Bugs statistic

Alexey Popov

18 июля 2018 года

## Баги и статистика

### Что мы хотим узнать?

Мы работает используя Agile/Scrum подход с недельными спринтами и оценкой задач в идеальных часах по шкале (1,2,4,8,16,24,40,..). На спринты редко попадают задачи в 24 часа, так как опытным путем установлено, что 24 идеальных часа это несколько больше 48 астрономических, то есть такая задача априори имеет хорошие шансы быть недоделанной - такие задачи просто не влезают в недельные спринты. Однако бывает и так: такие задачи на спринт все-таки берутся и, как ни странно, выполняются. Не всегда, но достаточно часто. Почему? Потому что 24 - это усредненная оценка, содержащая ошибку, то есть ее можно рассматривать как вероятностную величину. Внимание! Далее везде используются оценки в идеальных часах, а не астрономических. Даже если происходит переоценка тикета после выполнения, делается это в идеальных часах.

Что происходит, если поддержка приносит на спринт баг (воспроизводимый и оформленный, с подготовленной средой, проанализированными логами, развернутыми дампами и т.д.)? Есть два варианта. Вариант первый: баг простой и понятный (ошибка в орфографии, не в том формате выводиться поле в отчет и т.д.) и тогда разработчики сразу говорят - 1 час, или 2 часа, или "ну поставь 4, там надо повозиться..". Вариант второй: "а хрен его знает, надо посмотреть..." И на это "посмотреть" может уйти несколько часов, а то и дней. В этом случае мы ставим 4 часа (иногда, если сразу понятно, что работы будет много - 8) и рассматриваем эту задачу как спайк: разработчик должен изучить вопрос и а) исправить баг если трудозатраты в сумме меньше 4(8) часов или б) описать суть проблемы изнутри и предложить решение, предложить новую оценку и отложить фиксацию до следующего спринта. Однако на практике часто бывает так, что на исследование может уйти сильно больше 4(8) часов, зато исправление оказывается простым или относительно простым, и откладывать его еще на неделю как-то глупо. В этой ситуации исправление производится сразу, а задача переоценивается. Очевидно, что такие задачи будут "подрастать" по ходу спринта, и это не хорошо. С другой стороны, если баг казался сложным (4-х часовым), а исправили его за час, то есть происходит компенсация увеличение трудозатрат на сложные баги за счет снижение трудозатрат на простые. Конечно бывает и так, что баг, казавшийся простым и оцененный, скажем, в 1 час, на деле оказался сложнее и был исправлен, скажем, за 4 идеальных часа. К сожалению, на данный момент нет возможности получить априорную оценку по всем багам и сравнить, но по ощущению такое случается достаточно редко.

Исходная оценка в 4 часа для сложных багов была выбрана интуитивно 2 года назад, также, как и правила работы с бажными тикетами. Это было примерно спустя полгода после того как мы перешли на scrum. К текущему моменту накопилось уже достаточно приличная статистика по фиксации бажных тикетов. Поэтому первая группа вопросов ставиться так: правильно ли была выбрана исходная оценка в 4 часа? Происходит ли компенсация на практике? Правильно ли вообще мы работаем с бажными тикетами, не ли здесь какой-то систематической проблемы?

Вторая группа вопросов касается свойств распределения случайной величины (то есть оценки трудозатрат на исправление бага) и происходит из моего спора с одним известным бизнес-тренером. Вкратце суть спора такова. Я задал вопрос: что он может порекомендовать (какие инструменты, методики) для планирования в том случае, когда априори трудозатраты на задачу не известны и по определению не могут быть оценены до начала работ. Ответ тренера был примерно таков: трудозатраты можно рассматривать как вероятностную величину (тут я согласен) распределенную нормально (а вот тут я не был согласен) и просто брать среднюю данного распределения. Простые задачи будут компенсировать сложные и все будет хорошо (тут я тоже не согласился, еще не имея конкретных цифр на руках, но помня, что происходит на спринтах). Поэтому вторая группа вопросов ставиться так: является ли распределение трудозатрат на исправление багов нормальным? Можно ли в данном случае использовать свойства нормального распределения (критерий 3 сигмы, оценку вероятности редких событий и т.д.)?

