R Notebook

library(tidyverse)

## -- Attaching packages -------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 0.8.5  
## v tidyr 1.0.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts ----------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
library(ggplot2)  
# This sets the size of plots to a good default.  
options(repr.plot.width = 5, repr.plot.height = 4)  
  
getwd()

## [1] "C:/Users/dolug/Documents/R/Candy Crush Difficulty"

candy\_crush <- read.csv("candy\_crush.csv")  
head(candy\_crush)

## player\_id dt level num\_attempts num\_success  
## 1 6dd5af4c7228fa353d505767143f5815 2014-01-04 4 3 1  
## 2 c7ec97c39349ab7e4d39b4f74062ec13 2014-01-01 8 4 1  
## 3 c7ec97c39349ab7e4d39b4f74062ec13 2014-01-05 12 6 0  
## 4 a32c5e9700ed356dc8dd5bb3230c5227 2014-01-03 11 1 1  
## 5 a32c5e9700ed356dc8dd5bb3230c5227 2014-01-07 15 6 0  
## 6 b94d403ac4edf639442f93eeffdc7d92 2014-01-01 8 8 1

#the amount of unique ids  
print("Number of players:")

## [1] "Number of players:"

length(unique(candy\_crush$player\_id))

## [1] 6814

#the time frame we will be evaluating  
print("Period for which we have data:")

## [1] "Period for which we have data:"

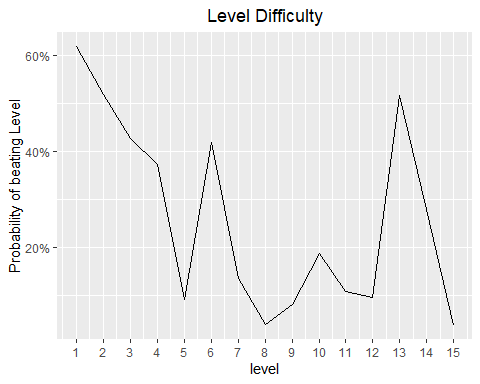
range(candy\_crush$dt)

## [1] "2014-01-01" "2014-01-07"

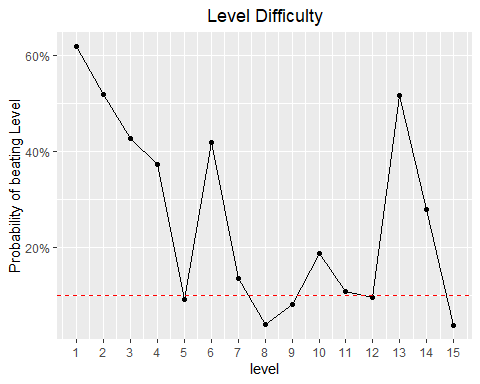
#calculating level difficulty  
difficulty <- candy\_crush %>% group\_by(level) %>% summarise(  
 attempts = sum(num\_attempts), success = sum(num\_success)) %>%   
 mutate(p\_success = success/attempts)  
  
print(difficulty)

## # A tibble: 15 x 4  
## level attempts success p\_success  
## <int> <int> <int> <dbl>  
## 1 1 1322 818 0.619   
## 2 2 1285 666 0.518   
## 3 3 1546 662 0.428   
## 4 4 1893 705 0.372   
## 5 5 6937 634 0.0914  
## 6 6 1591 668 0.420   
## 7 7 4526 614 0.136   
## 8 8 15816 641 0.0405  
## 9 9 8241 670 0.0813  
## 10 10 3282 617 0.188   
## 11 11 5575 603 0.108   
## 12 12 6868 659 0.0960  
## 13 13 1327 686 0.517   
## 14 14 2772 777 0.280   
## 15 15 30374 1157 0.0381

#plotting the level difficulty profile  
ggplot(difficulty,aes(level,p\_success)) +  
 geom\_line() + scale\_x\_continuous(breaks = 1:15) +  
 scale\_y\_continuous(label = scales::percent) +  
 ylab("Probability of beating Level") +  
 ggtitle("Level Difficulty") +  
 theme(plot.title = element\_text(hjust = 0.5))



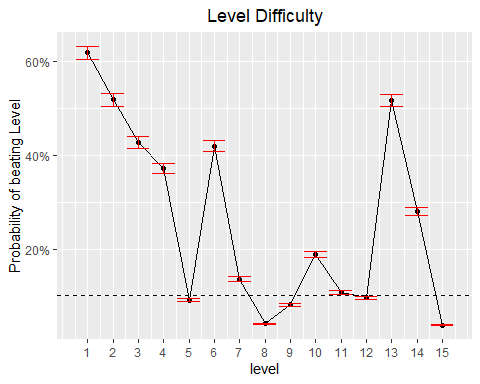
#to identity "hard" levels by defining a threshold  
#that levels with p\_success < 10% are considered hard.  
ggplot(difficulty,aes(level,p\_success)) +  
 geom\_line() + scale\_x\_continuous(breaks = 1:15) +  
 scale\_y\_continuous(label = scales::percent) +  
 ylab("Probability of beating Level") +  
 ggtitle("Level Difficulty") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 geom\_point() +  
 geom\_hline(yintercept=0.1, linetype='dashed', color='red')



#computing standard error of p\_success of each level  
error\_fn <- difficulty %>% mutate(error = sqrt(p\_success \* (1 - p\_success)/attempts))  
(error\_fn)

## # A tibble: 15 x 5  
## level attempts success p\_success error  
## <int> <int> <int> <dbl> <dbl>  
## 1 1 1322 818 0.619 0.0134   
## 2 2 1285 666 0.518 0.0139   
## 3 3 1546 662 0.428 0.0126   
## 4 4 1893 705 0.372 0.0111   
## 5 5 6937 634 0.0914 0.00346  
## 6 6 1591 668 0.420 0.0124   
## 7 7 4526 614 0.136 0.00509  
## 8 8 15816 641 0.0405 0.00157  
## 9 9 8241 670 0.0813 0.00301  
## 10 10 3282 617 0.188 0.00682  
## 11 11 5575 603 0.108 0.00416  
## 12 12 6868 659 0.0960 0.00355  
## 13 13 1327 686 0.517 0.0137   
## 14 14 2772 777 0.280 0.00853  
## 15 15 30374 1157 0.0381 0.00110

#Adding std error bars  
ggplot(error\_fn,aes(level,p\_success)) +  
 geom\_line() + scale\_x\_continuous(breaks = 1:15) +  
 scale\_y\_continuous(labels = scales::percent) +  
 ylab("Probability of beating Level") +  
 ggtitle("Level Difficulty") +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 geom\_point() +  
 geom\_hline(yintercept=0.1, linetype='dashed') +  
 geom\_errorbar(aes(ymin= p\_success - error, ymax = p\_success + error),   
 color='red')



# The probability of completing the episode without losing a single time  
p <- prod(difficulty$p\_success)  
  
# Printing it out  
p

## [1] 9.447141e-12