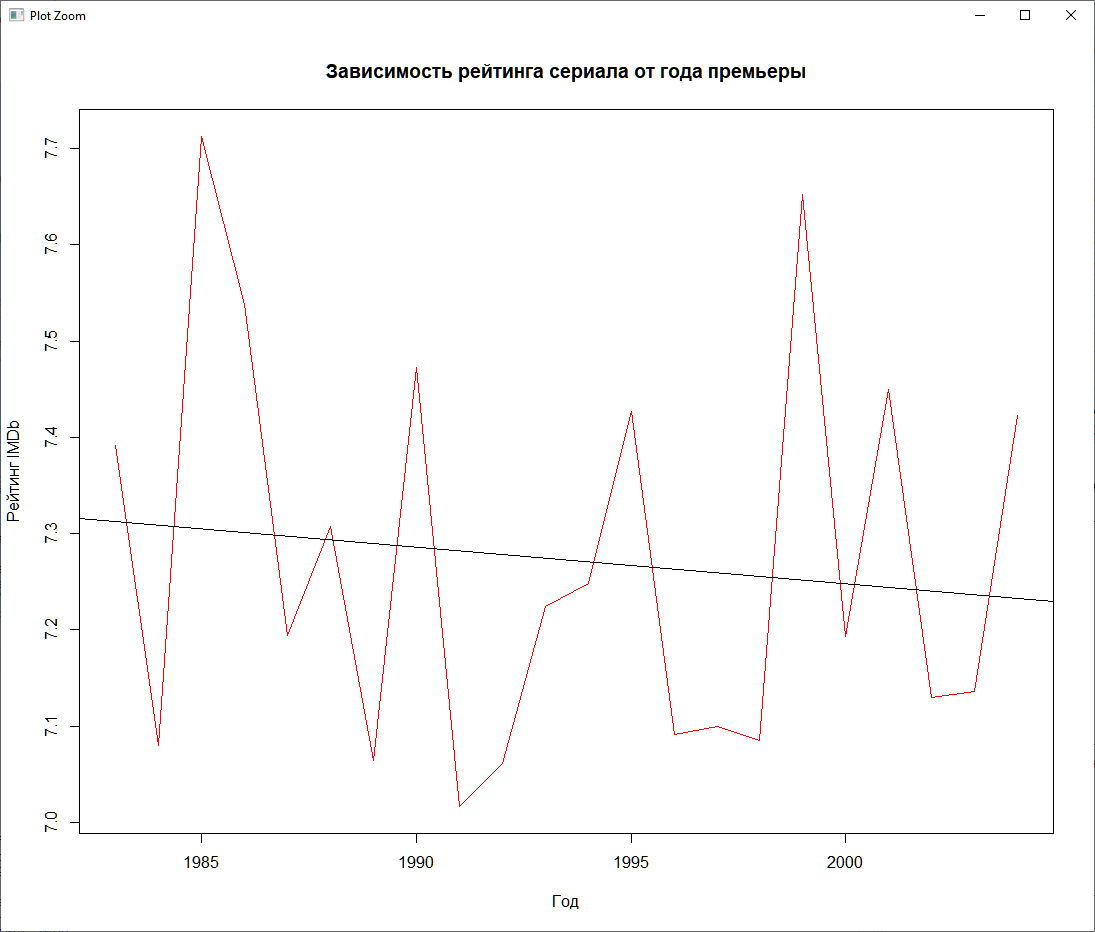


Рис. 7 – Зависимость рейтинга сериала от года премьеры (неполная выборка).

Средний рейтинг сериалов падал и за этот период, хоть и меньшими темпами, чем в предыдущих выборках.

# Выводы

В ходе выполнения расчетно-графической работы была проанализирована выборка сериалов 1934-2020 гг. Результаты показали, что с течением времени сериалы получают все более низкие оценки. Это можно объяснить тем, что о современных сериалах знает больше людей, и поэтому их средняя оценка ниже, чем у сериалов, которым более 70 лет. Также может влиять и то, что среди пользователей есть многие, кто специально занижают оценку современным сериалам, просто потому что они популярны.

Помимо этого, был опробован бутстреппинг с помощью средств R. Было замечено, что при количестве итераций, равном 500 (минимально рекомендуемое количество [2]), вычисления выполняются намного быстрее, чем при количестве итераций, равно 10 000 (что подтверждает ЭВМ-нишу применения бутстреппинга). Разница в результатах вычислений при варьировании количества итераций также имеет место быть.

# Список литературы

1. Шитиков В.К., Розенберг Г.С. Рандомизация и бутстреп: статистический анализ в биологии и экологии с использованием R. - Тольятти: Кассандра, 2013. - 314 с.
2. Орлов А.И. О реальных возможностях бутстрепа как статистического метода. // Заводская лаборатория. 1987. Т.53. No.10. С.82-85.
3. Bradley Efron. Bootstrap Methods: Another Look at the Jackknife (англ.) // Annals of Statistics. — 1979. — Vol. 7, no. 1. — P. 1—26. — ISSN 0090-5364.
4. University of California, Institute of Digital Research & Education – Statistical Consulting Web Resources URL: https://stats.idre.ucla.edu/r/ (дата обращения: 16.12.2023).
5. R Documentation and manuals URL: https://www.rdocumentation.org/ (дата обращения: 16.12.2023)
6. TV shows on Netflix, Prime Video, Hulu and Disney+ URL: <https://www.kaggle.com/datasets/ruchi798/tv-shows-on-netflix-prime-video-hulu-and-disney> (дата обращения: 16.12.2023)

# Приложение А

# Листинг

library("boot")

library(dplyr)

library(magrittr)

# функция для нахождения корреляции методом Пирсона (по умолчанию)

fc <- function(d, i){

d2 <- d[i,]

return(cor.test(d2$Year, d2$IMDb)$p.value)

}

# читаем датафрейм

df <- read.csv("C:\ tv\_shows.csv",

header = TRUE,

sep = ",")

# делаем выборку, в которой исключаем NA-значения и неактуальные года

filtered <- subset(df, !is.na(IMDb) & Year >= 1983 & Year <= 2004)

# смотрим на количество фильмов по годам

filtered %>% count(Year)

# выполняем бутстреппинг

boot(filtered, fc, R = 500)

# группируем датафрейм по годам и находим средние значения для каждой группы

aggregated <- aggregate(filtered[, ], list(filtered$Year), mean)

# рисуем линию регрессии

regression <- lm(IMDb ~ Year, data = aggregated)

summary(regression)

with(aggregated, plot(aggregated$Year, aggregated$IMDb,

main = "Зависимость рейтинга сериала от года премьеры",

xlab = "Год", ylab = "Рейтинг IMDb",

type = "l", col = "red"))

abline(regression)