### Основные параметры реальной выборки

Итак, я взял все бажные тикеты, закрытые одной командой за последние 2.5 года в рамках одного большого проекта. Состав команды за это время сменился примерно на половину, но это происходило плавно. Костяк команды не менялся, в него входит 4 разработчика с опытом работы больше 15 лет (продуктовая команды, С++/С#). Общая численность команды колебалась от 8 до 11 человек включая тестировщика и дизайнера/UX-специалиста.

Надо отметить, что оценка тикетов производится либо методом покер-планирования, либо методом экспертной оценки. При закрытии спринта оценка корректируется, если реальные трудозатраты отличались от плановых. В любом случае всегда единицы измерения - "идеальные" трудочасы.

Полная выгрузка из Redminе включала около 1050 записей. После процедуры очистки осталось чуть большое 900. Вполне приличная выборка для статистического анализа. В ходе очистки из выборки были исключены тикеты без оценки или с 0-ой оценкой (такое бывает, если баг исчез вместе с модулем, или его исправление стало следствие рефакторинга, проведенного в рамках другой задачи и т.п.). Пара десятков тикетов имели "нестандартные" оценки (например, 5 часов), которые были приведены к ближайшему "стандартному" целому. Некоторые тикеты были "потеряны" - я привязал их к соответствующему спринту по дате закрытия. В выборку попали все баги, закрытые данной командой за последние 2.5 года. Это важно, иначе пришлось бы говорить об учете эффекта селекции.

Загрузим очищенные данные и проверим итоговый размер выборки:

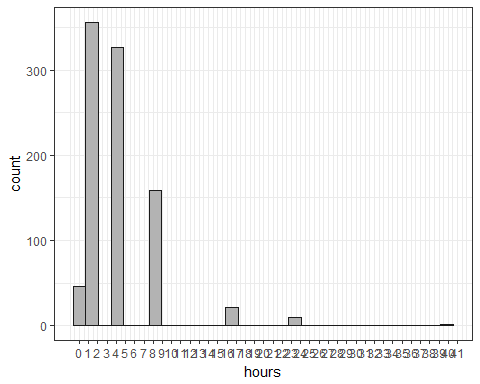
bugs <- readRDS("bugs.RDS")  
nrow(bugs)

## [1] 919

Посмотрим на распределение:

ggplot(data = bugs) + geom\_histogram(aes(x = hours), fill = "gray70", colour = "gray10") +  
 scale\_x\_continuous(breaks=seq(0,41,by=1)) +  
 theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Ну, на первый неискушенный взгляд кажется, что гистограмма выглядит весьма "нормально", хотя и подрезано слева.

Проверим самые важные параметры. Итак... Момент Истины! Барабанная дробь....

median(bugs$hours)

## [1] 4

bugs\_m <-mean(bugs$hours)  
bugs\_m

## [1] 4.062242

Просто замечательно! Во-первых, медиана и среднее лежат очень близко друг к другу. А во-вторых, предложение взять 4 часа в качестве значения по умолчанию (сделанное на глазок, без всяких расчетов) оказалось "не в бровь, а в глаз"! Лично я верю, что это интуиция очень опытного человека, а не само сбывающийся прогноз.

А что с разбросом?

bugs\_sd <-sd(bugs$hours)  
bugs\_sd

## [1] 3.809011

Имеем среднеквадратическое отклонение около 4 часов. Мы пока не знаем, имеем ли мы дело с нормальным распределением или нет, но давайте опредлим границы по критерию 3-сигмы.

mean(bugs$hours) + sd(bugs$hours)\*3

## [1] 15.48927

mean(bugs$hours) - sd(bugs$hours)\*3

## [1] -7.36479

Вот уже в этом месте стоит призадуматься. Во-первых, совершенно очевидно, что работы с отрицательными трудозатратами не бывает. Соответственно наша выборка будет иметь большую асимметрию. Во-вторых, 16-ти и 24-х часовые тикеты оказываются за границами инетрвала в 3-сигмы. На тестовых данных мы видим, что

z <- rnorm(1000, mean = 4, sd = 4)  
sum(z>=16)

## [1] 2

sum(z>=14)

## [1] 8

таких тикетов почти нет. Вероятность их появления в случае нормального распределения будет равна

1 - pnorm(24, mean = bugs\_m, sd = bugs\_sd)

## [1] 8.277573e-08

1 - pnorm(16, mean = bugs\_m, sd = bugs\_sd)

## [1] 0.0008619581

соответственно. Про 40-ка часовые и говорить нечего.... Однако мы знаем, что на практике это не так - такие задачи встречаются, причем регулярно. А что в реальной выборке?

sum(bugs$hours==16)

## [1] 21

sum(bugs$hours==24)

## [1] 9

sum(bugs$hours==40)

## [1] 1

sum(bugs$hours>=16)

## [1] 31

В реальной выборке мы видим, что такие тикеты есть и их относительно много. Самое время найти ответ на второй вопрос (тем более, что он проще)...

### Является ли данное распределение нормальным?

Проверим еще две характеристики, асимметрию (должна быть равна 0) и эксцесс (должен быть близок к 3). Асимметрия:

skewness(bugs$hours)

## [1] 3.046948

далека от нуля и положительна. Это говорит о том, что распределение "завалено" вправо. Эксцесс:

kurtosis(bugs$hours)

## [1] 18.55184

что в несколько раз больше 3. Что в свою очередь говорит о том, что распределение "приподнято" по краям, "размазано". Таким образом вероятность "поймать" баг с трудозатратами более 16-ти часов больше, чем это было бы характерно для нормального распределения.

Используем критерий Хи-квадрат.

# теоретическая плотность вероятности  
chi\_c <- c(pnorm(0, mean = 4, sd = 4),  
 pnorm(0.25, mean = 4, sd = 4) - pnorm(0, mean = 4, sd = 4),  
 pnorm(0.75, mean = 4, sd = 4) - pnorm(0.25, mean = 4, sd = 4),  
 pnorm(1.5, mean = 4, sd = 4) - pnorm(0.75, mean = 4, sd = 4),  
 pnorm(3, mean = 4, sd = 4) - pnorm(1.5, mean = 4, sd = 4),  
 pnorm(6, mean = 4, sd = 4) - pnorm(3, mean = 4, sd = 4),  
 pnorm(12, mean = 4, sd = 4) - pnorm(6, mean = 4, sd = 4),  
 pnorm(20, mean = 4, sd = 4) - pnorm(12, mean = 4, sd = 4),  
 1 - pnorm(20, mean = 4, sd = 4))  
  
chi\_o <- c(0,   
 sum(bugs$hours==0.1),   
 sum(bugs$hours==0.5),   
 sum(bugs$hours==1),   
 sum(bugs$hours==2),   
 sum(bugs$hours==4),   
 sum(bugs$hours==8),  
 sum(bugs$hours==16),   
 sum(bugs$hours>=24))  
  
chisq.test(x = chi\_o, p = chi\_c)

## Warning in chisq.test(x = chi\_o, p = chi\_c): Chi-squared approximation may  
## be incorrect

##   
## Chi-squared test for given probabilities  
##   
## data: chi\_o  
## X-squared = 3930.7, df = 8, p-value < 2.2e-16

Критерий Хи-квадрат просто огромен, а p-значение практически равно нулю, что говорит о том, что наша выборка не является нормально распределенной.

Прогоним тесты Шапиро-Уилка и Харке-Бера.

shapiro.test(bugs$hours)

##   
## Shapiro-Wilk normality test  
##   
## data: bugs$hours  
## W = 0.69913, p-value < 2.2e-16

jarque.test(bugs$hours)

##   
## Jarque-Bera Normality Test  
##   
## data: bugs$hours  
## JB = 10683, p-value < 2.2e-16  
## alternative hypothesis: greater

lillie.test(bugs$hours)

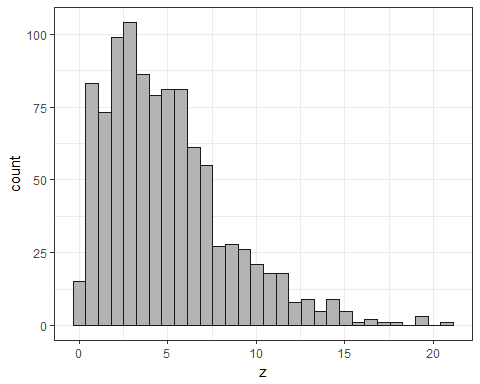
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: bugs$hours  
## D = 0.29977, p-value < 2.2e-16

Тесты подтверждают наш вывод. **Данное распределение никак нельзя считать нормальным.** Соответственно, нельзя использовать свойства нормально распределения для оценки вероятности, нельзя использоваться критерий 3-сигмы, а в силу большой асимметрии надо отдельно разбираться с вопросом компенсации трудозатрат на практике.

А что это за распределение? Я обсудил этот вопрос с коллегами на конференции SECR и Виктор Вальчук, эксперт по ТОС и ТРИЗ посоветовал мне обратить внимание на beta-распределение. Beta-распеределение с параметрами (1.7,16.7) нормированное на диапазон 0-53 часа дает действительно довольно похожий результат.

sh1 <- 1.7  
sh2 <- 16.7  
kn <- 53  
  
  
z <- rbeta(n = 1000, shape1 = sh1, shape2 = sh2)\*kn  
ggplot() + geom\_histogram(aes(x = z), fill = "gray70", colour = "gray10")+  
 theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



mean(z)

## [1] 4.941107

sd(z)

## [1] 3.442376

skewness(bugs$hours)

## [1] 3.046948

kurtosis(bugs$hours)

## [1] 18.55184

### Что же получается на практике?

Пришло время понять, правильно ли выбрана оценка сложного бага по умолчанию в 4 часа и происходит ли компенсация увеличения трудозатрат на сложные баги за счет экономии времени на простых багах.

Для начала давайте поймем, какую доля нашей работы приходится на легкие и на сложные баги в целом.

# доля вклада багов разной сложности и вероятности появления багов разной сложности исходя из фактической частотности событий  
  
total <- sum(bugs$hours)  
#общее количество трудозатрат на баги за весь период  
print(total)

## [1] 3733.2

bugs\_sum <- matrix(0, ncol = 3, nrow = 4)  
colnames(bugs\_sum) <- c("t<4", "t=4", "t>4")  
rownames(bugs\_sum) <- c("N", "H (hours)", "N, %", "H, %")  
  
# количество простых багов и трудозатраты на них  
bugs\_sum[1,1] <- sum(bugs$hours<4)  
bugs\_sum[2,1] <- sum(bugs$hours[bugs$hours<4])  
  
# количество багов средней сложности и трудозатраты на них  
bugs\_sum[1,2] <- sum(bugs$hours==4)  
bugs\_sum[2,2] <- sum(bugs$hours[bugs$hours==4])  
  
# количество багов высокой сложности и трудозатраты на них  
bugs\_sum[1,3] <- sum(bugs$hours>4)  
bugs\_sum[2,3] <- sum(bugs$hours[bugs$hours>4])  
  
# доля багов каждой категории от общего количества в процентах  
bugs\_sum[3,] <- 100 \* bugs\_sum[1,]/nrow(bugs)  
# доля трудозатрат на баги каждой категории от общих трудозатрат в процентах  
bugs\_sum[4,] <- 100 \* bugs\_sum[2,]/total  
  
print(bugs\_sum,digits = 3)

## t<4 t=4 t>4  
## N 402.0 327.0 190.0  
## H (hours) 561.2 1308.0 1864.0  
## N, % 43.7 35.6 20.7  
## H, % 15.0 35.0 49.9

Получается довольно интересно: 50% трудозатрат порождает лишь 20% сложных багов. Еще 50% простых багов порождают лишь 15% трудозатрат.

Но пока это нам ничего не дает. Посмотрим на ситуацию по-другому. Сложные баги изначально чаще всего оценивались по умолчанию в 4 часа. А уже после исправления подросли. В результате на таких багах по факту мы получаем перепланирование , где - фактические трудозатраты на подросшие баги, - количество таких багов. Иными словами - взяли на спринт больше чем можем сделать, ситуация переполнения, и это плохо. Или, иными словами, получается перерасход часов на исправление сложных багов.

Простые баги обычно сразу оцениваются менее чем в 4 часа. Если бы мы их тоже оценивали в 4 часа, то по факту имели бы недопланирование (или экономию, то есть на спринте оказывалось бы меньше часов, чем мы можем сделать, и это в общем-то хорошо). , где - фактические трудозатраты на простые баги, - количество простых багов.

Общая сумма затрат , где - сумма затрат на баги, оценка которых не изменилась и осталась прежней, 4 часа. А интересует нас прежде всего величина , определяющая баланс (или, компенсацию). Отрицательная величина говорит о том, что компенсация не происходит и на сложные баги времени уходит больше, чем мы могли бы экономить на простых. Дальнейшее рассмотрение основывается на этой концепции экономии на простых багах и перерасходе на сложных относительно средней оценки в 4 часа.

Если бы все баги были априорно оценены в 4 часа, то:

# фактически недоучтенное время на спринтах (перерасход на сложных багах) составило бы:  
t1 <- bugs\_sum[1,3]\*4 - bugs\_sum[2,3]  
print(t1)

## [1] -1104

# недопланированное время на спринтах (время, которое было бы сэкономлено, если бы исходная оценка была стандартной - 4 часа)  
t2 <- bugs\_sum[1,1]\*4 - bugs\_sum[2,1]  
print(t2)

## [1] 1046.8

# баланс в виде доли от общего объема в процентах  
print(100\*abs(t1+t2)/(total), digits = 3)

## [1] 1.53

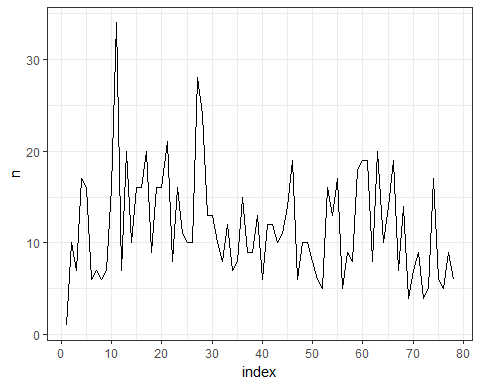
**Важный промежуточный вывод: в среднем, на достаточно длинном интервале времени происходит компенсация потерь времени на сложных багах за счет экономии на простых**. Если бы мы всегда все баги априори оценивали в 4 часа, а после исправления корректировали бы оценку, мы все равно вышли бы примерно на туже итоговую сумму и, что более важно, мы имели бы баланс между сложными и простыми багами, то есть экономия на простых багах компенсировала бы перерасход времени на сложных. **И в этом случае априорная оценка в 4 часа выбрана правильно.**

Однако есть одно важное "но". Дело в том, что мы работаем недельными спринтами и на спринт обычно попадает небольшое количество багов. Соответственно, вопрос о том, происходит ли компенсация в рамках спринта остается открытым и требует дополнительного изучения. К счастью, у нас есть информация о том, на каком спринте какой баг был исправлен, и мы может постараться найти ответ и на этот вопрос.

### Гипотетическая компенсация в пределах спринта

Построим сводную таблицу по спринтам и посмотрим, как по ним распределяются баги.

# сводная таблица по спринтам  
stages <- bugs %>% group\_by(sprint) %>% summarise(mean = mean(hours), median = median(hours), n = n(), total = sum(hours))  
  
index <- seq(1,nrow(stages),1)  
ggplot(data = stages) + theme\_bw() + scale\_x\_continuous(breaks=seq(0,80,by=10)) +  
 geom\_line(aes(x = index, y = n))



Количество багов на спринте так же является случайной величиной и как мы видим колеблется достаточно сильно. На отдельных спринтах исправляется буквально 5-7 багов, но есть и такие, где было исправлено больше 20.

В цифрах:

# среднее, медиана и среднеквадратическое отклонение количества бажных тикетов на спринт  
mean(stages$n)

## [1] 11.78205

median(stages$n)

## [1] 10

sd(stages$n)

## [1] 5.892208

# среднее, медиана и среднеквадратическое отклонение общих трудозатрат на бажные тикетов на спринт  
mean(stages$total)

## [1] 47.86154

median(stages$total)

## [1] 40.05

sd(stages$total)

## [1] 28.75104

Получается, что в среднем мы имеем на спринте около 10 бажных тикетов примерно на 40 идеальных часов. Но разброс действительно весьма велик. Примерно для трети спринтов количество тикетов на спринте явно недостаточно, чтобы компенсация работала, но об этом ниже.

# проверим себя, среднее, медиана и среднеквадратическое отклонение средних трудозатрат на один баг, должны получить ~4 часа,   
mean(stages$mean)

## [1] 4.163898

median(stages$mean)

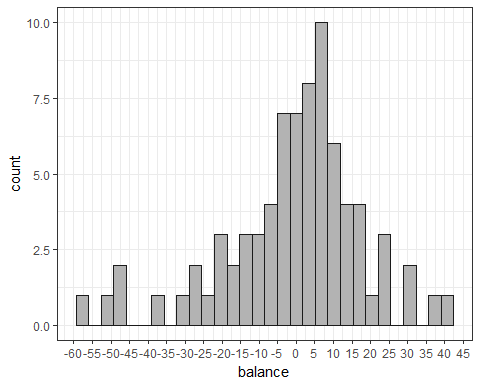
## [1] 3.798246

Сходится.

Хорошо, а что с компенсация по спринтам? Развернем исходную таблицу по спринтам и для каждого спринта вычислим перерасход (за счет недооценненых сложных багов, отрицательное число), экономию (которая могла бы быть за счет простых багов, положительное число) и баланс (он же компенсация = перерасход + экнономия). Построим распределение баланса по спринтам:

sprints <- spread(bugs, sprint, hours)  
sprints[is.na(sprints)] <- 0  
sprints <- sprints[,-1]  
sprints <- sprints[,-1]  
sprints <- t(as.matrix(sprints))  
  
sprints\_s <- sprints  
sprints\_s[sprints\_s>=4] <- 0  
sprints\_b <- sprints  
sprints\_b[sprints\_b<=4] <- 0  
  
sprints\_stat <- cbind.data.frame(rowSums(sprints<4 & sprints>0),   
 rowSums(sprints==4),   
 rowSums(sprints>4),   
 rowSums(sprints>0),  
 rowSums(sprints\_s),   
 rowSums(sprints\_b),   
 rowSums(sprints))  
rownames(sprints\_stat) <- rownames(sprints)  
colnames(sprints\_stat) <- c("small", "middle", "big", "all", "total\_s", "total\_b", "total")  
  
# сэкономленное время  
sprints\_stat$t1 <- sprints\_stat$small\*4 - sprints\_stat$total\_s  
# недопалнированное время  
sprints\_stat$t2 <- sprints\_stat$big\*4 - sprints\_stat$total\_b  
  
sprints\_stat$balance <- sprints\_stat$t1 + sprints\_stat$t2  
  
ggplot(data = sprints\_stat) + geom\_histogram(aes(x = balance), fill = "gray70", colour = "gray10") +  
 scale\_x\_continuous(breaks=seq(-60,60,by=5)) +  
 theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Перерасход имеет отрицательный знак, экономия положительный. Уже по гистограмме видно, что распределение тоже не очень-то нормальное. И мы имеет очень большой разброс баланса на спринтах. Бывали спринты, где было много сложных тикетов (недооцененных), но было и довольно много спринтов, когда могла иметь место существенная экономия на простых тикетах.

В цифрах:

# суммарный баланс по всем спринтам относительно общего объема трудозатрат в процентах (контрольно), должны выйти на цифру близкую к 0, как мы это видели раньше  
print(100\*abs(sum(sprints\_stat$balance))/total, digits = 3)

## [1] 1.53

# средний баланс  
print(mean(sprints\_stat$balance), digits = 2)

## [1] -0.73

# медианный баланс  
print(median(sprints\_stat$balance), digits = 3)

## [1] 2.5

# среднеквадратическое отклонение  
print(sd(sprints\_stat$balance), digits = 3)

## [1] 19

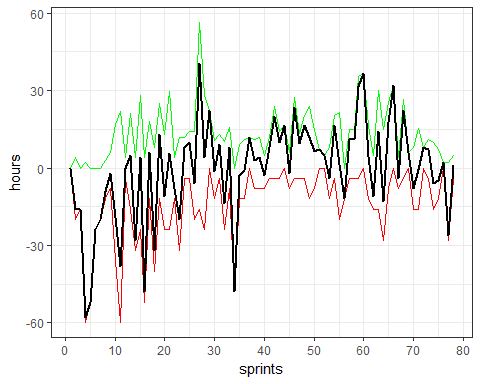
# количество спринтов, где баланс получился положительным или достаточно близким к нулю в процентах.   
print(100\*sum(sprints\_stat$balance>=-4)/nrow(sprints\_stat), digits = 1)

## [1] 68

Среднее -0.7, медиана 2.5, а мода судя по гистограмме >5. Среднеквадратическое отклонение очень большое - 20. И еще один важный промежуточный результат: **однако более чем на 2/3 спринтов баланс получился близким нулю или положительным!**

Построим график перерасхода (красный), экономии (зеленый) и балланса (черый) по спринтам.

sprints\_stat$name <- rownames(sprints\_stat)  
sprints\_stat <- arrange(sprints\_stat, name)  
  
  
ggplot(data = sprints\_stat) + theme\_bw() + scale\_x\_continuous(breaks=seq(0,80,by=10)) +  
 xlab("sprints") + ylab("hours") +   
# сэкономленное время   
 geom\_line(aes(x = index, y = sprints\_stat$t1), colour = "green", size = 0.5) +   
# недопланированное время   
 geom\_line(aes(x = index, y = sprints\_stat$t2), colour = "red", size = 0.5) +  
#баланс  
 geom\_line(aes(x = index, y = sprints\_stat$balance), size = 1)



Хорошо видно, что ситуация несколько улучшилась где-то в районе 35-го спринта (переход на недельные спринты?). И в последнее время мы чаще могли бы иметь положительный баланс, чем отрицательный.

**Таким образом, априорная оценка в 4 часа видится верной в случае, если она назначается всем тикетам на спринте, и на спринт берется достаточно большое количество бажных тикетов. Однако на практике простые баги априори получают низкие оценки трудозатрат и реальной экономии на простых багах практически нет.**  Наш фактический баланс близок к красной линии. Средний гипотетический перерасход по спринтам:

mean(sprints\_stat$t2[sprints\_stat$t2<0])

## [1] -16.47761

**Поэтому в нашем случае априорная оценка сложных багов в 4 часа видится заниженной.**

А сколько бажных тикетов мы должны иметь на спринте для того, чтобы компенсация устойчиво работала (допустим, мы всем бажным тикетам назначаем априорную оценку равную среднему, то есть 4 часа)? (In progress...)

### Результаты и обсуждения

1. Распределение трудозатрат на исправление ошибок в ПО не является нормальным. По крайней мере в данном конкретном случае. Оно имеет большую положительную асимметрию и просто колоссальный эксцесс. Поэтому оценка по методике 3-сигмы по формальным параметрам выборки будет давать неверные, сильно заниженные вероятности справа и ее нельзя применять.

Рассчитаем вероятность появления "черного лебедя" двумя способами: а) в рамках теоретического нормального распределения и б) опираясь на частоты в реальной выборке.

# "black swan" probability (out of 3-sigma ) in percents in within normal distribution   
(1-pnorm(mean(bugs$hours) + 3\*sd(bugs$hours), mean = mean(bugs$hours), sd = sd(bugs$hours)))\*100

## [1] 0.1349898

# the same within real sample (in percents):  
sum(bugs$hours>=16)/nrow(bugs)\*100

## [1] 3.373232

Таким образом, **вероятность "поймать черного лебедя" в реальности в 25 раз больше, чем если бы имели дело с нормальным распределением.**

Сравним плотности вероятности с точки зрения правила трех сигм справа.

# 1-сигма  
quantile(bugs$hours, probs = 0.683)

## 68.3%   
## 4

qnorm(0.683, mean = bugs\_m, sd = bugs\_sd)

## [1] 5.875728

# 2-сигмы  
quantile(bugs$hours, probs = 0.954)

## 95.4%   
## 8

qnorm(0.954, mean = bugs\_m, sd = bugs\_sd)

## [1] 10.4802

# 3-сигмы  
quantile(bugs$hours, probs = 0.997)

## 99.7%   
## 24

qnorm(0.997, mean = bugs\_m, sd = bugs\_sd)

## [1] 14.52857

Как можно видеть, вероятность попадания бага в 1-ю и 2-ю сигму отличаются, хотя и не сильно. Зато вероятность попадания в третью сигму на реальной выборке на много выше.

2.1 Баги с отрицательным временем исправления отсутствуют по определению (достаточно большая асимметрия). Но компенсация (редукция к среднему) в среднем на большой выборке должна происходить. На практике объем "сэкономленного времени" на простых багах (априори переоцениваемых) оказывается сопоставимым с объемом потерянного времени на сложных багах (априори недооцениваемых).

2.2 Мы работаем недельными спринтами и нас прежде всего интересует точность планирования, точнее возможность выполнить все задачи, взятые на спринт (добавить что-то еще мы всегда можем, ситуация, когда мы что-то не успеваем значительно хуже). Компенсация в рамках спринта происходила бы, если бы а) мы всегда для всех тикетов назначали исходную оценку в 4 часа и б) на спринт каждый раз попадало бы достаточное число тикетов. Однако на практике первое условие не выполняется в силу применяемой тактики априорной оценки тикетов, а второе условие выполняется в лучшем случае в половине случаев (смотря какой критерий использовать).

2.2.1 Если баг простой и понятный, то мы очень часто сразу прямо на планировании назначаем меньшую оценку (1, 2 часа). Но редко назначаем сложному багу большую чем 4 часа оценку. При этом если на баг уходит больше 4-х часов и это не "черный лебедь" он исправляется прямо на спинте, получая итоговую оценку в 8-24 часа.

2.2.2 Среднее количество багов, исправляемых за спринт около 10, но разброс большой. В то же время спринтов, на которых было исправлено менее 5-ти багов всего 6.

2.3 Таким образом на практике компенсация скорее всего происходит значительно реже, чем это могло бы быть в том случае, если бы на спринты бралось достаточно много багов и всем багам назначалась бы оценка равная среднему (4 часа). К сожалению, у нас есть только окончательная оценка тикета после исправления, и нет априорных, поэтому на данный момент невозможно точно вычислить фактический эффект компенсации.

3.1 **Надо отметить, что настоящие "черные лебеди" в эту статистику скорее всего вообще не попадают.** Речь идет о действительно сложных ситуациях, когда после длительных исследований и многократных попыток разобраться мы таки выясняем в чем дело и получаем несколько тикетов, в том числе с рефакторингом и доработками. При этом потери на исследования могут вообще не учитываться.

3.2 Бывает и так, что один и тот же баг исправляется несколько раз (неудачно) и воскресает спустя какое-то время как птица Феникс. Правильнее было бы суммировать трудозатраты по всем попыткам, но сейчас в статистику попадает несколько небольших исправлений, а не одно большое. Поэтому правый край распределения багов по трудозатратам должен быть еще выше.

Так что делать чтобы улучшить планирование?

Вариант 1. Оставить текущую тактику априорной оценки, но для сложных тикетов использовать завышенную априорную оценку, например, 6 или 8 часов. В частных случаях пытаться по возможности использовать и другие, бОльшие оценки, если к этому есть предпосылки.

Вариант 2. Использовать для всех багов оценку 4 часа, но при этом брать на спринт достаточно количество багов, чтобы компенсация работала. А это уже не просто организационно, так надо либо увеличить долю багфикса по отношению к новым разработкам, либо увеличивать длительность спринтов, либо увеличивать размер команды и/или производительность. У всех этих решений есть побочные эффекты и свои недостатки.

### Заключение

Таким образом, краткие ответы на вопросы, поставленные вначале будут такими:

1. Исследуемое распределение не является нормальным;
2. Вероятность поймать действительно сложный баг в жизни значительно больше, чем это предсказывает нормальное распределение;
3. Однако в среднем на длительном интервале времени компенсация увеличения трудозатрат на сложные баги за счет экономии на простых происходит;
4. Но только в рамках всей выборки, а в рамках рабочих спринтов ситуация хуже, на баланс рассчитывать не приходиться: примерно в трети спринтов гипотетический баланс существенно отрицательный;
5. В жизни ситуацию еще более осложняет текущая методика априорной оценки, которая практически исключает эффект компенсации; средняя оценка недопланирования на спринте составляет часов;
6. Черные лебеди конечно существуют, но как правило не попадают в статистику (аналог игровой ошибки по Талебу), так как в процессе "переваривания" раскладываются на множество тикетов, "размазанных" по нескольким спринтам, что искажает нашу оценку в лучшую сторону.